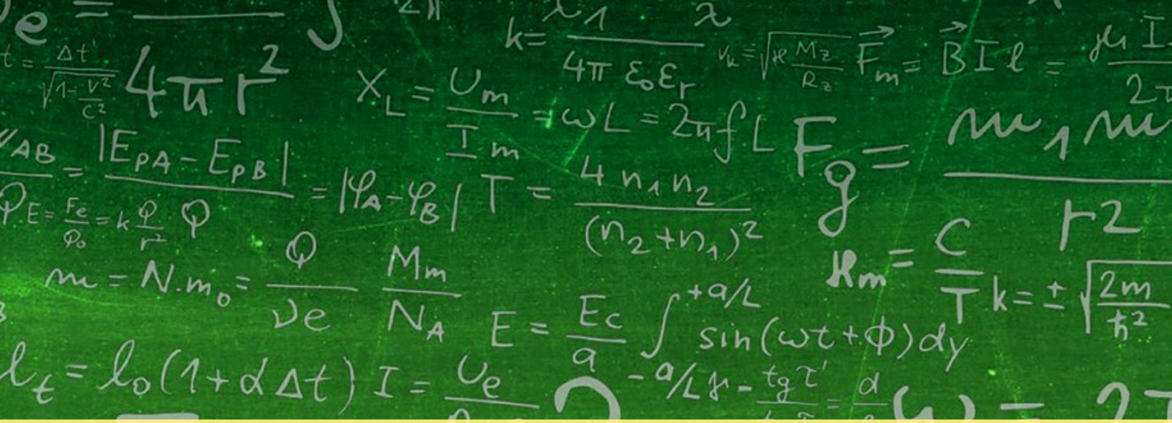


LECTURE NOTES IN COMPUTATIONAL SCIENCE AND ENGINEERING 139

Fred J. Vermolen · Cornelis Vuik *Editors*

Numerical Mathematics and Advanced Applications ENUMATH 2019

European Conference, Egmond aan Zee, The Netherlands, September 30 -October 4

Springer

Lecture Notes in Computational Science and Engineering

139

More information about this series at http://www.springer.com/series/3527

Fred J. Vermolen • Cornelis Vuik
Editors

Numerical Mathematics and Advanced Applications ENUMATH 2019

European Conference, Egmond aan Zee,
The Netherlands, September 30 - October 4

Springer

Editors
Fred J. Vermolen
DIAM
Delft University of Technology
Delft, The Netherlands

Cornelis Vuik
DIAM
Delft University of Technology
Delft, The Netherlands

ISSN 1439-7358 ISSN 2197-7100 (electronic)
Lecture Notes in Computational Science and Engineering
ISBN 978-3-030-55873-4 ISBN 978-3-030-55874-1 (eBook)
https://doi.org/10.1007/978-3-030-55874-1

Mathematics Subject Classification: 90-08, 35-XX, 65-XX, 68U20, 65M60, 68Uxx, 65M08

This Springer imprint is published by the registered company Springer Nature Switzerland AG.
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

Preface

The European Conference on Numerical Mathematics and Advanced Applications (ENUMATH) is an important conference series in Numerical Mathematics that is held every two years in a different European country. The series provides a platform for discussions about the latest insights in Numerical Analysis and its applications. This conference series is an important get-together event of computational scientists throughout Europe and the rest of the globe. Previous ENUMATH conferences were held in Paris (1995), Heidelberg (1997), Jyväskylä (1999), Ischia Porto (2001), Prague (2003), Santiago del Compostela (2005), Graz (2007), Uppsala (2009), Leicester (2011), Lausanne (2013), Ankara (2015) and Bergen (2017). We are proud to say that the reputation of the ENUMATH conferences is rated among the best in Numerical Mathematics and Scientific Computing worldwide. The publication of high standard conference proceedings has contributed largely to its reputation.

The current volume contains 120 proceedings papers from the event ENUMATH 2019 in Egmond aan Zee, in The Netherlands. The contributions are based on talks in minisymposia, contributed sessions and keynote talks. The level of the talks was excellent in general, and the atmosphere was very good and constructive. The number of delegates was 457, and the conference was characterised by its enormous diversity in topics within the science of Numerical Mathematics and Scientific Computing. This can also be seen by this booklet. Topics were within computational fluid dynamics, mathematical biology, quantum computing, traditional finite element analysis, isogeometric analysis, model order reduction, numerical linear algebra, optimisation, to mention some of them.

Scientific Computing is growing rapidly within the mathematical and computer-related disciplines. It has become a mature branch of science of its own, and it is engaged with the development of computational techniques that are needed to understand and to predict very diverse phenomena in Science, Finance, Medicine and Technology. The discipline focuses on the development of mathematical formalisms, as well as the development of efficient and robust algorithms. Next to these aforementioned aspects, Scientific Computing entails the analysis of convergence, efficiency, well-posedness and stability of the developed models and computational schemes. The numerical analytic aspects of the developed models and schemes are of utmost importance, and therefore we are happy to see numerous contributions that have been devoted to analytic aspects of Scientific Computing in this booklet.

The current ENUMATH conference organised in The Netherlands may be called a success. This success has been made possible by many people. In the first place, we thank the delegates for coming to the conference! You made this fantastic event possible with your presence, discussions, presentations and submission of papers! Thank you! Further, we thank the Enumath Programme Board, consisting of Barbara Wohlmuth, Franco Brezzi, Roland Glowinski, Gunilla Kreiss, Miloslav Feistauer, Yuri Kuznetsov, Pekka Neittäänmäki, Jacques Periaux, Alfio Quateroni, Rolf Rannacher and Endre Süli for giving us the opportunity to organise this event. We also thank the Scientific Committee and minisymposia organisers for reviewing the abstracts and conference papers. Furthermore, we thank the keynote speakers who all gave excellent presentations, where we personally thank Michele Benzi, Koen Bertels, Eduard Feireisl, Gitta Kutyniok, Maxim Olshanskii, Francesca Rapetti, Thomas Richter, Tuomo Rossi, Mishra Siddhartha, Stefan Vanderwalle and Karen Willcox.

The organisers are also thankful for the financial support from the sponsors: The Dutch Burns Foundation, NDNS+, 4TU.AMI, DCSE and the Delft University of Technology. Without their financial support, this conference would not have been possible. Last but not least, the administrational work was done by Marion van den Boer-Roggen (SciConf Scientific Conference Organisers, Eindhoven). You have helped us tremendously with all the paperwork and with very many of the emails that were sent to us. Thank you so much! Without you, the organisation and hence the conference could never have been this successful!

We conclude with thanking all the delegates again for their interesting contributions and we thank all the people who have been involved with reviewing abstracts and papers that made the excellent scientific level of this booklet and the conference possible.

Delft, The Netherlands Fred J. Vermolen
Delft, The Netherlands Kees Vuik
Delft, The Netherlands Matthias Möller

Contents

High Order Whitney Forms on Simplices and the Question of Potentials

Francesca Rapetti and Ana Alonso Rodríguez

Abstract In the frame of high order finite element approximations of PDEs, we are interested in an explicit and efficient way for constructing finite element functions with assigned gradient, curl or divergence in domains with general topology. Three ingredients, that bear the name of their scientific fathers, are involved: the de Rham's diagram and theorem, Hodge's decomposition for vectors, Whitney's differential forms. Some key images are presented in order to illustrate the mathematical concepts.

1 Introduction

The situation where a field is expressed with a differential operator d, such as **grad**, **curl** or div, applied to another field arises frequently in physics. In electromagnetic modeling, for example, it can be evocated by the physical model itself, as $\mathbf{curl}\,\mathbf{H} = \mathbf{J}$ in the Ampère's theorem relating the magnetic field $\mathbf{H}$ to the conducting current density $\mathbf{J}$, or artificially to simplify the problem solution, as $\mathbf{E} = -\mathbf{grad}\,V$ where $\mathbf{E}$ is a conservative electric (vector) field and V is the associated electric (scalar) potential. In both cases, the differentiated field, here $\mathbf{H}$, V, is called a potential, of $\mathbf{J}$, $\mathbf{E}$, respectively. The generalized Stokes' theorem $\int_M \mathrm{dw} = \int_{\partial M} \mathrm{w}$ establishes a duality between the functional differential operator d and the geometrical boundary operator ∂. It implies that potentials can exist only for fields w such that $\mathrm{dw} = 0$ (the closed forms in exterior calculus language). However, not all closed forms have potentials and this complication is correlated with the topological features of the domain. Due to the duality arising in Stokes' theorem, what matters is to

F. Rapetti (✉)
Dep. de Mathématiques J.-A. Dieudonné, Univ. Côte d'Azur, Nice, France
e-mail: Francesca.RAPETTI@univ-cotedazur.fr

A. A. Rodríguez
Dip. di Matematica, Università degli Studi di Trento, Trento, Italy
e-mail: ana.alonso@unitn.it

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_1

understand whether or not a part of the domain is the boundary of another part of the domain, and this is a homological question. Closed k-forms with no potential constitute the kth de Rham cohomology group H^k, which is by de Rham's theorem [19], isomorphic with the kth homology group H_k of the domain. The uniqueness question of potentials in a topologically simple unbounded domain $\Omega \subset \mathbb{R}^3$ was first settled by Helmholtz [11] about the decomposition of vector fields in fluid dynamics. One century later, Hodge [13] introduces, in the formalism of exterior calculus, a decomposition that generalizes the one proposed by Helmholtz to any space dimension n and to domain of general topology (in the literature, this decomposition is sometimes attributed to Ladyzhenskaya who refers to it as Weil's decomposition in [14]). The theory developed by Hodge to study algebraic geometry is built on the work of de Rham, on the de Rham cohomology.

When we pass to the discrete finite element setting, all these homological and cohomological concepts are not impacted by the discretization process (apart from perhaps the harmonic one). They have equivalents that neither depend on the size $h > 0$ of the mesh elements (simplices here), nor on the polynomial degree $q \geq 1$ of the basis functions adopted to reconstruct the fields. This is largely due to having adopted the correct formalism, the one of differential forms, and to the geometrical nature of Weil-Whitney forms [8, 16, 20, 21], that reconstruct fields. Graph theory and linear algebra are then sufficient to construct effective algorithms to complete the computational side. Stating the necessary and sufficient conditions for assuring that a function defined in a bounded set $\Omega \subset \mathbb{R}^3$ is the gradient of a scalar potential, the curl of a vector potential or the divergence of a vector field is one of the most classical problems of vector analysis (see for example [5]). In these pages we answer to the question of describing potentials in terms of finite element bases of high polynomial order in domain of general topology. The answer shows an interesting interplay of differential calculus and topology that is the goal of the present work.

2 The Continuous Side of the de Rham's Diagram

We introduce the minimal notation to present the question of potentials in terms of differential forms, referring to [6, 8] for more details.

Let us consider the n-dimensional Euclidean space $\mathbb{R}^n$, with $n \in \mathbb{N}$, and let $\Omega \subset \mathbb{R}^n$ be a (sufficiently) smooth n-manifold. We denote by $\Lambda^k(\Omega)$ the space of smooth differential k-forms on Ω. Scalar potentials, field intensities, flux densities, or densities are the so-called proxy fields of the corresponding differential k-forms. Among the linear operators acting on these forms, some are metric dependent others not. Differential forms can be integrated and differentiated on Ω, without involving any additional metric structure. If S is an oriented, piecewise smooth k-dimensional submanifold of Ω, and w is a piecewisely continuous k-form, then $\int_S w$ is well-defined. The notation $\int_S w$ is compact, in the sense that it stands for evaluating w at the point S if $k = 0$, or computing the line integral of w along S if $k = 1$, or estimating the surface integral of w over S if $k = 2$ and computing the volume integral of w over S when $k = 3$. In exterior calculus, we have particular operators acting on forms, three of them matter here. These are the exterior or wedge product $\wedge$, the Hodge's operator $\star$, and the exterior derivative d.

The operator $\wedge : \Lambda^k(\Omega) \times \Lambda^\ell(\Omega) \to \Lambda^{k+\ell}(\Omega)$ is a natural multiplicative map among forms such that $w \wedge z = (-1)^{k\ell} z \wedge w$, for all $w \in \Lambda^k(\Omega)$ and $z \in \Lambda^\ell(\Omega)$. It just generalizes to forms the dot and cross products among vectors. As an example, $f^0 \wedge u = fu$, $u^1 \wedge v^1 = u \times v$, $u^1 \wedge v^2 = u^2 \wedge v^1 = u \cdot v$. Note that we define the integral on Ω (only) of a differential n-form $u(x) = f(x) dx_1 \wedge \ldots \wedge dx_n$ such that

$$\int_\Omega u = \int_\Omega f(x)\, dx_1 \wedge \ldots \wedge dx_n$$

where the integral on the right is the standard integral on real functions f and $dx_1 \wedge \ldots \wedge dx_n$ plays the role of infinitesimal (oriented) volume $\det(dx_1, \ldots, dx_n)$. Recall that the metric defines an inner product for vectors. This notion also extends to forms: given a metric, one can define the product of two k-forms in $\Lambda^k(\Omega)$ which will measure, in a way, the projection of one onto the other, see [1] for a formal definition. Given this inner product denoted $\langle ., . \rangle$, the Hodge's operator $\star : \Lambda^k(\Omega) \to \Lambda^{n-k}(\Omega)$ is such that we may define

$$\int_\Omega w \wedge z = \int_\Omega \star w\, z \,\mathrm{vol} = \langle \star w, z \rangle \,\mathrm{vol}, \quad \forall w \in \Lambda^k,\ z \in \Lambda^{n-k}(\Omega).$$

Let $\mathrm{d} : \Lambda^k(\Omega) \to \Lambda^{k+1}(\Omega)$ denote the exterior derivative (where the term "exterior" is to indicate that d increases the degree of the form). It is linear and satisfies the two key properties $\mathrm{d} \circ \mathrm{d} = 0$ and the Leibniz's rule

$$\mathrm{d}^{k+\ell}(w \wedge z) = \mathrm{d}^k(w) \wedge z + (-1)^k w \wedge \mathrm{d}^\ell z, \quad w \in \Lambda^k(\Omega),\ z \in \Lambda^\ell(\Omega)\ .$$

The index k in $\mathrm{d}^k(w)$ does not indicate a derivative of order k on w, but an exterior derivative on the k-form w (in the following, we just write $\mathrm{d}\, w$). For $k = 0$, we have $\mathrm{d}\, w = \sum_{i=1}^n \frac{\partial f}{\partial x_i} dx_i$.

By (cochain) complex $(A^\bullet, \mathrm{f}^\bullet)$ we mean a sequence of algebraic objects with Abelian structure (*e.g.*, vector spaces) A^{-1}, A^0, A^1, A^2, …, connected by homomorphisms $\mathrm{f}^k : A^k \to A^{k+1}$ such that, for each k, it holds $\mathrm{f}^k \mathrm{f}^{k-1} = 0$. In the Euclidean space, differential forms on Ω constitute the complex

$$0 \xrightarrow{i} \Lambda^0(\Omega) \xrightarrow{\mathrm{d}^0} \Lambda^1(\Omega) \xrightarrow{\mathrm{d}^1} \Lambda^2(\Omega) \xrightarrow{\mathrm{d}^2} \Lambda^3(\Omega) \xrightarrow{0} 0$$

known as de Rham complex and denoted by $(\Lambda^\bullet(\Omega), \mathrm{d}^\bullet)$. In terms of proxies defined on Ω, the L^2 de Rham complex reads

$$0 \xrightarrow{i} \mathrm{H}^1 \xrightarrow{\mathbf{grad}} \mathrm{H}(\,\mathbf{curl}) \xrightarrow{\mathbf{curl}} \mathrm{H}(\mathrm{div}) \xrightarrow{\mathrm{div}} \mathrm{L}^2 \xrightarrow{0} 0.$$

A differential form $w \in \Lambda^k(\Omega)$ is said to be closed if $\mathrm{d}w = 0$, and exact if there exists a $(k-1)$-form z such that $w = \mathrm{d}z$. These concepts arise in physics. For example given a vector field $\mathbf{f}$ representing a force, one would like, if the force is conservative, to find a function p called the potential energy, such that $\mathbf{f} = \mathbf{grad}\, p$. With differential forms, to say that $\mathbf{f}$ is conservative is equivalent to say that the corresponding differential form f is exact. Since $\mathrm{d} \circ \mathrm{d} = 0$, we have $\mathrm{Im}\, \mathrm{d}^k \subset \mathrm{Ker}\, \mathrm{d}^{k+1}$, that is every exact form is closed. So, if $\mathbf{f}$ is conservative, the corresponding form f is also closed. It is natural to ask when a closed form is exact. The answer depends on the topology of the manifold Ω (see more details in [9], for example). In the de Rham complex, the Poincaré lemma addresses this question to a large extent.

The idea behind de Rham cohomology is to define equivalence classes of closed forms on a manifold. Let us denote by $Z^k(\Omega)$ the set $\mathrm{Ker}\, \mathrm{d}^k$ of closed k-forms and by $B^k(\Omega)$ the set $\mathrm{Im}\, \mathrm{d}^{k-1}$ of exact k-forms. We have in general that $B^k(\Omega) \subset Z^k(\Omega)$. The quotient

$$H^k = Z^k(\Omega)/B^k(\Omega) = \mathrm{Ker}\, \mathrm{d}^k / \mathrm{Im}\, \mathrm{d}^{k-1}$$

is the k-th cohomology group of Ω and we can write $Z^k(\Omega) = B^k(\Omega) \oplus H^k$. Indeed, one classifies two closed forms $u, v \in \Lambda^k(\Omega)$ as cohomologous if they differ by an exact form, that is, if $u - v$ is exact. This classification induces an equivalence relation on the space of closed forms in $\Lambda^k(\Omega)$ and H^k is the set of equivalent classes, namely the set of closed forms modulo the exact ones. The previous notions can be summarized in the diagram of Fig. 1, where the portion of de Rham complex between k- and $(k+1)$-forms is visualized. A complete de Rham diagram in three dimensions is given in Fig. 2. Horizontal lines show the Hodge's decomposition of $\Lambda^k(\Omega)$, that is $\Lambda^k(\Omega) = B^k \oplus H^k \oplus Y^k$ where Y^k is generally characterized as $Y^k = \delta^k \Lambda^{k+1}(\Omega)$ with the introduction of the coderivative operator $\delta^k : \Lambda^k(\Omega) \to \Lambda^{k-1}(\Omega)$ that for a smooth domain is defined as $\star \delta^k w = (-1)^k \mathrm{d} \star w$ for all $w \in \Lambda^k(\Omega)$, in order to have $\langle \mathrm{d}\alpha, \beta \rangle = \langle \alpha, \delta\beta \rangle$, for all $\alpha \in \Lambda^{k-1}(\Omega)$ and $\beta \in \Lambda^k(\Omega)$ vanishing near the boundary.

Theorem 1 (Hodge Decomposition for Forms) *Given a compact oriented Riemann manifold Ω, any k-form w on Ω uniquely splits into the sum of three terms, w_1, w_2, w_3, where w_1 is exact, w_2 is co-exact and w_3 is harmonic.*

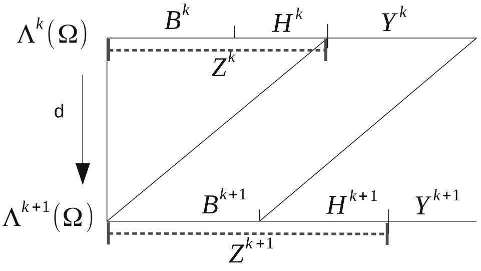


Fig. 1 The de Rham complex between $\Lambda^k(\Omega)$ and $\Lambda^{k+1}(\Omega)$. On the horizontal lines, the Hodge's decomposition of $\Lambda^\bullet(\Omega)$. Oblique lines mimic the action of the d operator: $\mathrm{d}\,(Z^k) = \{0\}$ and $\mathrm{d}(Y^k) = B^{k+1}$

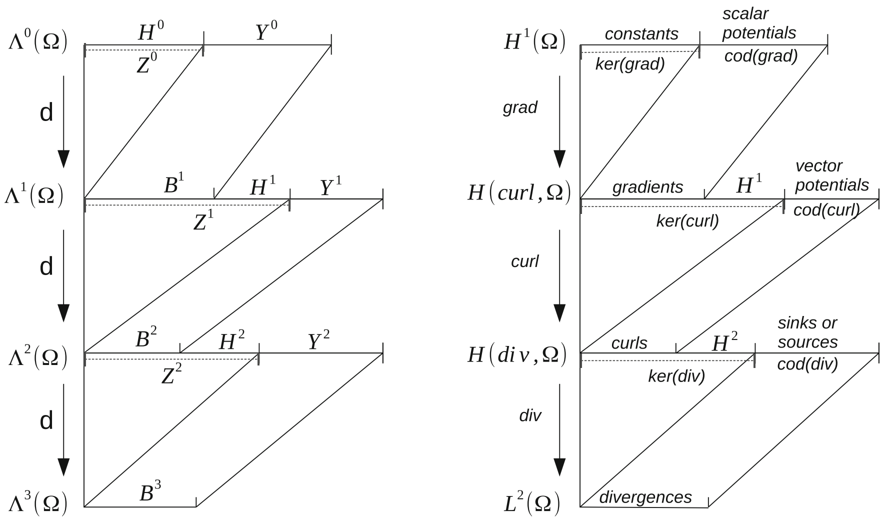


Fig. 2 The de Rham complex for $n = 3$, in the exterior calculus formalism (left) and in the vector formalism (right). The set H^1 (resp. H^2) on the left corresponds with the space $H^1 = \{\mathbf{u} \in L^2(\Omega)^3,\ \mathbf{curl}\,\mathbf{u} = \mathbf{0},\ \mathrm{div}\,\mathbf{u} = 0,\ \mathbf{u}\cdot\mathbf{n}_{\partial\Omega} = 0\}$ (resp. $H^2 = \{\mathbf{u} \in L^2(\Omega)^3,\ \mathbf{curl}\,\mathbf{u} = \mathbf{0},\ \mathrm{div}\,\mathbf{u} = 0,\ \mathbf{u}\times\mathbf{n}_{\partial\Omega} = \mathbf{0}\}$) on the right

So, w_1 exact (resp. w_2 co-exact) means that there exists a $(k-1)$-form α (resp. a $(k+1)$-form β) such that $w_1 = \mathrm{d}\alpha$ (resp. $w_2 = \delta\beta$). And, w_3 is harmonic if $\Delta w_3 = 0$ with $\Delta = \mathrm{d}\delta + \delta\mathrm{d}$. This follows by noting that exact and co-exact forms are orthogonal; the orthogonal complement consists in forms that are both closed and co-closed, thus harmonic. Orthogonality is defined with respect to the inner product $\langle .,. \rangle$ on $\Lambda^k(\Omega)$ (see, e.g., [6, 9]).

There exists a relation (of duality) between the exterior derivative d and the boundary operator ∂, stated by the Stokes's theorem

$$\int_S \mathrm{d}w = \int_{\partial S} w, \qquad \forall S \subset \Omega,\ \dim S = k, \quad \forall w \in \Lambda^k(\Omega).$$

k-forms are dual of k-manifolds S, and d is the adjoint of the boundary operator ∂. We have a (chain) complex denoted by $(C_\bullet, \partial_\bullet)$ where C_k if the set of chains of k-manifolds on Ω and $\partial_k : C_k \to C_{k-1}$, for $k \geq 1$. A k-cycle is a k-chain γ such that $\partial_k\gamma = 0$, thus, by definition, any k-chain that belongs to $\mathrm{Ker}\,\partial^k$. A k-chain γ is a k-boundary if there exists a $(k+1)$-chain σ such that $\partial_{k+1}\sigma = \gamma$. The set of k-boundaries coincides with $\mathrm{Im}\,\partial_{k+1}$. Due to the property $\partial_k\partial_{k+1} = 0$, we have $\mathrm{Im}\,\partial_{k+1} \subset \mathrm{Ker}\,\partial_k$. Two k-chains α, β are homologous if they differ for a boundary, that is, if $\alpha - \beta$ is a k-cycle. The set of equivalent classes in C_k is the k-th homology group H_k defined by

$$H_k = \mathrm{Ker}\,\partial^k / \mathrm{Im}\,\partial^{k+1}.$$

Theorem 2 (de Rham, See e.g. [10]) *For any value of the integer k, the k-th homology H_k is isomorphic to the k-th cohomology group H^k.*

The dimension of H_k is finite and defines the Betti's number b_k. In three dimensions, $b_0 = \dim H_0 = m$ is the number of connected components of Ω. Indeed, any function on Ω with zero derivative everywhere is separately constant on each of the connected components of Ω. In the following, we suppose Ω connected, thus $m = 1$. The first and second Betti's numbers, $b_1 = \dim H_1 = g$ and $b_2 = \dim H_2 = p$, correspond, resp., to the number of loops and cavities in Ω. Finally, $b_3 = 0$. These numbers are invariants, quantities that cannot change by continuous deformation and that characterize the topological space Ω. In other words, if Ω is a sphere $\mathcal{S}$, it will never be possible to deformate it continuously into a torus $\mathcal{T}$ since $b_1(\mathcal{S}) \neq b_1(\mathcal{T})$. The Euler's characteristic number $\chi(\Omega) = b_0 - b_1 + b_2 - b_3$ is also an invariant of Ω. For topologically trivial domains, we have $H_k = \{0\}$, for all $0 < k < n$ (this result is known as Poincaré's lemma).

3 The Discrete Side of the de Rham's Diagram

The key point in the de Rham's theorem is that the equivalence classes of H^k can be defined starting from those of H_k, therefore it expresses basic topological information about smooth manifolds in a form particularly adapted to computation. Indeed, thanks to the following result [17], we can rely on simplicial homology.

Lemma 1 *Let $\tau_h = (V, E, F, T)$ be a simplicial triangulation over Ω and $\Omega_h = \cup_{t \in T} t$. The k-th homology groups $H_k(\Omega)$ and $H_k(\Omega_h)$ are isomorphic.*

Even if τ_h is a simplicial triangulation of Ω, the topological properties computed on Ω_h are the same as those of Ω. For Ω connected, e.g., it holds

$$(\chi(\Omega) =) \qquad 1 - g + p = n_V - n_E + n_F - n_T \qquad (= \chi(\Omega_h))$$

where n_V, n_E,n_F, n_T are, respectively, the cardinalities of the sets of vertices V, edges E, faces F and tetrahedra T of the mesh τ_h. Given a simplicial mesh τ_h over $\bar{\Omega}$, we denote by $W^k_{r+1} = \mathcal{P}^-_{r+1}\Lambda^k(\tau_h)$ the set of Whitney differential k-forms of polynomial degree $r + 1$, where $k \in \{0, 1, 2, 3\}$ is the order of the form (see [6] for more details on the properties of these spaces). It is a compact notation to indicate space of polynomial functions which are well-known in finite elements. Indeed, for $k = 0$, we have $W^0_{r+1} = L_{r+1}$, the space of continuous, piecewise polynomials of degree $r + 1$; for $k = 1$, we obtain $W^1_{r+1} = N_{r+1}$ the first family of Nédélec edge element functions of degree $r + 1$; for $k = 2$, we get $W^2_{r+1} = RT_{r+1}$ the space of Raviart-Thomas functions of degree $r + 1$; for $k = 3$, we find $W^3_{r+1} = P_r$ discontinuous piecewise polynomials of degree r. The spaces W^k_{r+1} are connected in a complex by the linear operator d^k which can be represented by suitable matrices, namely G ($k = 0$), R ($k = 1$), D ($k = 2$) resp., with entries $0, \pm 1$, once a set of unisolvent dofs and consequently a basis in each space W^k_{r+1} have been fixed.

For $r = 0$, the dimension of the space W_1^k coincide with the number of k-simplices in the mesh, indeed $\dim L_1 = n_V$, $\dim N_1 = n_E$, $\dim RT_1 = n_F$ and $\dim P_0 = n_T$. Moreover, the matrices G, R, D are, resp., the edge-to-node, face-to-edge and tetrahedron-to-face connectivity matrices taking also into account respective orientations.

For $r > 0$, as explained in [16], by connecting the nodes of the principal lattice of degree $r+1$ in a n-simplex $t \in T$, we obtain a number of *small n-simplices* that are $1/(r+1)$-homothetic to t. The *small k-simplices*, $0 \leq k < n$, are all the k-simplices that compose the boundary of the small n-simplices. Any small k-simplex is denoted by a couple $\{\boldsymbol{\alpha}, s\}$, with s a k-simplex of τ_h and $\boldsymbol{\alpha}$ is a multi-integer $(\alpha_0, \dots, \alpha_n)$ with $\sum_{i=0}^n \alpha_i = r$, $\alpha_i \in \mathbb{Z}$ and $\alpha_i \geq 0$. The term *active* is to indicate all couples $\{\boldsymbol{\alpha}, s\}$ such that the function $\lambda^{\boldsymbol{\alpha}} \mathbf{w}^s$ belongs to a basis of W_{r+1}^k, where $\lambda^{\boldsymbol{\alpha}} = \lambda_0^{\alpha_0} \lambda_1^{\alpha_1} \cdots \lambda_n^{\alpha_n}$ and $\mathbf{w}^s \in W_1^k$. Indeed, by considering all possible multi-indices $\boldsymbol{\alpha}$ in the couples $\{\boldsymbol{\alpha}, s\}$, one generates more functions $\lambda^{\boldsymbol{\alpha}} \mathbf{w}^s$ than necessary. The dimension of the space W_{r+1}^k coincide with the number of *active small k-simplices* in the mesh, and the meaning of the matrices G, R, D is the same as for the case $r = 0$, provided that we work with the active small k-simplices instead of the k-simplices of the mesh τ_h. The small k-simplices were born to define a set of unisolvent dofs, the weights $\int_{\{\boldsymbol{\alpha}, s\}} u$, for functions $u \in W_{r+1}^k(t)$ when $r > 0$, that, differently from the classical moments, maintain a physical interpretation.

The cardinality of the set of weights on active small k-simplices coincides with $\dim W_{r+1}^k$ that is given below for $q = r + 1 \geq 1$ (the terms that are multiplied by $\dim \mathbb{P}_\ell(.)$ with $\ell < 0$ have to be neglected)

$$\begin{array}{rlllll}
d_L := \dim W_q^0 = & n_V & +n_E \dim \mathbb{P}_{q-2}(e) & +n_F \dim \mathbb{P}_{q-3}(f) & +n_T \dim \mathbb{P}_{q-4}(t) \\
d_N := \dim W_q^1 = & & +n_E \dim \mathbb{P}_{q-1}(e) & +n_F \dim \mathbb{P}_{q-2}(f)^2 & +n_T \dim \mathbb{P}_{q-3}(t)^3 \\
d_{RT} := \dim W_q^2 = & & & +n_F \dim \mathbb{P}_{q-1}(f) & +n_T \dim \mathbb{P}_{q-2}(t)^3 \\
d_P := \dim W_q^3 = & & & & +n_T \dim \mathbb{P}_{q-1}(t).
\end{array}$$

Proposition 1 *The identity $\chi(\Omega) = d_L - d_N + d_{RT} - d_P$ holds for all $r \geq 0$.*

Proof By a simple computation with factorials, for $q = r + 1$, it holds:

$$\begin{aligned}
& d_L - d_N + d_{RT} - d_P \\
& \quad = n_V + n_E(\dim \mathbb{P}_{r-1}(e) - \dim \mathbb{P}_r(e)) \\
& \quad +n_F(\dim \mathbb{P}_{r-2}(f) - 2 \dim \mathbb{P}_{r-1}(f) + \dim \mathbb{P}_r(f)) \\
& \quad +n_T(\dim \mathbb{P}_{r-3}(t) - 3 \dim \mathbb{P}_{r-2}(t) + 3 \dim \mathbb{P}_{r-1}(t) - \dim \mathbb{P}_r(t)) \\
& \quad = n_V + n_E \left[\binom{r}{1} - \binom{r+1}{1} \right] \\
& \quad +n_F \left[\binom{r}{2} - 2\binom{r+1}{2} + \binom{r+2}{2} \right] \\
& \quad +n_T \left[\binom{r}{3} - 3\binom{r+1}{3} + 3\binom{r+2}{3} - \binom{r+3}{3} \right] \\
& \quad = n_V - n_E + n_F - n_T = \chi(\Omega).
\end{aligned}$$

□

From now on, d_L (resp. d_N, d_{RT}, d_P) denotes the cardinality of the set of nodes or small nodes (resp. edges or active small edges, faces or active small faces, tetrahedra or small tetrahedra) whatever $r \geq 0$ is, and the terms *active* and *small* for k-simplices are taken for granted.

4 Notions from Graph Theory

Before continuing, we need a drop of graph theory (see, e.g., [18] for details).

Definition 1 The all-nodes incidence matrix $M^e \in \mathbb{Z}^{n\times m}$ of a directed graph $\mathcal{M} = (\mathcal{N}, \mathcal{A})$, with n nodes $\mathcal{N} = \{\mathfrak{n}_i\}_{i=0}^{n}$, m arcs $\mathcal{A} = \{\mathfrak{a}_j\}_{j=1}^{m}$, and with no self-loop is the matrix with entries

$$(M)^e_{i,j} = \begin{cases} 1 & \text{if } \mathfrak{a}_j \text{ is incident on } \mathfrak{n}_i \text{ and oriented toward it,} \\ -1 & \text{if } \mathfrak{a}_j \text{ is incident on } \mathfrak{n}_i \text{ and oriented away from it,} \\ 0 & \text{if } \mathfrak{a}_j \text{ is not incident on } \mathfrak{n}_i \text{ .} \end{cases}$$

Each column (arc) has exactly two entries (extreme nodes) different from zero: 1 and -1. The rows are not linearly independent because their sum is the zero vector. An incidence matrix of $\mathcal{M}$ is any submatrix of M^e with $n-1$ rows and m columns.

Definition 2 A tree of a graph $\mathcal{M} = (\mathcal{N}, \mathcal{A})$ is a connected acyclic subgraph of $\mathcal{M}$. A spanning tree $\mathcal{S}$ is a tree of $\mathcal{M}$ containing all its nodes (an example in Fig. 3).

Theorem 3 *Let $\mathcal{M} = (\mathcal{N}, \mathcal{A})$ be a connected directed graph with no self-loop and $M \in \mathbb{Z}^{(n-1)\times m}$ an incidence matrix of $\mathcal{M}$. Let $\mathcal{S} = (\mathcal{N}, \mathcal{B})$ be a spanning tree of $\mathcal{M}$ and M_{st} the submatrix of order $n-1$ of M given by the columns of M that correspond to the arcs in $\mathcal{S}$. Then M_{st} is invertible and the nonzero elements in each row of M_{st}^{-1} are either all 1 or all -1.*

Graph's theory matters in this context because D is an incidence matrix of the *dual* graph, with nodes that are the tetrahedra plus one additional node to represent the exterior of Ω, and arcs that are the faces of the mesh. Any interior face connects two tetrahedra and an face on the boundary connects a tetrahedron with the node representing the exterior of the domain. The node that corresponds to the exterior is the reference node of D. On the other hand, $G^\top$ is the all-node incidence matrix of the primal graph with nodes at the mesh nodes and arcs that are the edges of the mesh. For a multi-connected domain Ω, the associated graph has a connected component (thus a spanning tree) for each connected component of the domain. We refer to [3] for more details on the construction of these two graphs when $r > 0$, here we rather detail their use to solve the problem of potentials.

In Fig. 4, each horizontal line at level k collects all functions of W^k_{r+1} built on the $\lambda^{\boldsymbol{\alpha}}\mathbf{w}^s$ (resp. all chains in C_k built on *active small* k-simplices $\{\boldsymbol{\alpha}, s\}$) on the left (resp. right) diagram. For $k = 0$, the *end path nodes* are all the nodes in the mesh, due to the arbitrarity of the considered path, apart from the m roots (one node for each connected component of Ω). Here we suppose $m = 1$ for simplicity, but for

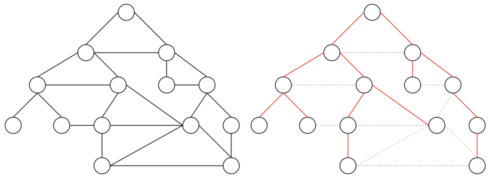


Fig. 3 (Left) the graph, (right) a spanning tree (resp. the co-tree) in solid line (resp. dashed line) with top node as *root* and bottom nodes as *leaves*, namely those nodes that have only one edge of the tree incident to them

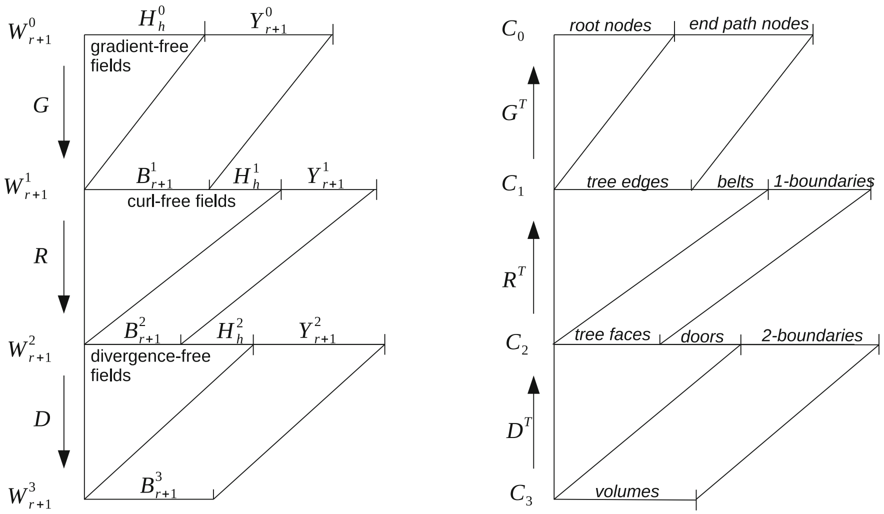


Fig. 4 Discrete cochain (left) and chain (right) de Rham complexes in three dimensions. Belts (resp. doors) are edges (resp. faces) lying on the loops (resp. cavities) which generate H_1 (resp. H_2), one for each class of equivalence. Note that on the discrete cochain side, $H^1_h = \{\mathbf{z}_h \in W^1_{r+1},\ \mathbf{z}_h \in \operatorname{Ker} R,\ \mathbf{z}_h \notin \operatorname{Im} G\}$ and $H^2_h = \{\mathbf{u}_h \in W^2_{r+1},\ \mathbf{u}_h \in \operatorname{Ker} D,\ \mathbf{u}_h \notin \operatorname{Im} R\}$

$m > 1$ it is sufficient to repeat the construction for each component of Ω. For $k = 1$, the *tree edges* are all the edges in a spanning tree, the set of *belts* collects one edge on each loop generating H_1 and the *1-boundaries* are the remaining edges, neither in the tree nor among the belts, but which are necessary to describe 1-chains bounding surfaces. For $k = 2$, the *tree faces* are all the faces in a spanning (dual) tree, the set of *doors* collects one face on each cavity generating H_2 and the *2-boundaries* are the remaining faces, neither in the tree nor among the doors, but necessary to describe 2-chains bounding volumes.

5 From Fields to Potentials

We wish to give an algorithm to construct finite element (scalar or vector) functions with assigned gradient, curl or divergence. We assume to know a basis $\{\sigma_j\}_{j=1,g}$ of $H_1(\bar{\Omega})$ and a basis $\{\theta_\ell\}_{\ell=1,p}$ of $H_2(\bar{\Omega})$. A suitable and easy way for constructing σ_j and θ_ℓ is presented in [12, 15]. Moreover, we suppose to have a spanning tree S_h of the graph $\mathcal{M}^G = (\mathcal{N}^G, \mathcal{A}^G)$ with $\mathcal{N}^G$ described by the small nodes and the arcs $\mathcal{A}^G$ by the active small edges, as explained in [3], where *spanning* means that S_h visits all nodes and *tree* stands to indicate that the arcs in S_h cannot connect in a loop. In addition to S_h, we need a spanning tree S_h^* of the graph $\mathcal{M}^D = (\mathcal{N}^D, \mathcal{A}^D)$ with the nodes $\mathcal{N}^D$ given by the small tetrahedra barycenters and the arcs $\mathcal{A}^D$ by the active small faces shared by neighbouring elements, as explained in [3]. Again, it is spanning, so it visits all small tetrahedra and it does not contain active small faces that can close chambers. The determination of a spanning tree is a standard procedure in graph theory [10]. With these tools, we can construct potentials whatever the approximation degree $r > 0$ is, generalizing the procedure detailed in [4] for $r = 0$.

Constructing a Function with Assigned Gradient The problem of finding a scalar function $\psi_h \in W^0_{r+1}$ such that $\mathbf{grad}\,\psi_h = \mathbf{f}_h$ with $\mathbf{f}_h \in W^1_{r+1}$ known, has not a unique solution: indeed, $\tilde{\psi}_h = \psi_h + c$, $c \in \mathbb{R}$, verifies $\mathbf{grad}\,\tilde{\psi}_h = \mathbf{f}_h$ too. However, it is enough to fix the value of ψ_h at one of the vertices, say n_1, to ensure uniqueness. With conditions on $\mathbf{f}_h$ stated in [7] and recalled in Fig. 2 (left), we consider the *grad problem*:

$$\text{Given } \mathbf{f}_h \in W^1_{r+1} \text{ s.t. } \mathbf{curl}\,\mathbf{f}_h = \mathbf{0},\ \oint_{\sigma_i} \mathbf{f}_h \cdot d\mathbf{s} = 0,\ \sigma_i \in H_1(\Omega),\ i = 1, \dots, g,$$

$$\text{find } \psi_h \in W^0_{r+1} \text{ s.t. } \mathbf{grad}\,\psi_h = \mathbf{f}_h \text{ in } \Omega \text{ and } \psi_h(n_*) = 0. \tag{1}$$

The fundamental theorem of calculus says that

$$\psi_h(n_b) - \psi_h(n_a) = \int_e \mathbf{grad}\,\psi_h \cdot \mathbf{t}_e = \int_e \mathbf{f}_h \cdot \mathbf{t}_e \tag{2}$$

for an edge $e = [n_a, n_b] \in E$. Equation (2) contains two unknowns, namely $\psi_h(n_a)$, $\psi_h(n_b)$. Starting from the equation with $n_a = n_*$, the root of the spanning tree S_h, where we have set $\psi_h(n_*) = 0$, we can compute the remaining value, say $\psi_h(n_b)$, as $\psi_h(n_b) = \psi_h(n_*) + \int_e \mathbf{f}_h \cdot \mathbf{t}_e$ for $e = [n_*, n_b] \in S_h$. At this point, n_b becomes root: the value of $\psi_h(n_b)$ is known and it can be used to compute the values of ψ_h at the remaining nodes in a neighborhood of n_b. Since S_h is a spanning tree, proceeding in this way (see the root-to-leaves algorithm in Fig. 5) we can visit all the nodes of τ_h. The spanning tree S_h is a tool for selecting the rows of the system equivalent to $\mathbf{grad}\,\psi_h = \mathbf{f}_h$ for which, using $\psi_h(n_*) = 0$, one can eliminate the unknowns one after the other. We have thus found a nodal function ψ_h such that its gradient has line

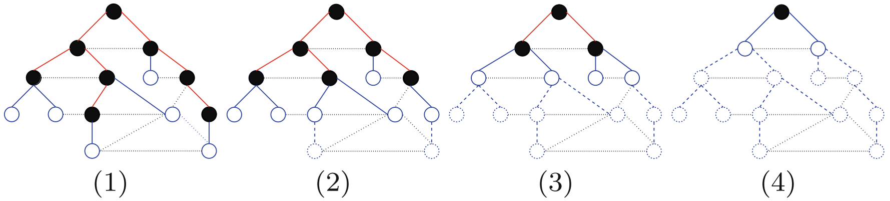


Fig. 5 Example of a 4-step *root-to-leaves algorithm* (4) → (1), where • are known values and ∘ unknown ones, that is used to solve the gradient problem. Example of a 4-step *leaves-to-root algorithm* (1) → (4), where arcs connecting a ∘ to a • denote unknown values and ∘ known ones, that is exploited to solve the divergence problem

integral on all the (small) edges $e \in S_h$ equal to the line integral on e of $\mathbf{f}_h$. Let us consider the edges $e \notin S_h$ (one of the 1-boundaries in Fig. 4). For each node $\bar{n} \neq n_*$, let $C_{\bar{n}}$ be the set of edges in S_h joining n_* to $\bar{n}$: then $\int_{C_{\bar{n}}} \mathbf{grad}\, \psi_h \cdot d\mathbf{s} = \psi_h(\bar{n}) - \psi_h(n_*)$. Given an edge $e = [n_a, n_b] \notin S_h$, we define the cycle $\sigma_e = C_{n_a} + e - C_{n_b}$. Since $\mathbf{f}_h$ is a gradient (it is curl-free and its line integral on all the loops $\sigma_j \in H_1(\Omega)$ vanishes), its line integral on σ_e vanishes too. Indeed, σ_e is either homotopic to 0 or is in $H_1(\Omega)$. Therefore,

$$0 = \oint_{\sigma_e} \mathbf{f}_h \cdot d\mathbf{s} = \psi_h(n_a) + \int_e \mathbf{f}_h \cdot \mathbf{t}_e - \psi_h(n_b) = \int_e \mathbf{f}_h \cdot \mathbf{t}_e - \int_e \mathbf{grad}\, \psi_h \cdot \mathbf{t}_e .$$

This yields $\mathbf{grad}\, \psi_h = \mathbf{f}_h$ also on $e \notin S_h$.

Constructing a Vector with Assigned Curl With conditions on $\mathbf{u}_h$ stated in [7] and recalled in Fig. 2 (left), we consider the *curl problem*:

$$\begin{aligned} &\text{Given}\, \mathbf{u}_h \in W_r^2 \text{ s.t. } \operatorname{div} \mathbf{u}_h = \mathbf{0} \text{ and } \int_{(\partial\Omega)_j} \mathbf{u}_h \cdot \mathbf{n} = 0,\ \forall\ j = 1, \dots, p, \\ &\text{find}\, \mathbf{z}_h \in W_r^1 \text{ s.t. } \mathbf{curl}\, \mathbf{z}_h = \mathbf{u}_h \text{ in } \Omega\ , \\ &\int_e \mathbf{z}_h \cdot \mathbf{t}_e = 0,\ \forall\ e \in S_h,\ \text{and} \oint_{\sigma_i} \mathbf{z}_h \cdot \mathbf{t}_i = 0,\ \ \sigma_i \in H_1(\Omega),\ i = 1, \dots, g. \end{aligned} \tag{3}$$

Concerning the conditions in the last line of (3), noting that the number of small edges $e \in S_h$ is $d_L - 1$, the first part on S_h can be seen as a filter for gradients. On the other hand, homology and cohomology are in duality, hence the last part on $H_1(\Omega)$ can be seen as a filter for cohomology (harmonic) fields. In matrix form, the *curl problem* reads $R\mathbf{Z} = \mathbf{U}$, with suitable conditions on $\mathbf{U}$. Indeed, if $D\mathbf{U} \neq \mathbf{0}$ the problem has no solution. If $\partial\Omega$ is connected the problem has a solution if and only if $D\mathbf{U} = 0$. If $\partial\Omega$ is not connected the problem has a solution if and only if $D\mathbf{U} = \mathbf{0}$ and $M\mathbf{U} = \mathbf{0}$ where the matrix $M \in \mathbb{R}^{p \times d_{RT}}$, with entries $M_{\ell j}$ equal to 1 or 0 depending if the face f_j is on $(\partial\Omega)_\ell$ or not. Let us set $\tilde{D} = \left[\begin{smallmatrix} D \\ M \end{smallmatrix}\right]$. We have that $\tilde{D} = D$ for $p = 0$. We recall that $G \in \mathbb{R}^{d_N \times d_L}$ and that $\dim \operatorname{Ker} G = 1$ if Ω is connected otherwise $\dim \operatorname{Ker} G = m$, where $m = \dim H_0$ (the number of connected components of Ω). Here we assume $m = 1$.

The solution of $R\mathbf{Z} = \mathbf{U}$ is not unique. Indeed, if Ω is simply connected then $\operatorname{Ker} R = \operatorname{Im} G$ therefore $R\,G\boldsymbol{\varphi} = \mathbf{0}$ for all $\boldsymbol{\varphi} \in \mathbb{R}^{d_L}$. When Ω is not simply connected, $R\boldsymbol{\rho}_h = \mathbf{0}$ for all $\boldsymbol{\rho}_h \in H_h^1$ where $H_h^1 = \{\boldsymbol{\rho}_h \in W_{r+1}^1 : \boldsymbol{\rho}_h \in \ker R,\ \boldsymbol{\rho}_h \notin \operatorname{Im} G\} \neq \emptyset$. Let G_r be a submatrix of G belonging to $\mathbb{R}^{d_N \times (d_L - 1)}$ such that $\operatorname{Im} G = \operatorname{Im} G_r$ and the columns of G_r are a basis of $\operatorname{Im} G_r$. Passing from G to G_r is equivalent to fix a small node as root for the tree S_h.

We suppose to know a basis $\{\boldsymbol{\rho}_{h,i}\}_{i=1,\ldots g}$ of H_h^1, whose weights on the edges are collected in the columns of a matrix $N \in \mathbb{R}^{n_E \times g}$. To have uniqueness of the solution of $R\,\mathbf{Z} = \mathbf{U}$ we have to find $\mathbf{Z}$ both in $(\operatorname{Im} G_r)^{\perp}$ (so, $G_r^{\top}\mathbf{Z} = \mathbf{0}$) and in $(H_h^1)^{\perp}$ (so, $N^{\top}\mathbf{Z} = \mathbf{0}$). We thus introduce the notation $\tilde{G}_r = [G_r, N]$, knowing that $\tilde{G}_r = G_r$ when Ω is simply connected. Note that $\tilde{G}_r$ has maximal rank $d_L - 1 + g$ ($\leq d_N$). The matrix

$$\begin{bmatrix} R & \tilde{D}^{\top} \\ \tilde{G}_r^{\top} & 0 \end{bmatrix}$$

has $d_{RT} + d_L - 1 + g$ rows and $d_N + d_P + p$ columns. It is thus square since $d_L - d_N + d_{RT} - d_P = 1 - g + p$ for Proposition 1.

Proposition 2 *The linear system*

$$\begin{bmatrix} R & \tilde{D}^T \\ \tilde{G}_r^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Z} \\ \Lambda \end{bmatrix} = \begin{bmatrix} \boldsymbol{U} \\ \boldsymbol{0} \end{bmatrix}$$

has a unique solution and, if $\tilde{\boldsymbol{D}}\boldsymbol{U} = 0$ then $\boldsymbol{R}\mathbf{Z} = \boldsymbol{U}$.

Proof Matrix $\tilde{D}$ has maximal rank, so $\operatorname{Ker}\tilde{D}^T = (\operatorname{Im} \tilde{D})^{\perp} = \{\mathbf{0}\}$ and

$$\begin{bmatrix} R & \tilde{D}^T \\ \tilde{G}_r^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Z} \\ \Lambda \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix} \Rightarrow R\mathbf{Z} = -\tilde{D}^T \Lambda \text{ and } \tilde{G}_r^T \mathbf{Z} = \mathbf{0}$$

$$R\mathbf{Z} = -\tilde{D}^T \Lambda \Rightarrow \mathbf{Z}^T R^T R\mathbf{Z} = -\,\mathbf{Z}^T R^T \tilde{D}^T \Lambda = 0 \Rightarrow R\mathbf{Z} = \mathbf{0}\,,$$

because $R^T \tilde{D}^T = (\tilde{D}\,R)^T$ and $\tilde{D}R = 0$ since $DR = 0$ (if $p > 0$, also $MR = 0$). This yields

$$R\mathbf{Z} = \mathbf{0} \Rightarrow \tilde{D}^T \Lambda = \mathbf{0} \Rightarrow \Lambda = \mathbf{0}.$$

Then

$$\tilde{G}_r^T \mathbf{Z} = \mathbf{0} \Rightarrow \mathbf{Z} \in \operatorname{Ker} \tilde{G}_r^T = (\operatorname{Im} \tilde{G}_r)^{\perp}.$$

$$R\mathbf{Z} = \mathbf{0} \text{ and } \mathbf{Z} \in (\operatorname{Im} \tilde{G}_r)^{\perp} \Rightarrow \mathbf{Z} = \mathbf{0}.$$

If $\tilde{D}\mathbf{U} = 0$ then $\tilde{D}(R\mathbf{Z} + \tilde{D}^T \Lambda) = \tilde{D}\mathbf{U} = 0$ so $\tilde{D}\tilde{D}^T \Lambda = \mathbf{0}$ thus $\tilde{D}^T \Lambda = \mathbf{0}$ hence $R\mathbf{Z} = \mathbf{U}$. This ends the proof. □

A spanning tree $\mathcal{S}$ of the graph $\mathcal{G}_{GT}$ has $d_L - 1$ arcs that correspond to $d_L - 1$ columns of G_r^T. We thus write $G_r^T = [G_t^T, G_c^T]$ with G_t^T invertible. A belted tree $\mathcal{S}_{bt}$ of the graph $\mathcal{G}_{GT}$ has $d_L - 1 + g$ arcs that correspond to $d_L - 1 + g$ columns of $\tilde{G}_r^T$. We thus write $\tilde{G}_r^T = [\tilde{G}_{bt}^T, \tilde{G}_{cbt}^T]$ with $\tilde{G}_{bt}^T$ invertible. A spanning tree $\mathcal{S}'$ of the graph $\mathcal{G}_{D_e}$ has d_P arcs that correspond to d_P columns of D, namely d_P rows of D^T. We thus set $\tilde{D} = [\tilde{D}_{t'}, \tilde{D}_{c'}]$ with $\tilde{D}_{t'}$ invertible. The curl problem in matrix form reads: given $\mathbf{U}_{t'}$, $\mathbf{U}_{c'}$, find $\mathbf{Z}_t$, $\mathbf{Z}_c$ and $\boldsymbol{\Lambda}$ such that

$$\begin{bmatrix} R_{t'\,bt} & R_{t'\,cbt} & \tilde{D}_{t'}^T \\ R_{c'\,bt} & R_{c'\,cbt} & \tilde{D}_{c'}^T \\ \tilde{G}_{bt}^T & \tilde{G}_{cbt}^T & 0 \end{bmatrix} \begin{bmatrix} \mathbf{Z}_t \\ \mathbf{Z}_c \\ \boldsymbol{\Lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{t'} \\ \mathbf{U}_{c'} \\ \mathbf{0} \end{bmatrix}.$$

Since the belted tree can be constructed starting from the edges lying on the loops generating H_1, we can consider a system as the one in Prop. 3, with the last block of lines replaced by the identity and zeros, as here below.

Proposition 3 *The linear system*

$$\begin{bmatrix} R_{t'\,bt} & R_{t'\,cbt} & \tilde{D}_{t'}^T \\ R_{c'\,bt} & R_{c'\,cbt} & \tilde{D}_{c'}^T \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{Z}}_{bt} \\ \widehat{\mathbf{Z}}_{cbt} \\ \widehat{\boldsymbol{\Lambda}} \end{bmatrix} = \begin{bmatrix} \boldsymbol{U}_{t'} \\ \boldsymbol{U}_{c'} \\ \boldsymbol{0} \end{bmatrix} \tag{4}$$

has a unique solution. If $\tilde{D}\boldsymbol{U} = \boldsymbol{0}$ then $\widehat{\boldsymbol{\Lambda}} = \boldsymbol{0}$ and $R\widehat{\mathbf{Z}} = \boldsymbol{U}$. In particular, if $\tilde{D}\boldsymbol{U} = 0$ then $\widehat{\mathbf{Z}}_{cbt} = R_{c'\,cbt}^{-1}\boldsymbol{U}_{c'}$.

Proof If we set

$$\begin{bmatrix} R_{t'\,bt} & R_{t'\,cbt} & \tilde{D}_{t'}^T \\ R_{c'\,bt} & R_{c'\,cbt} & \tilde{D}_{c'}^T \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{Z}}_{bt} \\ \widehat{\mathbf{Z}}_{cbt} \\ \widehat{\boldsymbol{\Lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix}$$

then

$$R\widehat{\mathbf{Z}} = -\tilde{D}^T \widehat{\boldsymbol{\Lambda}} \Rightarrow \mathbf{0} = \tilde{D}R\widehat{\mathbf{Z}} = -\tilde{D}\tilde{D}^T \widehat{\boldsymbol{\Lambda}} \Rightarrow \tilde{D}^T \widehat{\boldsymbol{\Lambda}} = \mathbf{0}.$$

Since $\tilde{D}$ is full rank, it follows that $\widehat{\boldsymbol{\Lambda}} = \mathbf{0}$ and $R\widehat{\mathbf{Z}} = \mathbf{0}$. So, $\widehat{\mathbf{Z}} = G_r\boldsymbol{\varphi}_r + \boldsymbol{\rho}$ with $\boldsymbol{\rho} \in H_h^1$. The last condition on $\mathbf{z}_h$ listed in the last line of (3), yields $\boldsymbol{\rho} = \mathbf{0}$ and thus $\widehat{\mathbf{Z}} = G_r\boldsymbol{\varphi}_r$. In particular $\mathbf{0} = \widehat{\mathbf{Z}}_t = G_t\boldsymbol{\varphi}_r$, hence $\boldsymbol{\varphi}_r = \mathbf{0}$ because G_t is invertible. So we have $\widehat{\mathbf{Z}} = G_r\boldsymbol{\varphi}_r = \mathbf{0}$. Hence the matrix in the linear system (4) is not singular and the linear system has a unique solution.
If $\tilde{D}\mathbf{U} = 0$ then $R\widehat{\mathbf{Z}} + \tilde{D}^T\widehat{\boldsymbol{\Lambda}} = \mathbf{U} \Rightarrow \tilde{D}\tilde{D}^T\widehat{\boldsymbol{\Lambda}} = \mathbf{0} \Rightarrow \widehat{\boldsymbol{\Lambda}} = \mathbf{0} \Rightarrow R\widehat{\mathbf{Z}} = \mathbf{U}$.

To conclude the proof note that $R_{c'\,cbt}$ is square since $d_{RT} - d_P - p = d_N - (d_L - 1) - g$ where the second part of the identity is the number of the edges out of the belted tree (block cbt). To prove that $R_{c'\,cbt}$ is invertible we will see that for each $\mathbf{U}_{c'} \in \mathbb{R}^{d_{RT}-d_P-p}$ the linear system $R_{c'\,cbt}\widehat{\mathbf{Z}}_{cbt} = \mathbf{U}_{c'}$ has a solution. In fact, let us set $\mathbf{U}_{t'} = -D_{t'}^{-1}D_{c'}\mathbf{U}_{c'}$. Then the linear system

$$\begin{bmatrix} R_{t'\,bt} & R_{t'\,cbt} & D_{t'}^T \\ R_{c'\,bt} & R_{c'\,cbt} & D_{c'}^T \\ I & 0 & 0 \end{bmatrix} \begin{bmatrix} \widehat{\mathbf{Z}}_{bt} \\ \widehat{\mathbf{Z}}_{cbt} \\ \widehat{\boldsymbol{\Lambda}} \end{bmatrix} = \begin{bmatrix} \mathbf{U}_{t'} \\ \mathbf{U}_{c'} \\ \mathbf{0} \end{bmatrix}$$

has a unique solution and $\widehat{\mathbf{Z}}_{bt} = \mathbf{0}$. Since, by construction, $\tilde{D}\,\mathbf{U} = \mathbf{0}$ then $\widehat{\boldsymbol{\Lambda}} = \mathbf{0}$, hence $R_{c'\,cbt}\widehat{\mathbf{Z}}_{cbt} = \mathbf{U}_{c'}$. □

Constructing a Vector with Assigned Divergence The problem of finding a vector function $\mathbf{v}_h \in W_r^2$ such that $\operatorname{div}\mathbf{v}_h = w_h$ with $w_h \in W_r^3$ known, has not a unique solution: indeed, $\tilde{\mathbf{v}}_h = \mathbf{v}_h + \mathbf{curl}\,\mathbf{z}$, $\mathbf{z} \in W_r^1$, verifies $\operatorname{div}\tilde{\mathbf{v}}_h = w_h$ too. It is however enough to fix the value $\int_f \mathbf{v}_h \cdot \mathbf{n}_f = 0$ at all the active small faces $f \notin S_h^*$ and $\int_{(\partial\Omega)_j} \mathbf{v}_h \cdot \mathbf{n}_{\partial\Omega} = 0$ on each connected component $(\partial\Omega)_i$ of $\partial\Omega$, $j = 1, p$, in order to set a filter for the cohomological (harmonic) fields. We thus have the following *divergence problem*:

$$\text{Given } w_h \in W_r^3, \text{ find } \mathbf{v}_h \in W_{r+1}^2 \text{ s.t. } \operatorname{div}\mathbf{v}_h = w_h \text{ in } \Omega,$$
$$\text{with } \int_f \mathbf{v}_h \cdot \mathbf{n}_f = 0,\ \forall f \notin S_h^*,\ \text{and } \int_{(\partial\Omega)_j} \mathbf{v}_h \cdot \mathbf{n}_{\partial\Omega} = 0,\ \forall j. \quad (5)$$

It is known that $\dim \operatorname{Im}\mathbf{curl} = d_N - \dim \operatorname{Ker}\mathbf{curl}$ therefore $\dim \operatorname{Im}\mathbf{curl} = d_N - (d_L - 1 + g)$ and for Prop. 1 we have $\dim \operatorname{Im}\mathbf{curl} = d_{RT} - d_P - p$. The system associated with (5) is square, in d_{RT} unknowns and equations. Indeed, $\operatorname{div}\mathbf{v}_h = w_h$ counts d_P equations (one for each tetrahedron) and the conditions, in the last line of (5), are, respectively, $d_{RT} - d_P - p$, as many the arcs in the dual co-tree (since the arcs in S_h^* are $d_P + p$) and p, as many as the number of $(\partial\Omega)_j$. We use the leaves-to-root algorithm presented in Fig. 5 (see [2] for a similar algorithm when using moments as degrees of freedom). Indeed, given Σ a small tetrahedron or a connected component of $\partial\Omega$, we denote by $F(\Sigma) = \{f \in F,\ f \in \partial\Sigma\}$, the set of active small faces in F that are on the boundary of Σ (arcs in the dual graph that connect to the point Σ). The *leaves* of the dual tree are tetrahedra that have only one face in S_h^*. If Σ is a leave of S_h^* and $f(\Sigma)$, with external unit normal $\mathbf{n}$, is the unique dual arc (face) in S_h^* incident to Σ, we can compute the dof on $f(\Sigma)$. In fact,

$$\int_{f(\Sigma)} \mathbf{v}_h \cdot \mathbf{n} = \begin{cases} \int_{\partial\Sigma} \mathbf{v}_h \cdot \mathbf{n} = \int_\Sigma w_h, & \text{if } \Sigma \in T \\ \int_{(\partial\Omega)_j} \mathbf{v}_h \cdot \mathbf{n} = 0, & \text{if } \Sigma = (\partial\Omega)_j \\ \int_{(\partial\Omega)_0} \mathbf{v}_h \cdot \mathbf{n} = \int_\Omega w_h, & \text{if } \Sigma = (\partial\Omega)_0. \end{cases}$$

by relying on the divergence theorem in the first and last identities. It is clear that if $\mathbf{v}_h \in W_{r+1}^2$ is such that $\operatorname{div} \mathbf{v}_h = 0$, $c_j = 0$ for all $j = 1, \ldots, p$ and $\int_f \mathbf{v}_h \cdot \mathbf{n}_f = 0$, for all $f \notin S_h^*$, then $\int_{f(\Sigma)} \mathbf{v}_h \cdot \mathbf{n}_f = 0$, for all $f(\Sigma)$ that are leaves of S_h^*. We can iterate as in the leaves-to-root algorithm: from (1) to (2) in Fig. 5, we remove from S_h^* the leaves Σ and the incident arcs $f(\Sigma)$ that we have used in (1), the remaining graph in (2) is still a tree. The arcs of this new tree are the faces where dofs are still unknown. We can thus repeat the previous procedure. After a finite number of steps, the tree reduces to one node (root), and we have obtained $\int_f \mathbf{v}_h \cdot \mathbf{n}_f = 0$, for all $f \in F$. Since the problem yields a square system, this proves that the solution is unique.

Acknowledgement This work was partially supported by PRIN's project NA-FROM-PDEs 201752HKH8.

References

1. R. Abraham, J. Marsden, T. Ratiu, *Manifolds, Tensor Analysis, and Applications*, Applied Mathematical Sciences **75**, Springer (1988).
2. A. Alonso Rodríguez, J. Camaño, E. De Los Santos, F. Rapetti, *A graph approach for the construction of high order divergence-free Raviart-Thomas finite elements*, Calcolo **55**:42, 2018.
3. A. Alonso Rodríguez, F. Rapetti, *Small trees for high order Whitney elements*, in "Spectral and High Order Methods for PDEs", S.J. Sherwin *et al.* eds., Icosahom 2018 procs., LNCSE Vol. 134, Springer-Verlag, 2020.
4. A. Alonso Rodríguez, A. Valli, *Finite element potentials*, Appl. Numer. Math., **95** (2015) 2–14.
5. C. Amrouche, C. Bernardi, M. Dauge, V. Girault, *Vector potentials in three-dimensional nonsmooth domains*, Math. Methods Appl. Sci., **21** (1998) 823–864.
6. D.N. Arnold, R.S. Falk, and R. Winther, *Finite element exterior calculus, homological techniques, and applications*, Acta Numer. **15** (2006) 1–155.
7. A. Bossavit, *Magnetostatic problems in multiply connected regions: some properties of the curl operator*, Phys. Sci., Meas. and Instr., Management and Education - Reviews, IEE Procs. **A, 135** (1988) 179–187.
8. A. Bossavit, *Computational electromagnetism*, Academic Press, Inc., San Diego, CA, 1998.
9. J. Cantarella, D. De Turck, H. Gluck, *Vector calculus and the topology of domains in 3-space*, Amer. Math. Monthly **109** (2002) 409–442.
10. J.R. Munkres, *Elements of algebraic topology*, Perseus Books, Cambridge, MA, 1984.
11. H. Helmohltz, *Über integrale der hydrodynamischen gleichungen welch den wirbelbewegungen*, J. Reine Agew. Math. **55** (1858) 25–55.
12. R. Hiptmair, J. Ostrowski, *Generators of $H_1(\Gamma_h, \mathbb{Z})$ for Triangulated Surfaces: Construction and Classification*, SIAM J. Comput. **31/5** (2002) 1405–1523.

13. W.V.D. Hodge, *The theory and applications of harmonic integrals*, Cambridge Univ. Press (1941).
14. O.A. Ladyzhenskaya, *The Mathematical Theory of Viscous Incompressible Flow*, Gordon and Breach (1963).
15. F. Rapetti, F. Dubois, A. Bossavit, *Discrete vector potentials for non-simply connected three-dimensional domains*, SIAM J. on Numer. Anal. **41/4** (2003) 1505–1527.
16. F. Rapetti, A. Bossavit, *Whitney forms of higher degree*, SIAM J. Numer. Anal. **47** (2009) 2369–2386.
17. J. Stillwell, *Classical topology and Combinatorial Group Theory*, Graduate Texts in Maths. **72**, Springer (1993).
18. K. Thulasiraman, M.N.S. Swamy, *Graphs: theory and algorithms*, A Wiley-Interscience Publication, John Wiley & Sons, Inc., New York (1992).
19. F.W.Warner, *Foundations of differentiable manifolds and Lie groups*, Graduate Texts in Maths. **94**, Springer (1983).
20. A. Weil, *Sur les théorèmes de de Rham*, in Commentarii Mathematici Helvetici **26** (1952) 119–145.
21. H. Whitney, *Geometric integration theory*, Princeton University Press, Princeton, N. J., 1957.

The Candy Wrapper Problem: A Temporal Multiscale Approach for PDE/PDE Systems

Thomas Richter and Jeremi Mizerski

Abstract We discuss the application of a multiscale scheme to a medical flow problem, the so called Candy Wrapper problem. This problem describes the restenosis of a stented blood vessel, which will take several months but which is governed by the rapidly oscillating dynamics of the blood flow. A long term simulation of this three dimensional free-boundary flow problem resolving the fast dynamics is not feasible. Our multiscale approach which has been recently published is based on capturing the fast dynamics by locally isolated periodic-in-time problems which have to be approximated once in each macro step of the long term process. Numerical results show the accuracy and efficiency of this multiscale approach.

1 Introduction: The Candy Wrapper Problem

The idea of opening or dilating occluded or narrowed coronary artery originates in the works of Andreas Gruentzig. First human application of percutaneous transluminal coronary angioplasty (PTCA) had been performed on September 16th 1977 at University Hospital in Zurich. The method was basically just putting the balloon catheter through narrowing and inflating it [24]. The immediate results were good, only about 1% of the patients suffered from immediate vessel closure and myocardial infarct. Later after the interventions 30% of the stenosis recurred accompanied by the symptoms of angina of the intensity close to those from before the intervention. That happened usually from 30 days to 6 months form the intervention [9]. At that time the cardiologists were convinced that only about 10%

T. Richter (✉)
University of Magdeburg, Institute for Analysis and Numerics, Magdeburg, Germany
e-mail: thomas.richter@ovgu.de

J. Mizerski
University of Magdeburg, Institute for Optimization, Magdeburg, Germany
e-mail: jeremi.mizerski@ovgu.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_2

of all the patients will be suitable for the method and the rest of coronary artery disease cases had to be referred to cardiac surgery for by-pass grafting. The remedy for the situation was to place stents intended as an internal scaffold for the artery to maintain it's patency. The method was introduced in 1986 with some success [15]. Soon after that a new set of complications came into the attention. The early and late onset of thrombosis started to haunt the patients undergoing procedures of bare metal stent (BMS) implantation. The BMS coped also with the problem of intimal hypertrophy which resulted in in-stent stenosis. From that moment on the era of drug eluting stents (DES) begins. Throughout the 90s different companies try different chemical compounds. The first successful application was reported by Serruys in 1998 [14]. That however did not solve the problem entirely and resulted in even more complex set of complications [1, 34]. The platelet dependent thrombosis resulted in explosion of anti-platelet drug development in following years. The problem defined as a "restenosis of treatment margins" or "candy wrapper" phenomenon was described by radiologists trying to apply the oncological brachytherapy principles to the neointimal overgrowth inside BMS [12]. Soon after that the molecular bases of the process started to be extensively studied [8]. The issue of stent edge stenosis had not been resolved by introduction of new materials and coatings [10, 11]. The biological effects of flow properties have been studied extensively since the introduction of extracorporeal circulatory system in early 50s. The body of evidence built on that experience showed large interdependencies between the local flow properties and the tissue response. The research areas branched towards optimization of stent struts geometry [31] and usage of different cytostatic drugs as a stent coating material [23]. The key elements of the milieu created by stents are usually considered separately. Some computational models allow to recreate and integrate more elements into the system [30, 41]. By means of computer simulations the researchers were able to simulate not only fluid dynamics around the stented area but also the effects of drug diffusion into the arterial walls [3, 44]. The edge restenosis phenomenon however did not find its' conclusive description. To fully understand that complex phenomenon we need to take the arterial wall mechanics and fluid-structure interactions into consideration. The specific challenge that is tackled in this work is the temporal multiscale character of this problem: While restenosis occurs after months, the driving mechanical forces come from the pulsating blood flow that requires a resolution in the order of centiseconds. Direct simulations of this long-term process are not feasible and we present temporal multiscale methods aiming efficient predictions.

2 Model Configuration

In this section we will briefly describe the mathematical model used to describe the stenosis growth effects. Medical, biological and chemical processes are strongly simplified. They do however still contain the specific couplings and scales that are characteristic for the underlying problem. We choose problem parameters as close to the medical configuration as possible and as known, which is an issue since good data is difficult to measure and only sparsely available.

The most important simplification in our present computational model is the assumption of a rigid vessel wall. Although deformations by dynamical fluid-structure interactions are small it is well known that the effects of elasticity should be taken into effect for an appropriate depiction of wall stresses, which are an essential ingredient in triggering stenosis growth. However, we give an outlook on techniques that are suitable to substantially increase the efficiency in medical fluid-structure interaction simulation that suffer from special instabilities by the added-mass effect due to similar masses of fluid and solid [7].

2.1 Governing Equations

We consider a system of partial differential equations that is inspired by Yang et al. [42, 43], where a model describing the interaction of mechanical fluid-structure interactions with bio/chemical reactions and active growth and material deformation is introduced. The mechanical system is described by a nonlinear fluid-structure interaction model, where the blood is modeled as incompressible Newtonian fluid, which is an adequate choice for the vessel sizes under consideration

$$\rho_f\big(\partial_t \mathbf{v} + (\mathbf{v}\cdot\nabla)\mathbf{v}\big) - \operatorname{div}\boldsymbol{\sigma}(\mathbf{v},p) = 0, \quad \operatorname{div}\mathbf{v} = 0 \text{ in } \mathcal{F}(t), \tag{1}$$

where $\mathcal{F}(t)$ is the (moving) fluid domain, the lumen, $\rho_f \approx 1.06\,\text{gcm}^{-3}$ the density of blood and $\boldsymbol{\sigma}(\mathbf{v},p) = \rho_f \nu_f(\nabla\mathbf{v}+\nabla\mathbf{v}^T) - pI$ the Cauchy stress tensor, depending on velocity $\mathbf{v}$ and pressure p, with the kinematic viscosity $\nu_f \approx 0.03\,\text{cm}^2\text{s}^{-1}$. The vessel walls are governed by an elastic material

$$J\rho_s\partial_t\mathbf{v} - \operatorname{div}\Big(\mathbf{F}\boldsymbol{\Sigma}\Big) = 0, \quad \mathbf{v} = \partial_t\mathbf{u} \text{ in } \mathcal{S}, \tag{2}$$

where ρ_s is the fluid's density (in current configuration), $\mathbf{v}$ the velocity, $\mathbf{u}$ the deformation, $\mathbf{F} := I + \nabla\mathbf{u}$ the deformation gradient with determinant $J := \det \mathbf{F}$. By $\mathcal{S}$ we denote the Lagrangian reference configuration. By $\boldsymbol{\Sigma}$ we denote the Piola Kirchhoff stresses. The proper modeling of the stresses within vessel walls is under active research [27]. In particular there is still little knowledge on the degree of complexity that is required for accurately predicting the behavior of the coupled system. To incorporate growth of the stenosis in the context of fluid-structure interactions, the technique of a multiplicative decomposition of the deformation gradient

$$\mathbf{F} = \mathbf{F}_e\mathbf{F}_g(c), \quad \mathbf{F}_e = \mathbf{F}\mathbf{F}_g(c)^{-1}, \quad V_0 \xrightarrow{\mathbf{F}_g} V_g \xrightarrow{\mathbf{F}_e} V$$

into active deformation $\mathbf{F}_g(c)$ and elastic response $\mathbf{F}_e$ can be applied, see [20, 40]. The idea is to introduce an intermediate configuration that includes the growth $\hat{S} \to \hat{S}_g$ and that is mediated by $\mathbf{F}_g(c)$ depending directly on the exterior growth trigger c but that is not physical, i.e. it is stress free but not necessarily free of strain. The stresses then depend on the elastic part only, to be precise on $\mathbf{F}_e = \mathbf{F}_g(c)^{-1}(I + \nabla \mathbf{u})$. Such models are successfully used in describing the formation of plaques [42, 43].

In this work we considerably simplify the model by neglecting all elastic effects. The Navier-Stokes equations are solved in the domain $\mathcal{F}$ that directly depends on a growth variable c by prescribing normal growth

$$\partial\mathcal{F}\big(c(t)\big) = \{\mathbf{x} - c(\mathbf{x}, t) \cdot \mathbf{n}_{\hat{\mathcal{F}}}(\mathbf{x}) \;:\; \mathbf{x} \in \partial\hat{\mathcal{F}}\},$$

where $\hat{\mathcal{F}}$ is the non-grown fluid domain in reference state and $\mathbf{n}_{\hat{\mathcal{F}}}$ the outward facing unit normal vector. The description of the coupled problem we will be based on an ALE formulation, where all quantities are given on the undeformed reference domain $\hat{\mathcal{F}}$, see [36, Chapter 5]. This reference domain is a straight pipe of length 7 cm and diameter 0.2 cm. A typical curvature, irregularities, the effect of the stent and in particular of the stenosis will be augmented by the ALE deformation $T(t) : \hat{\mathcal{F}} \to \mathcal{F}(t)$.

The growth variable c will live on the surface $\partial\hat{\mathcal{F}}$. The evolution of c is governed by a simple surface diffusion equation

$$d_t c - \lambda_c \Delta_\Gamma c = R(c, \boldsymbol{\sigma}) \text{ on } \partial\hat{\mathcal{F}}, \quad c(0) = 0 \tag{3}$$

with the Laplace Beltrami operator Δ_Γ and a small diffusion constant $\lambda_c \approx 5 \cdot 10^{-7}\,\mathrm{m}^2/\mathrm{s}$. In a detailed model, this simple reaction diffusion equation is replaced by a cascade of chemical reaction systems that trigger growth, see [42]. Due to the very slow evolution of the plaque, the motion of the evolving surface can be neglected in the temporal derivative. There is no experimental data on the role of diffusion and the size of λ_c. We will hence consider λ_c as a procedure for stabilization and choose is small enough to cancel any effects on the macroscopic evolution of the growth. In lack of relevant parameters equation (3) can be considered to be dimensionless. By $R(c, \boldsymbol{\sigma})$ we denote the coupling term triggering growth of the stenosis

$$R(c, \boldsymbol{\sigma}; \mathbf{x}) = \frac{\alpha}{1 + \beta c(\mathbf{x})} \gamma\big(\sigma_{WSS}\big(\boldsymbol{\sigma}(\mathbf{x}); \mathbf{x}\big)\big). \tag{4}$$

The parameter α controls the rate of the stenosis growth and it can be considered as the scale parameter separating the fast scale of the fluid problem from the slow scale of the growth, by β we control some saturation of the growth. By σ_{WSS} we denote the wall shear stress that is acting close to the tips of the stent at s_0 and s_1 (in direction of the main flow direction $\mathbf{x}_1$, where injuring of the vessel wall will trigger stenosis growth

$$\sigma_{WSS}(\boldsymbol{\sigma};\mathbf{x}) = |\boldsymbol{\sigma}(\mathbf{x})\mathbf{n}(\mathbf{x})\cdot\mathbf{n}(\mathbf{x})|\Big(\Theta(s_0;\mathbf{x}_1)+\Theta(s_1;\mathbf{x}_2)\Big),$$

with

$$\Theta(s;x) = \Big(1+\exp\big(2(s_0-1-x)\big)\Big)^{-1}\Big(1+\exp\big(2(x-s_0-1)\big)\Big)^{-1}.$$

Only wall shear stresses in a certain range above and below activation limits are responsible for plaque growth, hence we introduce the scaling function $\gamma(\cdot)$ as

$$\gamma(S) = \Big(1+\exp\big(3(\sigma_{min}-S)\big)\Big)^{-1}\Big(1+\exp\big(3(S-\sigma_{max})\big)\Big)^{-1}.$$

2.2 Parameters

All computations are carried out on the reference domain, a vessel of diameter 0.2cm and length 7cm. Deformations, imposed by the stent T_{stent}, the general curvature of the configuration $T_{geometry}$ and the stenosis $T_{stenosis}$ are realized by mappings

$$T = T_{geometry}\circ T_{stenosis}\circ T_{stent}.$$

All units are given in cm, g, s.

T_{stent} models the impact of the stent, a slight extension of the vessel at the tips s_l and s_r

$$T_{stent}(x) = \begin{pmatrix}x_1\\0\\0\end{pmatrix} + \Big(1+\rho_{stent}\mathrm{e}^{-\gamma_{stent}(x_1-s_0)^2}+\mathrm{e}^{-\gamma_{stent}(x_1-s_1)^2}\Big)\begin{pmatrix}0\\x_2\\x_3\end{pmatrix} \qquad (5)$$

with $\rho_{stent}=0.1$ and $\gamma_{stent}=50$. Growth of the stenosis is assumed to be in normal direction only. We prescribe $T_{stenosis}$ by the simple relation

$$T_{stenosis}(c;x) = \begin{pmatrix}x_1\\0\\0\end{pmatrix} + \big(1-c(x)\big)\begin{pmatrix}0\\x_2\\x_3\end{pmatrix}.$$

The overall vessel geometry is curved in the x/y plane for $x_1 < s_m = 3.5\,\text{cm}$ which is the left half of the vessel and in the x/z plane for $x_1 > s_m$

$$T_{geo}(x)\Big|_{x_1<s_m} = \begin{pmatrix} x_1 - \tau(x_1)\big(1+\tau(x_1)^2\big)^{-\frac{1}{2}}x_2 \\ \tau'(x_1) + \big(1+\tau(x_1)^2\big)^{-\frac{1}{2}}x_2 \\ x_3 \end{pmatrix},$$

$$T_{geo}(x)\Big|_{x_1>s_m} = \begin{pmatrix} x_1 - \tau(x_1)\big(1+\tau(x_1)^2\big)^{-\frac{1}{2}}x_3 \\ x_2 \\ \tau'(x_1) + \big(1+\tau(x_1)^2\big)^{-\frac{1}{2}}x_3 \end{pmatrix}$$

where $\tau(x_1)$ describes the center-line of the deformed vessel, given by $\tau(x_1) = 4 \cdot 10^{-3}(x_1 - s_m)^4$. The mapping is chosen to give a curvature that is realistic in coronary arteries with a straight middle-section describing the stented area. As further parameters we consider the fluid density $\rho_f = 1.06\,\text{g}\cdot\text{cm}^{-3}$, the viscosity $\nu = 0.03\,\text{cm}^2\cdot\text{s}^{-1}$. The stent starts at $s_0 = 2\,\text{cm}$, extends over 3 cm to $s_1 = 5\,\text{cm}$. The geometric parameters for the impact of the stent, see 5, are $\gamma_{stent} = 50$ and finally, the reaction term uses the limits $\sigma_{min} = 5$ and $\sigma_{max} = 8$.

The flow problem is driven by enforcing a periodic relative pressure profile (inflow to outflow) condition that is inspired from the usual pressure drops in stented coronary arteries suffering from a stenosis. On the inflow boundary Γ_{in} we prescribe the time-periodic average pressure

$$P_{in}(t) = \begin{cases} 10 + 25t & 0 \le t < 0.4\,\text{s} \\ 140/3 - 200t/3 & 0.4\,\text{s} \le t < 0.7\,\text{s} \\ 100t/3 - 70/3 & 0.7\,\text{s} \le t < 1\,\text{s} \end{cases}, \quad \text{periodically extended over } [0,1]$$

2.3 ALE Formulation and Discretization

Based on the mapping $T(x) = T_{geometry}(x) \circ T_{stenosis}(x) \circ T_{stent}(x)$ the Navier-Stokes equations and the surface growth equation are transformed to ALE coordinate, e.g. by introducing reference values $\hat{v}(\hat{x},t) = v(x,t)$, $\hat{p}(\hat{x},t) = p(x,t)$ and $\hat{c}(\hat{x},t) = c(x,t)$. The resulting set of equations is given on the reference domain $\hat{\mathcal{F}}$ and in variational formulation it takes the form

$$\begin{gathered} \Big(J\rho_f\big(\partial_t\hat{\mathbf{v}} + (\hat{\mathbf{F}}^{-1}\hat{\mathbf{v}}\cdot\hat{\nabla})\hat{\mathbf{v}}\big), \phi\Big)_{\hat{\mathcal{F}}} + \Big(J\hat{\boldsymbol{\sigma}}\mathbf{F}^{-T}, \hat{\nabla}\hat{\phi}\Big)_{\hat{\mathcal{F}}} = 0 \\ \Big(J\mathbf{F}^{-1} : \hat{\nabla}\hat{\mathbf{v}}, \xi\Big)_{\hat{\mathcal{F}}} = 0, \qquad \Big(c', \psi\Big)_{\partial\hat{\mathcal{F}}} + \Big(\lambda_c\nabla_\Gamma c, \nabla_\Gamma\psi\Big)_{\partial\hat{\mathcal{F}}} = R(\hat{c}, \hat{\boldsymbol{\sigma}}). \end{gathered} \tag{6}$$

Several simplifications in comparison to an exact ALE formulation have been applied: due to the very slow evolution of the surface we neglect inertia terms by its motion. Further, since surface diffusion will only serve as numerical stabilization we refrain from an exact transformation of the surface Laplace.

The discretization of system (6) is by standard techniques. In time, we use the θ-time stepping method

$$u' = f(t, u) \quad \rightarrow \quad u_n - u_{n-1} = \Delta t\theta f(t_n, u_n) + \Delta t(1-\theta) f(t_{n-1}, u_{n-1}),$$

with constant step sizes Δ and the choice $\theta = \frac{1}{2} + \mathcal{O}(k)$ to achieve second order accuracy with good stability properties, see [32, 38]. Spatial discretization is by means of stabilized equal order tri-quadratic finite elements on a hexahedral mesh. For stabilization of the inf-sup condition and of convective regimes the local projection stabilization is used [4, 5]. The surface PDE is continued into the fluid domain and can be considered as a weakly imposed boundary condition. We refer to [36] for details on the discretization and implementation in Gascoigne 3D [6].

3 Temporal Multiscales

The big challenge of the candy wrapper problem is in the range of temporal scales that must be bridged. While the flow problem is driven by a periodic flow pattern with period 1 s the growth of the stenosis takes months. The growth model comprises the parameter α, see (4) that indicates exactly this scale separation, since $|R(c, \boldsymbol{\sigma})| = \mathcal{O}(\alpha)$. In [19] we have recently introduced and analysed a temporal multiscale scheme for exactly such long-scale/short-scale problems governed by a PDE/ODE system and driven by a periodic-in-time micro process. Here we extend this technique for handling 3d PDE/PDE couplings.

We briefly sketch the layout of the multiscale approximation. To begin with, we identify the growth parameter $c(\mathbf{x}, t)$ as the main variable of interest. Furthermore, as we are interested in the long term behavior of the growth only, we introduce the (locally) averaged growth variable

$$\bar{c}(\mathbf{x}, t) = \int_t^{t+1\,\mathrm{s}} c(\mathbf{x}, s)\,\mathrm{d}s, \tag{7}$$

where the averaging extends over one period only.

Next, to decouple slow and fast scales we make the essential assumption that the flow problem on a fixed domain $\mathcal{F}(\bar{c}_f)$, where $\bar{c}_f := \bar{c}(t_f)$ for one point in time t_f

admits a periodic in time solution

$$\begin{aligned}\Big(J(\bar c_f)\rho_f\big(\partial_t^{\bar c_f}\mathbf{v}^{\bar c_f} + (\mathbf{F}(\bar c_f)^{-1}\mathbf{v}^{\bar c_f}\cdot\nabla)\mathbf{v}^{\bar c_f}\big),\phi\Big)_{\mathcal{F}} \\ + \Big(J(\bar c_f)\boldsymbol{\sigma}^{\bar c_f}\mathbf{F}(\bar c_f)^{-T},\nabla\phi\Big)_{\mathcal{F}} + \Big(J(\bar c_f)\mathbf{F}(\bar c_f)^{-1} : \nabla\mathbf{v}^{\bar c_f},\xi\Big)_{\mathcal{F}} = 0 \\ \mathbf{v}^{\bar c_f}(\cdot,0) = \mathbf{v}^{\bar c_f}(\cdot,1)\end{aligned} \tag{8}$$

Only very few theoretical results exist on periodic solutions to the Navier-Stokes equations, see [21]. They only hold in the case of small data which is not given in the typical candy wrapper configurations with Reynolds numbers going up to about $Re = 1000$. Computational experiments however do suggest the existence of stable limit cycles in the regime of interest.

Multiscale Algorithm Given such periodic solutions, the computational multiscale method is based on a subdivision of $I = [0, T]$ (where $T \approx$ months is large) into macro time-steps t_n for $n = 0, \dots, N$ with $t_0 = 0$ and $T_N = T$ and the step size $K = t_n - t_{n-1}$. The small interval of periodicity $I_P = [0, 1]$ is partitioned into micro time-steps τ_n for $n = 0, \dots, M$ with $\tau_0 = 0$, $\tau_M = 1$ and the step size $k = \tau_m - \tau_{m-1} \ll K$. A simple explicit/implicit multiscale iteration is then as follows:

Algorithm 1 (First order explicit/implicit multiscale iteration) *Let $\bar c_0$ be the initial value for the slow component. For $n = 1, 2, \dots$ iterate*

1. *Solve the periodic flow problem $(\mathbf{v}^{\bar c_{n-1}}, p^{\bar c_n})$ on the domain $\mathcal{F}(\bar c_{n-1})$*
2. *Compute the average of the reaction term*

$$\bar R(\bar c_{n-1}) := \int_0^1 R(\bar c_{n-1}, \boldsymbol{\sigma}^{\bar c_{n-1}}(s); \boldsymbol{x})\,ds$$
$$= \frac{\alpha}{1+\beta\bar c_{n-1}(\boldsymbol{x})}\int_0^1 \gamma\Big(\sigma_{WSS}\big(\boldsymbol{\sigma}^{\bar c_{n-1}}(\boldsymbol{x},s);\boldsymbol{x}\big)\Big)\,ds$$

3. *Make an semi-explicit step of the stenosis growth problem*

$$K^{-1}\big(\bar c_n - c_{n-1},\psi\big)_{\partial\hat{\mathcal{F}}} + \big(\lambda_c\nabla_\Gamma\bar c_n,\nabla_\Gamma\psi\big)_{\partial\hat{\mathcal{F}}} = \big(\bar R(\bar c_{n-1}),\psi\big)_{\partial\hat{\mathcal{F}}}$$

The discretization of the growth problem in Step 3. can easily be replaced by a second order explicit scheme like the Adams-Bashforth formula, see [19]. A fully implicit time-integration can be realized by adding a sub-iteration for steps 2–4. However, since the diffusion parameter is very small, explicit schemes are appropriate in this setting.

Within every step of the iteration it is necessary to solve the periodic-in-time flow problem (even multiple solutions are required in a fully implicit setting). This is the main effort of the resulting scheme, since the sub interval [0, 1] must be integrated several times to obtain a suitable periodic solution. In principle it is possible to just compute several cycles of the periodic problem until the periodicity error

$$\|\mathbf{v}^{\bar{c}_n}(T + 1\,\mathrm{s}) - \mathbf{v}^{\bar{c}_n}(T)\| < \epsilon_P$$

falls below a given threshold $\epsilon_P > 0$. Usually however this error is decreasing with an exponential rate only that depends on parameters like the viscosity and the domain size. For acceleration several methods are discussed in literature, based on optimization problem [39], on the idea of the shooting method [28], on Newton [25] or on space time techniques [33]. Here we quickly present a very efficient novel scheme that converges with a fixed rate that does not depend on any further parameters. We note however that although the computational efficiency is striking, the theoretical validation extends to the linear Stokes equation only, see [37].

Solution of the Periodic Flow Problem The idea of the averaging scheme for the rapid identification of periodic flow problems is to split the periodic solution into average and oscillation, see also [37]

$$\mathbf{v}^\pi(t) = \bar{\mathbf{v}}^\pi + \tilde{\mathbf{v}}^\pi(t), \quad \int_0^1 \tilde{\mathbf{v}}^\pi(s)\,\mathrm{d}s = 0.$$

In a nonlinear problem like the Navier-Stokes equations it is not possible to separate the average from the oscillations. But, by averaging the Navier-Stokes equation, we derive

$$-\operatorname{div}\bar{\boldsymbol{\sigma}}_f^\pi + (\bar{\mathbf{v}}^\pi \cdot \nabla)\bar{\mathbf{v}}^\pi = \underbrace{-\int_0^1 \Big\{(\tilde{\mathbf{v}}^\pi(s)\cdot\nabla)\tilde{\mathbf{v}}^\pi + (\tilde{\mathbf{v}}^\pi\cdot\nabla)\tilde{\mathbf{v}}^\pi(s)\Big\}\mathrm{d}s}_{=:N(\bar{\mathbf{v}}^\pi,\tilde{\mathbf{v}}^\pi)}, \quad \operatorname{div}\bar{\mathbf{v}}^\pi = 0.$$

If we average a solution $(\mathbf{v}(t), p(t))$ to the Navier-Stokes problem for arbitrary initial $\mathbf{v}_0$ (that does not yield the periodic solution) we get

$$-\operatorname{div}\bar{\boldsymbol{\sigma}}_f + (\bar{\mathbf{v}}\cdot\nabla)\bar{\mathbf{v}} = \mathbf{v}(0) - \mathbf{v}(1) + N(\bar{\mathbf{v}}, \tilde{\mathbf{v}}), \quad \operatorname{div}\bar{\mathbf{v}} = 0.$$

The difference $\mathbf{w} := \mathbf{v}^\pi - \mathbf{v}$, $q := p^\pi - p$ between dynamic solution and periodic solution satisfies the averaged equation

$$\begin{aligned} -\operatorname{div}\bar{\boldsymbol{\sigma}}_f(\bar{\mathbf{w}}, \bar{q}) + (\bar{\mathbf{w}}\cdot\nabla)\bar{\mathbf{w}} + (\mathbf{w}\cdot\nabla)\bar{\mathbf{v}} + (\bar{\mathbf{v}}\cdot\nabla)\mathbf{w} \\ = \mathbf{v}(1) - \mathbf{v}(0) + N(\bar{\mathbf{v}}^\pi, \tilde{\mathbf{v}}^\pi) - N(\bar{\mathbf{v}}^\pi, \tilde{\mathbf{v}}^\pi), \quad \operatorname{div}\bar{\mathbf{w}} = 0. \end{aligned}$$

We assume that we start with a good guess $\mathbf{v}$ that is already close to the periodic solution $\mathbf{v}^\pi$, i.e. $\|\mathbf{w}\|$ is small. If no initial approximation is available, e.g. in the very first step of the multiscale scheme, we still can perform a couple for forward simulations. Given that $\|\mathbf{w}\|$ is small, we will neglect both the nonlinearity $(\bar{\mathbf{w}}\cdot\nabla)\bar{\mathbf{w}}$

and all fluctuation terms $N(\cdot,\cdot)$ involving the oscillatory parts. We approximate the difference between average of the dynamic solution and average of the desired periodic solution by the linear equation

$$(\mathbf{w}\cdot\nabla)\bar{\mathbf{v}} + (\bar{\mathbf{v}}\cdot\nabla)\mathbf{w} - \text{div}\ \bar{\boldsymbol{\sigma}}_f(\bar{\mathbf{w}},\bar{q}) = \mathbf{v}(1) - \mathbf{v}(0) \tag{9}$$

The averaging scheme for finding the periodic solution is then given by the following iteration.

Algorithm 2 (Averaging scheme for periodic-in-time problems) *Let $\boldsymbol{v}_0^0$ be a guess for the initial value. If no approximation is available, $\mathbf{v}_0^0$ can be obtained by computing several cycles of the dynamic flow problem. For $l = 1, 2, \dots$ iterate*

1. *Based on the initial $\boldsymbol{v}^l(0) = \boldsymbol{v}_0^{l-1}$ solve once cycle of the dynamic flow problem on $I_P = [0, 1]$.*
2. *Solve the averaging equation for $\bar{\boldsymbol{w}}^l$ and $\bar{q}^l$, Eq. (9)*
3. *Update the initial value by correcting the average*

$$\boldsymbol{v}_0^l := \boldsymbol{v}^l(1) + \bar{\boldsymbol{w}}^l.$$

The analysis of this averaging scheme is open for the Navier-Stokes equations but simple for linear problems with symmetric positive definite operator like the Stokes equations. Here the convergence estimate

$$\|\mathbf{v}_0^l - \mathbf{v}_0^\pi\| \le \rho_{avg}\cdot\|\mathbf{v}_0^{l-1} - \mathbf{v}_0^\pi\|$$

holds, with $\rho_{avg} < 0.3$ in the continuous and $\rho_{avg} < 0.42$ in the discrete setting, for further results we refer to [37].

4 Numerical Results

We present a numerical study on the multiscale scheme and give a first discussion on its accuracy and efficiency. In [19] simple two-dimensional problems have been studied that also allow for resolved simulations such that a direct comparison of computational times for forward simulations and multiscale simulations can be performed. These demonstrated speedups reaching from $1:200$ to $1:10{,}000$. Here it was shown that the multiscale scheme benefits from larger scale separation. To be precise: to reach the same relative accuracy in a multiscale computation as compared to a direct forward computation, the speedup behaves like $1:\alpha^{-1}$.

Before presenting results for the multiscale method we briefly discuss the averaging scheme for finding periodic solutions

Table 1 Number of cycles required to reduce the periodicity error to $\|\mathbf{v}(t+1\,\mathrm{s})-\mathbf{v}(t)\| < 10^{-8}$ for the direct forward simulation and the averaging scheme. Variation in the viscosity ν

ν	Forward			Averaging		
	0.1	0.05	0.025	0.1	0.05	0.025
Cycles	40	74	140	15	15	18

4.1 Convergence of the Averaging Scheme for Periodic Flow Problems

We consider a 3d problem that is inspired by the driven cavity problem. On the cube $\Omega = (-2, 2)^3$ we drive the Navier-Stokes equation by a 1-periodic forcing

$$\mathbf{f}(\mathbf{x}, t) = \frac{\sin\big(2\pi t\big)}{6}\begin{pmatrix} 3\tanh(\mathbf{x}_2) \\ 2\tanh(\mathbf{x}_3) \\ \tanh(\mathbf{x}_1) \end{pmatrix}$$

Since the data is periodic in time we can expect to obtain a time-periodic solution (if the Reynolds number is sufficiently small). In Table 1 we show the performance of the averaging scheme in comparison to a simple forward iteration. We give the number of cycles required to reach the periodicity error $\|\mathbf{v}(T_n+1\,\mathrm{s})-\mathbf{v}(T_n)\| < 10^{-8}$. The results show a strong superiority of the averaging scheme, both in terms of robustness (with respect to ν) and in terms of the overall computational complexity. For the forward iteration, the number of cycles approximately doubles with each reduction of ν. The performance of the averaging scheme slightly deteriorates for $\nu = 0.025$ due to the higher Reynolds number regime. For $\nu = 0.01$ we cannot identify a stable periodic solution. The computational overhead of the averaging scheme is very low, one additional stationary problem must be solved in each cycle. A detailed study of the averaging scheme with an analysis of the sensitivity to various further parameters is given in [37].

4.2 Simulation of the Candy Wrapper Problem

Figure 1 shows the evolution of the stenosis at three different points in time. In addition we show the outflow rate as function over time (one period). Several effects known from the medical practice can be identified: The growth of the stenosis is non-symmetric and mostly centered on the inflow-tip of the stent. This shows the necessity of considering full three dimensional models. Further, the simulations show an extension and growth of the stenosis to both sides which is also typical. Since the flow is pressure driven, the outflow rate decreases with the development of the stent.

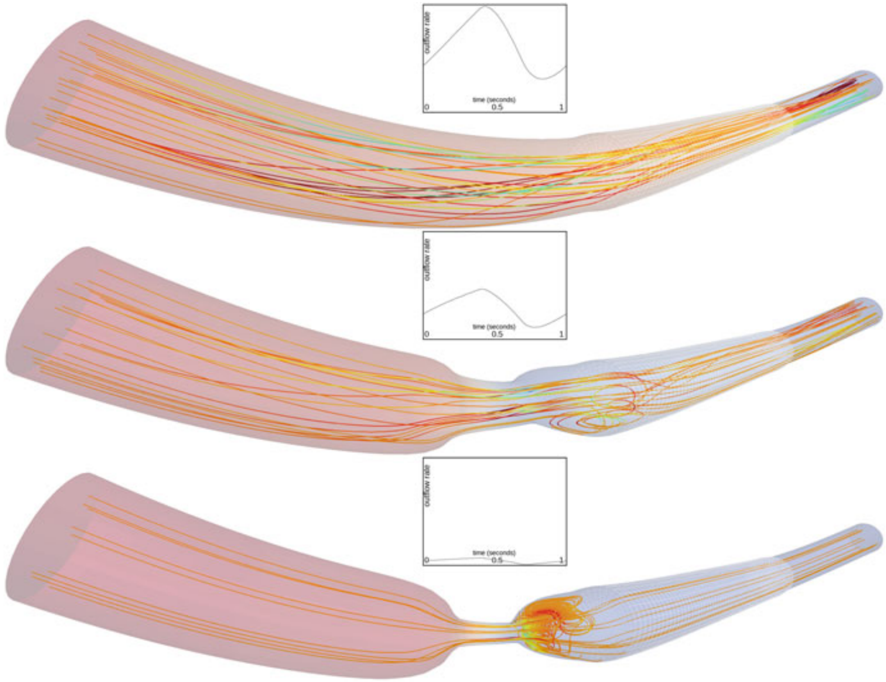


Fig. 1 Development of the stenosis at initial time, at $T = 33$ days and $T = 67$ days. The average and the oscillation of the flow rate get smaller while the stenosis develops

Table 2 Outflow J_{out} at time $T \approx 18$ days and extrapolation including numerical convergence order for $K \to 0$ (k fixed) and $k \to 0$ (K fixed). 55 years ($*$) computational time result from a projection of the computational time for a resolved simulation without the multiscale scheme

K	k	J_{out}	Time	K	k	J_{out}	Time
144,000	0.02	0.9359	9 min	72,000	0.04	0.9132	15 min
72,000	0.02	0.9138	18 min	72,000	0.02	0.9138	18 min
36,000	0.02	0.9043	40 min	72,000	0.01	0.9140	41 min
Extra $K \to 0$		0.8971 (1.22)	55 years$^{(*)}$	Extra $k \to 0$		0.9131 (1.58)	

In Table 2 we compare the results of the multiscale scheme for different values of k and K. We observe convergence in both parameters. Numerical extrapolation yields $O(k^{1.58} + K^{1.22})$, slightly off the expected rates $O(k^2 + K)$. We also indicate the computational times required for running the multiscale scheme till $T \approx 18$ days. A corresponding resolved simulation would require about 55 years computational time. This value is predicted based on the average time for computing a complete cycle of the periodic problem and based on an average three iterations required for approximating the periodic flow problem. Assuming that the extrapolated value for $K \to 0$ is accurate, the simulation based on $K \approx 36\,000$ s carries a multiscale error of about 1%. This approximation is achieved in 40 min instead of 55 years. The results in Table 2 indicate that it is worthwhile to consider a second order time stepping scheme for the plaque growth problem, since the error in K dominates. We refer to [19] for a realization in the context of a PDE/ODE long-scale/short scale problem.

5 Outlook and Discussion

We have demonstrated a numerical framework for simulating complex multiphysics/multiscale problems in hemodynamics. For the first time we could demonstrate an efficient numerical scheme for a long-scale/short-scale problem coupling different partial differential equations. We are able to include both temporal and spatial effects in bio-medical growth applications. The combination of a temporal multiscale method with fast solvers and efficient discretizations for the (periodic) micro problems gives substantial speedups such that three dimensional problems can be treated. Two main challenges remain for future work:

Fluid-Structure Interactions The main challenge in including elastic vessel walls lies in the increased complexity of the resulting system due to nonlinearities coming from the domain motion and the coupling to the hyperbolic solid equation that, by introducing the deformation as additional variable, blows up the problem size. In hemodynamical applications the coupling is governed by the added mass instability that usually calls for strongly coupled solution approaches, see [7, 26]. Although some progress has been made in recent years [2, 18, 29, 35], the design of efficient solvers for the resulting algebraic problems is still not satisfactory.

Considering monolithic solution approaches in combination with Newton-Krylov solvers make the use of large time steps possible. In all of the just mentioned approaches for designing linear solvers it has shown to be essential to partition the linear system when it actually comes to inversion of matrices, either within a preconditioner or within a multigrid smoother. This is mainly due to the very large condition numbers of the coupled system matrix that by far exceeds those of the subproblems, see [2, 35].

A second difficulty coming with fluid-structure interactions lies in the derivation of the effective growth equation described in Sect. 3. If elastic fluid-structure interactions are taken into account, the domain undergoes oscillations in the scale of the fast problem, i.e. during each pulsation of the blood flow. However, we can nevertheless introduce the averaged growth variable $\bar{c}(\mathbf{x}, t)$ as in (7) and simply average the growth equation (the third equation of Eq. (6)) as this is stated on the fixed reference domain. We note however that we have chosen a very simple growth model given as surface equation. Considering the detailed system introduced in [42], growth takes place within the solid, which is a three dimensional domain $\mathcal{S}(t) \subset \mathbb{R}^3$ undergoing deformation from the coupled fsi problem. A corresponding equation mapped to the fixed reference domain $\hat{S}$ (taken from [42]) reads

$$\left(\frac{\partial}{\partial t}(J\hat{c}), \psi\right)_{\hat{S}} + \left(\lambda_c J\mathbf{F}^{-1}\mathbf{F}^{-T}\nabla\hat{c}, \nabla\psi\right)_{\hat{S}} = R(\hat{c}, \hat{\boldsymbol{\sigma}}).$$

Since J and $\mathbf{F}$ oscillate with the frequency of the fast scale problem, derivation of an effective equation is still subject to future work.

Patient Specific Simulation The second open problem is to incorporate patient specific data into the simulations for generating specific predictions. Flow and geometry data can easily be measured during the stenting process. This process however is strongly invasive and causes subsequent adaptions of the vessel and the surrounding tissue interacting with the stent. Further data on the resulting configurations are not easily available without additional interventions. With a diameter of only a few millimeters, coronary arteries are small, such that measurements at good accuracy cannot be obtained.

Medical Application The edge stenosis accompanying the implementation of DES (Drug Eluting Stents) is great starting point for development of the further numerical experiments in the field of the plaque formation and biochemical processes ongoing in the vessel walls exposed to other types of interventions. Explosive growth of the intravascular interventions in recent decade is, inevitably, going to demand more advanced studies on the nature of vascular wall response to the implantable devices [13]. Novel numerical methods may also shade new light on well-established surgical procedures and augment the awareness of the potential benefits or hazards that are not yet fully understood or identified [16]. On the other hand the population of the patients is changing dramatically and that process is soon to accelerate. According to the recent report published by European Commission, diseases of the circulatory system are the most common cause of death in elderly population aged over 75 years [17]. In addition to that gruesome information the ageing of the European population in the years to come is growing concern of the governments. Poland belongs to the group of the countries that may become affected by the population ageing the most [22]. Due to that we face the necessity of development the most efficient treatment strategies for the elderly population. One of those treatment procedures is TAVR (Transcatheter Aortic Valve Replacement). The procedure addresses aortic valve stenosis that is quite often ailment in the aforementioned group of patients. By application of the fluid structure interaction methods it might be possible to tailor the design of the medical devices to the stiffer tissues usually present in the elderly patients in the way that may augment long time outcome of the procedure. Just such a small improvement may diminish the risk of repeated procedures undertaken in frail patients.

The methodology presented in our work should also find it’s application in optimization of the classic surgery for the coronary artery disease. The position of the vascular anastomosis in relation to the existing vascular wall lesions may find new rationale when understood through the knowledge of the mechanotransduction phenomena. Also the strategic planning of the target vessels and “landing sites” for the aorto-coronary by-pass grafts may find its' new understanding. Those perspective studies could be undertaken only by the means of model based planning.

Acknowledgement We acknowledge support by the Federal Ministry of Education and Research of Germany (project number 05M16NMA).

References

1. D. J. Angiolillo, M. Sabata, F. Alfonso, and C. Macaya. "Candy wrapper" effect after drug-eluting stent implantation: deja vu or stumbling over the same stone again? *Catheter Cardiovasc. Interv.*, 61(3):387–91, 2004.
2. E. Aulisa, S. Bna, and G. Bornia. A monolithic ale Newton-Krylov solver with Multigrid-Richardson–Schwarz preconditioning for incompressible fluid-structure interaction. *Computers & Fluids*, 174:213–228, 2018.
3. Brinda Balakrishnan, Abraham R. Tzafriri, Philip Seifert, Adam Groothuis, Campbell Rogers, and Elazer R. Edelman. Strut position, blood flow, and drug deposition. *Circulation*, 111(22):2958–2965, 2005.
4. R. Becker and M. Braack. A finite element pressure gradient stabilization for the Stokes equations based on local projections. *Calcolo*, 38(4):173–199, 2001.
5. R. Becker and M. Braack. A two-level stabilization scheme for the Navier-Stokes equations. In et. al. M. Feistauer, editor, *Numerical Mathematics and Advanced Applications, ENUMATH 2003*, pages 123–130. Springer, 2004.
6. R. Becker, M. Braack, D. Meidner, T. Richter, and B. Vexler. The finite element toolkit GASCOIGNE. http://www.gascoigne.uni-hd.de.
7. P. Causin, J.F. Gereau, and F. Nobile. Added-mass effect in the design of partitioned algorithms for fluid-structure problems. *Comput. Methods Appl. Mech. Engrg.*, 194:4506–4527, 2005.
8. M.A. Costa and D. Simon I. Molecular basis of restenosis and drug-eluting stents. *Circulation*, 111(17):2257–2273, 2005.
9. D.R. Holmes et al. Restenosis after percutaneous transluminal coronary angioplasty (PTCA): A report from the PTCA registry of the national heart, lung, and blood institute. *The American Journal of Cardiology*, 53(12):C77–C81, 1984.
10. F. LaDisa J., Jr et al. Stent design properties and deployment ratio influence indexes of wall shear stress: a three-dimensional computational fluid dynamics investigation within a normal artery. *Journal of Applied Physiology*, 97(1):424–430, 2004.
11. L. Jian et al. An integrated TAXUS IV, V, and VI intravascular ultrasound analysis of the predictors of edge restenosis after bare metal or paclitaxel-eluting stents. *The American Journal of Cardiology*, 103(4):501–506, 2009.
12. M. Sabaté et al. Geographic miss. *Circulation*, 101(21):2467–2471, 2000.
13. M.J. Mack et al. Transcatheter aortic-valve replacement with a balloon-expandable valve in low-risk patients. *N. Engl. J. Med.*, 380(18):1695–1705, 2019.
14. P.W. Serruys et al. Randomised comparison of implantation of heparin-coated stents with balloon angioplasty in selected patients with coronary artery disease (Benestent II). *The Lancet*, 352(9129):673–681, 1998.
15. U. Sigwart et al. Intravascular stents to prevent occlusion and re-stenosis after transluminal angioplasty. *New England Journal of Medicine*, 316(12):701–706, 1987.
16. V.H. Thourani et al. Contemporary real-world outcomes of surgical aortic valve replacement in low-risk, intermediate-risk, and high-risk patients. *Ann. Thorac. Surg.*, 99(1):55–61, 2015.
17. EUROSTAT. Ageing Europe: looking at the lives of older people in the EU. Technical report, European Union, 2019.

18. L. Failer and T. Richter. A parallel newton multigrid framework for monolithic fluid-structure interactions. *Journal of Scientific Computing*, 82(2), 2020.
19. S. Frei and T. Richter. Efficient approximation of flow problems with multiple scales in time. *SIAM Multiscale Modeling and Simulation*, 18(2), 942–969.
20. S. Frei, T. Richter, and T. Wick. Long-term simulation of large deformation, mechano-chemical fluid-structure interactions in ALE and fully Eulerian coordinates. *J. Comp. Phys.*, 321:874–891, 2016.
21. G.P. Galdi and M. Kyed. Time-periodic solutions to the Navier-stokes equations. In *Giga Y., Novotny A. (eds) Handbook of Mathematical Analysis in Mechanics of Viscous Fluids*, pages 1–70. Springer, 2016.
22. K. Giannakouris. Ageing characterises the demographic perspectives of the European societies. In *EUROPOP2008*. European Union, 2008.
23. E. Grube, U. Gerckens, R. Müller, and L. Büllesfeld. Drug eluting stents: initial experiences. *Zeitschrift für Kardiologie*, 91(3):44–48, 2002.
24. Andreas Grüntzig. Transluminal dilatation of coronary-artery stenosis. *The Lancet*, 311(8058):263, 1978.
25. F.M. Hante, M.S. Mommer, and A. Potschka. Newton-Picard preconditioners for time-periodic, parabolic optimal control problems. *SIAM J. Num. Ana.*, 53(5):2206–2225, 2015.
26. M. Heil, A.L. Hazel, and J. Boyle. Solvers for large-displacement fluid-structure interaction problems: Segregated vs. monolithic approaches. *Computational Mechanics*, 43:91–101, 2008.
27. G.A. Holzapfel. *Nonlinear Solid Mechanics: A Continuum Approach for Engineering*. Wiley-Blackwell, 2000.
28. L. Jiang, L. T. Biegler, and V. G. Fox. Simulation and optimization of pressure-swing adsorption systems for air separation. *AIChE Journal*, 49(5):1140–1157, 2003.
29. D. Jodlbauer, U. Langer, and T. Wick. Parallel block-preconditioned monolithic solvers for fluid-structure interaction problems. *International Journal for Numerical Methods in Engineering*, 117(6):623–643, 2019.
30. K. C. Koskinas, Y. S. Chatzizisis, A. P. Antoniadis, and G. D. Giannoglou. Role of endothelial shear stress in stent restenosis and thrombosis: pathophysiologic mechanisms and implications for clinical translation. *J Am Coll Cardiol*, 59(15):1337–49, 2012.
31. J.F. LaDisa, I. Guler, L.E. Olson, D.A. Hettrick, Judy R. Kersten, D.C. Warltier, and P.S. Pagel. Three-dimensional computational fluid dynamics modeling of alterations in coronary wall shear stress produced by stent implantation. *Annals of Biomedical Engineering*, 31(8):972–980, 2003.
32. M. Luskin and R. Rannacher. On the smoothing property of the Crank-Nicholson scheme. *Applicable Anal.*, 14:117–135, 1982.
33. F. Platte, D. Kuzmin, C. Fredebeul, and S. Turek. Novel simulation approaches for cyclic-steady-state fixed-bed processes exhibiting sharp fronts and shocks. In M. de Bruin, D. Mache, and J. Szabados, editors, *Trends and applications in constructive approximations*, volume 151 of *International series of numerical mathematics*, pages 207–233. Birkhäuser, 2005.
34. T. C. Poerner, K. K. Haase, B. Wiesinger, J. Wiskirchen, and S. H. Duda. Drug-coated stents. *Minimally Invasive Therapy & Allied Technologies*, 11(4):185–192, 2002.
35. T. Richter. A monolithic geometric multigrid solver for fluid-structure interactions in ALE formulation. *Int. J. Numer. Meth. Engrg.*, 104(5):372–390, 2015.
36. T. Richter. *Fluid-structure Interactions. Models, Analysis and Finite Elements*, volume 118 of *Lecture notes in computational science and engineering*. Springer, 2017.
37. T. Richter. An averaging scheme for the approximation of periodic-in-time flow problems. *Computers and Fluids*, accepted 2020. https://arxiv.org/abs/1806.00906.
38. T. Richter and T. Wick. On time discretizations of fluid-structure interactions. In T. Carraro, M. Geiger, S. Körkel, and R. Rannacher, editors, *Multiple Shooting and Time Domain Decomposition Methods*, volume 9 of *Contributions in Mathematical and Computational Science*, pages 377–400. Springer, 2015.
39. T. Richter and W. Wollner. Optimization framework for the computation of time-periodic solutions of partial differential equations. *Viet. J. Math.*, 46(4):949–966, 2019.

40. E.K. Rodriguez, A. Hoger, and A.D. McCulloch. Stress-dependent finite growth in soft elastic tissues. *J. Biomechanics*, 4:455–467, 1994.
41. Tuoi T. N. Vo, Sarah Morgan, Christopher McCormick, Sean McGinty, Sean McKee, and Martin Meere. Modelling drug release from polymer-free coronary stents with microporous surfaces. *International Journal of Pharmaceutics*, 544(2):392–401, 2018.
42. Y. Yang, W. Jäger, M. Neuss-Radu, and T. Richter. Mathematical modeling and simulation of the evolution of plaques in blood vessels. *J. of Math. Biology*, 72(4):973–996, 2016.
43. Y. Yang, T. Richter, W. Jaeger, and M. Neuss-Radu. An ALE approach to mechano-chemical processes in fluid-structure interactions. *Int. J. Numer. Math. Fluids.*, 84(4):199–220, 2017.
44. P. Zunino, C. D'Angelo, L. Petrini, C. Vergara, C. Capelli, and F. Migliavacca. Numerical simulation of drug eluting coronary stents: Mechanics, fluid dynamics and drug release. *Computer Methods in Applied Mechanics and Engineering*, 198(45):3633–3644, 2009.

Systematisation of Systems Solving Physics Boundary Value Problems

Tuomo Rossi, Jukka Räbinä, Sanna Mönkölä, Sampsa Kiiskinen, Jonni Lohi, and Lauri Kettunen

Abstract A general conservation law that defines a class of physical field theories is constructed. First, the notion of a general field is introduced as a formal sum of differential forms on a Minkowski manifold. By the action principle the conservation law is defined for such a general field. By construction, particular field notions of physics, e.g., magnetic flux, electric field strength, stress, strain etc. become instances of the general field. Hence, the differential equations that constitute physical field theories become also instances of the general conservation law. The general field and the general conservation law together correspond to a large class of relativistic hyperbolic physical field models. The parabolic and elliptic models can thereafter be derived by adding constraints. The approach creates solid foundations for developing software systems for scientific computing; the unifying structure shared by the class of field models makes it possible to implement software systems which are not restricted to certain predefined problems. The versatility of the proposed approach is demonstrated by numerical experiments with moving and deforming domains.

1 Introduction

In this paper we focus on second-order boundary value problems (BVP's) related to physical field theories. BVP's and their numerical solution methods is an extensively studied field of science. Still, many practical challenges remain, e.g.: (1) One may have a problem to which there is no software system available. (2) The software systems are laborious if not hard to extend beyond their original purpose and such extensions increase the complexity of the system. (3) In case of incorrect results,

T. Rossi (✉) · J. Räbinä · S. Mönkölä · S. Kiiskinen · J. Lohi · L. Kettunen
Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
e-mail: tuomo.rossi@jyu.fi; tuomo.j.rossi@jyu.fi; jukka.rabina@jyu.fi; sanna.monkola@jyu.fi; sampsa.kiiskinen@jyu.fi; jonni.lohi@jyu.fi; lauri.kettunen@jyu.fi

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_3

it is tedious to distinguish between simple user errors and errors in reasoning. (4) Users often have to learn many software specific details.

While practical challenges will always remain, the aforementioned issues reflect the traditional architectural view on mathematical software from the fifties and sixties. Nowadays there exists more powerful mathematical and programming language-theoretic knowledge that can be exploited in developing systems for boundary value problems. Thus, there is a call for a systematic mathematical analysis to combine the knowledge in BVP's and modern programming and computing. The software systems can be established more systematically on the mathematical structures on which BVP's are built.

We aim to present a class of BVP's that covers classical physics, such as Maxwell's equations, Schrödinger equation etc. The specialized models are obtained by adding constraints (e.g., omitting terms, linking terms together etc.) to the general model. This resembles object-oriented style in programming; a generic class is instantiated and made more concrete by adding constraints. The finite-dimensional models can all be constructed with the so-called discrete exterior calculus (DEC) from the models expressed with differential forms. The approach is not limited to ordinary differential forms. Vector valued (E-valued) and matrix valued (End(E)-valued) differential forms can also be utilised making it possible to conveniently construct, for example, the equations of elasticity or the Yang–Mills equations with the same approach.

This research is, therefore, directly linked to several fields: partial differential equations, differential geometry, manifolds and cell complexes, algebraic topology (homology and cohomology theories, fiber spaces and bundles), global analysis of manifolds, numerical analysis, and computer science.

The state-of-the-art in field theories is gauge theory [3, 4, 23]. It is about classical and quantum fields whose configurations are cocycles in differential cohomology. We focus on ordinary gauge theories whose field configurations are vector bundles with connection. Their main principles [3]—Lagrangians, actions, the action principle [4, 11] manifolds, vector bundles, sections of bundles, connections, etc.—form a cornerstone of the work.

Our general presentation of various field-theoretic space-time models provides us with significant advantages. In classical physics and in engineering different fields use different concepts, notation and terminology. This results in scattered knowledge, in waste of resources, and in redundancy in software. Classical and quantum field theories appear quite distinct. In classical theories the effect of the fundamental forces is averaged into the mesoscopic constitutive laws. The corresponding material laws can be embedded into the Hodge operator [5, 6, 17, 18], and thus they describe the metric properties of spatial space. For this reason, the metric structure is essential in classical theories.

Powerful commercial and academic software for classical multi-physics exist, such as COMSOL Multiphysics [9] or GetDP [10], but there is no encompassing mathematical theory available to guide the software development. Our aim is to employ the presented approach as the guiding theory in systemizing development of software in scientific computing.

2 Differential Geometric Models

In the late 1990's and early 2000's, Bossavit et al. developed and introduced the so called "geometric approach" into electromagnetism [7]. In addition, in 1997 the idea of a "discrete Hodge operator" [36] was introduced to reveal the key mathematical structures behind finite difference and finite element kind of methods [8]. At the time the finite difference method [34] was commonly explained in a rather elementary manner in Cartesian coordinate systems following K. Yee's original paper [39] from 1966. Bit later the scientific community in elasticity picked the idea and the geometric approach became known also as "discrete exterior calculus" (DEC) after Hirani [16].

We have further developed the geometric approach and created a generic software system based on it. The system can be employed to solve hyperbolic application problems from classical and quantum physics [22, 26, 28], such as electromagnetic, elastic, and acoustic wave problems, the Schrödinger equation [11], or Gross–Pitaevskii equations [27], and so on. We explain the mathematical foundations of the software system in [20]. The implementation of the simulation software and the various mesh structures which we have employed are described in detail in [26].

To explain the methodology, we will first outline a theory of ordinary gauge theories on form bundles. Thereafter we will briefly discuss the extension to Clifford and tensor algebra. Exterior (or Grassmann) algebra [12, 21] is the Clifford algebra [15], where the quadratic form is identically zero, and Clifford algebra itself is a quotient algebra of tensor algebra [24]. We assume a Minkowski manifold [3, 13], and describe the proposed methodology in steps from the foundations.

2.1 *Formal Sums of Field Configurations*

The field-notion in physics involves an idea of assigning numbers to geometrical objects of space-time, such as to points, (virtually) small segments of oriented lines, etc. These numbers represent observations made by measurements, and they can be interpreted as the values differential forms yield on p-vectors [5].

Let us start from ordinary differential forms, which come with a degree from zero to the dimension n of the manifold. Since we are not after any particular field configuration, forms of a particular degree are not in our interest. We hide the information of the degree by introducing a formal sum of differential forms of all degrees:

$$F = \alpha_0 f^0 + \alpha_1 f^1 + \ldots + \alpha_n f^n \in \bigoplus_{p=0}^{n} \bigwedge{}^p T^*\Omega,$$

where $\alpha_p \in \{0, 1\}$ and $T^*\Omega$ is the dual space of the tangent space. Note that with ordinary differential forms and in the n-dimensional case the number of p-forms is $\binom{n}{p}$ and the formal sum has the total of 2^n degrees of freedom.

By operating with F the emphasis is shifted from particular degrees to the property that all forms map some p-vector, $0 \leq p \leq n$, to scalars.

2.2 Differentiation and the Action Principle

Next, we need to introduce differentiation for F. This is straightforward as smooth p-forms are differentiated with the exterior derivative d, and so is also F. Ordinary gauge theories are characterized by pairs of differential equations, such as electromagnetic theory [33] is described by Maxwell's equations. The gauge-theoretic view is that differential equations follow from the action principle [3]. An action is the integral of a Lagrangian $\mathcal{L}$ over a manifold, and differential equations correspond to the critical points of the action.

A large class of models in ordinary gauge theories have to do with the conservation of some quadratic notion. We equip the Minkowski manifold with a metric tensor providing us also with a Hodge operator $\star$. Then, we assume F is an exact field, $F = dH$ where $H = h^0 + \ldots + h^n$ is a potential. In addition, for the source terms we introduce another formal sum $G = g^0 + \ldots + g^n$. Now, an action of the desired type can be given by

$$\mathcal{A} = \bigoplus_{p=0}^{n} \Big(\frac{1}{2} \int\limits_{\Omega} f^p \wedge \star f^p + \int\limits_{\Omega} h^{p-1} \wedge \star g^{p+1} \Big).$$

The differential equations are then obtained as follows. The variation of action $\mathcal{A}$ is

$$\delta\mathcal{A} = \frac{d}{d\alpha}\mathcal{A}(H_\alpha)\Big|_{\alpha=0},$$

where $H_\alpha = H + \alpha\delta H$ and by insisting on the variation δA to vanish for all δH yields the critical points of $\mathcal{A}$ and the corresponding differential equations. Hence, the action principle implies that at all (ordinary) points on the Minkowski manifold the following differential equations $\mathrm{d}F = 0$ and $\star\, \mathrm{d} \star F = \star\star G$ should hold. These equations can be expressed as the diagram in Fig. 1.

Fig. 1 Diagram of differential equations

$$\begin{tikzcd}
0 & F & \star F & & \star\star G \\
\bullet & \bullet \arrow[l, "\mathrm{d}"'] \arrow[r, bend left, "\star"] & \bullet \arrow[l, bend left, "\star^{-1}"] \arrow[r, "\mathrm{d}"] & \bullet \arrow[r, bend left, "\star"] & \bullet \arrow[l, bend left, "\star^{-1}"]
\end{tikzcd}$$

Let us also express the action principle as a diagram. For brevity, to introduce such a diagram we assume G to vanish. Then, the Lagrangian of the action becomes $\mathcal{L} = \bigoplus_{p=0}^{n} \frac{1}{2} f^p \wedge \star f^p$, and the definition of the Hodge operator implies, that each component $\mathcal{L}_p$ satisfies

$$\mathcal{L}_p = \frac{1}{2} f^p \wedge \star f^p \; = \; \frac{1}{2} \langle f^p, f^p \rangle \, \omega^0 \; = \; q_p(f^p) \, \omega^0,$$

where ω^0 is the unit n-volume of Minkowski space and q_p is the quadratic refinement of the Minkowski bilinear form $\langle \cdot, \cdot \rangle$. The $\mathcal{L}_p$'s form a product space $L = D_f \times D_f^*$ equipped with projections $\pi_f \in L \to D_f$ and $\pi_f^* \in L \to D_f^*$ satisfying the following universal property: For every action $\mathcal{A}$ and Lagrangian $\mathcal{L}$ there is a unique map $a \in \mathcal{A} \to \mathcal{L}$ and $l \in \mathcal{L} \to L$ such that the diagram of Fig. 2 is commutative.

The combination of the two diagrams of Figs. 1 and 2 results in a diagram presenting how the action with the Lagrangian defines differential equations for a pair of fields, which are in a Hodge relation to each other. We call this diagram by the name DGOrd (designating that it involves ordinary differential forms), simplify it a bit –object L is left out– and draw it in Fig. 3. Ordinary gauge theories include also other type of differential forms than ordinary ones, which are also essential

$$\begin{tikzcd}
 & \mathcal{A} \arrow[d, "a"] \arrow[ddl, bend right, "a_f"'] \arrow[ddr, bend left, "a_f^*"] & \\
 & \mathcal{L} \arrow[d, "l"] \arrow[dl, bend right, "l_f"'] \arrow[dr, bend left, "l_f^*"] & \\
D_f & L \arrow[l, "\pi_f"'] \arrow[r, "\pi_f^*"] & D_f^*
\end{tikzcd}$$

Fig. 2 Diagram of the action principle

$$\begin{tikzcd}
 & & \bullet \arrow[d, "a"] \arrow[ddl, bend right, "a_f"'] \arrow[ddr, bend left, "a_f^*"] & & & \\
 & & \bullet \arrow[dl, "l_f"'] \arrow[dr, "l_f^*"] & & & \\
\bullet & \bullet \arrow[l, "d"'] \arrow[rr, bend left, "\star"] & & \bullet \arrow[ll, bend left, "\star^{-1}"] \arrow[r, "d"] & \bullet \arrow[r, bend left, "\star"] & \bullet \arrow[l, bend left, "\star^{-1}"]
\end{tikzcd}$$

Fig. 3 Diagram DGOrd involves ordinary differential forms and is commutative. Its vertices are unique and they exist for all objects

in mathematical physics. For instance, in elasticity [1] E-valued forms, vector and covector-valued forms [3, 14, 21] are needed [19, 29–32].

Let us next extend the idea of formal sums of differential forms to E-valued forms. By construction, such formal sums of E-valued forms can be differentiated with the exterior covariant derivative d_∇, where ∇ is the connection. To introduce the Lagrangian as a quadratic refinement of the Minkowski metric, the Hodge operator should be extended to E-valued forms such that $\mathcal{L} = \bigoplus_{p=0}^{n} \frac{1}{2} f^p \wedge \star f^p$ becomes a formal sum of scalars. We denote such a Hodge operator by $\star_E$. In the same manner as in the case of ordinary forms, in case E-valued forms the action principle yields differential equations $\mathrm{d}_\nabla F = 0$, $\star_E \, \mathrm{d}_\nabla \star_E F = \star_E \star_E G$.

End(E)-valued [3] (i.e., matrix valued) forms are needed for example in Yang–Mills theory [2, 35, 37, 38]. The Hodge operator $\star_{End}$ is now extended to End(E)-valued forms so that the Lagrangian becomes a formal sum of scalars.

Formally, there exists an abstract diagram DGA shown in Fig. 4, and mappings $M_0 \in$ DGA $\rightarrow$ DGOrd, $M_{\mathrm{E}} \in$ DGA $\rightarrow$ DGE and $M_{\mathrm{End}} \in$ DGA $\rightarrow$ DGEnd. They map the abstract diagram DGA to more concrete diagrams DGOrd of ordinary forms, DGE of vector valued forms and DGEnd of matrix valued forms. They also map the hodge duality to operators $\star$, $\star_E$, and $\star_{End}$, respectively. M_O maps the differentiation to exterior derivative d. M_{E} and M_{End} map it to d_∇.

This construction suggests that mappings M_O, M_E and M_{End} represent various models of the "theory of differential geometric models" represented by DGA. This is a step towards a category-theoretic representation of physical field theories.

Remark Hyperbolic wave problems in physics are particular examples of our models. DGA can also be concretized to elliptic and parabolic models. Later, as an example, we show how to concretize the Schrödinger equation from the general setting. As we use differential geometric formalism, the canonical way to discretize all the considered models is to use DEC.

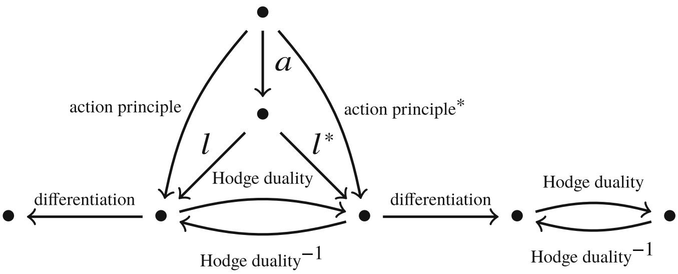


Fig. 4 Diagram DGA

3 Concretization of Particular Models

Next, to verify the usefulness of the theory and its models in scientific computing, let us exemplify how particular models are concretized from the theory. This also highlights the pragmatic significance of a proper mathematical theory; resources become more efficiently exploited, if software systems are designed to realize theories instead of particular models.

Let us start by concretizing DGOrd to four dimensional differentiable manifold Ω with Minkowski metric, signature $(-, +, +, +)$, and a decomposition of space-time into space and time-like components; $\Omega = \Omega_t \times \Omega_s$. Symbols F and G denote formal sums of p-forms, $F = f^0 + \ldots + f^4$ and $G = g^0 + \ldots + g^4$ and consequently, differential equations $\mathrm{d}F = 0$ and $\star\mathrm{d}\star f = \star\star G$ can be written as:

$$\begin{bmatrix} \cdot & -\star\mathrm{d}\star & \cdot & \cdot & \cdot \\ \mathrm{d} & \cdot & \star\mathrm{d}\star & \cdot & \cdot \\ \cdot & \mathrm{d} & \cdot & -\star\mathrm{d}\star & \cdot \\ \cdot & \cdot & \mathrm{d} & \cdot & \star\mathrm{d}\star \\ \cdot & \cdot & \cdot & \mathrm{d} & \cdot \end{bmatrix} \begin{bmatrix} f^0 \\ f^1 \\ f^2 \\ f^3 \\ f^4 \end{bmatrix} = \begin{bmatrix} g^0 \\ g^1 \\ g^2 \\ g^3 \\ g^4 \end{bmatrix}.$$

By (i) decomposing p-forms into time-like components and only space-like components, (ii) and exterior derivative to space and time-like components, and (iii) applying the Leibniz rule, we obtain an equivalent system of Eq. (1) [20].

$$\begin{bmatrix} \partial_t & \cdot & \cdot & \cdot & \cdot & \star\mathrm{d}\star & \cdot & \cdot \\ \cdot & \partial_t & \cdot & \cdot & -\mathrm{d} & \cdot & \star\mathrm{d}\star & \cdot \\ \cdot & \cdot & \partial_t & \cdot & \cdot & -\mathrm{d} & \cdot & \star\mathrm{d}\star \\ \cdot & \cdot & \cdot & \partial_t & \cdot & \cdot & -\mathrm{d} & \cdot \\ \cdot & \star\mathrm{d}\star & \cdot & \cdot & -\star\partial_t\star & \cdot & \cdot & \cdot \\ \mathrm{d} & \cdot & \star\mathrm{d}\star & \cdot & \cdot & \star\partial_t\star & \cdot & \cdot \\ \cdot & \mathrm{d} & \cdot & \star\mathrm{d}\star & \cdot & \cdot & -\star\partial_t\star & \cdot \\ \cdot & \cdot & \mathrm{d} & \cdot & \cdot & \cdot & \cdot & \star\partial_t\star \end{bmatrix} \begin{bmatrix} f^0 \\ f^1 \\ f^2 \\ f^3 \\ f^0_s \\ f^1_s \\ f^2_s \\ f^3_s \end{bmatrix} = \begin{bmatrix} g^0_s \\ g^1_s \\ g^2_s \\ g^3_s \\ g^0 \\ g^1 \\ g^2 \\ g^3 \end{bmatrix}. \qquad (1)$$

Here d and $\star$ are now the exterior derivative and the Hodge operator, respectively, in the space-like component Ω_s of manifold Ω. Subscript s in the f^p and g^p's denotes the space-like component f^p_s of $(p+1)$-form $f_t dt \wedge f^p_s$. This system of equations and its natural transformations cover a wide class of physical field theories. By construction, all the models covered by the theory are relativistic. Each particular model corresponds to a choice of F and G as demonstrated next with some examples.

For *Maxwell's equations* [33] in space and time, F is chosen to be the Faraday field and G the source charges q and currents j [3]: $F = b + e \wedge \mathrm{d}t = b + \mathrm{d}t \wedge (-e)$ and $G = \star j - \mathrm{d}t \wedge \star q$. This corresponds to setting $f^1_s = -e$, $f^2 = b$, $g^1 = \star j$, and $g_s^0 = -\star q$, and by substituting these to the system of Eq. (1), we obtain

$$\begin{aligned} \mathrm{d}\, b &= 0\,, \quad \text{8th row}, & \mathrm{d}e + \partial_t b &= 0, \quad \text{3rd row},\\ -\star\partial_t\star_\epsilon e + \star\mathrm{d}\star_\mu b &= \star j, \quad \text{6th row}, & -\star\mathrm{d}\star_\epsilon e &= -\star q, \quad \text{1st row}. \end{aligned}$$

We have considered permittivity ϵ and permeability μ as properties of Ω_s. Thus, they are embedded into the Hodge operators $\star_\epsilon$ and $\star_\mu$ [6].

The *non-relativistic Schrödinger equation* [11] can also be concretized from DGOrd by imposing some simplifying constraints on the general model. For this, we choose

$$\begin{aligned} f^0 &= \hbar\varphi_R, & f_s^0 &= \hbar\varphi_I, & f^1 &= \frac{\hbar}{2m}q_R, & f_s^1 &= \frac{\hbar}{2m}q_I,\\ g^0 &= V\varphi_R, & g_s^0 &= V\varphi_I, & g^1 &= q_R, & g_s^1 &= -q_I, \end{aligned}$$

where $\hbar$ is the reduced Planck constant and m is particle's mass. By substitution to the general system one obtains:

$$\begin{aligned} \partial_t\hbar\varphi_R + \star\mathrm{d}\star\frac{\hbar}{2m}q_I &= V\varphi_I, \ \text{1st row}, & \partial_t\frac{\hbar}{2m}q_R - \mathrm{d}\hbar\varphi_I &= -q_I, \text{2nd row},\\ \star\mathrm{d}\star\frac{\hbar}{2m}q_R - \star\partial_t\star\hbar\varphi_I &= V\varphi_R, \text{5th row}, & \mathrm{d}\hbar\varphi_R + \star\partial_t\star\frac{\hbar}{2m}q_I &= q_R, \ \text{6th row},\\ -\mathrm{d}\frac{\hbar}{2m}q_I &= 0, \quad \text{3rd row}, & \mathrm{d}\frac{\hbar}{2m}q_R &= 0, \quad \text{7th row}. \end{aligned}$$

The relativistic property is next lost by a modelling decision. Terms $\partial_t q_R$ and $\partial_t q_I$ are assumed to vanish. Now the bottom equations become tautologies and the system is reduced to the pair

$$\partial_t\hbar\varphi_R + \frac{\hbar^2}{2m}\star\mathrm{d}\star\mathrm{d}\,\varphi_I = V\varphi_I, \qquad \star\partial_t\star\hbar\varphi_I - \frac{\hbar^2}{2m}\star\mathrm{d}\star\mathrm{d}\,\varphi_R = -V\varphi_R.$$

By mapping differential forms to vector fields and mapping the exterior derivative to the corresponding differential operators of vector analysis, the textbook version $\hbar\,\partial_t\varphi - \mathrm{i}\frac{\hbar^2}{2m}\,\mathrm{div}\,\mathrm{grad}\,\varphi = -\mathrm{i}V\varphi$ results. It is defined using complex arithmetic which restricts it to flat Minkowski manifold only. The relativistic intermediate stage obtained from the general model can also be implemented on curved space-time. This is an interesting topic to be numerically tested.

Small-strain elasticity is naturally modelled using E-valued forms. Recall that in this case the differential equations on the manifold $\Omega_t \times \Omega_s$ take the form $\mathrm{d}_\nabla F = 0$ and $\star_E\,\mathrm{d}_\nabla\star_E F = \star_E\star_E G$. Analogously to previous, Leibniz rule and the space-time split of forms and exterior covariant derivative d_∇ results in structurally similar general system as in the case of ordinary forms. The exterior derivative d is simply replaced by d_∇ and $\star$ is replaced by $\star_E$.

The model of elasticity now arises by the choice $f_s^0 = u$, $f^1 = \varepsilon$, $g^0 = -\star f_v$, where the vector-valued 0-form u is the time-derivative of displacement ν, $u = \partial_t \nu$, the vector-valued 1-form ϵ is linearized strain, and g^0 is the source force term. Substituting this choice back to the system of equations yields

$$\partial_t \varepsilon - \mathrm{d}_\nabla u = 0 \;\; \text{2nd row}, \quad \mathrm{d}_\nabla \varepsilon = 0 \;\; \text{7th row}, \quad \star \mathrm{d}_\nabla \star^C \varepsilon - \star \partial_t \star^\rho u = -\star f_v \;\; \text{5th row}.$$

The Hodge operator $\star^C$ contains the parameters of the stress-strain relation, and density ρ is embedded to $\star^\rho$. Since $u = \partial_t \nu$, the first equation states that $\epsilon = \mathrm{d}_\nabla \nu$ and the second equation is automatically satisfied. As a result, we get the elasticity equations which are, for convenience, written out also in Euclidian space and using vector analysis notation:

$$\begin{aligned} -\partial_t \varepsilon + \mathrm{d}_\nabla u &= 0\,, & \sigma &= \star^C \varepsilon\,, & -\partial_t \overline{\varepsilon} + \mathrm{grad}\,\overline{u} &= 0\,, & \overline{\sigma} &= C\overline{\varepsilon}\,, \\ \star^\rho \partial_t u - \mathrm{d}_\nabla \sigma &= f_v\,, & u &= \partial_t \nu\,, & \rho \partial_t \overline{u} - \mathrm{div}\,\overline{\sigma} &= \overline{f}_v\,, & \overline{u} &= \partial_t \overline{\nu}\,. \end{aligned}$$

The final example is *Yang–Mills equations* [38] where the field configurations are End(E)-valued forms. As Yang and Mills developed their theory as an extension to Maxwell's theory, Yang–Mills equations are concretized from the system of differential equations for End(E)-valued forms in the same manner as Maxwell's equations are concretized from the system for ordinary forms. Such a process results in

$$\begin{aligned} \mathrm{d}_\nabla b &= 0, & \mathrm{d}_\nabla e + \nabla_t b &= 0, \\ -\star_{End} \nabla_t \star_{End}\, e + \star_{End}\, \mathrm{d}_\nabla \star_{End}\, b &= \star_{End}\, j, & \star_{End}\, \mathrm{d}_\nabla \star_{End}\, e &= \star_{End}\, \rho\,. \end{aligned}$$

Clifford and Tensor Algebra The "theory of differential geometric models" presented is not complete for the needs of software design. First, tensor algebra is the most general algebra for vector spaces over scalars, and all field configurations share the structure of a vector space. Second, a Clifford algebra is unital associative algebra generated by a vector space equipped with a quadratic form [15]. Third, exterior algebra is the Clifford algebra when the quadratic form is zero. Clifford algebra seems to provide us with a better starting point as certain Clifford algebras, such as Pauli or Dirac algebra [13], are very important in mathematical physics.

We construct the universal Clifford algebra as a subalgebra of the algebra of linear transformations [15]. Let $\mathbb{F}$ be a scalar field and denote $\Lambda^0(V) = \mathbb{F}$, $\Lambda^1(V) = V$ and $\Lambda^p(V)$ contains the sums of products $v_1 \wedge \cdots \wedge v_p$. The Grassmann algebra over vector space V is then $\Lambda(V) = \oplus_{p=0}^n \Lambda^p(V)$. In the algebra $\mathcal{L}(\Lambda(V))$ of linear transformations of $\Lambda(V)$ map M_v is defined as the linear extension of $M_v(1) = v$, and $M_v(v_1 \wedge \cdots \wedge v_p) = v \wedge v_1 \wedge \cdots \wedge v_p$. Another map, δ_v, is defined as the linear extension of $\delta_v(1) = v$, and $\delta_v(v_1 \wedge \cdots \wedge v_p) = \sum_{k=1}^p (-1)^{k-1} B(v, v_k)\, v_1 \wedge \cdots \wedge v_k' \wedge$

$\cdots \wedge v_p$, where v_k' denotes the term to be omitted from the product, and where $B(\cdot, \cdot)$ is the Minkowski bilinear form. Thus, M_v is exterior multiplication by v and δ_v is interior multiplication with respect to the inner product induced on $V \times V$ by $B(\cdot, \cdot)$. Define $\eta \in V \to \mathcal{L}(\Lambda(V))$, $v \mapsto M_v + \delta_v$. The subalgebra of $\mathcal{L}(\Lambda(V))$ defined by $\{\eta(v) \mid v \in V\}$ and $\{\lambda 1 \mid \lambda \in \mathbb{F}\}$ is a universal Clifford algebra for $(V, Q)$where the quadratic form Q is subject to the condition $\eta(v)^2 = (M_v + \delta_v)^2 = Q(v)I$.

Let the (metric compatible) covariant derivative be mapped by functor C from the tensor bundle to the Clifford bundle. The image of the covariant derivative in the Clifford bundle is denoted by ∇. The codomain of map $\nabla \in \mathcal{L}(\Lambda(V)) \to \mathcal{L}(\Lambda(V))$ can be decomposed into components corresponding to the exterior and interior multiplication, and consequently we may write $\mathrm{cod}(\nabla) = \mathrm{cod}(\nabla_e) \oplus \mathrm{cod}(\nabla_i)$. If the covariant derivative is mapped to the exterior bundle with functor D, then the image of the covariant derivative is $\mathrm{d} \pm \star\mathrm{d}\star$, where the sign depends on grade p and dimension n.

Tensor bundles and tensor algebra provide us with a starting point general enough for the theory needed in software design for ordinary gauge theories. The theory should not, however, be tied to the category of sets. We seek a category that just condenses the essentials of differentiation, of the metric properties of space-time, and of the action principle. This approach resembles reverse mathematics in the sense that we are looking for a minimal set of assumptions needed to define the theory. Software based on such assumptions should not be bound to any specific algebra. This enables end-users to employ algebras that fit their needs the best.

4 Some Numerical Experiments with Space-Time Models

In our earlier papers, we have demonstrated the proposed approach with several numerical experiments. Such experiments include simulations with acoustic, elastodymanic, electromagnetic and quantum mechanic waves [25–28]. For the extended accuracy, the mesh structures play an essential role [26]. The numerical scheme can also be optimized by locally adaptive time-stepping and by tuning the discrete Hodge operator, e.g., for time-harmonic waves. In certain cases such optimizations can improve the efficiency of the simulation even by orders of magnitude as reported in [25].

The formulation of the general model in Minkowski space provides additional benefits. It is namely possible to simulate the wave propagation in moving (and even deforming) spatial domains. In the papers [25–27] and [28], the spatial mesh generation together with the associated spatial finite difference approximation and time-stepping were considered as separate entities, without emphasizing the fact that the usual leap-frog time integration scheme for first order systems could also be derived from geometrical principles analogous to the spatial mesh generation which is based on the Delaunay–Voronoi duality.

4.1 Transforming Cavity

This chapter contains numerical experiments that demonstrate how the general model of the Minkowski space can be discretized. The construction of the space-time model begins by creating a mesh that fills the entire space-time domain. When generating a mesh, one should ensure that a valid dual mesh is available. The dual mesh is made up of cells each having an orthogonal counterpart in the (primal) mesh. Orthogonality is defined such that the Minkowskian bilinear form between any vector from a primal cell and any vector from the corresponding dual cell equals to zero.

Figure 5 illustrates the solution of the one-dimensional time-dependent wave problem in a moving cavity. We build a simplicial mesh in space-time in dimension two. Then we attach a floating point number to each primal 1-cell (edge) to construct a discrete version of F including only 1-form term. The initial values are set at time $t = 0$ (at the bottom of the figure) to trigger a wave pulse. Elsewhere, the values of F are explicitly solved by following the equation $dF = 0$. Since the dual mesh and the discrete Hodge operators are constructed using Minkowskian metric, the solution is a traveling wave with a propagation speed of 1 in both directions (see the right-hand-side of the Fig. 5).

Figure 6 shows a numerical experiment where the same approach has been applied to solve a two-dimensional time-dependent wave problem in a rotating cavity. In this case, the mesh is three-dimensional and the shape of the two-

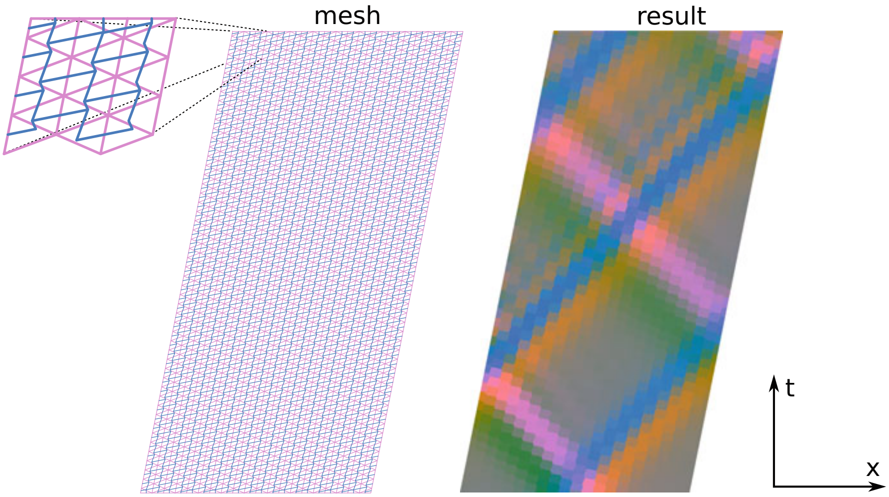


Fig. 5 A space-time approach to solve of a time-dependent wave problem in a moving cavity: The mesh with simplicial cells (purple edges) and corresponding Minkowski dual mesh (blue edges) are illustrated on the left. The solution of a wave problem is shown on the right. The color components red and green correspond to dx and dt components of the resulting 1-form. The colors are normalized such that grey indicates the zero field

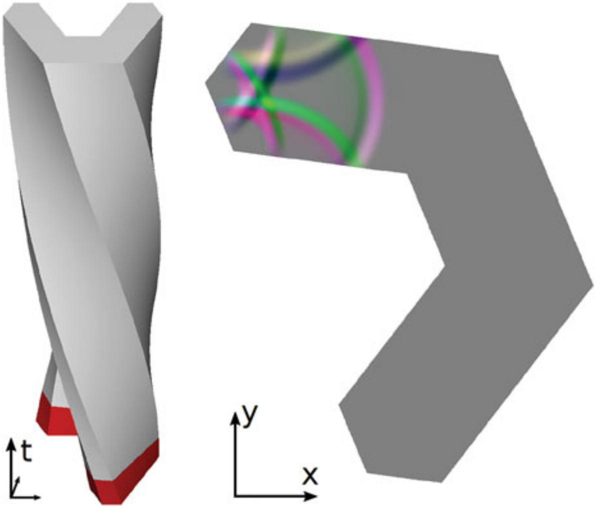


Fig. 6 Simulation of wave propagation in rotating two-dimensional cavity: The (2 + 1)-dimensional space-time mesh is illustrated on the left. The red color (dark) at the bottom indicates the past time and the cross section of the mesh at the current time is shown on the right. The color components red, green, and blue represent the components $dx \wedge dy$, $dx \wedge dt$, and $dy \wedge dt$ of the resulting 2-form, respectively. The figure is normalized such that the grey color indicates the zero field

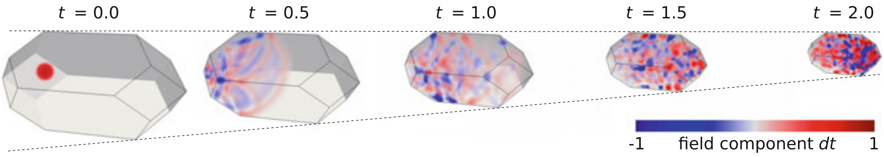


Fig. 7 Wave propagation in a shrinking three-dimensional cavity: Cross-section of the space-time mesh and dt-component of the resulting field are presented at five instances of time

dimensional base mesh (spatial cross-section of the space-time mesh) resembles a boomerang. The space-time mesh is twisted around the time axis, causing the cross-section to rotate as time progresses. The field F to be solved is a 2-form which is discretized by attaching one floating-point number to each 2-cell (face) of the mesh. By initializing F as an impulse at the initial time and solving $dF = 0$ inside the computational domain, we detect a wavefront propagating at speed 1 and reflecting from the moving walls. A video of this numerical experiment can be found at the following url: https://urly.fi/1oxH.

To prove the generalizability of the method, we present yet another experiment where we solve a three-dimensional acoustic-like wave problem in a shrinking computational domain. We create a (3 + 1)-dimensional simplicial mesh that, at time $t = 0$, fills a three-dimensional spatial volume as illustrated on the left of Fig. 7. The element lengths of the mesh are proportional to the term $1 - 0.3t$. This means that the element sizes decrease exponentially in terms of the number of time steps. The point $(0, 0, 0, \frac{1}{3})^T$ of convergence is never reached in the simulation.

In order to reduce the amount of memory required, the mesh duration over time is chosen as short as possible. We integrate 1-form F over mesh by explicitly solving $dF = 0$. When integration over mesh is completed, the last calculated terms are copied as the initial values of the next iteration and the integration is repeated. In this way, the task can be integrated as long as desired, without having to store the entire mesh in memory. The resulting field of time-integration is illustrated in Fig. 7. A video of this numerical experiment can be found online at url: https://urly.fi/1oWx, and the source codes of simulations of Figs. 6 and 7 can be found at url: https://github.com/juolrabi/gfd.

4.2 Local Time-Stepping and Stability

Traditionally, the Courant–Friedrichs–Lewy (CFL) condition sets an upper limit for the length of maximal time step. The smaller the spatial element size is, the shorter the time step must be in order to achieve numerical stability. When the spatial element length is not constant, local time-stepping can speed up the integration of time-dependent wave problems. This section shows how to create local time-stepping methods using the space-time integration.

Let's start with a $(1 + 1)$-dimensional example and consider a one-dimensional spatial mesh consisting of unevenly distributed nodes and line-segments (edges) between them. Nodes of the spatial mesh are copied at regular intervals in the time direction using individual step sizes Δt for the nodes. The length of the time step is set to the maximum length that obeys the inequality $\Delta t < c\Delta x$, where Δx is the length of the shortest neighboring edge and c is a constant. The space-time structure is completed as the Delaunay mesh. The mesh is 2 units wide in spatial direction and 1 unit high in time-direction. The mesh and its dual mesh are illustrated in Fig. 8.

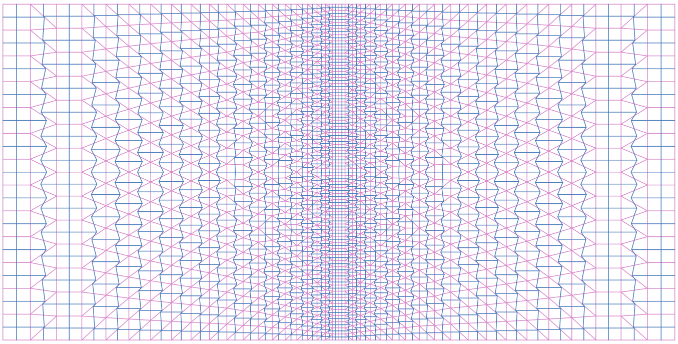


Fig. 8 A $(1 + 1)$-dimensional mesh with variable spatial edge lengths Δx. The condition for time step size is $\Delta t < 1.0\Delta x$. Primal and dual edges are illustrated with purple and blue colors, respectively

The field under consideration is a 1-form and it is formatted and integrated in the same way as in the previous section. When the integration over the mesh is completed, the last calculated terms are copied as the initial values and the integration is repeated. In this way, we are able to reuse the mesh again and integrate over time as long as necessary.

We consider stability of the time-integration in long term simulations with two different constants $c = 1.0$ and $c = 0.9$. The results are illustrated in Fig. 9. The conclusion is that the time integration is not stable with the constant of $c = 1.0$. The noise in the resulting field is visible already after 50 iterations. However, with the constant of $c = 0.9$, the system is stable because no dispersion is detectable even after 200 iterations. The condition used for the time step length seems to be a good first guess to replace the CFL condition in the asynchronous space-time integration.

We also investigate local time-stepping in a $(2 + 1)$-dimensional wave problem. The mesh is constructed by creating a two-dimensional circular base mesh with varying element sizes. We limit the individual time step Δt of each node by the relation $\Delta t < c\Delta x$, where Δx is length of the shortest spatial edge next to the node. The structure of the space-time mesh is determined as a Delaunay mesh and a truncated mesh is visualized in Fig. 10.

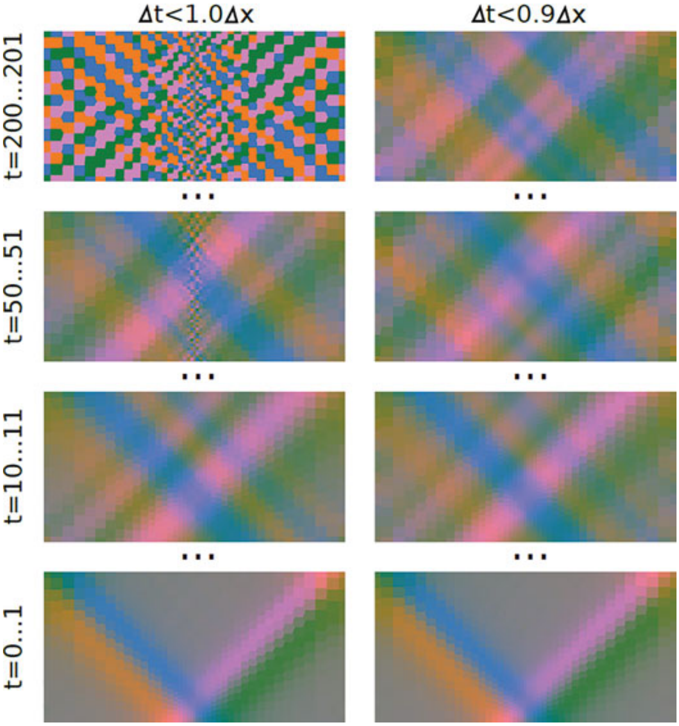


Fig. 9 The resulting fields during the various stages of integration. The color components red and green correspond to dx and dt components of the field and grey color indicates the zero field

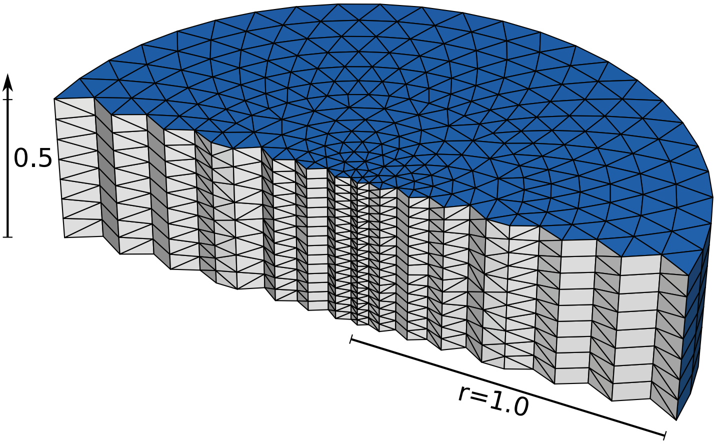


Fig. 10 A (2 + 1)-dimensional mesh with variable spatial edge lengths Δx and with condition $\Delta t < 0.6\Delta x$ for the time step size

We integrate 1-form over time and consider the numerical stability of long-term simulations. From Fig. 11, we find that by limiting the time step length with the constant of $c = 0.7$, numerical stability is not achieved. We observe noise in the resulting field already after 5 units of time. Instead, using the constant $c = 0.6$, we keep the integration stable and do not observe any dispersion in the resulting field even after 100 units of time.

5 Conclusions

In this paper we have considered the common structure of boundary value problems. The structure is based on ordinary gauge theories on form bundles. We have presented models from classical and modern physics as particular examples of the system. The finite-dimensional models are constructed with generalized finite-difference, that is, discrete exterior calculus (DEC) type of approach. The pair of cell complexes is based on Delaunay–Voronoi duality with Minkowskian metric. A consistent construction of the discrete Hodge operator enables also moving deformable domains. Adaptive time stepping can also be implemented by utilizing the geometry of space-time mesh. Numerical results show that the software system based on the systematic structure is applicable in boundary value problems in one, two, and three spatial dimensions.

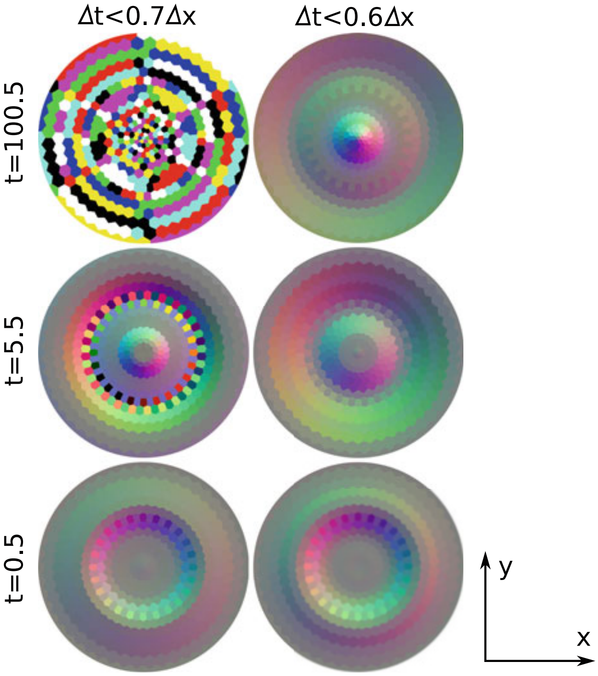


Fig. 11 Cross-sections of fields at different time instances and under different conditions for a time step length. The color components red, green, and blue correspond to dx, dy, and dt components of the resulting field, respectively. The grey color indicates the zero field

References

1. R. Abraham and J. E. Marsden. *Foundations of mechanics, 2 ed*. Addison-Wesley, 1987.
2. M.F. Atiyah. *Duality in Mathematics and Physics*. http://www.iecl.univ-lorraine.fr/~Wolfgang.Bertram/Atiyah-Duality.pdf.
3. J. Baez and J.P. Muniain. *Gauge Fields, Knots and Gravity*. Series on Knots and Everything, vol. 4, World Scientific, 1994.
4. D. Bleecker. *Gauge theories and variational principles*. Addison-Wesley, 1981.
5. A. Bossavit. *Computational electromagnetism*. Academic Press, 1997.
6. A. Bossavit. Compel 20 (2001), no. 1, 233–239.
7. A. Bossavit and L. Kettunen. International Journal of Numerical Modelling: Electronic Networks, Devices and Fields 12 (1999), 129–142.
8. P. G. Ciarlet. *The finite element method for elliptic problems*. Artech House Publishers, 1978.
9. Comsol Inc. *COMSOL Multiphysics*. http://www.comsol.com.
10. P. Dular and C. Geuzaine. *GetDP a General Environment for the Treatment of Discrete Problems*. http://getdp.info.
11. R. P. Feynman et al. *The Feynman lectures on physics, vol. 2 & 3*. Addison-Wesley Pub. Co., 1963.
12. H. Flanders. *Differential forms with application to the physical sciences*. Dover,1989.
13. T. Frankel. *The geometry of physics, an introduction, 3. ed*. Cambridge Univ. Press, 2012.
14. A. Frölicher and A. Nijenhuis. Indagationes Mathematicae (Proceedings) 59 (1956), 338–350.

15. J. E. Gilbert and M. A. M. Murray. *Clifford algebras and Dirac operators in harmonic analysis, vol.26*. Cambridge Univ. Press, 1991.
16. A. N. Hirani. *Discrete exterior calculus*. Ph.D. thesis, Caltech, Pasadena, California, 5 2003.
17. W. V. D. Hodge. Proc. London Math. Soc. 36 (1934), 257–303.
18. W. V. D. Hodge. *The theory and applications of harmonic integrals*. Cambridge Univ. Press, 1941.
19. E. Kanso, M. Arroyo, Y. Tong, A. Yavari, J. E. Marsden, and M. Desbruni. Z. Angew. Math. Phys. 58 (2007), 1–14.
20. L. Kettunen, S. Mönkölä, J. Parkkonen, and T. Rossi. arXiv:1908.10634v1, Submitted.
21. S. Kobayashi and K. Nomizu. *Foundations of differential geometry, vol. 1*. Wiley Interscience, 1963.
22. H. Lindqvist et al. Journal of Quantitative Spectroscopy and Radiative Transfer 217 (2018), 329–337.
23. nLab authors. *Gauge theory*. http://ncatlab.org/nlab/show/gauge%20theory, August 2019, Revision 56.
24. P. Petersen. *Riemannian geometry, 3. ed*. Springer, 2016.
25. J. Räbinä et al. SIAM Journal on Scientific Computing, 37 (2015), no. 6, B834-B854.
26. J. Räbinä et al. ESAIM: Mathematical Modelling and Numerical Analysis 52 (2018), no. 3, 1195–1218.
27. J. Räbinä et al. Phys. Rev. A. 98 (2018), no. 2.
28. J. Räbinä et al. Journal of Quantitative Spectroscopy and Radiative Transfer 178 (2016), 295–305.
29. R. Segev. Arch. Ration. Mech. Anal. 154 (2000), 183–198.
30. R. Segev. J. Math. Phys. 43 (2002), 3220–3231.
31. R. Segev and G. Rodnay. J. Elasticity 56 (1999), 129–144.
32. P. Stefanov and G. Vodev. Commun. Math. Phys. 176 (1996), 645–659.
33. J. A. Stratton. *Electromagnetic theory*. McGraw-Hill Company, 1941.
34. A. Taflove and S. C. Hagness. *Computational electrodynamics: The finite-difference time-domain method, 3. ed*. North-Holland Publishing Company, 2005.
35. T. Tao and G. Tian. J. Am. Math. Soc. 17 (2004), no. 3, 557–593.
36. T. Tarhasaari et al. IEEE Transactions on Magnetics 35 (1999), no. 3, 1494–7.
37. C. N. Yang. Physics today 67 (2014), no. 11, 45–51.
38. C. N. Yang and R. Mills. Phys. Rev. 96 (1954), no. 1, 191–195.
39. K. Yee. IEEE Transactions on Antennas and Propagation 14 (1966), 302–307.

On the Convergence of Flow and Mechanics Iterative Coupling Schemes in Fractured Heterogeneous Poro-Elastic Media

Tameem Almani, Kundan Kumar, and Abdulrahman Manea

Abstract In this work we establish the convergence of an adaptation of the fixed-stress split coupling scheme in fractured heterogeneous poro-elastic media. Here, fractures are modeled as possibly non-planar interfaces, and the flow in the fracture is described by a lubrication type system. The flow in the reservoir matrix and in the fracture are coupled to the geomechanics model through a fixed-stress split iteration, in which mass balance equations (for both flow in the matrix and in the fracture) are augmented with fixed-stress split regularization terms. The convergence proof determines the appropriate localized values of these regularization terms.

1 Introduction

The coupling of flow and mechanics is required to simulate different natural and induced physical phenomena including reservoir deformation, pore collapse, wellbore stability, fault activation, and hydraulic fracturing [2]. Fractures have significant effects on reservoir flow profiles. Moreover, the fractures are also the vulnerable regions for mechanical integrity of the system. Therefore, it is important to study the coupled flow and mechanics in a fractured heterogeneous (flow parameters are given functions of spatial variables) media. In this work, we extend the previous results for iterative coupling approaches in fractured medium to fractured—heterogeneous porous matrix system. In particular, we establish the convergence of an adaptation of the fixed-stress split scheme in heterogeneous poro-elastic media. The convergence of different iterative coupling schemes, including the fixed-stress, fixed-strain, drained, and undraind split schemes, was established in

T. Almani (✉) · A. Manea
Saudi Aramco, Dhahran, Saudi Arabia
e-mail: tameem.almani@aramco.com; almanitm@utexas.edu; abdulrahman.manea@aramco.com

K. Kumar
Department of Mathematics, University of Bergen, Bergen, Norway
e-mail: kundan.kumar@uib.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_4

the work of [3, 7–9]. Interpreting the fixed-stress split scheme as a preconditioner for the simultaneously coupled system was presented in the work of [10, 11]. Multirate extensions of the fixed-stress and undrained split iterative schemes were established in the work of [12]. Moreover, multiscale and nonlinear extensions of the fixed-stress split scheme were formulated and analyzed in the work of [13] and [15]. A parallel in time extension of the scheme was established in the work of [16]. The convergence in heterogeneous media was established in [4, 14] for the fixed-stress split scheme, and in [5]. Here, Banach fixed-point contraction results will be derived for this flow-mechanics coupled system by studying the equations satisfied by the difference of iterates. Geometric convergence to the unique solution of the system follows immediately as the sequence of iterates represents a convergent Cauchy sequence.

2 Model

We assume a linear, elastic, and isotropic fractured porous medium $\Omega \subset \mathbb{R}^3$, saturated with a slightly compressible single phase fluid. Following the lubrication fracture model, as described in [2], fractures are treated as non-planar interfaces denoted by $\mathcal{C}$ as shown in Fig. 1. As shown in Fig. 1, we introduce an auxiliary partition of Ω into two non-overlapping subdomains Ω^+ and Ω^-. The interface between the two subdomains is assumed to be Lipschitz and denoted by Γ. The fracture $\mathcal{C}$ is contained within Γ: $\mathcal{C} \subset \Gamma$. We will distinguish the two sides (or faces) of the fracture, $\mathcal{C}$, by the superscripts $+$ and $-$, and we will use the superscript $\star$ to denote either $+$ or $-$. Let $\Omega^\star$ denote the part of Ω adjacent to $\mathcal{C}^\star$ and let $\boldsymbol{n}^\star$ denote the unit normal vector to $\mathcal{C}$ exterior to $\Omega^\star$, $\star = +, -$. The fracture is represented by two coincident sides/surfaces, so we have $\boldsymbol{n}^- = -\boldsymbol{n}^+$. Moreover, we let $\Gamma^\star = \partial\Omega^\star \backslash \Gamma$. For any function g defined in $\Omega \setminus \mathcal{C}$ with a trace, let $g^\star$ denote the trace of g on $\mathcal{C}^\star$, $\star = +, -$. The jump of g on $\mathcal{C}$ in the direction of $\boldsymbol{n}^+$ is defined by $[g]_{\mathcal{C}} = g^+ - g^-$.

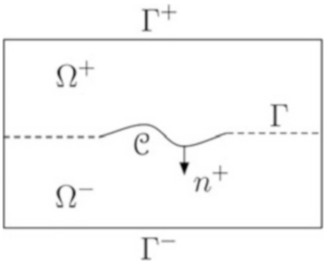


Fig. 1 Reservoir and fracture domains (image courtesy of [2])

We also assume a quasi-static Biot model for coupling flow with mechanics, ignoring the second order time derivative for the displacement. Our coupled model is as follows: Find $\boldsymbol{u}$, p_r, and p_f satisfying the equations below for all time $t \in]0, T[$:

$$\begin{aligned}
\text{Balance of Linear Momentum: } & -\operatorname{div} \boldsymbol{\sigma}^{\text{por}}(\boldsymbol{u}, p_r) = \boldsymbol{f} \quad \text{in } \Omega \setminus \mathcal{C} \\
\text{Cauchy Stress Tensor: } & \boldsymbol{\sigma}^{\text{por}}(\boldsymbol{u}, p_r) = \boldsymbol{\sigma}(\boldsymbol{u}) - \alpha\, p_r\, \boldsymbol{I} \\
\text{Effective Linear Elastic Stress Tensor: } & \boldsymbol{\sigma}(\boldsymbol{u}) = \lambda(\nabla \cdot \boldsymbol{u})\boldsymbol{I} + 2\, G\boldsymbol{\varepsilon}(\boldsymbol{u}) \\
\text{Reservoir Flow Model: } & \frac{\partial}{\partial t}\Big(\big(\frac{1}{M} + c_r\varphi_0\big)p_r + \alpha\nabla \cdot \boldsymbol{u}\Big) + \nabla \cdot \mathbf{Q}_r = \tilde{q} \quad \text{in } \Omega \setminus \mathcal{C}, \\
& \mathbf{Q}_r = -\frac{1}{\mu}\boldsymbol{K}\big(\nabla p_r - \rho g \nabla \eta\big) \quad \text{in } \Omega \setminus \mathcal{C}. \\
\text{Fracture Flow Model: } & c_f \frac{\partial p_f}{\partial t} + \frac{\partial}{\partial t} w + \overline{\nabla} \cdot \mathbf{Q}_f = \tilde{q}_W + [\mathbf{Q}_r]_{\mathcal{C}} \cdot \boldsymbol{n}^+ \quad \text{in } \mathcal{C}, \\
& \mathbf{Q}_f = -\frac{K_{\mathcal{C}}}{12\mu}(\overline{\nabla} p_f - \rho g \overline{\nabla} \eta) \quad \text{in } \mathcal{C}.
\end{aligned}$$

In the above, p_r and $\mathbf{Q}_r$ represent the pressure and flux unknowns in the reservoir matrix, p_f and $\mathbf{Q}_f$ represent the pressure and flux unknowns in the fracture (p_f is the trace of p_r on the fracture surface), and $\boldsymbol{u}$ is the the solid's displacement. In addition, $\boldsymbol{I}$ is the identity tensor, $\alpha > 0$ is the Biot coefficient, $\lambda > 0$ and $G > 0$ are the Lamé constants, $\boldsymbol{f}$ is a body force (in our case, the gravity loading term), $\mu > 0$ is the constant fluid viscosity, $\rho > 0$ is a constant reference density (relative to the reference pressure p_0), η is the distance in the vertical direction (pointing downwards), φ_0 is the initial porosity, M is the Biot constant, $\tilde{q} = \frac{q}{\rho}$ where q is a mass source or sink term, $\boldsymbol{K}$ and $\boldsymbol{K}_{\mathcal{C}}$ are the permeability tensors in the matrix and fracture respectively, c_r and c_f are the fluid compressibilities in the matrix and the fracture respectively, $\overline{\nabla}$ is the tangential derivative along the fracture, $w = -[\boldsymbol{u}]_{\mathcal{C}} \cdot \boldsymbol{n}^+$ represents the width of the fracture, $\tilde{q}_W = q_W/\rho$ is the injection term for flow in the fracture, and $[\mathbf{Q}_r]_{\mathcal{C}} \cdot \boldsymbol{n}^+$ is the leakage term which connects the flow in matrix to the flow in the fracture.

Notation In what follows, we will adopt the following notation: n denotes the flow/mechanics iterative coupling iteration index, k denotes the time step index. $\Delta t = t_k - t_{k-1}$ stands for the time step size, where $t_k = k\Delta t$, $0 \leqslant k \leqslant N$, and N is the total number of time steps, $T = N\Delta t$.

3 Fixed Stress Split in Fractured Poro-Elastic Media

Following the formulation of the fixed stress split scheme in fractured media as given in [1], we first solve the flow problem in the reservoir and the fracture in a monolithic manner (Step (a)), then we solve the mechanics problem (Step (b)), and we iterate:

Step (a) [Flow] Given $\boldsymbol{u}^n$, we solve for $p_r^{n+1}, \mathbf{Q}_r{}^{n+1}, p_f^{n+1}, \mathbf{Q}_f{}^{n+1}$

$$\left(\tfrac{1}{M} + c_r\varphi_0 + \tfrac{\alpha^2}{\lambda}\right)\tfrac{\partial}{\partial t}p_r^{n+1} - \nabla \cdot \mathbf{Q}_r{}^{n+1} = \tfrac{\alpha^2}{\lambda}\,\tfrac{\partial}{\partial t}p_r^n - \alpha\nabla \cdot \tfrac{\partial}{\partial t}\boldsymbol{u}^n + \tilde{q},$$
$$\mathbf{Q}_r{}^{n+1} = \tfrac{1}{\mu}\boldsymbol{K}\big(\nabla\, p_r^{n+1} - \rho g \nabla\, \eta\big)\text{in } \Omega \setminus \mathcal{C},$$

$$\big(\gamma_c + c_f\big)\,\tfrac{\partial}{\partial t}\, p_f^{n+1} + \tfrac{\partial}{\partial t}w^n - \overline{\nabla} \cdot \mathbf{Q}_f{}^{n+1} = \gamma_c\,\tfrac{\partial}{\partial t}p_f^n + \tilde{q}_W + [\mathbf{Q}_r]_{\mathcal{C}}^{n+1} \cdot \boldsymbol{n}^+,$$
$$\mathbf{Q}_f{}^{n+1} = \tfrac{K_{\mathcal{C}}}{\mu}(\overline{\nabla}\, p_f^{n+1} - \rho g \overline{\nabla}\, \eta) \text{ in } \mathcal{C},$$

Step (b) [Mechanics] Given $p_r^{n+1}, \mathbf{Q}_r{}^{n+1}, p_f^{n+1}, \mathbf{Q}_f{}^{n+1}$, we solve for $\boldsymbol{u}^{n+1}$ satisfying

$$-\operatorname{div}\boldsymbol{\sigma}^{\text{por}}(\boldsymbol{u}^{n+1}, p_r^{n+1}) = \boldsymbol{f} \text{ in } \Omega \setminus \mathcal{C}$$
$$\boldsymbol{\sigma}^{\text{por}}(\boldsymbol{u}^{n+1}, p_r^{n+1}) = \boldsymbol{\sigma}(\boldsymbol{u}^{n+1}) - \alpha\, p_r^{n+1}\,\boldsymbol{I} \text{ in } \Omega \setminus \mathcal{C}$$
$$(\boldsymbol{\sigma}^{\text{por}}(\boldsymbol{u}^{n+1}, p_r^{n+1}))^{\star}\boldsymbol{n}^{\star} = -p_f^{n+1}|_{\mathcal{C}}\boldsymbol{n}^{\star}\ ,\ \star = +, - \text{ on } \mathcal{C}$$

We note here that the mass conservation equations for the flow in the matrix and in the fracture are augmented with the fixed-stress split regularization terms ($\frac{\alpha^2}{\lambda}$ for the matrix, and γ_c for the fracture) which will vanish upon the convergence of the iteration for every time step, recovering the consistency of the scheme.

4 Convergence in Heterogeneous Media

4.1 Assumptions

In our analysis, we assume homogeneous elastic parameters (G, λ), and heterogeneous flow parameters. The spatial domain of the reservoir matrix will be denoted by $\Omega \subset \mathbb{R}^d$, d = 1, 2, or 3, and the spatial domain of the fracture will be denoted by $\mathcal{C} \subset \mathbb{R}^{d-1}$. Furthermore, The matrix is discretized into N_Ω conforming grid elements $E_i(\Omega)$ such that: $\overline{\Omega} = \bigcup_{i=1}^{N_\Omega} E_i(\Omega)$. Similarly, The $d-1$ fracture is discretized into $N_{\mathcal{C}}$ conforming grid elements $E_i(\mathcal{C})$ such that: $\overline{\mathcal{C}} = \bigcup_{i=1}^{N_{\mathcal{C}}} E_i(\mathcal{C})$. By heterogeneous flow parameters, we mean that each matrix grid element $E_i(\Omega)$ has its own, independent, set of flow parameters: $\boldsymbol{K}_i, M_i, c_r$ and φ_{0_i}. In a similar manner, each fracture grid element $E_i(\mathcal{C})$ has its own, independent, set of flow parameters: $\boldsymbol{K}_{\mathcal{C}_i}, c_{f_i}$, and μ. Here we denote, $\boldsymbol{K}_i = \boldsymbol{K}_i/\mu$, $\boldsymbol{K}_{\mathcal{C}_i} = \boldsymbol{K}_{\mathcal{C}_i}/\mu$.

Furthermore, the outward normal vector for each grid element E_i will be denoted by $\boldsymbol{n}_i$ and for every two adjacent grid elements E_i & E_{i-1}, $\boldsymbol{n}_i = -\boldsymbol{n}_{i-1}$ across the common boundary (for both matrix and the fracture).

4.2 Discretization

For spatial discretization, we assume a mixed finite element discretization for the flow in the reservoir matrix and the fracture, and a Conformal Galerkin method for mechanics. For temporal discretization, we assume a simple Backward-Euler scheme. Let $\mathfrak{T}_h$ denote a regular family of conforming triangular elements of the domain of interest, $\overline{\Omega}$. Using the lowest order RT (Raviart and Thomas, 1977) spaces, we have the discrete spaces [2]:

$$\begin{aligned}\boldsymbol{u}_h \in \boldsymbol{V}_h = \{&\boldsymbol{v}_h \in H^1(\Omega^+ \cup \Omega^-)^d\,;\ \forall T \in \mathfrak{T}_h, \boldsymbol{v}_{h|T} \in {\mathbb{P}_1}^d, [\boldsymbol{v}_h]_{\Gamma \setminus \mathcal{C}} = \mathbf{0},\\ &\boldsymbol{v}_{h|\Gamma^\star}^{\star} = \mathbf{0}, \star = +, -\}\end{aligned}$$

$$p_{r_h} \in Q_h = \{p_{r_h} \in L^2(\Omega)\,;\ \forall T \in \mathfrak{T}_h, p_{r\,h|T} \in \mathbb{P}_0\}$$

$$p_{f_h} \in Q_{c_h} = \{p_{f_h} \in H^{1/2}(\mathcal{C}_h)\,;\ \forall T \in \mathfrak{T}_h, p_{c_{h\,|T}} \in \mathbb{P}_1\}$$

$$\begin{aligned}\mathbf{Q}_{r\,h} \in \boldsymbol{Z}_h = \{&\boldsymbol{q}_h \in H(\text{div};\, \Omega^+ \cup \Omega^-)^d\,; \forall T \in \mathfrak{T}_h, \boldsymbol{q}_{h|T} \in {\mathbb{P}_1}^d,\\ &[\boldsymbol{q}_h] \cdot \boldsymbol{n}^+ = 0 \text{ on } \Gamma \setminus \mathcal{C} \quad \boldsymbol{q}_h \cdot \boldsymbol{n} = 0 \text{ on } \partial\Omega\}\end{aligned}$$

$$\mathbf{Q}_{f\,h} \in \boldsymbol{Z}_{\mathcal{C}h} = \{\boldsymbol{\mu}_{f\,h} \in \boldsymbol{Z}_{\mathcal{C}}\,;\ \forall T \in \mathfrak{T}_h, \boldsymbol{\mu}_{f\,h|T} \in {\mathbb{P}_1}^d\}$$

where $\mathbb{P}_0$, and $\mathbb{P}_1$ are the spaces of polynomials of degrees zero and one respectively, and ${\mathbb{P}_1}^d$ is the space of polynomials of degree one in $\mathbb{R}^d$. In the above, $\boldsymbol{Z}_{\mathcal{C}}$ represents the space of continuous velocities in the fracture: $\boldsymbol{Z}_{\mathcal{C}} = \{\boldsymbol{\mu}_f \in L^2(\mathcal{C})^{d-1}\,; \overline{\nabla} \cdot \boldsymbol{\mu}_f \in H^{-1/2}(\mathcal{C})\}$. It is normed by: $\|\boldsymbol{q}_f\|_{\boldsymbol{Z}_{\mathcal{C}}} = \big(\|\boldsymbol{q}_f\|^2_{L^2(\mathcal{C})} + \|\overline{\nabla} \cdot \boldsymbol{q}_f\|^2_{H^{-1/2}(\mathcal{C})}\big)^{1/2}$. In addition, the space Q_{c_h} is equipped with the norm: $\|v\|_{H^{1/2}(\mathcal{C})} = \big(\|v\|^2_{L^2(\mathcal{C})} + |v|^2_{H^{1/2}(\mathcal{C})}\big)^{1/2}$, where $|v|_{H^{1/2}(\mathcal{C})} = \big(\int_{\mathcal{C}}\int_{\mathcal{C}} \frac{|v(\boldsymbol{x})-v(\boldsymbol{y})|^2}{|\boldsymbol{x}-\boldsymbol{y}|^d} d\boldsymbol{x}\, d\boldsymbol{y}\big)^{1/2}$. Other spaces use the usual corresponding norms. We also note that the discrete leakage term $\tilde{q}_L$ is in the same space as the discrete fracture flux space ($\frac{1}{\mu}[\mathbf{Q}_{r\,h}]_{\mathcal{C}} \cdot \boldsymbol{n}^+ = -\tilde{q}_L$ on $\mathcal{C}$). We also assume that the solution at time t_{k-1} to be known (the values of $\boldsymbol{u}^{k-1}$, ${p_r}^{k-1}$, ${p_f}^{k-1}$, ${\mathbf{Q}_r}^{k-1}$, and ${\mathbf{Q}_f}^{k-1}$ are computed from last time step) with given corresponding initial values for the first time step. Furthermore, if the domain of integration is not indicated, it is understood to be over $\Omega^+ \cup \Omega^-$.

Now, we list the Banach contraction result for the fixed stress split scheme described in Sect. 3 in heterogeneous poroelastic media.

5 Banach Contraction Result

Theorem 1 (Localized Contraction Estimate for Fractured Heterogeneous Media) *For* $\gamma_{c_i} = \frac{2c_{f_i}}{(\lambda-2)}$ *and* $\chi_i = \left(\frac{\gamma_{c_i}}{\lambda}\right)^{1/2}$ *for each* $E_i \in \mathcal{C}$*, and if the conditions* (6)*,* (7)*, and* (8) *are satisfied, and for homogeneous elastic parameters* G *and* λ*, and heterogenous (localized) flow parameters, the localized iterative scheme is a contraction given by*

$$\begin{aligned}
&\sum_{i=1}^{N_\Omega} \left\| \delta\sigma_v^{n+1,k} \right\|_{E_i(\Omega)}^2 + \sum_{i=1}^{N_\mathcal{C}} \left\| \delta\sigma_f^{n+1,k} \right\|_{E_i(\mathcal{C})}^2 + \sum_{i=1}^{N_\Omega} \lambda^2 \left\| \nabla \cdot \delta\boldsymbol{u}_h^{n+1,k} \right\|_{E_i(\Omega)}^2 \\
&+ \sum_{i=1}^{N_\Omega} 2\Delta t \left\| K_i^{-1/2} \delta\boldsymbol{Q}_{r\,h}^{\,n+1,k} \right\|_{E_i(\Omega)}^2 + \sum_{i=1}^{N_\mathcal{C}} \frac{\Delta t}{6} \left\| K_{\mathcal{C}_i}^{-1/2} \delta\boldsymbol{Q}_{f\,h}^{\,n+1,k} \right\|_{E_i(\mathcal{C})}^2 \\
&+ \sum_{i=1}^{N_\mathcal{C}} \left(4G\lambda C^* - \frac{\lambda(\lambda-2)}{2c_{f_i}} \right) \left\| \delta w_h^{n+1,k} \right\|_{E_i(\mathcal{C})}^2 + \sum_{i=1}^{N_\Omega} \alpha_i^2(\beta_i - 1) \left\| \delta\sigma_v^{n+1,k} \right\|_{E_i(\Omega)}^2 \cdots \\
&\qquad\qquad \leq \max\left\{ \max_{1\leq i\leq N_\Omega} \left(\frac{1}{\beta_i\lambda^2}\right), \max_{1\leq i\leq N_\mathcal{C}} \left(\frac{\chi_i^2}{\beta_{c_i}}\right) \right\} \left(\sum_{i=1}^{N_\Omega} \left\| \delta\sigma_v^{n,k} \right\|_{E_i(\Omega)}^2 + \sum_{i=1}^{N_\mathcal{C}} \left\| \delta\sigma_f^{n,k} \right\|_{E_i(\mathcal{C})}^2 \right).
\end{aligned}$$

The contraction coefficient can be shown to be <1.

5.1 Proof

We will follow the same steps as outlined in the work of [4, 5], and as outlined below:

- **Step (1):** Write the continuous-in-space weak formulation locally for each grid element.
- **Step (2):** Sum up these local weak formulations to get a global weak formulation. All inner boundary terms will get cancelled.
- **Step (3):** Write a corresponding discrete-in-space global weak formulation by mimicking the continuous-in-space global weak formulation.
- **Step (4):** Match coefficients as in the homogeneous case to ensure contraction.

We proceed directly to the third step as the first two steps are identical to the ones employed in [4, 5].

Step (3): Fully Discrete Weak Form
Now, we mimic the continuous-in-space global weak formulation to reach to the fully discrete weak form as follows:

Flow Solve Find $\boldsymbol{u}_h^{n+1,k} \in \boldsymbol{V}_h$, $p_{r_h}^{n+1,k} \in Q_h$, $p_{f_h}^{n+1,k} \in Q_{c_h}$, $\mathbf{Q}_{r\,h}^{\,n+1,k} \in \boldsymbol{Z}_h$, and $\mathbf{Q}_{f\,h}^{\,n+1,k} \in \boldsymbol{Z}_{\mathcal{C}h}$ such that:

$$\forall \theta_h \in Q_h\,,\ \sum_{i=1}^{N_\Omega}\Big(\big(\frac{1}{M} + c_{r_i}\varphi_{0i} + \frac{\alpha_i^2}{\lambda}\big)\big(\frac{p_{r_h}^{n+1,k} - p_{r_h}^{k-1}}{\Delta t}\big), \theta_h\Big)_{E_i(\Omega)} + \sum_{i=1}^{N_\Omega}\big(\nabla\cdot\mathbf{Q}_{r\,h}^{\,n+1,k}, \theta_h\big)_{E_i(\Omega)} =$$

$$\sum_{i=1}^{N_\Omega}\Big(\frac{\alpha_i^2}{\lambda}\big(\frac{p_{r_h}^{n+1,k} - p_{r_h}^{k-1}}{\Delta t}\big) - \alpha_i\nabla\cdot\big(\frac{\boldsymbol{u}_h^{n+1,k} - \boldsymbol{u}_h^{k-1}}{\Delta t}\big), \theta_h\Big)_{E_i(\Omega)} + \sum_{i=1}^{N_\Omega}\big(\tilde{q}_h, \theta_h\big)_{E_i(\Omega)} \tag{1}$$

$$\begin{aligned}\forall \theta_{c_h} \in Q_{c_h}\,,\ &\sum_{i=1}^{N_\mathcal{C}}\Big(\big(c_{f_i} + \gamma_{c\,i}\big)\frac{p_{f_h}^{n+1,k} - p_{f_h}^{k-1}}{\Delta t}, \theta_{c_h}\Big)_{E_i(\mathcal{C})} + \sum_{i=1}^{N_\mathcal{C}}\frac{1}{12}\big(\overline{\nabla}\cdot(\mathbf{Q}_{f\,h}^{\,n+1,k}), \theta_{c_h}\big)_{E_i(\mathcal{C})} \\ &- \sum_{i=1}^{N_\mathcal{C}}\big([\mathbf{Q}_{r\,h}^{\,n+1,k}]_\mathcal{C}\cdot\boldsymbol{n}^+, \theta_{c_h}\big)_{E_i(\mathcal{C})} = \sum_{i=1}^{N_\mathcal{C}}\Big(\gamma_{c\,i}\,\frac{p_{f_h}^{n,k} - p_{f_h}^{k-1}}{\Delta t}, \theta_{c_h}\Big)_{E_i(\mathcal{C})} \\ &+ \Big(\frac{[\boldsymbol{u}_h^{n,k}]_\mathcal{C}\cdot\boldsymbol{n}^+ - [\boldsymbol{u}_h^{k-1}]_\mathcal{C}\cdot\boldsymbol{n}^+}{\Delta t}, \theta_{c_h}\Big)_{E_i(\mathcal{C})} + \big(q\tilde{_W}_h, \theta_{c_h}\big)_{E_i(\mathcal{C})}\end{aligned} \tag{2}$$

$$\begin{aligned}\forall \boldsymbol{q}_h \in \boldsymbol{Z}_h\,,\ \sum_{i=1}^{N_\Omega}\big(\boldsymbol{K}_i^{-1}\mathbf{Q}_{r\,h}^{\,n+1,k}, \boldsymbol{q}_h\big)_{E_i(\Omega)} &= \sum_{i=1}^{N_\Omega}(p_{r_h}^{n+1,k}, \nabla\cdot\boldsymbol{q}_h)_{E_i(\Omega)} - \sum_{i=1}^{N_\mathcal{C}}\big(p_{f_h}^{n+1,k}, [\boldsymbol{q}_h]_\mathcal{C}\cdot\boldsymbol{n}^+\big)_{E_i(\mathcal{C})} \\ &+ \sum_{i=1}^{N_\Omega}\big(\nabla(\rho g\eta), \boldsymbol{q}_h\big)_{E_i(\Omega)}\end{aligned} \tag{3}$$

$$\forall \boldsymbol{\mu}_{f\,h} \in \mathbf{Z}_{\mathcal{C}h}\,,\ \sum_{i=1}^{N_\mathcal{C}}\big(K_{\mathcal{C}_i}^{-1}\mathbf{Q}_{f\,h}^{\,n+1,k}, \boldsymbol{\mu}_{f\,h}\big)_{E_i(\mathcal{C})} = \sum_{i=1}^{N_\mathcal{C}}\big(p_{f_h}^{n+1,k}, \overline{\nabla}\cdot(\boldsymbol{\mu}_{f\,h})\big)_{E_i(\mathcal{C})} + \sum_{i=1}^{N_\mathcal{C}}\big(\overline{\nabla}(\rho g\eta), \boldsymbol{\mu}_{f\,h}\big)_{E_i(\mathcal{C})}. \tag{4}$$

The mechanics equations are standard and left for brevity. The system is complimented by the initial condition.

Step (4): Proceed as in The Homogeneous Case
Now, we proceed as in the homogeneous coefficients case. Let $\beta_i = (\frac{1}{M_i\alpha_i^2} + \frac{c_{r_i}}{\alpha_i^2}\varphi_{0i} + \frac{1}{\lambda})$, for each reservoir grid element $E_i(\Omega)$, and $\beta_{c_i} = c_{f_i} + \gamma_{c_i}$ for each fracture grid element $E_i(\mathcal{C})$. In what follows, we will take the difference between iterative coupling iterations for Eqs. (1)–(4), and mechanics discrete equation, and denote the corresponding differences in the unknowns as $\delta\xi^{n+1,k}$, where $\delta\xi^{n+1,k} = \xi^{n+1,k} - \xi^{n,k}$, in which ξ may stand for any unknown variable we are solving for. Now, following the same approach as outlined in [1, 6], for the flow part, we test (1), (2), (3), and (4) with $\delta p_{r_h}^{n+1,k}$, $\delta\mathbf{Q}_{r\,h}^{\,n+1,k}$, $\delta p_{f_h}^{n+1,k}$, and $\delta\mathbf{Q}_{f\,h}^{\,n+1,k}$ respectively, and combine the results. For the mechanics part, we test mechanics equation with

$\boldsymbol{v}_h = \delta \boldsymbol{u}_h^{n+1,k}$, and multiply the whole equation by 2λ (recall that G and λ are homogeneous throughout the whole domain). Further we use the following estimate [1],

$$C^\star \|w_h\|^2_{L^2(\mathcal{C})} = C^\star \|[\boldsymbol{u}_h]_{\mathcal{C}} \cdot \boldsymbol{n}^+\|^2_{L^2(\mathcal{C})} \leq \|\boldsymbol{\varepsilon}(\boldsymbol{u}_h)\|^2_{L^2(\Omega\backslash\mathcal{C})} \leq \|\boldsymbol{\varepsilon}(\boldsymbol{u}_h)\|^2_{L^2(\Omega)}$$

where $C^\star = (2C^2\big(\max(\mathcal{P}_{\Gamma^+}, \mathcal{P}_{\Gamma^-})^2 + 1\big)C_\kappa^2)^{-1}$, and C, $\mathcal{P}_\Gamma$, and C_κ denote respectively the constants of the trace, Poincaré, and Korn inequality in $\Omega^\star$, $\star = +, -$: $\|\boldsymbol{u}|^\star_\Omega\|_{L^2(\mathcal{C})} \leq C\|\boldsymbol{u}\|_{H^1(\Omega^\star)}$, $\|\boldsymbol{u}\|_{L^2(\Omega^\star)} \leq \mathcal{P}_{\Gamma^\star}|\boldsymbol{u}|_{H^1(\Omega^\star)}$, and $\|\boldsymbol{u}|_{H^1(\Omega^\star)} \leq C_\kappa\|\boldsymbol{\varepsilon}(\boldsymbol{u})\|_{L^2(\Omega^\star)}$. Now, we put together all the steps above, together with an application of Young's inequality. This gives

$$\sum_{i=1}^{N_\Omega} \Big\{ \Big\|\alpha_i \delta p_{r_h}^{n+1,k}\Big\|^2_{E_i(\Omega)} - 2\lambda\big(\alpha_i \delta p_{r_h}^{n+1,k}, \nabla \cdot \delta \boldsymbol{u}_h^{n+1,k}\big)_{E_i(\Omega)} + \lambda^2 \|\nabla \cdot \delta \boldsymbol{u}_h^{n+1,k}\|^2_{E_i(\Omega)} \Big\}$$

$$+\lambda^2 \|\nabla \cdot \delta \boldsymbol{u}_h^{n+1,k}\|^2_\Omega + 2\Delta t \sum_{i=1}^{N_\Omega} \|K_i^{-1/2} \delta \mathbf{Q}_{r\,h}^{\,n+1,k}\|^2_{E_i(\Omega)} + \frac{\Delta t}{6} \sum_{i=1}^{N_\mathcal{C}} \|K_{\mathcal{C}_i}^{-1/2} \delta \mathbf{Q}_{f\,h}^{\,n+1,k}\|^2_{E_i(\mathcal{C})}$$

$$+\sum_{i=1}^{N_\mathcal{C}} \Big\{ 2\beta_{c_i} \|\delta p_{f_h}^{n+1,k}\|^2_{E_i(\mathcal{C})} + 4G\lambda C^* \|\delta w_h^{n+1,k}\|^2_{E_i(\mathcal{C})} - 2\lambda\big(\delta p_{f_h}^{n+1,k}, \delta w_h^{n+1,k}\big)_{E_i(\Omega)} \Big\}$$

$$\leq \sum_{i=1}^{N_\Omega} \frac{1}{\beta_i \lambda^2} \Big\|\delta \sigma_v^{n,k}\Big\|^2_{E_i(\Omega)} + \sum_{i=1}^{N_\mathcal{C}} \frac{1}{\beta_{c_i}} \Big\|\gamma_{c_i} \delta p_{f_h}^{n,k} - \delta w_h^{n,k}\Big\|^2_{E_i(\mathcal{C})} \tag{5}$$

Let $\delta \sigma_f^{n,k}|_{E_i(\mathcal{C})} = \frac{\gamma_{c_i}}{\chi_i} \delta p_{f_h}^{n,k} - \delta w_n^k$ for a free parameter χ_i as determined below for each grid element in the fracture $E_i(\mathcal{C})$. Recall that $\delta \sigma_v^{n,k}|_{E_i(\Omega)} = \lambda \nabla \cdot \delta \boldsymbol{u}_h^{n,k} - \alpha_i \delta p_{r_h}^{n,k}$ for each grid element $E_i(\Omega)$ in the reservoir matrix. By matching the coefficients of the expanded squares of $\Big\|\delta \sigma_v^{n,k}\Big\|^2_{E_i(\Omega)}$ and $\Big\|\delta \sigma_f^{n,k}\Big\|^2_{E_i(\mathcal{C})}$ with the corresponding coefficients on the left hand side of (5), and for $\gamma_{c_i} = \frac{2c_{f_i}}{(\lambda-2)}$, $\chi_i = \left(\frac{\gamma_{c_i}}{\lambda}\right)^{1/2}$, we can show that the scheme contracts on $\Big\|\delta \sigma_v^{n,k}\Big\|^2_{E_i(\Omega)}$ and $\Big\|\delta \sigma_f^{n,k}\Big\|^2_{E_i(\mathcal{C})}$ locally for each reservoir and fracture grid elements $E_i(\Omega)$ and $E_i(\mathcal{C})$ respectively provided

$$\beta_i > \max(1, \frac{1}{\lambda^2}), \quad \text{for all} \quad E_i \in \Omega \quad \text{(Condition on Reservoir Flow).} \tag{6}$$

$$8GC^* > \frac{\lambda - 2}{c_{f_i}} \quad \text{for all} \quad E_i \in \mathcal{C} \quad \text{(Condition on Fracture Flow)} \tag{7}$$

$$\lambda^2 > 2. \quad \text{(Condition on Mechanics)} \tag{8}$$

With the above choices of γ_{c_i} and χ_i, the localized fixed stress regularization terms for the flow in the matrix and fracture are given by $\frac{\alpha_i^2}{\lambda}$, and $\frac{2c_{f_i}}{(\lambda-2)}$ respectively, the contraction result above is established.

References

1. V. Girault, K. Kumar, and M. F. Wheeler. Convergence of iterative coupling of geomechanics with flow in a fractured poroelastic medium. *Computational Geosciences*, 20 (5), 997–101, 2016.
2. V. Girault, M. F. Wheeler, B. Ganis, and M. E. Mear. A Lubrication Fracture Model in a Poro-elastic Medium. Mathematical Models and Methods in Applied Sciences. *Mathematical Models and Methods in Applied Sciences*, 25 (04), 587–645, 2015.
3. A. Mikelić and M. F. Wheeler. Convergence of iterative coupling for coupled flow and geomechanics. *Computational Geosciences*, 17:455–461, 2013.
4. T. Almani, K. Kumar, and M. F. Wheeler. Convergence Analysis of Single Rate and Multirate Fixed Stress Split Iterative Coupling Schemes in Heterogeneous Poroelastic Media. *Ices report 17–23*, Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas, 2017.
5. T. Almani, A. Manea, K. Kumar, and A. H. Dogru. Convergence of the undrained split iterative scheme for coupling flow with geomechanics in heterogeneous poroelastic media. *Computational Geosciences*, 24:551–569, 2020.
6. T. Almani. Efficient algorithms for flow models coupled with geomechanics for porous media applications. *PhD Dissertation*, Institute for Computational Engineering and Sciences, The University of Texas at Austin, Austin, Texas, 2017.
7. J. Kim, H. A. Tchelepi, and R. Juanes. Stability and convergence of sequential methods for coupled flow and geomechanics: drained and undrained splits In *Computer Methods in Applied Mechanics and Engineering*, 200(23–24):2094–2116, 2011.
8. J. Kim, H. A. Tchelepi, and R. Juanes. Stability and convergence of sequential methods for coupled flow and geomechanics: fixed-stress and fixed-strain splits In *Computer Methods in Applied Mechanics and Engineering*, 200(13–16):1591–1606, 2011.
9. J. Kim, H. A. Tchelepi, and R. Juanes. Stability, accuracy, and efficiency of sequential methods for coupled flow and geomechanics. In *The SPE Reservoir Simulation Symposium, Houston, Texas*, February 2–4, 2009. SPE119084.
10. X. Gai, R. H. Dean, M. F. Wheeler, and R. Liu. Coupled geomechanical and reservoir modeling on parallel computers. In *The SPE Reservoir Simulation Symposium, Houston, Texas*, Feb. 3–5, 2003.
11. N. Castelletto, J. A. White, and H. A. Tchelepi. Accuracy and convergence properties of the fixed-stress iterative solution of two-way coupled poromechanics. In *International Journal for Numerical and Analytical Methods in Geomechanics*, 39:1593–1618, 2015.
12. T. Almani, K. Kumar, A. Dogru, G. Singh, and M. F. Wheeler. Convergence analysis of multirate fixed-stress split iterative schemes for coupling flow with geomechanics. In *Computer Methods in Applied Mechanics and Engineering*, 311:180–207, 2016.
13. S. Dana, B. Ganis, and M. F. Wheeler. A multiscale fixed stress split iterative scheme for coupled flow and poromechanics in deep subsurface reservoirs. In *Journal of Computational Physics*, 352:1–22, 2018.
14. J. W. Both, M. Borregales, J. M. Nordbotten, K. Kumar, and F. A. Radu. Robust fixed stress splitting for Biots equations in heterogeneous media. In *Applied Mathematics Letters*, 68:101–108, 2017.

15. M. Borregales, F. A. Radu, K. Kumar, and J. M. Nordbotten. Robust iterative schemes for non-linear poromechanics. In *Computational Geosciences*, 17:1573–1499, 2018.
16. M. Borregales, K. Kumar, F. A. Radu, C. Rodrigo, F. J. Gaspar. A parallel-in-time fixed-stress splitting method for Biot's consolidation model. In *Computers and Mathematics with Applications*, 77:1466–1478, 2019.

Finite Difference Solutions of 2D Magnetohydrodynamic Channel Flow in a Rectangular Duct

Sinem Arslan and Münevver Tezer-Sezgin

Abstract The magnetohydrodynamic (MHD) flow of an electrically conducting fluid is considered in a long channel of rectangular cross-section along with the z-axis. The fluid is driven by a pressure gradient along the z-axis. The flow is steady, laminar, fully-developed and is influenced by an external magnetic field applied perpendicular to the channel axis. So, the velocity field $\mathbf{V} = (0, 0, V)$ and the magnetic field $\mathbf{B} = (0, B_0, B)$ have only channel-axis components V and B depending only on the plane coordinates x and y on the cross-section of the channel which is a rectangular duct. The finite difference method (FDM) is devised to solve the problem tackling mixed type of boundary conditions such as no-slip and insulated walls and both slipping and variably conducting walls. Thus, the numerical results show the effects of the Hartmann number Ha, the conductivity parameter c and the slipping length α on both of the velocity and the induced magnetic field, especially near the walls. It is observed that the well-known characteristics of the MHD flow are also caught.

1 Introduction

Magnetohydrodynamics is arisen from the main results of fluid mechanics and electrodynamics. It considers the flow of an electrically conducting fluid exposed to an external magnetic field and/or an electric current [6]. Thus, it investigates the influence of these external effects on the behavior of the flow of electrically conducting fluids. The study of magnetohydrodynamics is introduced by Hartmann [5] who studied the MHD flow between parallel planes and thanks to his results, there is an insight for understanding the working principles of MHD flow. MHD has applications in almost every area of our daily life and in engineering such as magnetic cooling systems, magnetic refrigerators, water treatment devices.

S. Arslan (✉) · M. Tezer-Sezgin
Department of Mathematics, Middle East Technical University, Ankara, Turkey
e-mail: arsinem@metu.edu.tr; munt@metu.edu.tr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_5

Basically, there are several devices whose working principles are based on MHD effects such as MHD pumps, generators, brakes, flow meters and blood flow measurement. Analytical solutions for MHD duct flow are available only for a simple geometry of the duct and simple wall conditions as insulated and no-slip velocity [4]. The main numerical studies of MHD flow problem come into play with the use of some methods such as FDM [2, 7], FEM [8, 11], and BEM [3, 10] again for no-slip and partly conducting partly insulated walls. FDM solution for the general mixed boundary conditions has been given in [1].

In this paper, the MHD flow of an electrically conducting fluid which is viscous and incompressible is considered in a long channel of rectangular cross-section (duct) and the flow is laminar, steady, and fully-developed in the channel-axis direction. Fluid starts to move with a pressure gradient in the z-direction. The interaction between the electrically conducting fluid and vertically applied external magnetic field induces also a magnetic field inside the fluid. Thus, the total magnetic and velocity fields become $\mathbf{B} = (0, B_0, B)$ and $\mathbf{V} = (0, 0, V)$ varying in the duct only, that is $V = V(x, y)$ and $B = B(x, y)$ since the flow is fully-developed. The governing equations of MHD duct flow are solved using FDM with the most general type of boundary conditions corresponding to slipping velocity and variably conducting walls. The influences of the slipping and the conductivity changes on the velocity and the induced magnetic field are illustrated with equivelocity and the current lines for increasing values of Hartmann number, slip length, and the conductivity parameter.

2 Mathematical Formulation

The governing MHD duct flow equations result from the combination of Navier-Stokes equations of hydrodynamics and Maxwell's equations of electromagnetism through Ohm's law. The pipe-axis components of the momentum and the magnetic induction equations give

$$\mu\nabla^2 V + \mu_e H_0 \frac{\partial H}{\partial y} = \tfrac{\partial P}{\partial z}, \tag{1}$$

$$\nabla^2 H + \sigma\mu_e H_0 \frac{\partial V}{\partial y} = 0. \tag{2}$$

Dimensionless variables are introduced as $V' = \frac{V}{U_0}$, $B' = \frac{B}{U_0\mu_e\sqrt{\sigma\mu}}$, $x' = \frac{x}{L_0}$ and $y' = \frac{y}{L_0}$, where $U_0 = -\frac{L_0^2}{\mu}\frac{\partial P}{\partial z}$ is the characteristic velocity and L_0 is the characteristic length and $\mathbf{H} = (0, H_0, H)$, P, σ, μ_e, μ are magnetic field, pressure, electrical conductivity, magnetic permeability, and the viscosity of the fluid, respectively. The coupled MHD equations governing the 2D channel flow in dimensionless form become

$$\nabla^2 V + Ha\frac{\partial B}{\partial y} = -1 \tag{3}$$

$$\nabla^2 B + Ha\frac{\partial V}{\partial y} = 0, \tag{4}$$

where $Ha = B_0 L_0 \sqrt{\sigma/\mu}$ is the Hartmann number and the domain $\Omega = \{-1 \leq x \leq 1, -1 \leq y \leq 1\}$ is the dimensionless cross-section of the duct and B_0 is the intensity of the external magnetic field. The problem is considered with the most general form of wall conditions such as slipping velocity and variably conducting walls.

$$V \pm \alpha\frac{\partial V}{\partial y} = 0, \quad B \pm c\tfrac{\partial B}{\partial y} = 0 \quad \text{when} \quad y = \pm 1, \tag{5}$$

$$V \pm \alpha\frac{\partial V}{\partial x} = 0, \quad B \pm c\tfrac{\partial B}{\partial x} = 0 \quad \text{when} \quad x = \pm 1. \tag{6}$$

Here, the constants α and c denote the slipping length of the velocity and the conductivity parameter, respectively. Thus, $c \to 0$ corresponds to electrically insulating walls and $c \to \infty$ to electrically perfectly conducting walls. Also, $\alpha = 0$ indicates that we have no-slip velocity at the duct walls.

3 Implementation of FDM and Boundary Conditions

The MHD flow equations (3)–(4) are coupled in V and B and should be solved together in Ω. Firstly, discretizing the MHD equations as a whole by central finite differences for both the Laplace operator ∇^2 and the convection operator $\partial/\partial y$ we obtain the following discretized equations

$$\begin{aligned} V_{i+1,j} - 4V_{i,j} + V_{i-1,j} + V_{i,j+1} + V_{i,j-1} + \tfrac{hHa}{2}\left(B_{i,j+1} - B_{i,j-1}\right) &= -h^2 \\ B_{i+1,j} - 4B_{i,j} + B_{i-1,j} + B_{i,j+1} + B_{i,j-1} + \tfrac{hHa}{2}\left(V_{i,j+1} - V_{i,j-1}\right) &= 0 \end{aligned} \tag{7}$$

for $i, j = 2, \ldots, N$. Here, N is the number of subintervals taken on each side and $h = 2/N$ is the step size. The approximation of mixed type boundary conditions (5–6) is carried in such a way that we use forward difference on the walls $x = y = -1$ and use backward difference on the walls $x = y = 1$ in order to define the boundary values in terms of inner mesh point values. Then, inserting the boundary conditions into the discretized equations (7), we obtain M unknowns in M equations where

$M = 2(N-1)^2$ for a general N. These equations are written in a matrix-vector system with the coefficient matrix Q of size $M \times M$. Thus, we have

$$Qx = w \tag{8}$$

where the unknown vector x of size $M \times 1$ is ordered as

$$x = \left[V_{2,2}\ B_{2,2} \cdots V_{2,N}\ B_{2,N}\ \cdots\ V_{N,2}\ B_{N,2} \cdots V_{N,N}\ B_{N,N}\right]^{\mathsf{T}}.$$

The right hand-side vector w of size $M \times 1$ is

$$w = \left[-h^2\ 0\ -h^2\ 0 \cdots -h^2\ 0\right]^{\mathsf{T}}.$$

The coefficient matrix Q of size $M \times M$ is a block diagonal matrix which includes two different block matrices Q_1 and Q_2 of sizes $2(N-1) \times 2(N-1)$ on the main diagonal. Also, the block matrices Q_1 and Q_2 are the matrices including the Hartmann number Ha, step-size h, slipping length α and the conductivity parameter c in their entries. Finally, the unknown vector x at the discretized points from the solution of the system (8) is obtained giving $V(x, y)$ and $B(x, y)$ at the mesh points.

4 Numerical Results and Discussion

The velocity and the induced magnetic field are simulated for increasing values of Ha, α and c. It is observed that we need to increase the number of nodes N with an increasing Ha since it causes convection dominance in the MHD equations. So, we use $N = 30, 40, 60, 80, 100$ with the corresponding values of $Ha = 5, 10, 30, 50, 100$. As Ha increases, boundary layers of $O(1/Ha)$ and of $O(1/\sqrt{Ha})$ are developed near the Hartmann (perpendicular) and side (parallel) walls for both V and B as the well-known behavior of MHD duct flow [4]. The slipping fluid is also observed on the duct walls. However, the slip diminishes with a further increase in Ha because of the formation of these boundary layers.

The graphs of Fig. 1 show that, the velocity magnitudes increase when α rises. This is a theoretically known behavior [9]. As the slipping parameter α increases, the slip on the walls increases and we see much more slip on the Hartmann walls than on the side walls. The increase in the slipping parameter α has not much effect on the profile of the induced magnetic field when the walls are insulated but it only causes a decrease in the induced magnetic field magnitude.

It is observed from the graphs of the Fig. 2 that as the conductivity parameter c increases, the velocity magnitudes decrease for no-slip velocity ($\alpha = 0$) whereas the induced magnetic field magnitudes increase. But, the increase in the induced magnetic field magnitude becomes weak when c increases further. The profiles of the induced magnetic field reveal that it tries to become perpendicular to the side walls as c increases but this orthogonality behavior is weakened for small values of c. That is, for $c \approx 10$ the side walls become almost electrically perfectly conducting.

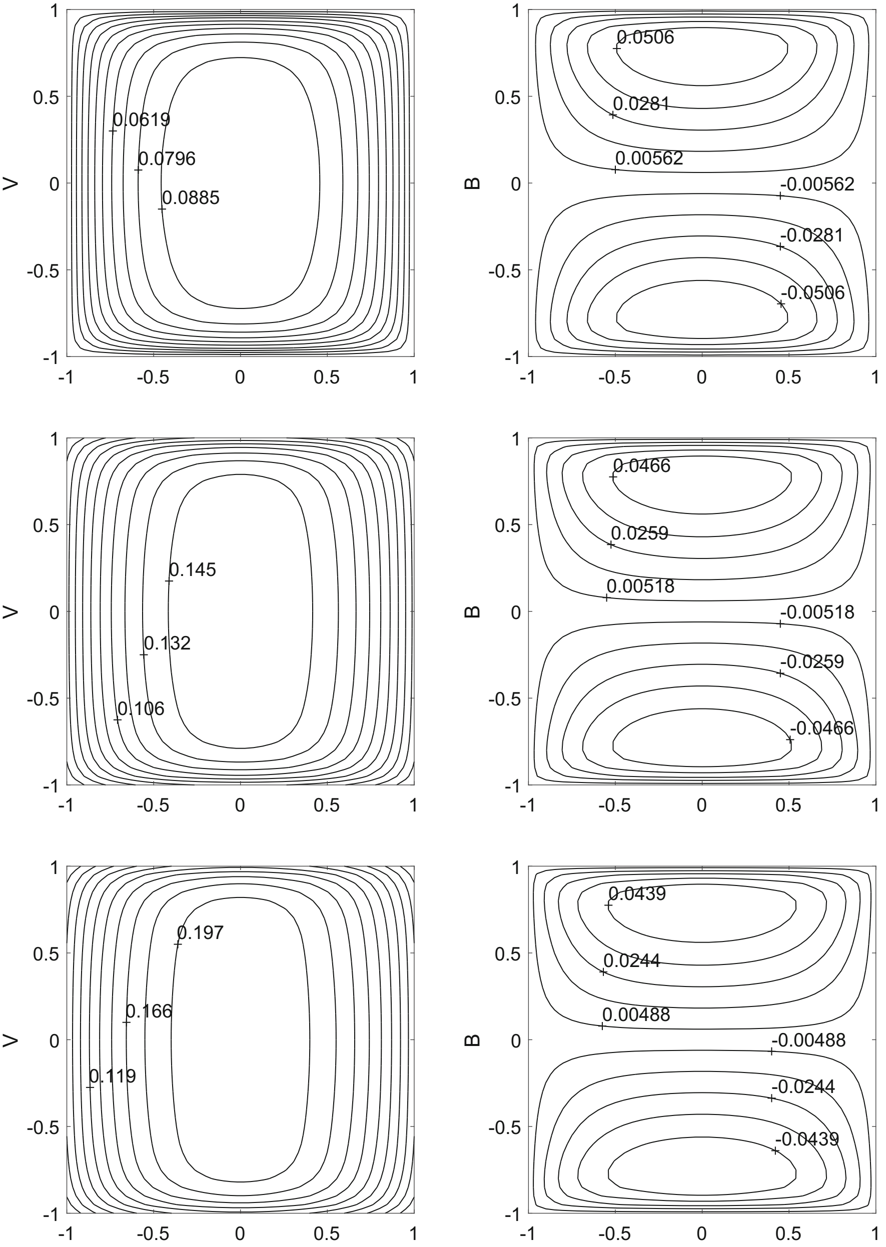


Fig. 1 Velocity and current lines for $Ha = 10$ and $c = 0$. Top $\alpha = 0$, middle $\alpha = 0.1$, and bottom $\alpha = 0.2$

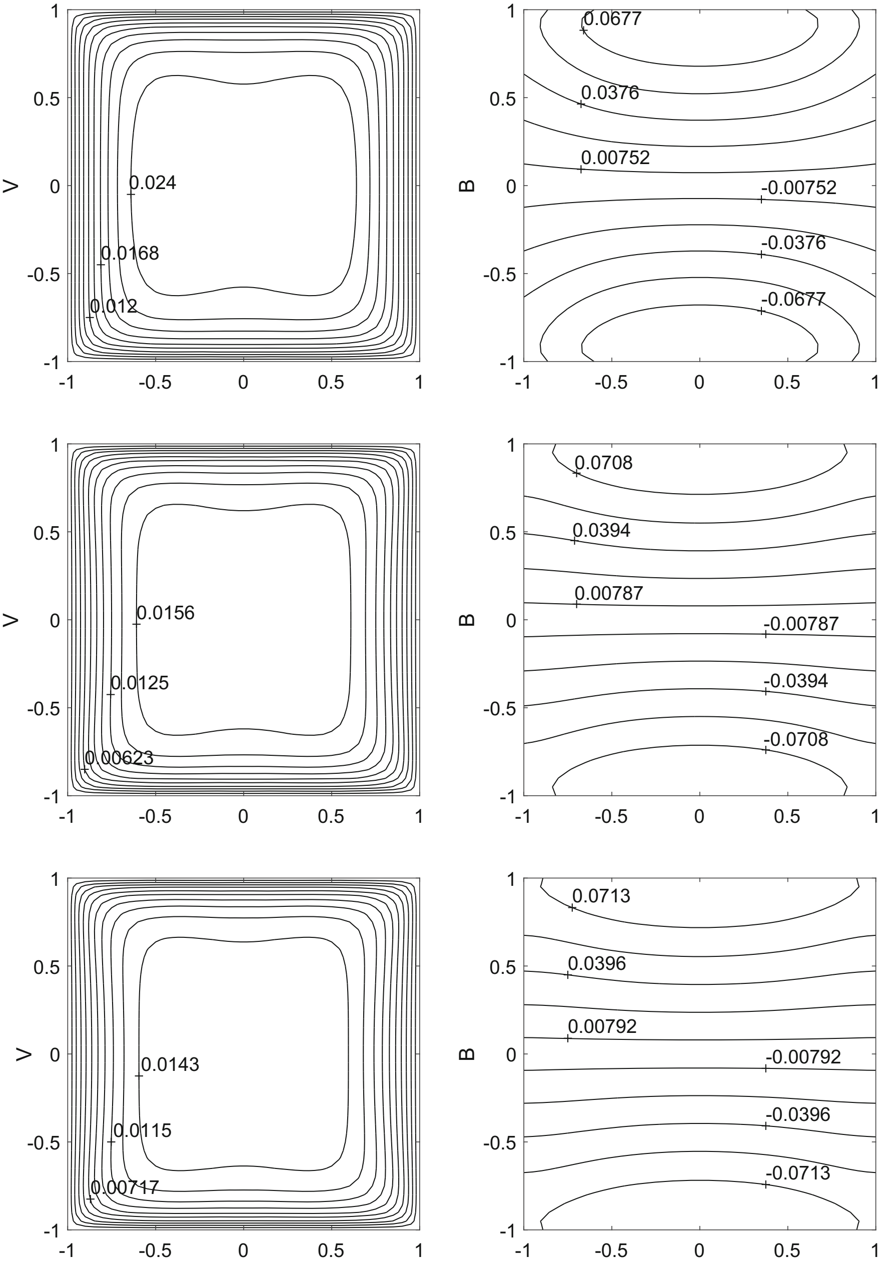


Fig. 2 Velocity and current lines for $Ha = 10$ and $\alpha = 0$. Top $c = 1$, middle $c = 5$, and bottom $c = 10$

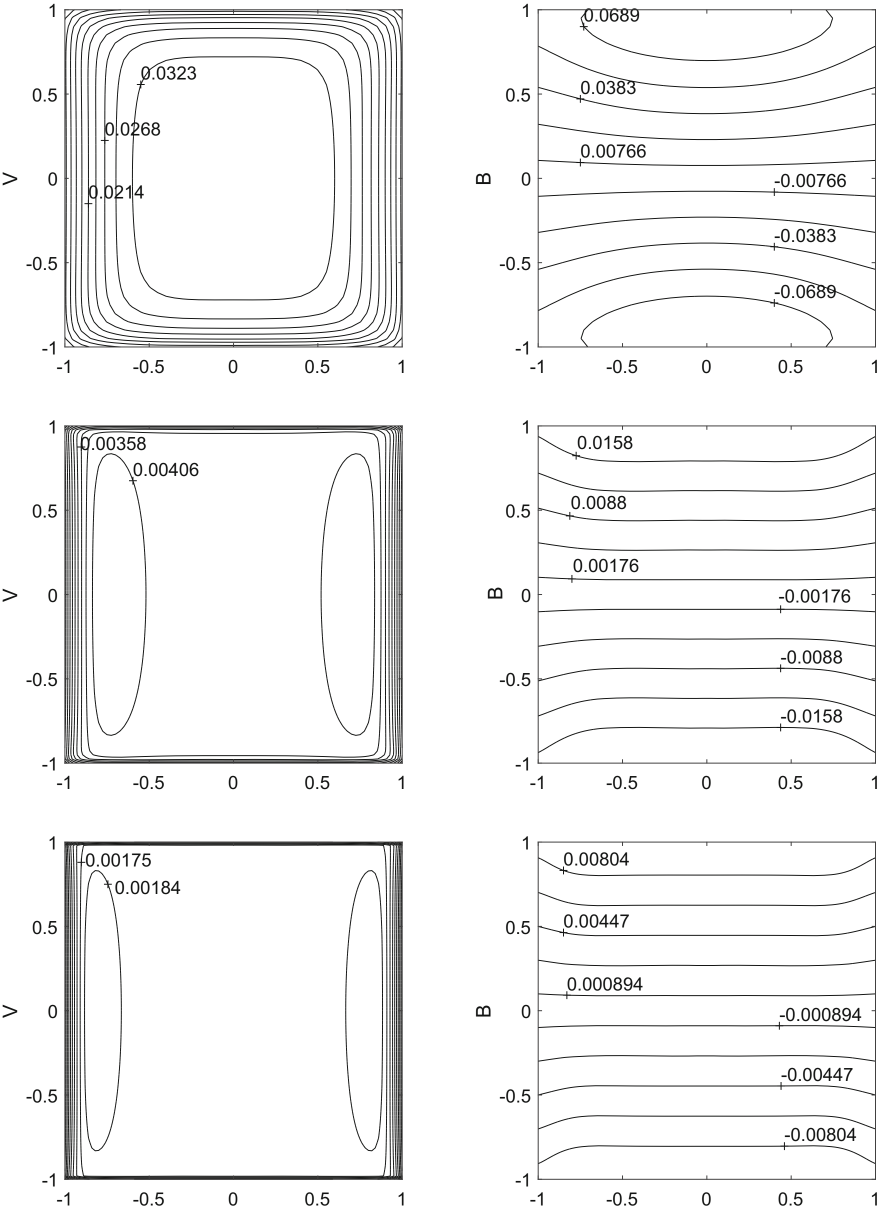


Fig. 3 Velocity and current lines for $\alpha = 0.1$ and $c = 2$. Top $Ha = 10$, middle $Ha = 50$, and bottom $Ha = 100$

Lastly, considering the effects of both α and c with Ha increase in Fig. 3, we see that for a small value of Hartmann number ($Ha = 10$), the slip is seen on the Hartmann walls but it disappears for large values of Hartmann number

($Ha = 50, 100$) for variably conducting wall case. It is observed from the velocity profiles that as Ha increases, the Hartmann layers become very thin obeying the order $1/Ha$, the core region increases, and the fluid flows near the side walls. Also, the induced magnetic field becomes perpendicular to the side walls with an increase in both wall conductivity c and Ha.

5 Conclusion

In this study, the 2D MHD flow in a rectangular duct is investigated. Mixed type boundary conditions are considered for both the velocity and the induced magnetic field which contain no-slip to slipping velocity and insulated to perfectly conducting induced current wall conditions. The effects of the Hartmann number, slip length and boundary conductivity on the flow and induced current are shown in terms of equivelocity and equal induced magnetic field lines. An increase in Ha causes to flatten the flow and the induced current. As Ha increases, one needs to make the mesh finer due to the convection dominance of the MHD equations.

It has been also shown that as Ha increases boundary layers are formed near the Hartmann walls and the side walls. The increase in the slip length causes an increase in the velocity magnitude, which is weakened for large values of Hartmann number, that is, the slip of the velocity on the walls tends to diminish when Ha rises. When the slipping length is kept fixed, the induced magnetic field magnitude increases with an increase in the conductivity parameter whereas the velocity magnitude drops. Consequently, we see that the well-known characteristics of the MHD flow are caught and the effects of slip and varying conductivity on the walls are very well depicted with the numerical results obtained using the FDM arranged especially to handle mixed boundary conditions. The FDM is easy to implement and gives accurate results at a cheap expense.

Acknowledgement The support of The Scientific and Technological Research Council of Turkey (TUBITAK) is acknowledged.

References

1. Arslan, S., Tezer, M.: Finite difference method solution of magnetohydrodynamic flow in channels with electrically conducting and slipping walls. [Electronic resource]. METU (2018)
2. Arslan, S., Tezer-Sezgin, M.: Exact and FDM solutions of 1D MHD flow between parallel electrically conducting and slipping plates. Advances in Computational Mathematics **45**, 1923–1938 (2019)
3. Carabineanu, A., Dinu, A., Oprea, I.: The application of the boundary element method to the magnetohydrodynamic duct flow. Zeitschrift für angewandte Mathematik und Physik ZAMP **46**(6), 971–981 (1995)
4. Dragos, L.: Magnetofluid dynamics. Abacus Press (1975)

5. Hartmann, J., Lazarus, F.: Hg-dynamics. Levin & Munksgaard Copenhagen (1937)
6. Müller, U., Bühler, L.: Magnetofluiddynamics in Channels and Containers. Springer, New York (2001)
7. Singh, B., Lal, J.: MHD axial-flow in a triangular pipe under transverse magnetic-field parallel to a side of the triangle. Indian Journal of Technology **17**(5), 184–189 (1979)
8. Singh, B., Lal, J.: Finite element method in magnetohydrodynamic channel flow problems. International Journal for Numerical Methods in Engineering **18**(7), 1104–1111 (1982)
9. Smolentsev, S.: MHD duct flows under hydrodynamic "slip" condition. Theoretical and Computational Fluid Dynamics **23**(6), 557 (2009)
10. Tezer-Sezgin, M.: Boundary element method solution of MHD flow in a rectangular duct. International journal for numerical methods in fluids **18**(10), 937–952 (1994)
11. Tezer-Sezgin, M., Köksal, S.: Finite element method for solving MHD flow in a rectangular duct. International journal for numerical methods in engineering **28**(2), 445–45 (1989)

Applications of the PRESB Preconditioning Method for OPT-PDE Problems

Owe Axelsson

Abstract Optimal control problems constrained by partial differential equations arise in a multitude of important applications. They lead mostly to the solution of very large scale algebraic systems to be solved, which must be done by iterative methods. The problems should then be formulated so that they can be solved fast and robust, which requires the construction of an efficient preconditioner. After reduction of a variable, a two-by-two block matrix system with square blocks arises for which such a preconditioner, PRESB is presented, involving the solution of two algebraic systems which are a linear combination of the matrix blocks. These systems can be solved by inner iterations, involving some available classical solvers to some relative, not very demanding tolerance.

1 Introduction

As is widely accepted, analyses and solutions of partial differential equations are mostly merely just part of a general solution process that includes some kind of optimization and sensitivity analyses where the PDE equation acts as a constraint. For example, one may want to control an equipment to have a desired behaviour as close as possible to some target function. In other applications one must identify some coefficient, see i.e. [10], such as describing the unknown material properties or boundary values at an inaccessible part of the boundary of the domain, which is important to enable to control that various safety requirements are satisfied, see e.g. [12].

The control and observation domains can be identical, possibly equal to the whole domain of definition or can be separate subdomains.

After a presentation of the basic properties of the PRESB, i.e. preconditioned square block matrix and its application for the common subdomain case, a boundary

O. Axelsson (✉)
The Czech Academy of Sciences, Institute of Geonics, Ostrava, Czech Republic
e-mail: owe.axelsson@it.uu.se

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_6

optimal control problem is presented which leads to separate subdomains and for which the standard PRESB method must be modified. Theoretical backgrounds are included in the paper. For numerical results, see [3].

2 The PRESB Preconditioning Method

For basic optimal control problems one can use a very efficient preconditioner, named preconditioned square block (PRESB) method, which arose as a simple method to avoid complex arithmetics when solving symmetric complex valued systems. Consider, see e.g. [7],

$$(A + iB)(x + iy) = f + ig,$$

where A, B etc. are real valued and we assume that $A + B$ is nonsingular. It can be rewritten in real valued form

$$\mathcal{A}\begin{bmatrix} x \\ y \end{bmatrix} := \begin{bmatrix} A & -B \\ B & A \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix},$$

which as has been shown in [7] and elsewhere, can be solved easily and efficiently by the use of a PRESB preconditioned iteration method.

Consider a more general problem, such as arises in Maxwell's equation for eddy current electromagnetic problems, (see e.g. [6])

$$\mathcal{A}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} A & -B^* \\ B & A \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{1}$$

where B^* denotes the complex conjugate of B. We assume that A is symmetric and positive semi-definite, that $B + B^*$ is positive semi-definite and that $\mathcal{N}(A) \cap \mathcal{N}(B + B^*) = \{0\}$.

Proposition 1 *Under the above conditions, $\mathcal{A}$ is nonsingular.*

Proof For a singular equation (1) it holds $Ax = B^*y$ and $Bx + Ay = 0$. Hence $x^*Ax + y^*Ay = 0$, that is, x, y belongs to $\mathcal{N}(A)$, because A is semidefinite. This implies that $B^*y = 0$, $Bx = 0$ that is, x, y belongs also to $\mathcal{N}(B)$, therefore, $x = y = 0$. □

As preconditioner to $\mathcal{A}$ we take the PRESB matrix

$$\mathcal{B} = \begin{bmatrix} A & -B^* \\ B & A + B + B^* \end{bmatrix}.$$

As is readily seen, $\mathcal{B}$ can be factorized as

$$\begin{bmatrix} I & -I \\ 0 & I \end{bmatrix} \begin{bmatrix} A+B & 0 \\ B & A+B^* \end{bmatrix} \begin{bmatrix} I & I \\ 0 & I \end{bmatrix}, \tag{2}$$

which shows that, besides a matrix vector multiplication with B and some vector additions, an action of $\mathcal{B}^{-1}$ involves just solving a linear system with matrix $A + B^*$ and one with $A + B$.

In many problems there exist efficient solution methods for such systems, such as based on algebraic multigrid, modified incomplete factorization, or for very large problems, use of a domain decomposition method, see e.g. [13, 15] for efficient implementations of AGMG methods.

Let $A^\dagger$ denote a generalized inverse of A.

Proposition 2 *Under the stated conditions, the eigenvalues* λ *of* $\mathcal{B}^{-1}\mathcal{A}$ *are contained in the interval* $\frac{1}{2} \leq 1 - \varrho(D_0) \leq \lambda \leq 1$, *where*

$$D_0 = ((A + B)A^\dagger(A + B^*))^{-1}(B + B^*).$$

If $B^* = B$ *and* B *is spsd, then*

$$\frac{1}{2} \leq \lambda(\mathcal{B}^{-1}\mathcal{A}) \leq \frac{1}{2}\left(1 + \max_{\mu((A+B)^{-1}B)} (1 - 2\mu)^2\right) \leq 1\,,$$

where $\mu(\,)$ *denotes eigenvalues.*

Proof For a proof, see [8]. For a proof of the eigenvalue interval $[\frac{1}{2}, 1]$, see the Remark 1 in Sect. 4. □

Corollary 1 *Let* A *be spd,* $A + B$ *nonsingular and assume that* $Re(\mu) \geq 0$ *where* $\mu Ax = Bx$, $\|x\| \neq 0$. *Then the eigenvalues of* $\mathcal{B}^{-1}\mathcal{A}$ *satisfy* $1 \geq \lambda \geq \frac{1}{1+\alpha}$, *where* $\alpha = \max\limits_{\mu} \frac{2\,Re(\mu)}{1+|\mu|^2}$.

Proof It follows $1 - \lambda = \frac{2\mathrm{Re}(\mu)}{1+|\mu|^2+2\mathrm{Re}(\mu)}$. Hence $\lambda = \frac{1+|\mu|^2}{1+|\mu|^2+2\mathrm{Re}(\mu)} \geq \frac{1}{1+\alpha}$. Note that $\alpha \leq 1$. □

It follows that the preconditioned iteration method converges fast and, since the eigenvalue bounds are known, it can even be efficient to apply the Chebyshev iteration method. The rate of convergence factor is then bounded above by $\frac{\sqrt{2}-1}{\sqrt{2}+1} = \frac{1}{3+2\sqrt{2}} \approx \frac{1}{6}$.

As has been shown in [6], see also [5, 11], for time-harmonic eddy-current problems, the ratio $2\mathrm{Re}(\mu)/(1 + |\mu|^2)$ becomes very small for large values of the frequency ω, i.e. where $|\mu|$ is large, which implies that the eigenvalues cluster at unity, and implies a superlinear rate of convergence.

Since we solve inner systems by iteration to some flexible accuracy, it can be efficient to use a variable preconditioned Krylov subspace method, see [9, 14]. As has been shown in [1, 2, 6, 8] when the eigenvalue bounds are known one can also use a Chebyshev acceleration method.

3 A Basic Optimal Control Problem

We consider first a constrained optimal control PDE problem with identical observation and control subdomains $\Omega_0 \subset \Omega$, where Ω is the whole domain of definition for the given partial differential equations, see e.g. [4]. Hence we want to compute

$$\inf_{u,v} J(u,v), \qquad J(u,v) = \frac{1}{2}\|u - u_0\|^2_{\Omega_0} + \frac{1}{2}\beta\|v\|^2_{\Omega_0} \qquad \text{s.t. } \mathcal{L}u = f + v.$$

Here u is the state solution defined in Ω, v is the control defined in Ω_0, u_d is the target solution, f is a given source function, $\mathcal{L}$ is a 2'nd order coercive elliptic operator, i.e. spd and $\beta > 0$ is a regularization parameter. We assume that proper boundary conditions hold.

The corresponding Lagrange functional with multiplier w, that is the adjoint variable to u, takes the form:
Seek the $\inf_{u,v}$ / $\sup_w$, i.e. the saddle point solution of

$$J(u,v) + \int_\Omega w(\mathcal{L}u - f - v).$$

Note that the control acts to modify the source function f. Discretizing the problem in a finite element subspace and applying the first order necessary optimality conditions, lead to the coupled algebraic system,

$$\begin{bmatrix} \widetilde{M}_0 & 0 & K^T \\ 0 & \beta M_0 & -N^T \\ K & -N & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \widetilde{M}_0 u_d \\ 0 \\ f \end{bmatrix},$$

where M_0 is the mass matrix corresponding to the discrete subdomain Ω_0 and $\widetilde{M}_0 = \begin{bmatrix} M_0 & 0 \\ 0 & 0 \end{bmatrix}$ is the matrix extended to Ω, $N = \begin{bmatrix} M_0 \\ 0 \end{bmatrix}$, K is the finite element discretization of $\mathcal{L}$ and the vectors correspond to the discrete versions of the corresponding continuous functions.

After elimination of the control vector v we get

$$\begin{bmatrix} \widetilde{M}_0 & K^T \\ K & -\beta^{-1}\widetilde{M}_0 \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} \widetilde{M}_0 u_d \\ f \end{bmatrix},$$

which we scale and reorder,

$$\widetilde{A}\begin{bmatrix}\widetilde{w}\\ u\end{bmatrix} := \begin{bmatrix}\widetilde{K} & \widetilde{M}_0\\ -\widetilde{M}_0 & \widetilde{K}\end{bmatrix}\begin{bmatrix}\widetilde{w}\\ u\end{bmatrix} = \begin{bmatrix}\widetilde{f}\\ \widetilde{M}u_d\end{bmatrix},$$

where $\widetilde{K} = \sqrt{\beta}K$, $\widetilde{w} = \frac{1}{\sqrt{\beta}}w$ and $\widetilde{f} = \sqrt{\beta}f$. For this square block matrix we can apply the PRESB preconditioner,

$$\widetilde{\mathcal{B}} = \begin{bmatrix}\widetilde{K} & \widetilde{M}_0\\ -\widetilde{M}_0 & \widetilde{K} + 2\widetilde{M}_0\end{bmatrix},$$

which can be factorized as in (2).

Proposition 3 *The eigenvalues of* $\widetilde{\mathcal{B}}^{-1}\widetilde{\mathcal{A}}$ *are contained in the interval* $[\frac{1}{2}, 1]$.

Proof Let λ be an eigenvalue, then

$$(1-\lambda)\widetilde{\mathcal{B}}\begin{bmatrix}\xi\\ \eta\end{bmatrix} = (\widetilde{\mathcal{B}} - \widetilde{\mathcal{A}})\begin{bmatrix}\xi\\ \eta\end{bmatrix} = \begin{bmatrix}0\\ 2\widetilde{M}_0\eta\end{bmatrix},$$

so $\lambda = 1$ if and only if $\eta \in \mathcal{N}(\widetilde{M}_0)$, any ξ. (Note the large dimension of the unit eigenvalue!) For $\lambda \neq 1$ it follows that $\widetilde{K}\xi + \widetilde{M}_0\eta = 0$ and

$$(1-\lambda)(\widetilde{M}_0\widetilde{K}^{-1}\widetilde{M}_0 + \widetilde{K})\eta = \lambda 2\widetilde{M}_0\eta.$$

Hence $\lambda < 1$ and

$$(1-\lambda)\widetilde{\eta}^T(\widehat{M}_0^2 + I)\widehat{\eta} = 2\lambda\widehat{\eta}^T\widehat{M}_0\widehat{\eta},$$

where $\widehat{M}_0 = \widetilde{K}^{-1/2}\widetilde{M}_0\widetilde{K}^{-1/2}$, $\widehat{\eta} = \widetilde{K}^{1/2}\eta$.

It follows that $1-\lambda \leq \lambda$, that is, $\lambda \geq \frac{1}{2}$. □

Previously, for a fully distributed control function v in Ω, the following PDE problems have been analysed and illustrated numerically:

(i) convection diffusion: $Ku = -\Delta u + \mathbf{c}\cdot\nabla u = f$, where $\nabla \cdot c \leq 0$, see [4].

(ii) time-harmonic problems,

$$\frac{\partial u}{\partial t} - \Delta u + \sigma u = f, 0 < t < T, \text{ where } f = f_0 e^{i\omega t}, \omega = k\frac{2\pi}{T}, k = 1, 2, \cdots$$

which leads to $Ku = -\Delta u + \sigma u + i\omega u = f_0 + v$, see [5, 11].
Here one can solve for each frequency ω in parallel.

(iii) The similar, Maxwell's eddy current electromagnetic equation, see [6].

4 An Inverse Identification Problem for a Non-selfadjoint Problem

4.1 Problem Formulation

There exist several types of inverse problems in the form of identification problems, such as identification of a material coefficient in a PDE problem or identification of some inaccessible boundary part of the unknown solution. Here we consider identification of part of the boundary conditions, namely at a for measurement inaccessible part of a physical boundary, such as hidden by other structures, for practical examples, see e.g. [12].

Let Ω be a given domain where the boundary part $\partial\Omega_1$ is assumed to be inaccessible and let $\partial\Omega_2$ be the other part, $\partial\Omega_2 = \partial\Omega/\partial\Omega_1$, of the boundary, see Fig. 1 for an illustration.

In order to find the missing boundary condition we overimpose, that is, we assume that both the Dirichlet values u_d and the Neuman conditions $\frac{\partial u}{\partial n}$ are given, e.g. have been measured on $\partial\Omega_2$. Let the differential operator problem be

$$Ku = -\Delta u + \mathbf{c}\nabla u + \sigma u = f, \quad \sigma - \frac{1}{2}\nabla \cdot c > 0, \quad \text{with } g = \frac{\partial u}{\partial n} \text{ given on } \partial\Omega_2.$$

To find an approximation of $v = \frac{\partial u}{\partial n}$ on $\partial\Omega_1$ we imbed the problem in an optimal control framework, that is, the aim is to solve

$$\min_{u,v} J(u,v), \quad J(u,v) = \frac{1}{2}\|u - u_d\|^2_{\partial\Omega_2} + \frac{1}{2}\beta\|v\|^2_{\partial\Omega_1},$$

which is subject to $Ku = f$ in Ω, $\frac{\partial u}{\partial n} = g$ on $\partial\Omega_2$.

Here the Dirichlet values u_d on $\partial\Omega_2$ are used as target solution, v acts as a control function and $\beta > 0$ is a standard regularization parameter.

Letting w be the Lagrange multiplier to impose the differential equation constraint, the variational formulation becomes:
Find

$$\inf_{u,v} \sup_{w} \left\{ J(u,v) + \int_\Omega (\nabla u \cdot \nabla w + \mathbf{c} \cdot \nabla u\, w + \sigma u\, w - f w) - \oint_{\partial\Omega_1} v w - \oint_{\partial\Omega_2} g w \right\}$$

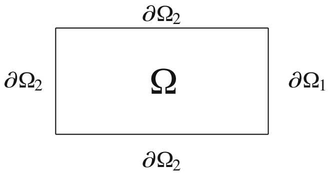


Fig. 1 A domain Ω with an inaccessible part $\partial\Omega_1$ of its boundary, with overimposed boundary conditions on $\partial\Omega_2 = \partial\Omega/\partial\Omega_1$. The aim is to find $v = \frac{\partial u}{\partial n}$ on $\partial\Omega_1$

After discretization, the Karush–Kuhn–Tucker first order optimality conditions (for notational simplicity we keep the notation K, etc for the discrete operator and vectors) give,

$$\begin{bmatrix} \widetilde{M}_2 & 0 & K^T \\ 0 & \beta \widetilde{M}_1 & -N^T \\ K & -N & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} \widetilde{M}_2 u_d \\ 0 \\ \widehat{f} \end{bmatrix} \tag{3}$$

where $\widetilde{M}_1 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & M_1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $\widetilde{M}_2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & M_2 \end{bmatrix}$, $N = \begin{bmatrix} 0 \\ M_1 \\ 0 \end{bmatrix}$ and M_i are the mass matrices for $\partial\Omega_i$, $i = 1, 2$ and $\widehat{f} = f + [\oint_{\partial\Omega_2} g\varphi_i]$, where $\{\varphi_i\}$ are the set of basis functions on $\partial\Omega_2$.

Note that all vectors have the same dimension. Here we have used the ordering, interior nodepoints followed by nodepoints on $\partial\Omega_i$, $i = 1, 2$.

There are two major issues associated with the solution of optimal control problems:

(i) Construction of an efficient iterative solution method to solve (3), that is, in particular the construction of a preconditioner.

(ii) The estimate of errors in the solution and control function, which depends on both the discrete mesh parameter h and the parameter β.

Due to limited space of this paper we shall consider only topic (i). For a more complete presentation, see [3].

4.2 The Reduced Matrix System and Its Nonsingularity

After elimination of the control variable $v = \frac{1}{\beta} w$, we get the reduced system,

$$\begin{bmatrix} \widetilde{M}_2 & K^T \\ K & -\frac{1}{\beta}\widetilde{M}_1 \end{bmatrix} \begin{bmatrix} u \\ w \end{bmatrix} = \begin{bmatrix} \widetilde{M}_2 u_d \\ \widetilde{f} \end{bmatrix}$$

which we reorder and scale to obtain

$$\mathcal{A} \begin{bmatrix} u \\ \widetilde{w} \end{bmatrix} := \begin{bmatrix} \widetilde{M}_2 & -\widetilde{K}^T \\ \widetilde{K} & \widetilde{M}_1 \end{bmatrix} \begin{bmatrix} u \\ \widetilde{w} \end{bmatrix} = \begin{bmatrix} \widetilde{M}_2 u_d \\ \widetilde{f} \end{bmatrix} \tag{4}$$

where $\widetilde{K} = \sqrt{\beta}K$, $\widetilde{f} = \sqrt{\beta}\widehat{f}$ and $\widetilde{w} = -\frac{1}{\sqrt{\beta}}w$, which is the equation to be solved. Guided by the PRESB method in Sect. 2, (4) will be solved by iteration with the preconditioner,

$$\mathcal{B} = \begin{bmatrix} \widetilde{M}_2 & -\widetilde{K}^T \\ \widetilde{K} & \widetilde{M}_1 + \widetilde{K} + \widetilde{K}^T \end{bmatrix}.$$

Systems with this matrix can be solved by inner iterations using the PRESB preconditioner, i.e. where $\widetilde{M}_1$ above has been replaced with $\widetilde{M}_2$, which will converge rapidly.

The eigenvalues of $\mathcal{B}^{-1}\mathcal{A}$ satisfy

$$(1-\lambda)\mathcal{B}\begin{bmatrix} \xi \\ \eta \end{bmatrix} = (\mathcal{B}-\mathcal{A})\begin{bmatrix} \xi \\ \eta \end{bmatrix} = \begin{bmatrix} 0 \\ (\widetilde{K} + \widetilde{K}^T)\eta \end{bmatrix}, \qquad \text{where } \|\xi\|+\|\eta\| \neq 0. \tag{5}$$

Proposition 4 *The eigenvalues of $\mathcal{B}^{-1}\mathcal{A}$ are located in the interval* $(0, 1]$. *In particular, $\mathcal{A}$ is nonsingular.*

Proof Since by assumption made, $\widetilde{K} + \widetilde{K}^T$ is nonsingular, it follows that $\lambda = 1$ if and only if $\eta = 0$, arbitrary ξ. For $\lambda \neq 1$, (5) shows that

$$\begin{cases} \widetilde{M}_2\xi = \widetilde{K}^T\eta \\ (1-\lambda)(\widetilde{K}\xi + \widetilde{M}_1\eta) = \lambda(\widetilde{K} + \widetilde{K}^T)\eta. \end{cases}$$

If $\widetilde{M}_2\xi = 0$, then $\eta = 0$, i.e. $\widetilde{K}\xi = 0$ so also $\xi = 0$. Hence $\xi \in \mathcal{N}(\widetilde{M}_2)^{\perp}$. Let $\widetilde{M}_2^{\dagger}$ be a generalized inverse of $\widetilde{M}_2$. Then $\xi = \widetilde{M}_2^{\dagger}\widetilde{K}^T\eta$ and

$$(\widetilde{K}\widetilde{M}_2^{\dagger}\widetilde{K}^T + \widetilde{M}_1)\eta = \mu(\widetilde{K} + \widetilde{K}^T)\eta, \tag{6}$$

where $\mu = \lambda/(1-\lambda)$. It follows that μ is positive so $0 < \lambda \leq 1$ and $\mathcal{A}$ is nonsingular. □

In order to find how the eigenvalues depend on β as $\beta \to 0$, we rewrite (6) as,

$$2\mu\widehat{\eta} = (\widehat{M}_2^{\dagger} + \widehat{M}_1)\widehat{\eta},$$

where $\widehat{M}_2^{\dagger} = S^{-1/2}\widetilde{K}\widetilde{M}_2^{\dagger}\widetilde{K}^T S^{-1/2}$, $\widehat{M}_1 = S^{-1/2}\widetilde{M}_1 S^{-1/2}$ and $S = \frac{1}{2}(\widetilde{K} + \widetilde{K}^T)$. We note that $\widehat{\eta}^T\widehat{M}_2^{\dagger}\widehat{\eta}/\widehat{\eta}^T\widehat{\eta}$ is contained in the interval $\big(|O(\beta^{1/2})|, O(1)\big)$ and $\widehat{\eta}^T\widehat{M}_1\widehat{\eta}/\widehat{\eta}^T\widehat{\eta}$ in $[0, O(\beta^{-1/2})]$. It follows that μ is contained in the interval $(|O(\beta^{1/2})|, |O(\beta^{-1/2})|)$.

From the lower bound values it is seen that the corresponding eigenvalues $\lambda = |O(\sqrt{\beta})|$ and for the upper bound values that $\lambda = 1/(1+|O(\sqrt{\beta})|) = 1-|O(\sqrt{\beta})|$.

Hence the eigenvalues cluster at unity for eigenvectors $\widehat{\eta} \in \mathcal{N}(\widehat{M}_1)^{\perp}$, which subspace has a large dimension. The small eigenvalues are taken for $\eta \in \mathcal{N}(\widehat{M}_1)$.

Numerical tests in [3] show that the iteration method converges rapidly. The approximation errors decrease as $O(\beta)$ when $\beta \to 0$ and as $O(h^2)$ as the mesh parameter $h \to 0$.

Remark 1 If the control and observation subdomains are identical, then it follows readily from (6) that the eigenvalues λ are located in the interval $[\frac{1}{2}, 1]$, which gives a proof of the related Proposition in Sect. 2.

Acknowledgments The research of O. Axelsson was supported by the Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project "IT4 Innovations excellence in science LQ1602"

References

1. Axelsson, O.: Iterative Solution Methods. Cambridge University Press, Cambridge (1994)
2. Axelsson, O., Liang, Z.-Z., Kruzik, J., Horak, D.: Inner product free iterative solution and elimination methods for linear systems of a three-by-three block matrix form. J. Comput. Appl. Math. **383**, 113–117 (2021)
3. Axelsson, O., Blaheta R., Béreš, M.: A boundary optimal control identification problem. Uppsala University, Department of Technology, TR 2020-002, May 2020.
4. Axelsson, O., Farouq, S., Neytcheva, M.: Comparison of preconditioned Krylov subspace iteration methods for PDE-constrained optimization problems. Poisson and convection-diffusion control. Numer. Alg. **73**, 631–663 (2016)
5. Axelsson, O., Liang, Z.-Z.: A note on preconditioning methods for time-periodic eddy current optimal control problems. J. Comput. Appl. Math. **352**, 262–277 (2019)
6. Axelsson, O., Lukáš, D.: Preconditioning methods for eddy-current optimally controlled time-harmonic electromagnetic problems. J. Numer. Math. **27**, 1–21 (2019)
7. Axelsson, O., Neytcheva, M., Ahmad, B.: A comparison of iterative methods to solve complex valued linear algebraic systems. Numerical Algorithms **66**, 811–841 (2014)
8. Axelsson, O., Salkuyeh, D.K.: A new version of a preconditioning method for certain two-by-two block matrices with square blocks. BIT Numer. Math. **59**, 321–342 (2019)
9. Axelsson, O., Vassilevski, P.S.: Algebraic multilevel preconditioning methods. I. Numerische Mathematik **56**, 157–177 (1989)
10. Isakov, V.: Inverse Problems for Partial Differential Equations. Springer, New York (2006)
11. Liang, Z.-Z., Axelsson, O., Neytcheva, M.: A robust structured preconditioner for time-harmonic parabolic optimal control problems. Numer. Algor. **79**, 575–596 (2018)
12. Martin, T.J., Dulikravich, G.S.: Inverse determination of temperatures and heat fluxes on inaccessible surfaces. WIT Transactions on Modelling and Simulation **8**, (1994)
13. Notay, Y.: AGMG software and documentation. See http://homepages.ulb.ac.be/~ynotay/ (2015)
14. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM J. Sci. Comput. **14**, 461–469 (1993)
15. Vassilevski, P.S.: Multilevel Block Factorization Preconditioners. Springer-Verlag, New York (2008)

Model Order Reduction Framework for Problems with Moving Discontinuities

H. Bansal, S. Rave, L. Iapichino, W. Schilders, and N. van de Wouw

Abstract We propose a new model order reduction (MOR) approach to obtain effective reduction for transport-dominated problems or hyperbolic partial differential equations. The main ingredient is a novel decomposition of the solution into a function that tracks the evolving discontinuity and a residual part that is devoid of shock features. This decomposition ansatz is then combined with Proper Orthogonal Decomposition applied to the residual part only to develop an efficient reduced-order model representation for problems with multiple moving and possibly merging discontinuous features. Numerical case-studies show the potential of the approach in terms of computational accuracy compared with standard MOR techniques.

1 Introduction

Hyperbolic partial differential equations (PDEs) are ubiquitous in science and engineering. Applications encompassing the fields of chemical industry, nuclear industry, drilling industry, etc., fall within this class. Model Order Reduction of systems of non-linear hyperbolic PDEs is a challenging research topic and is an active area of research in the scientific community. Moving discontinuities (such as shock-fronts) are representative features of this class of models and pose a major hindrance to obtain effective reduced-order model representations [1]. As a result, standard MOR techniques [2] do not fit the requirements for real-time estimation and control or multi-query simulations of such problems. This motivates us to investigate and propose efficient, advanced and automated approaches to

H. Bansal (✉) · L. Iapichino · W. Schilders · N. van de Wouw
Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: h.bansal@tue.nl; l.iapichino@tue.nl; w.h.a.schilders@tue.nl; n.v.d.wouw@tue.nl

S. Rave
University of Muenster, Münster, Germany
e-mail: stephanrave@uni-muenster.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_7

obtain reduced models, while still guaranteeing the accurate approximation of wave propagation phenomena.

A lot of research is in progress to improve the state of the art of MOR for transport-dominated problems: *(i)* (data-based and model-based) time and space-dependent coordinate transformation/symmetry reduction framework [3–8], *(ii)* optimal transport [9–11], *(iii)* interpolation/dictionary/tracking framework [12–14], *(iv)* adaptive and stabilization strategies [15, 16], and, *(v)* deep learning/neural network concepts [17, 18]. These works have mainly focused on resolving transport along a single direction [3] and multiple directions [4] for linear and non-linear classes of (parameterized) problems.

Effective reduction of non-linear transport-dominated problems in the context of multiple moving (and merging) discontinuous features is still challenging. Few notable works that aim at mitigating this problem are [4, 12, 13]. The works [12, 13] are based on the concept of (low and high resolution) transformed snapshot interpolation. Such an approach has been particularly tested in the regions near (and at) the singularity, induced upon merging of the wavefronts. Another work in this direction is the concept of freezing multiple frames [19]. However, their performance, demonstrated for parabolic problems, does not carry over to less regular hyperbolic problems and suffers from additional travelling structures or numerical instabilities in the decomposed components. Moreover, the existing methods [4, 19] lack the (online-efficient) automated identification of switching point from multiple wavefront setting to single wavefront setting upon merging of wavefronts.

We propose an approach that is a stepping stone towards resolving the aforementioned issues. The main contribution of the work is to propose a new decomposition ansatz that decomposes the solution into a basis function that tracks the evolving discontinuity and a residual part that is expected to be devoid of shock features. This decomposition renders the residual part to be amenable for reduced-order approximation. We, then, use these generated bases to apply Proper Orthogonal Decomposition (POD) on the residual part and later reconstruct the solution by lifting it to the high-dimensional problem space. We finally assess the combined performance of decomposition, reduction and reconstruction approach (as opposed to conventional reduction and reconstruction approach) in the scope of transport-dominated problems with moving and interacting discontinuities.

2 Mathematical Formulation

We consider a scalar 1D conservation equation of the form:

$$\partial_t u(x,t) + \partial_x f(u(x,t)) = 0, \quad u(x,0) = u_0(x). \tag{1}$$

We assume that $u(x,0) = u_0(x)$ already has S number of discontinuities at locations $x_1(0), \ldots, x_S(0)$ with values $u^-(x_s(0),0), \quad s = 1, \ldots, S$ from the left and values $u^+(x_s(0),0), \quad s=1,\dots,S$ from the right. We associate a single basis function $\sigma_s(x-x_s(t))$ to each discontinuity at their respective locations. This basis function has a jump of height 1, i.e., $\sigma_s^+(0)-\sigma_s^-(0)=1$, at the location of the discontinuity and can have any (preferably continuous and smooth) shape away from the discontinuity.

We now decompose the solution of (1) in the following way:

$$u(x,t)=\sum_{s=1}^{S} j_s(t)\sigma_s(x-x_s(t))+u_r(x,t),$$

$$j_s(t)=u^-(x_s(t),t)-u^+(x_s(t),t). \tag{2}$$

If $x_s(t)$ exactly matches the shock locations and (2) is exactly fulfilled, then $u_r(x,t)$ does not contain any discontinuities and is amenable to a low-rank approximation.

The time-stepping scheme is defined in the following way. In each time step, we:

- Compute updated shock locations $x_s(t^{n+1})$ using the Rankine Hugoniot condition.
- Compute $u^{\pm}(x_s(t^{n+1}),t^{n+1})$ in a neighborhood of $x_s(t^{n+1})$ and define jumps, $j_s(t^{n+1})$, via (2).
- Compute the residual part $u_r(x,t^{n+1})$ from

$$\begin{aligned}&u_r(x,t^{n+1})-u_r(x,t^n)=\\&\sum_{s=1}^{S} j_s(t^n)\sigma_s(x-x_s(t^n))-\Delta t\partial_x f(u(x,t^n))-\sum_{s=1}^{S} j_s(t^{n+1})\sigma_s(x-x_s(t^{n+1})).\end{aligned} \tag{3}$$

The standard way to construct a reduced-order model (ROM) is to reduce (1) by applying Galerkin projection on u. Instead, we reduce (3) via Galerkin projection onto $V_N\subseteq V_h$, where V_N is a N-dimensional reduced space spanned by the functions obtained from a truncated singular value decomposition of the u_r snapshot matrix, and V_h is a h-dimensional high-fidelity space. Upon considering the projection operator $P_N:V_h\to V_N$, the reduced scheme takes the following form:

$$\begin{aligned}u_{r,N}^{k+1}=u_{r,N}^{k}+P_N\Big(&\sum_{s=1}^{S} j_{s,N}(t^k)\sigma_s(x-x_{s,N}(t^k))-\Delta t\partial_x f(P_N^{'}u_N^k)-\\&\sum_{s=1}^{S} j_{s,N}(t^{k+1})\sigma_s(x-x_{s,N}(t^{k+1}))\Big),\end{aligned} \tag{4}$$

where $u^k_{r,N} \in V_N$ and $u^0_{r,N} = P_N(u^0_r)$ with u^k_N defined in the following form:

$$P^{'}_N u^k_N = \sum_{s=1}^{S} j_{s,N}(t^k)\sigma_s(x - x_{s,N}(t^k)) + P'_N u^k_{r,N}, \tag{5}$$

and, $j_{s,N}$ and $x_{s,N}$ are, respectively, the jumps and shock locations computed during the ROM time-stepping. $j_{s,N}$ and $x_{s,N}$ can be obtained in a manner similar to the steps carried out during the full-order model (FOM) time-stepping.

It is well known that projection alone is not sufficient to reduce the costs of computing the solution of a reduced-order model if the Finite Volume operators are non-linear in nature. Empirical Operator Interpolation [20] can be used here as a recipe for hyper-reduction. We do not delve into the full and efficient offline and online decomposition as its discussion is not within the scope of this work. However, we mention that we need to know $j_{s,N}(t^k)$ and $u_{r,N}(x_{s,N}(t^k), t^k)$ for computing $x_{s,N}(t^{k+1})$. In a reduced scheme this means that we need to keep the entire reduced basis in memory. However, the basis vectors are only evaluated at the shock locations at each time step. The same consideration holds for the computation of the $j_{s,N}(t^{k+1})$.

3 Numerical Experiments

We numerically test the new approach and show its potential as a reduced-order modelling technique. We reduce Burgers equation, which is given by:

$$\partial_t u + \partial_x(\frac{u^2}{2}) = 0, x \in [0, L]. \tag{6}$$

The case studies consider that the shock is already present in the initial data, which for single and multiple wavefront scenarios, is respectively given by:

$$u(x,0) = u_0(x) = \begin{cases} x, & 0 \le x \le 1, \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad u(x,0) = u_0(x) = \begin{cases} x-2, & 2 \le x \le 4, \\ \frac{(x-5)}{2}, & 5 \le x \le 7, \\ 0, & \text{otherwise.} \end{cases}$$

We consider only periodic boundary conditions. Furthermore, we consider the spatial domain to be $L = 10$ and use an upwind finite volume (FV) scheme for the spatial discretization and first-order Forward Euler for the time-stepping. We take 8000 steps in time for the scenarios under consideration i.e., $t \in [0, 4]$ with a timestep of 0.0005. We consider three different spatial mesh resolutions (spatial step size of 0.005, 0.002 and 0.001) to assess the performance of the standard (POD without decomposition) and the proposed approach.

We quantify the performance of the standard and the proposed approach by computing the reduced-order modeling (ROM) error. We consider L^2 in space and L^2 in time (absolute) error and define it in the following manner (for a basis-size N):

$$e_{rom} = \sqrt{\Delta t \sum_{k=1}^{N_T+1} \Delta x \sum_{i=1}^{N_x} \mid u_{i,k} - (P_N' u_N^k)_{i,k} \mid^2}. \tag{7}$$

where Δt is the time-step, Δx is the spatial step, N_T is the number of time-steps and N_x is number of Finite Volume elements. $u_{i,k}$ means u at $x = x_i$ and $t = t_k$ (similarly for $(P_N' u_N^k)_{i,k}$). Herewith, (7) expresses the error between the full-order model (Finite Volume solution) governed by (1) and the reconstruction given by (5)).

3.1 Single Wavefront Scenario

We first consider the scenario where only a single discontinuous front evolves across the spatial domain. Here, we use the following shape function for $\sigma_s(x - x_s)$:

$$\sigma_s(x - x_s) = \begin{cases} 1 + x - x_s, & x_s - 1 \leq x \leq x_s, \\ 0, & \text{otherwise.} \end{cases} \quad, \quad s = 1, .., S, \tag{8}$$

with $S = 1$, $x_s(t = 0) = 1$.

3.2 Multiple Wavefront Scenario

Here, we consider the setting where multiple (discontinuous) wavefronts evolve across the spatial domain and also interact non-linearly with each other. We study the scenario where two wavefronts are present in the spatial domain and the left front propagates faster than the right one. We, however, restrict the study to only assess the performance of the proposed approach in dealing with the interaction of the head of one wavefront with the tail of the other one. We postpone the discussion of automatically dealing with the merging of wavefronts for future work. We use the following shape function for $\sigma_s(x - x_s)$ to study this scenario.

$$\sigma_s(x - x_s) = \begin{cases} 1 + \frac{1}{2}(x - x_s), & x_s - 2 \leq x \leq x_s, \\ 0, & \text{otherwise} \end{cases} \quad, \quad s = 1, .., S, \tag{9}$$

with $S = 2$, $x_{s=1}(t = 0) = 4$ and $x_{s=2}(t = 0) = 7$.

3.3 Discussion

Interpolation of $\sigma_s(x - x_s(t))$ onto the FV mesh results in numerical approximation error. As a result, we observe residual jumps in the residual part, u_r, during FOM simulation. The aim is to build a reduced space by applying POD on the residual part. One option could be to build the bases (or reduced space) from the computed residual part (with residual jumps). An other alternative could be to post-process the residual part (computed during FOM) in order to get rid of the residual jumps. This post-processed residual part, which is even more low-rank approximable than the residual part with residual jumps, can be then used to build the (effective) reduced space. We invoke one of these ways to generate the bases and build a ROM.

We, first, consider the setting where the shock locations and jumps computed during FOM simulation are used during the ROM time-stepping i.e., we assume that $j_{s,N} = j_s$ and $x_{s,N} = x_s$. We, further, use the computed residual part (with residual jumps) to generate the bases. We can clearly see the benefits of the proposed approach in Fig. 1, which shows the behavior of the ROM error for increasing basis sizes N across different mesh resolutions. Firstly, the initial error incurred via the proposed approach is clearly lower than that of the standard approach. This is attributed to the fact that our decomposition approach associates a basis function corresponding to the travelling discontinuity. Secondly, the rate of decay of the error is better for the proposed approach compared to the standard approach. We also see that the ROM error for the standard approach is larger for finer mesh-sizes. This occurs as the effect of the shock becomes more pronounced for finer meshes. Also, the finer mesh implies less numerical viscosity. We also observe that the ROM errors could even increase with an increment in the basis size. It can be argued that this could occur as a result of insufficiently many basis functions. However, the ROM error for the proposed approach decreases with an increment in basis size. Moreover, the ROM error is lower (and stagnates later) for finer mesh-sizes. This can be argued from the fact that the proposed approach is able to resolve

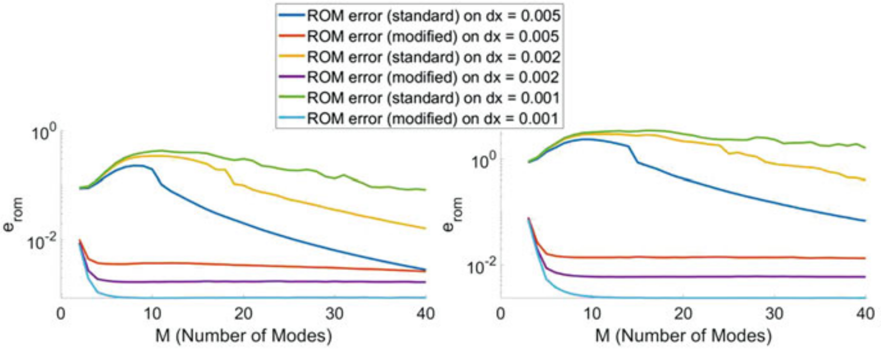


Fig. 1 ROM error upon using shock locations and jumps computed during FOM simulation: (left) single wavefront scenario and (right) multiple wavefront scenario

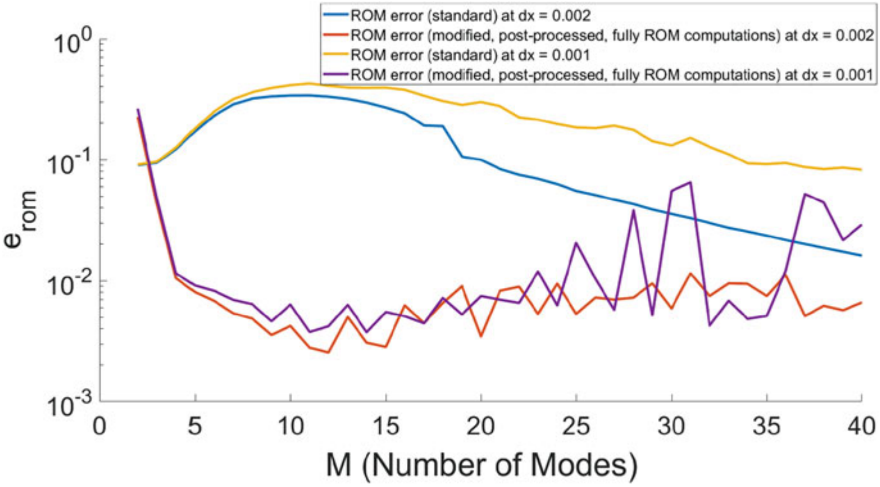


Fig. 2 ROM error under fully ROM computations for the single wavefront scenario

the shock more accurately at finer meshes. This error behavior is clearly in contrast to that of the standard approach which fails to efficiently capture the shock. As a result, the difference between the ROM error (at a certain number of basis function) computed via standard and proposed approach becomes even more pronounced for finer meshes.

Figure 2 demonstrates the performance for fully ROM computations, i.e., shocks locations, $x_{s,N}$ and jumps, $j_{s,N}$ are computed during ROM time-stepping. We perform post-processing on the residual part computed during FOM. u_r is post-processed by linear interpolation between the locations x_s^+, x_s^- where the local minimum u^+ and maximum u^- in a neighborhood of x_s is attained. We, then, generate the bases from this post-processed residual part. The post-processing was not needed in an earlier setting (discussed in the paragraph above) as accurate shock locations and jumps from the FOM simulation were used. However, it becomes essential here in order to approximate $x_{s,N}$ and $j_{s,N}$ within the ROM time-stepping with good accuracy. We observe that the proposed approach still performs better than the standard approach. However, the proposed approach seems to incur larger ROM error for larger POD mode numbers. Similar issues (not included in this paper) are observed for the multiple wavefront scenario. Such issues did not exist when we used the shock locations and jumps from FOM during the ROM time-stepping. Hence, the issues could be caused from a poor approximation of the shock. A possible explanation could be that we have more oscillations (around the shock position in the residual part) as the number of POD modes increases. The oscillations, which appear due to the reduced regularity of the residual part, lead to wrong computation of $x_{s,N}$ and $j_{s,N}$. It is clear that $x_{s,N}$ (and $j_{s,N}$) need to be approximated with good accuracy. The error in $x_{s,N}$, which would increase over time, should be in the order of the discretization error to achieve an overall

ROM error in the order of the discretization error. A mitigating measure could be to improve shock approximation similar to [14]. The high-frequency modes could also be a source of the problem. The potential solution could be to filter out the high-frequency modes when advancing the shock.

4 Conclusions

We have proposed a decomposition ansatz and used it in conjunction with POD. We have show-cased the performance of the proposed approach on the Burgers equation. The proposed approach is able to resolve the discontinuities and also offers reduction in ROM error. Future work will deal with resolving issues that exist in the proposed approach for larger POD mode numbers. Moreover, we will adapt the discussed formulation for system of conservation laws. We will also assess the performance of the method for parametrized scalar and system of conservation laws.

Acknowledgments The first author has been funded by the Shell NWO/ FOM PhD Programme in Computational Sciences for Energy Research. The second author has been funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2044–390685587, Mathematics Muenster: Dynamics–Geometry–Structure. The authors are also grateful for the support of COST Action TD1307, EU-MORNET (www.eu-mor.net).

References

1. M. Ohlberger and S. Rave, "Reduced basis methods: Success, limitations and future challenges," Proceedings of the Conference Algoritmy, pp. 1–12, 2016.
2. P. Benner, W. H. A. Schilders, S. Grivet-Talocia, A. Quarteroni, G. Rozza, and M. Silveira Luís, Model Order Reduction, Volume 2: Snapshot-Based Methods and Algorithms. Berlin, Boston: De Gruyter, 2020.
3. M. Ohlberger and S. Rave. Nonlinear reduced basis approximation of parameterized evolution equations via the method of freezing. C R Math, 351(23–24):901–906, 2013.
4. J. Reiss, P. Schulze, J. Sesterhenn, and V. Mehrmann, "The Shifted Proper Orthogonal Decomposition: A Mode Decomposition for Multiple Transport Phenomena," SIAM J. Sci. Comput., vol. 40, no. 3, pp. A1322–A1344, Jan. 2018.
5. N. Cagniart, Y. Maday, and B. Stamm. Model order reduction for problems with large convection effects, pages 131–150. Springer International Publishing, Cham, Switzerland, 2019.
6. C. W. Rowley, I. G. Kevrekidis, J. E. Marsden, and K. Lust, "Reduction and reconstruction for self-similar dynamical systems," Nonlinearity, vol. 16, no. 4, p. 1257, 2003.
7. J.-F. Gerbeau and D. Lombardi, "Approximated Lax pairs for the reduced order integration of nonlinear evolution equations," J. Comput. Phys, vol. 265, pp. 246–269, May 2014.
8. N. J. Nair and M. Balajewicz, "Transported snapshot model order reduction approach for parametric, steady-state fluid flows containing parameter-dependent shocks," Int. J. Numer. Meth. Eng, vol. 117, no. 12, pp. 1234–1262, 2019.

9. A. Iollo and D. Lombardi, “Advection modes by optimal mass transfer,” Phys. Rev. E, vol. 89, 022923, Feb. 2014.
10. D. Rim and K. T. Mandli, “Displacement Interpolation Using Monotone Rearrangement,” SIAM/ASA J. Uncertainty Quantification, vol. 6, no. 4, pp. 1503–1531, Jan. 2018.
11. V. Ehrlacher, D. Lombardi, O. Mula, and F.-X. Vialard, “Nonlinear model reduction on metric spaces. Application to one-dimensional conservative PDEs in Wasserstein spaces,” arXiv:1909.06626 [cs, math], Sep. 2019.
12. G. Welper, “Interpolation of Functions with Parameter Dependent Jumps by Transformed Snapshots,” SIAM J. Sci. Comput., vol. 39, no. 4, pp. A1225–A1250, Jan. 2017.
13. G. Welper, “Transformed Snapshot Interpolation with High Resolution Transforms,” arXiv:1901.01322 [math], Jan. 2019.
14. T. Taddei, S. Perotto, and A. Quarteroni, “Reduced basis techniques for nonlinear conservation laws,” ESAIM: M2AN, vol. 49, no. 3, pp. 787–814, May 2015.
15. K. Carlberg, “Adaptive h-refinement for reduced-order models,” Int. J. Numer. Meth. Eng, vol. 102, no. 5, pp. 1192–1210, 2015.
16. B. Peherstorfer, “Model reduction for transport-dominated problems via online adaptive bases and adaptive sampling,” arXiv:1812.02094 [cs, math], Dec. 2018.
17. K. Lee and K. Carlberg, “Deep Conservation: A latent dynamics model for exact satisfaction of physical conservation laws,” arXiv:1909.09754 [physics], Sep. 2019.
18. J. S. Hesthaven and S. Ubbiali, “Non-intrusive reduced order modeling of nonlinear problems using neural networks,” J. Comput. Phys, vol. 363, pp. 55–78, Jun. 2018.
19. W. J. Beyn, S. Selle, and V. Thummler. Freezing multipulses and multifronts. SIAM J. Appl. Dyn. Syst, 7(2):577–608, 2008.
20. M. Drohmann, B. Haasdonk, and M. Ohlberger, “Reduced Basis Approximation for Nonlinear Parametrized Evolution Equations based on Empirical Operator Interpolation,” SIAM J. Sci. Comput., vol. 34, no. 2, pp. A937–A969, Jan. 2012.

Numerical Simulation of a Phase-Field Model for Reactive Transport in Porous Media

Manuela Bastidas, Carina Bringedal, and Iuliu Sorin Pop

Abstract We consider a Darcy-scale model for mineral precipitation and dissolution in a porous medium. This model is obtained by homogenization techniques starting at the scale of pores. The model is based on a phase-field approach to account for the evolution of the pore geometry and the outcome is a multi-scale strongly coupled non-linear system of equations. In this work we discuss a robust numerical scheme dealing with the scale separation in the model as well as the non-linear character of the equations. We combine mesh refinement with stable linearization techniques to illustrate the behaviour of the multi-scale iterative scheme.

1 Introduction

Soil salinization and harvesting of geothermal energy are examples from real life in which the pore-scale geometry can be affected by mineral precipitation and dissolution. While these processes are active at the pore scale (micro scale) and affect the pore-scale structures, their effects are reflected in the Darcy-scale (macro-scale) parameters such as the porosity and permeability.

Several approaches are available to account for the evolution of the micro-scale geometry. To locate the micro-scale interfaces a layer thickness function is proposed in [8, 15], whereas a level set approach is considered in [2, 13, 14]. In both approaches, upscaled models can be derived by solving micro-scale problems involving moving interfaces. This makes the development of numerical schemes a

M. Bastidas (✉) · I. S. Pop
Faculty of Sciences, Hasselt University, Diepenbeek, Belgium
e-mail: manuela.bastidas@uhasselt.be; sorin.pop@uhasselt.be

C. Bringedal
Institute for Modelling Hydraulic and Environmental Systems, University of Stuttgart, Stuttgart, Germany
e-mail: carina.bringedal@iws.uni-stuttgart.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_8

challenging task as it requires a very fine mesh reproducing the micro-scale details such as tracking the movement of the interfaces.

Here we model the evolution of the micro-scale boundary through a phase-field equation. Then, moving interfaces are approximated by a thin diffuse interface layer described by a phase-field variable ϕ. This variable is an approximation of the characteristic function and approaches 1 in the fluid phase and 0 in the mineral. Using the phase-field approach we avoid the difficulties related to discontinuities in the domain. Building on the idea of minimizing the free energy developed in [5], a phase-field model for dissolution and precipitation processes is developed in [16]. In [12] this is extended to two fluid phases and the mineral phase. We consider the phase-field model proposed in [3] which considers one fluid phase and the mineral phase, but includes fluid flow. This model is hence defined over the entire domain where the evolution of the phase field accounts the moving fluid-mineral interface.

Since the main interest is the behaviour of the system at the macro scale, homogenization techniques are employed to derive upscaled models. The outcome is a coupled and non-linear system of equations addressing flow, chemistry and the phase-field evolution. Focusing on the two-scale model in [3], the main goal of this paper is to develop a robust numerical scheme accounting for both scale separation and the non-linearities in the model. This multi-scale iterative scheme borrows ideas from [4], where a stabilized iterative coupling scheme is introduced for a phase-field approach for fracture propagation.

This paper is organized as follows. In Sect. 2 the two-scale model is presented briefly, where the governing equations in two different scales are displayed and the strong coupling between the scales is discussed. In Sect. 3 we introduce the multi-scale iterative scheme in order to solve the upscaled model. There we give some details about handling the non-linearities and the convergence of the multi-scale iterative scheme. Finally, Sect. 4 provides a numerical example and the discussion of the results.

2 The Two-Scale Phase-Field Model

We consider the two-scale phase-field model formulation of single-phase fully saturated flow with constant density and viscosity introduced in [3]. There, the details about the formal homogenization procedure can be found. Here we restrict to presenting the upscaled model only.

We consider a periodic porous medium $\Omega \subseteq \mathbb{R}^2$. At each $\mathbf{x} \in \Omega$ we identify the variations at the micro-scale defining a fast variable. In other words, for each macro-scale point $\mathbf{x} \in \Omega$ we use one micro-scale cell $Y := [0, 1]^2$ to capture the fast changes encountered locally.

The unknowns $\mathbf{q}(\mathbf{x},t)$, $p(\mathbf{x},t)$ denote the macro-scale velocity and pressure in the fluid and $u(\mathbf{x},t)$ is the upscaled solute concentration. The macro-scale flow and solute transport problems are

$$\begin{aligned}
(\mathcal{P}_1^M) \quad & \begin{cases} \nabla \cdot \mathbf{q} = 0, & \text{in } \Omega_\mathrm{T} := \Omega \times (0,\mathrm{T}], \\ \mathbf{q} = -\mathcal{K}\nabla p, & \text{in } \Omega_\mathrm{T}, \end{cases} \\
(\mathcal{P}_2^M) \quad & \begin{cases} \partial_t(\overline{\phi}(u-u^\star)) + \nabla\cdot(\mathbf{q}u) = D\nabla\cdot(\mathcal{A}\nabla u), & \text{in } \Omega_\mathrm{T}, \end{cases}
\end{aligned} \tag{1}$$

completed by initial and boundary conditions. Here D is the solute diffusivity and $u^\star > u$ is the constant concentration of the species as part of the immobile mineral phase. Moreover, the variable $\overline{\phi}$ defines the porosity and it is nothing but the average of the phase field ϕ over the micro-scale Y. The matrices $\mathcal{A}$ and $\mathcal{K}$ are the effective diffusion and permeability, respectively.

For all $\mathbf{x} \in \Omega$ the phase field $\phi(\mathbf{x},\mathbf{y},t)$ is updated by solving the following micro-scale problem

$$(\mathcal{P}_\phi^\mu) \quad \begin{cases} \lambda^2\partial_t\phi + \gamma P'(\phi) = \gamma\lambda^2\nabla^2\phi - 4\lambda\phi(1-\phi)\dfrac{1}{u^\star}f(u), & \text{for } \mathbf{y} \in Y,\ t>0, \\ \phi \text{ is } Y\text{-periodic.} \end{cases} \tag{2}$$

This problem is defined for $\mathbf{y} \in Y$, while $\mathbf{x}$ enters as a parameter. Therefore, the spatial derivatives should be understood w.r.t. $\mathbf{y}$. The phase field ϕ has a smooth transition layer of width $\lambda > 0$ separating the phases. This equation is coupled with the macro scale through the reaction rate, which is chosen as $f(u) := \frac{u^2}{u_\mathrm{eq}^2} - 1$ with u_eq being a given equilibrium concentration. The term $P(\phi) = 8\phi^2(1-\phi)^2$ is the double-well potential, which ensures that the phase field approaches 0 and 1. The parameter γ is the diffusivity of the interface that separates the fluid and the mineral.

The macro-scale porosity in (1) is defined by the phase field $\overline{\phi}(\mathbf{x},t) := \int_Y \phi(\mathbf{x},\mathbf{y},t)d\mathbf{y}$. The elements of the effective matrices $\mathcal{A}(\mathbf{x},t)$ and $\mathcal{K}(\mathbf{x},t)$ are given by

$$\mathcal{A}_{\mathtt{rs}}(\cdot,t) = \int_Y \phi_\delta\left(\delta_{\mathtt{rs}} + \partial_{\mathtt{r}}\omega^{\mathtt{s}}\right)d\mathbf{y} \quad \text{and} \quad \mathcal{K}_{\mathtt{rs}}(\cdot,t) = \int_Y \phi_\delta\, \mathbf{w}_{\mathtt{r}}^{\mathtt{s}} d\mathbf{y} \tag{3}$$

for $\mathtt{r}, \mathtt{s} = 1, \ldots, \mathtt{d}$. The functions $\omega^{\mathtt{s}}$ and $\mathbf{w}^{\mathtt{s}} = [\mathbf{w}_1^{\mathtt{s}}, \ldots, \mathbf{w}_{\mathtt{d}}^{\mathtt{s}}]^t$ solve the following cell problems, defined for each $\mathbf{x} \in \Omega$ and $t > 0$

$$
\begin{aligned}
&(\mathcal{P}_A^{\mu}) \quad \begin{cases} \nabla \cdot (\phi_\delta(\nabla\omega^{\mathtt{s}} + \mathbf{e}_{\mathtt{s}})) = 0, & \text{in } Y, \\ \omega^{\mathtt{s}} \text{ is } Y\text{-periodic} \quad \text{and} \quad \displaystyle\int_Y \omega^{\mathtt{s}} d\mathbf{y} = 0, & \end{cases} \\
&(\mathcal{P}_K^{\mu}) \quad \begin{cases} (\nabla\Pi^{\mathtt{s}} + \mathbf{e}_{\mathtt{s}}) + \mu_f \nabla^2(\phi_\delta \mathbf{w}^{\mathtt{s}}) = \dfrac{g(\phi, \lambda)}{\phi_\delta}\mathbf{w}^{\mathtt{s}}, & \text{in } Y, \\ \nabla \cdot (\phi_\delta \mathbf{w}^{\mathtt{s}}) = 0, & \text{in } Y, \\ \Pi^{\mathtt{s}} \text{ is } Y\text{-periodic} \quad \text{and} \quad \displaystyle\int_Y \Pi^{\mathtt{s}} d\mathbf{y} = 0. & \end{cases}
\end{aligned} \tag{4}
$$

As before, in (4) the derivatives are w.r.t the $\mathbf{y}$ variable. The function $g(\phi, \lambda)$ in (4) ensures that the flow in the mineral phase becomes zero. This function is such that $g(1, \lambda) = 0$ and $g(0, \lambda) > 0$ (see [6]). Also observe the presence of a regularized phase-field $\phi_\delta := \phi + \delta$ where $\delta > 0$ is a regularization parameter which is included to avoid singularities in (4).

3 The Multi-Scale Iterative Scheme

We propose an iteratively coupled scheme to simulate the multi-scale behaviour of the phase-field model presented in Sect. 2. In [4, 10] similar approaches can be found but we remark that in the present case the coupling of different scales is encountered.

We let $N \in \mathbb{N}$ be the number of time steps and $\Delta t = \mathrm{T}/N$ be the time step size. For $n \in 1, \ldots, N$ define $t^n = n\Delta t$ and denote the time discrete solutions by $v^n := v(\cdot, t^n)$ for $v \in \{\phi, \mathcal{A}, \mathcal{K}, p, \mathbf{q}, u\}$.

Applying the Euler implicit discretization, at each time step a fully coupled nonlinear system of equations has to be solved. For each $n > 0$, the iterative algorithm defines a multi-scale sequence $\left\{\phi_j^n, \mathcal{A}_j^n, \mathcal{K}_j^n, p_j^n, \mathbf{q}_j^n, u_j^n\right\}$ with $j \geq 0$ being the iteration index. Naturally, the initial guess for ϕ_0^n and u_0^n are ϕ^{n-1} and u^{n-1}.

The iterative scheme follows the idea in [4]. We let $L_\phi > 0$ be a stabilization parameter and for $j > 0$ with given u_{j-1}^n and ϕ_{j-1}^n, one performs the following steps:

Step 1. *For each* $\mathbf{x} \in \Omega$, *find* ϕ_j^n *such that*

$$
\begin{aligned}
&\phi_j^n + \Delta t \gamma \nabla \cdot \phi_j^n - \frac{\Delta t}{\lambda^2} F(\phi_j^n, u_{j-1}^n) + L_\phi \left(\phi_j^n - \phi_{j-1}^n\right) = \phi^{n-1}, \quad inY \\
&\qquad \phi_j^n \text{ is } Y - periodic.
\end{aligned} \tag{5}
$$

where $F(\phi_j^n, u_{j-1}^n) := -\gamma P'(\phi_j^n) - 4\lambda\phi_j^n(1 - \phi_j^n)\frac{1}{u^\star} f(u_{j-1}^n)$.

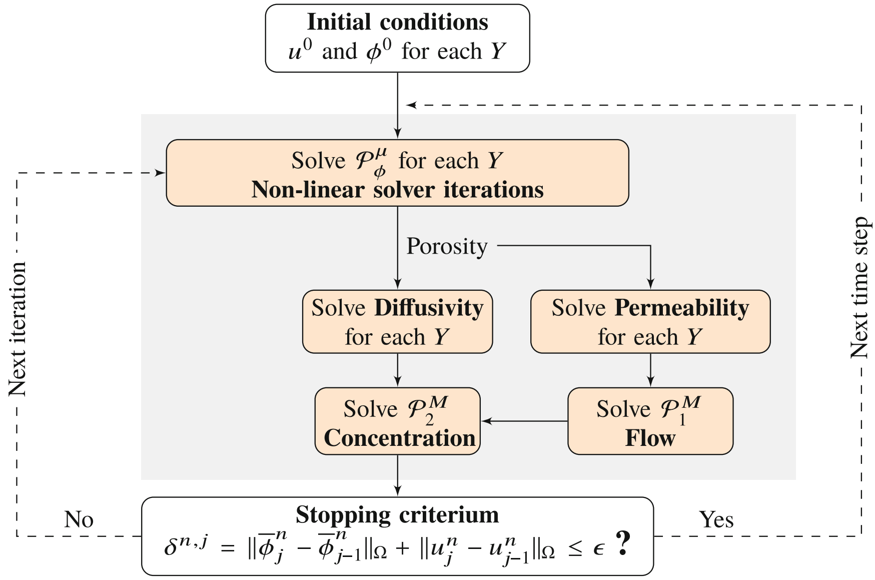

Initial conditions
u^0 and ϕ^0 for each Y
Solve $\mathcal{P}^{\mu}_{\phi}$ for each Y
Non-linear solver iterations
Porosity
Solve **Diffusivity** for each Y
Solve **Permeability** for each Y
Solve $\mathcal{P}^M_2$ **Concentration**
Solve $\mathcal{P}^M_1$ **Flow**
Stopping criterion
$\delta^{n,j} = \|\overline{\phi}^n_j - \overline{\phi}^n_{j-1}\|_\Omega + \|u^n_j - u^n_{j-1}\|_\Omega \leq \epsilon$ **?**
No
Yes
Next iteration
Next time step

Fig. 1 Sketch of the iterative scheme

Step 2. *Given ϕ_j^n find the effective matrices $\mathcal{A}_j^n$ and $\mathcal{K}_j^n$ in* (3) *by solving the cell problems* (4).

Step 3. *Given $\mathcal{K}_j^n$ and $\mathcal{A}_j^n$ find p_j^n, $\mathbf{q}_j^n$ and u_j^n by solving the system* (1).

In Fig. 1 we sketch the multi-scale iterative scheme. Here it is important to remark that the behaviour of an efficient and robust non-linear solver for (5) affects directly the convergence of the complete multi-scale iterative scheme. To deal with the non-linearities we use a fixed-point iteration scheme, called L-scheme (see [9, 11]).

The convergence of the iterative scheme in Steps 1–3 is a non-trivial task as it involves multiple scales and couples non-linear and possible degenerate systems of equations. Preliminary results are obtained in a simplified setting. Specifically, we assume that the pore space is never clogged and that the mineral never disappears completely. In other words, there exists two constants $\overline{\phi}_m, \overline{\phi}_M \in (0, 1)$ such that $0 < \overline{\phi}_m \leq \overline{\phi}(\mathbf{x}) \leq \overline{\phi}_M < 1$ for a.e $\mathbf{x} \in \Omega$. Moreover, the flow component is disregarded and the diffusion tensor is assumed not depending on the phase field. With $M_1 = \max\limits_{\substack{\phi\in[0,1]\\ u>0}} \{|\partial_1 F(\phi, u)|\}$, $M_2 = \max\limits_{\substack{\phi\in[0,1]\\ u>0}} \{|\partial_2 F(\phi, u)|\}$ and $\bar{u} = \max\limits_{\substack{\mathbf{x}\in\Omega\\ n\in\mathbb{N}}} \{|u^\star - u^n(\mathbf{x})|\}$ one can prove the following.

Proposition 1 *Let M_1, M_2, $\bar{u}$ and $\overline{\phi}_m$ be as above. If the time step is small enough, namely*

$$\Delta t \le 2\lambda^2 \min\left\{\frac{1}{M_1 + M_2 + 0.5\bar{u}}, \frac{\overline{\phi}_m^2}{M_2}\right\}$$

the scheme in Steps 1–3 is convergent.

The proof uses contraction arguments, we omit the details here.

4 A Numerical Example

We consider a simplified 2D situation where the processes are expected to be uniform in the vertical direction. The macro-scale domain is $\Omega = [0, 1]^2$, where a dissolution process is triggered by imposing a Dirichlet condition for the concentration on the right boundary of Ω. This configuration is displayed in Fig. 2, while Table 1 shows the parameters used for the simulation. In the following, all the solutions are computed using the lowest order Raviart–Thomas elements.

Figure 3 shows the evolution of the phase field corresponding to the macro scale location $(0.5, 0.5)$. At the micro scale we use a mesh refinement strategy to capture the movement of the phase-field transition zone.

At each time step we construct a micro-scale mesh with 800 elements. This mesh is refined in the first iteration of the scheme by following a prediction-correction strategy. We refer to [1, 7] for more details about handling similar meshes.

As mentioned before, we use an L-scheme dealing with the non-linearities at the micro scale. The non-linear term $F(\phi, u)$ needs to be split in a convex and concave part. Only the concave part is treated implicitly and the linearization parameter

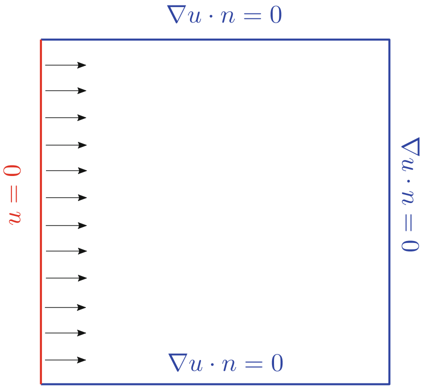


Fig. 2 The configuration of the macro-scale problem

Table 1 The parameters

Solute diff.	$D = 1$
Mineral concent.	$u^\star = 1$
Equilibrium	$u_{\text{eq}} = 0.5$
Diffusivity	$\gamma = 0.01$
Transition zone	$\lambda = 0.08$
Initial porosity	$\overline{\phi}_0 = 0.5$
Max. porosity	$\overline{\phi}_M = 0.87$
Initial condition	$u_0 = 0.5$
Stabilization	$L_\phi = 1\text{E}{-4}$

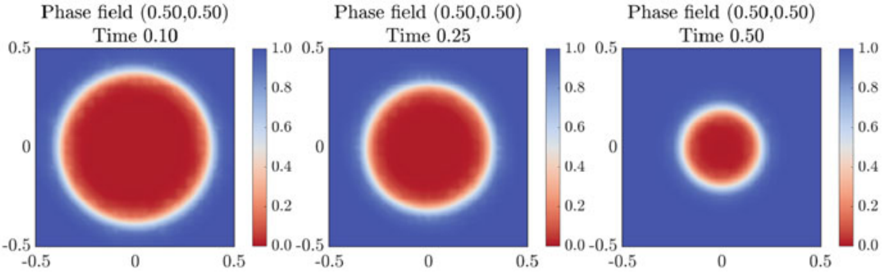


Fig. 3 The phase-field evolution at the macro-scale location (0.5, 0.5). From left to right, the phase field at $t = 0.2$, 0.25 and 0.5

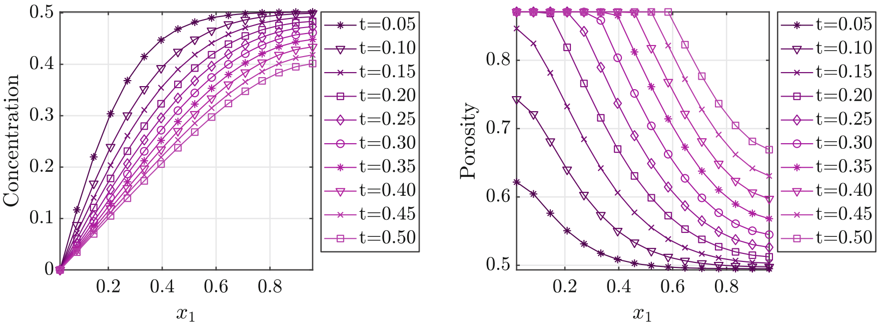


Fig. 4 The 1D projection of the concentration and the porosity at different times

corresponds to the Lipschitz constant of F with respect to ϕ (which depends on the concentration u) at every multi-scale iteration.

The Darcy-scale solute concentration is displayed in Fig. 4. Due to the chosen boundary and initial conditions, this solution does not depend on the vertical component and therefore the 1D projection in the horizontal direction is sufficient. The results for the porosity and the effective parameters are shown in Figs. 4 and 5.

We highlight that even if we are not computing the flow in this case, the effective permeability can still be calculated. Where the concentration decreases, it induces a

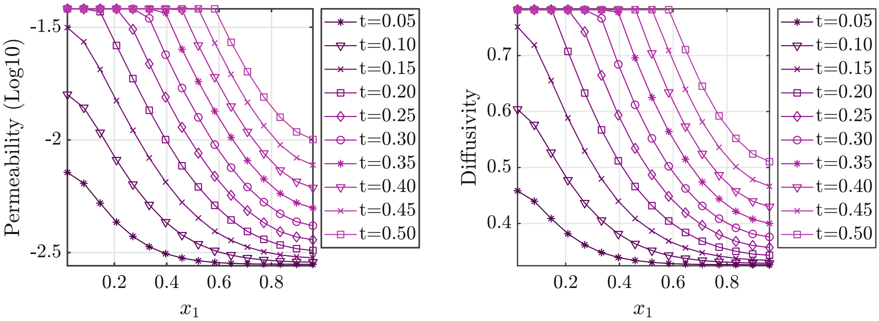


Fig. 5 The 1D projection of the effective parameters at different times

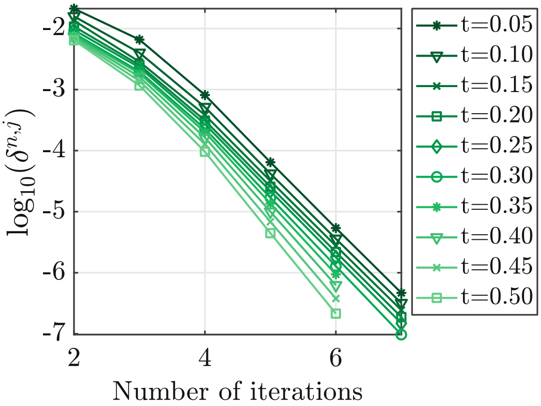


Fig. 6 The convergence of the multi-scale iterative scheme

dissolution of the mineral, which then increases the diffusivity and the permeability until the system reaches the maximum porosity $\overline{\phi}_M$.

Finally, in Fig. 6 we show the convergence of the norm of $\delta^{n,j} = \|\overline{\phi}^n_j - \overline{\phi}^n_{j-1}\|_\Omega + \|u^n_j - u^n_{j-1}\|_\Omega$ at different time steps. The non-linear solver at each micro-scale domain Y is stopped once the convergence criterion is below 1E−10.

In this numerical example the averaged number of degrees of freedom is 7,623,300 per time step. At the macro scale we have 512 elements and for each element the porosity and the effective parameters must be updated at each iteration. Due to the local mesh refinement the micro-scale degrees of freedom vary between 1200 and 1400. However, the micro-scale problems are solved in parallel and this can be improved by employing an adaptive strategy at the macro scale (see [12]).

We conclude that the multi-scale iterative scheme presented here is a valid approach to solve the two-scale phase-field model of precipitation and dissolution processes. This scheme can easily be parallelized and the resulting simulations show the influence of the micro-scale structural changes on the macro-scale parameters.

The next research steps are in the direction of proving the convergence of the full numerical scheme, including the error analysis of the micro-cell problems. Moreover, the study of the optimal choice of the stabilization parameter L_ϕ and the macro-scale adaptivity are important to enhance the efficiency of the scheme.

Acknowledgments The authors are supported by the Research Foundation—Flanders (FWO) through the Odysseus programme (Project G0G1316N). We thank Thomas Wick and Florin Adrian Radu who made valuable suggestions and contributed to the ideas behind this manuscript.

References

1. M. Bastidas, C. Bringedal, I. S. Pop, and F. A. Radu, *Adaptive numerical homogenization of non-linear diffusion problems*, arXiv preprint arXiv:1904.10665, (2019).
2. C. Bringedal, I. Berre, I. S. Pop, and F. A. Radu, *Upscaling of non-isothermal reactive porous media flow with changing porosity*, Transport in Porous Media, 114 (2016), pp. 371–393.
3. C. Bringedal, L. von Wolff, and I. S. Pop, *Phase field modeling of precipitation and dissolution processes in porous media: Upscaling and numerical experiments*, Multiscale Modeling & Simulation, (2020). Accepted.
4. M. K. Brun, T. Wick, I. Berre, J. M. Nordbotten, and F. A. Radu, *An iterative staggered scheme for phase field brittle fracture propagation with stabilizing parameters*, Computer Methods in Applied Mechanics and Engineering, 361 (2020), p. 112752.
5. G. Caginalp and P. C. Fife, *Dynamics of layered interfaces arising from phase boundaries*, SIAM Journal on Applied Mathematics, 48 (1988), pp. 506–518.
6. H. Garcke, C. Hecht, M. Hinze, and C. Kahle, *Numerical approximation of phase field based shape and topology optimization for fluids*, SIAM Journal on Scientific Computing, 37 (2015), pp. A1846–A1871.
7. T. Heister, M. F. Wheeler, and T. Wick, *A primal-dual active set method and predictor-corrector mesh adaptivity for computing fracture propagation using a phase-field approach*, Computer Methods in Applied Mechanics and Engineering, 290 (2015), pp. 466–495.
8. K. Kumar, T. Van Noorden, and I. S. Pop, *Effective dispersion equations for reactive flows involving free boundaries at the microscale*, Multiscale Modeling & Simulation, 9 (2011), pp. 29–58.
9. F. List and F. A. Radu, *A study on iterative methods for solving richards' equation*, Computational Geosciences, 20 (2016), pp. 341–353.
10. A. Mikelić and M. F. Wheeler, *Convergence of iterative coupling for coupled flow and geomechanics*, Computational Geosciences, 17 (2013), pp. 455–461.
11. I. S. Pop, F. A. Radu, and P. Knabner, *Mixed finite elements for the richards' equation: linearization procedure*, Journal of Computational and Applied Mathematics, 168 (2004), pp. 365–373.
12. M. Redeker, C. Rohde, and I. S. Pop, *Upscaling of a tri-phase phase-field model for precipitation in porous media*, IMA Journal of Applied Mathematics, 81 (2016), pp. 898–939.
13. R. Schulz, N. Ray, F. Frank, H. Mahato, and P. Knabner, *Strong solvability up to clogging of an effective diffusion–precipitation model in an evolving porous medium*, European Journal of Applied Mathematics, 28 (2017), pp. 179–207.

14. T. Van Noorden, *Crystal precipitation and dissolution in a porous medium: effective equations and numerical experiments*, Multiscale Modeling & Simulation, 7 (2009), pp. 1220–1236.
15. T. Van Noorden, *Crystal precipitation and dissolution in a thin strip*, European Journal of Applied Mathematics, 20 (2009), pp. 69–91.
16. T. Van Noorden and C. Eck, *Phase field approximation of a kinetic moving-boundary problem modelling dissolution and precipitation*, Interfaces and Free Boundaries, 13 (2011), pp. 29–55.

A Structure-Preserving Approximation of the Discrete Split Rotating Shallow Water Equations

Werner Bauer, Jörn Behrens, and Colin J. Cotter

Abstract We introduce an efficient split finite element (FE) discretization of a y-independent (slice) model of the rotating shallow water equations. The study of this slice model provides insight towards developing schemes for the full 2D case. Using the split Hamiltonian FE framework (Bauer et al., A structure-preserving split finite element discretization of the rotating shallow water equations in split Hamiltonian form (2019). https://hal.inria.fr/hal-02020379), we result in structure-preserving discretizations that are split into topological prognostic and metric-dependent closure equations. This splitting also accounts for the schemes' properties: the Poisson bracket is responsible for conserving energy (Hamiltonian) as well as mass, potential vorticity and enstrophy (Casimirs), independently from the realizations of the metric closure equations. The latter, in turn, determine accuracy, stability, convergence and discrete dispersion properties. We exploit this splitting to introduce structure-preserving approximations of the mass matrices in the metric equations avoiding to solve linear systems. We obtain a fully structure-preserving scheme with increased efficiency by a factor of two.

W. Bauer (✉)
Inria Rennes, France, and Imperial College London, London, UK
e-mail: w.bauer@imperial.ac.uk

J. Behrens
CEN/Department of Mathematics, Universität Hamburg, Hamburg, Germany
e-mail: joern.behrens@uni-hamburg.de

C. J. Cotter
Imperial College London, London, UK
e-mail: colin.cotter@imperial.ac.uk

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_9

1 Introduction

The notion of structure-preserving schemes describes discretizations that preserve important structures of the corresponding continuous equations: e.g. (i) the conservation of invariants such as energy, mass, vorticity and enstrophy in the case of the rotating shallow water (RSW) equations, (ii) the preservation of geometric structures such as $\mathsf{div}\ \mathsf{curl} = \mathsf{curl}\ \mathsf{grad} = 0$ or the Helmholtz decomposition of vector fields, and (iii) the conservation of large scale structures such as geostrophic or hydrostatic balances [15]. Their conservation is important to avoid, for instance, biases in the statistical behavior of numerical solutions [10] or to get models that correctly transfer energy and enstrophy between scales [12].

The construction of such schemes is an active area of research and various approaches to develop structure-preserving discretizations exist: e.g. variational discretizations [5, 6, 14] or compatible FE methods [9, 11]. In particular FE methods are a very general, widely applicable approach allowing for flexible use of meshes and higher order approximations. When combined with Hamiltonian formulations, they allow for stable discretizations of the RSW equations that conserve energy and enstrophy [2, 11]. However, they usually apply integration by parts to address the regularity properties of the FE spaces in use, which introduces additional errors and non-local differential operators. Moreover, FE discretizations usually involve mass matrices which are expensive to solve, while approximations of the mass matrices have to be designed carefully in order to preserve structure.

To address these disadvantages, we introduced in [3, 4] the split Hamiltonian FE method based on the split equations of Geophysical Fluid Dynamics [1], in which pairs of FE spaces are used such that integration by parts is avoided, and we derived structure-preserving discretizations of a y-independent RSW slice-model that preserve both the Hamiltonian and the split structures. Our method shares some basic ideas with *mimetic discretizations* (e.g. [7–9, 13]) in which PDEs are written in differential forms, but stresses a distinction between topological and metric parts and the use of a proper FE space for each variable.

Here, we address the disadvantage of FE methods arising from mass matrices. In the framework of split FEM [3, 4], we introduce approximations of the mass matrices in the metric equations resulting in a structure-preserving discretization of the split RSW slice-model that is more efficient than the original schemes introduced in [4]. To this end, we recall in Sect. 2 the split Hamiltonian framework and the split RSW slice-model, and we introduce the approximation of the metric equations. In Sect. 3, we present numerical results and in Sect. 4 we draw conclusions.

2 Split Hamiltonian FE Discretization of the RSW Slice-Model

On the example of a y-independent slice model of the RSW equations, we recall the split Hamiltonian FE method of [4]. For pairs of height fields (straight 0-form $h^{(0)}$ and twisted 1-form $\widetilde{h}^{(1)}$), of velocity fields in x-direction (twisted 0-form $\widetilde{u}^{(0)}$ and

Fig. 1 Relation between operators and spaces

$$\begin{tikzcd}
h^{(0)}, v^{(0)} \in \Lambda^0 \arrow[r, "\mathrm{d}"] \arrow[d, leftrightarrow, "\tilde{\star}"] & \Lambda^1 \ni u^{(1)} \arrow[d, leftrightarrow, "\tilde{\star}"] \\
\widetilde{h}^{(1)}, \widetilde{v}^{(1)} \in \widetilde{\Lambda}^1 & \widetilde{\Lambda}^0 \ni \widetilde{u}^{(0)} \arrow[l, "\mathrm{d}"']
\end{tikzcd}$$

straight 1-form $u^{(1)}$), and of velocity fields in outer slice direction (straight 0-form $v^{(0)}$ and twisted 1-form $\widetilde{v}^{(1)}$), this RSW slice-model reads [4]

$$\frac{\partial u^{(1)}}{\partial t} - q^{(1)} F_v^{(0)} + \mathrm{d}\, B^{(0)} = 0, \quad \frac{\partial \widetilde{v}^{(1)}}{\partial t} + q^{(1)} \widetilde{F}_u^{(0)} = 0, \quad \frac{\partial \widetilde{h}^{(1)}}{\partial t} + \mathrm{d}\widetilde{F}_u^{(0)} = 0,$$
$$\widetilde{u}^{(0)} = \tilde{\star} u^{(1)}, \qquad v^{(0)} = \tilde{\star}\widetilde{v}^{(1)}, \qquad \widetilde{h}^{(1)} = \tilde{\star} h^{(0)}, \tag{1}$$

in which $\widetilde{F}_u^{(0)} := h^{(0)}\widetilde{u}^{(0)}$ and $F_v^{(0)} := h^{(0)} v^{(0)}$ are mass fluxes, $B^{(0)} := gh^{(0)} + \frac{1}{2}(\widetilde{u}^{(0)})^2 + \frac{1}{2}(v^{(0)})^2$ is the Bernoulli function with gravitational constant g. $q^{(1)} = \tilde{\star}\widetilde{q}^{(0)} = \widetilde{q}^{(0)}\widetilde{dx}$ is the potential vorticity (PV) defined implicitly via $\widetilde{q}^{(0)}\widetilde{h}^{(1)} = \mathrm{d}v^{(0)} + f dx$ with Coriolis parameter f. All variables are functions of x and t: for instance, $u^{(1)}(x, t)$ is the coefficient function of the 1-form $u^{(1)} = u^{(1)}(x, t)dx$.

The pairs of variables are connected via the twisted Hodge-star operator $\tilde{\star}$: $\Lambda^k \to \widetilde{\Lambda}^{(1-k)}$ (see definition in [1]) that maps from straight k-forms to twisted $(1 - k)$-forms (or vice versa) with $k = 0, 1$ in one dimension (1D). The index $^{(k)}$ denotes the degree, and Λ^k, $\widetilde{\Lambda}^k$ the space of all k-forms. Note that straight forms do not change their signs when the orientation of the manifold changes in contrast to twisted forms. The exterior derivative d is a mapping $\mathrm{d} : \Lambda^k \to \Lambda^{k+1}$ (or $\mathrm{d} : \widetilde{\Lambda}^k \to \widetilde{\Lambda}^{k+1}$). Here in 1D, it is simply the total derivative of a smooth function $g^{(0)} \in \Lambda^0$, $\mathrm{d}\, g^{(0)} = \partial_x g(x) dx \in \Lambda^1$ (see [1] for full details). Figure 1 illustrates the relations between the operators and spaces.

Galerkin Discretization To substitute FE for continuous spaces, we consider $\Lambda_h^0, \widetilde{\Lambda}_h^0 = CG_p$ and $\Lambda_h^1, \widetilde{\Lambda}_h^1 = DG_{p-1}$ with polynomial order p. We allow the discrete Hodge star operators $\tilde{\star}_h^0 : \widetilde{\Lambda}_h^1 \to \Lambda_h^0$ and $\tilde{\star}_h^1 : \Lambda_h^1 \to \widetilde{\Lambda}_h^0$ to be non-invertible. The split FE discretization of Eqs. (1) seeks solutions $u_h^{(1)}, \widetilde{v}_h^{(1)}, \widetilde{h}_h^{(1)} \in (\Lambda_h^1(L), \widetilde{\Lambda}_h^1(L), \widetilde{\Lambda}_h^1(L))$ of the *topological equations* (as trivial projections)

$$\langle \chi_h^{(1)}, \frac{\partial}{\partial t} u_h^{(1)} \rangle - \langle \chi_h^{(1)}, q_h^{(1)} F_{v_h}^{(0)} \rangle + \langle \chi_h^{(1)}, \mathrm{d}\, B_h^{(0)} \rangle = 0, \qquad \forall \chi_h^{(1)} \in \Lambda_h^1, \tag{2}$$

$$\langle \widetilde{\chi}_h^{(1)}, \frac{\partial \widetilde{v}_h^{(1)}}{\partial t} \rangle + \langle \widetilde{\chi}_h^{(1)}, q_h^{(1)} \widetilde{F}_{u_h}^{(0)} \rangle = 0, \qquad \forall \widetilde{\chi}_h^{(1)} \in \widetilde{\Lambda}_h^1, \tag{3}$$

$$\langle \widetilde{\chi}_h^{(1)}, \frac{\partial \widetilde{h}_h^{(1)}}{\partial t} \rangle + \langle \widetilde{\chi}_h^{(1)}, \mathrm{d}\, \widetilde{F}_{u_h}^{(0)} \rangle = 0, \qquad \forall \widetilde{\chi}_h^{(1)} \in \widetilde{\Lambda}_h^1, \tag{4}$$

subject to the *metric closure equations* (as non-trivial Galerkin projections (GP))

$$\mathrm{GP}(1\text{-}i)_u: \int_L \tau^{(i)} \wedge \begin{cases} \tilde{\star}(\tilde{\star}_h^1 u_h^{(1)}) \\ \tilde{\star}_h^1 u_h^{(1)} \end{cases} = \int_L \tau^{(i)} \wedge \begin{cases} \tilde{\star}\widetilde{u}_h^{(0)} & \text{if } i=0 \\ \widetilde{u}_h^{(0)} & \text{if } i=1 \end{cases} \forall \tau^{(i)} \in \Lambda_h^i \tag{5}$$

as realizations of $\tilde{\star}_h^1 : \Lambda_h^1 \to \widetilde{\Lambda}_h^0$ and, similarly defined, $\mathrm{GP}(1\text{-}i)_v$ and $\mathrm{GP}(1\text{-}j)_h$, $j = 0, 1$, as realizations of $\tilde{\star}_h^0 : \widetilde{\Lambda}_h^1 \to \Lambda_h^0$. $q_h^{(1)} := \tilde{\star}\widetilde{q}_h^{(0)}$ is a discrete 1-form with coefficient function $\widetilde{q}_h^{(0)}(x) \in \widetilde{\Lambda}_h^0$ determined by $\langle \tilde{\star}\widetilde{\phi}_h^{(0)}, \widetilde{q}_h^{(0)}\widetilde{h}_h^{(1)}\rangle + \langle \mathrm{d}\,\widetilde{\phi}_h^{(0)}, \widetilde{v}_h^{(1)}\rangle - \langle \tilde{\star}\widetilde{\phi}_h^{(0)}, f\,dx\rangle = 0, \forall \widetilde{\phi}_h^{(0)} \in \widetilde{\Lambda}_h^0$. $B_h^{(0)}$, $\widetilde{F}_{u_h}^{(0)}$, $F_{v_h}^{(0)}$ follow from the definitions above. $\langle \cdot, \cdot \rangle := \int_L \cdot \wedge \tilde{\star} \cdot$ is the L_2 inner product on the domain L. We distinguish between continuous and discrete Hodge star operators $\tilde{\star}$ and $\tilde{\star}_h$, respectively. $\tilde{\star}$ is used in $\langle , \rangle$ such that k-forms of the same degree are multiplied, while $\tilde{\star}_h$ is realized as in Eqs. (5) via non-trivial GP between 0- and 1-forms, cf. [3]. As the prognostic equations (2)–(4) hold, as those in (1), pointwise and consist of forms, we denote them as topological.

2.1 Continuous and Discrete Split Hamiltonian RSW Slice-Model

Both the continuous split RSW slice-model of Eqs. (1) and the corresponding weak (discrete) form of (2)–(5) can equivalently be written in Hamiltonian form, as shown in [4]. Considering the discrete version, the Hamiltonian with metric equations reads

$$\begin{aligned} \mathcal{H}[u_h^{(1)}, \widetilde{v}_h^{(1)}, \widetilde{h}_h^{(1)}] &= \frac{1}{2}\langle u_h^{(1)}, \tilde{\star}h_h^{(0)}\widetilde{u}_h^{(0)}\rangle + \frac{1}{2}\langle \widetilde{v}_h^{(1)}, \tilde{\star}h_h^{(0)} v_h^{(0)}\rangle + \langle \widetilde{h}_h^{(1)}, \tilde{\star}g h_h^{(0)}\rangle \\ \widetilde{u}_h^{(0)} &= \tilde{\star}_h^1 u_h^{(1)},\ v_h^{(0)} = \tilde{\star}_h^0 \widetilde{v}_h^{(1)},\ h_h^{(0)} = \tilde{\star}_h^0 \widetilde{h}_h^{(1)} \text{ (metric eqns.)} \end{aligned} \tag{6}$$

while the almost *Poisson bracket* {, } is defined as

$$\{\mathcal{F}, \mathcal{G}\} := -\langle \frac{\delta \mathcal{F}}{\delta \widetilde{h}_h^{(1)}}, \mathrm{d}\,\tilde{\star}\frac{\delta \mathcal{G}}{\delta u_h^{(1)}}\rangle - \langle \frac{\delta \mathcal{F}}{\delta u_h^{(1)}}, \mathrm{d}\,\tilde{\star}\frac{\delta \mathcal{G}}{\delta \widetilde{h}_h^{(1)}}\rangle + \langle \frac{\delta \mathcal{F}}{\delta u_h^{(1)}}, q_h^{(1)}\tilde{\star}\frac{\delta \mathcal{G}}{\delta \widetilde{v}_h^{(1)}}\rangle - \langle \frac{\delta \mathcal{F}}{\delta \widetilde{v}_h^{(1)}}, q_h^{(1)}\tilde{\star}\frac{\delta \mathcal{G}}{\delta u_h^{(1)}}\rangle \tag{7}$$

with $q_h^{(1)}$ defined as above. Then, the dynamics for any functional $\mathcal{F}[u_h^{(1)}, \widetilde{v}_h^{(1)}, \widetilde{h}_h^{(1)}]$ is given by $\frac{d}{dt}\mathcal{F} = \{\mathcal{F}, \mathcal{H}\}$.

Splitting of Schemes Properties The split Hamiltonian FE method results in a family of schemes in which the schemes' properties split into *topological and metric dependent ones*, cf. [4].

The **topological properties** hold for all double FE pairs that fulfill the double complex structure of diagram (1) (shown in [4]). In particular,

- the total energy is conserved, because $\frac{d}{dt}\mathcal{H} = \{\mathcal{H}, \mathcal{H}\} = 0$ which follows from the antisymmetry of (7);
- the Casimirs $C = M, PV, PE$ are conserved as $\frac{d}{dt}C = \{C, \mathcal{H}\} = 0$ for $\{C, \mathcal{G}\} = 0 \forall \mathcal{G}$ with $C = \langle \widetilde{h}_h^{(1)}, \tilde{\star} F(\widetilde{q}_h^{(0)})\rangle$ for $F = 1(M)$, $F = \widetilde{q}_h^{(0)}\tilde{1}(PV)$, $F = (\widetilde{q}_h^{(0)})^2(PE)$;
- $\{,\}$ is independent of $\tilde{\star}_h$, hence $\mathcal{H}, C$ are conserved for any metric equation.

The **metric properties** are associated to a certain choice of FE spaces. In particular, this choice determines

- the dispersion relation which usually depends on Δx between degrees of freedom (DoFs),
- the stability, because the inf-sup condition depends on the norm, and
- convergence and accuracy, which both are measured with respect to norms.

2.2 *Family of Structure-Preserving Split RSW Schemes*

Besides the splitting into topological and metric properties, another remarkable feature of the split FE framework is that one choice of compatible FE pairs leads to a family of split schemes, cf. [3, 4]. In the following, we consider for $p = 1$ the piecewise linear space $\Lambda_h^0, \widetilde{\Lambda}_h^0 = CG_p = \text{P1}$ with basis $\{\phi_l(x)\}_{l=1}^N$ and the piecewise constant space $\Lambda_h^1, \widetilde{\Lambda}_h^1 = DG_{p-1} = \text{P0}$ with basis $\{\chi_m(x)\}_{m=1}^N$. Being in a 1D domain with periodic boundary, both have N independent DoFs. We approximate 0-forms in P0 and 1-forms in P1, e.g. $u_h^{(1)}(x,t) = \sum_{m=1}^N u_m(t)\chi_m(x)$. The split framework [4] leads to *one set of discrete topological equations* for Eqs. (2)–(4), and *four combinations of discrete metric equations* for (5) (using the Hadamard product $\circ$):

$$\textbf{topological momentum eqns.}: \quad \frac{\partial}{\partial t}\mathbf{u}_e^1 - \mathbf{q}_e^1 \circ \mathbf{F}_{\mathbf{v}_n}^0 + \mathbf{D}^{en}\mathbf{B}_n^0 = 0, \quad \frac{\partial}{\partial t}\tilde{\mathbf{v}}_e^1 + \mathbf{q}_e^1 \circ \widetilde{\mathbf{F}}_{\mathbf{u}n}^0 = 0,$$

$$\textbf{topological continuity eqns.}: \quad \frac{\partial}{\partial t}\tilde{\mathbf{h}}_e^1 + \mathbf{D}^{en}\widetilde{\mathbf{F}}_{\mathbf{u}n}^0 = 0,$$

$$\textbf{metric closure eqns.}: \quad \begin{tikzcd}
\mathbf{h}_n^0 \in \Lambda_h^0 \subset P1 \arrow[r, "\mathbf{D}^{en}"] & \Lambda_h^1 \subset P0 \ni \mathbf{u}_e^1, (\tilde{\mathbf{v}}_e^1 \in \widetilde{\Lambda}_h^1) \arrow[d, "\text{GP1}_u:\ \mathbf{M}^{nn}\tilde{\mathbf{u}}_n^0 = \mathbf{P}^{ne}\mathbf{u}_e^1\ \&\ \mathbf{M}^{nn}\mathbf{v}_n^0 = \mathbf{P}^{ne}\tilde{\mathbf{v}}_e^1"] \\
{} \arrow[u, "\text{GP1}_h:\ \mathbf{M}^{nn}\mathbf{h}_n^0 = \mathbf{P}^{ne}\tilde{\mathbf{h}}_e^1"'] & {} \arrow[d, "\text{GP0}_u:\ \mathbf{M}^{en}\tilde{\mathbf{u}}_n^0 = \mathbf{u}_e^1\ \&\ \mathbf{M}^{en}\mathbf{v}_n^0 = \tilde{\mathbf{v}}_e^1"] \\
\tilde{\mathbf{h}}_e^1 \in \widetilde{\Lambda}_h^1 \subset P0 \arrow[u, "\text{GP0}_h:\ \mathbf{M}^{en}\mathbf{h}_n^0 = \tilde{\mathbf{h}}_e^1"'] & \widetilde{\Lambda}_h^0 \subset P1 \ni \tilde{\mathbf{u}}_n^0, (\mathbf{v}_n^0 \in \Lambda_h^0). \arrow[l, "\mathbf{D}^{en}"']
\end{tikzcd} \tag{8}$$

We used the following $(N \times N)$ matrices with index n for nodes and e for elements: (i) mass matrices $\mathbf{M}^{nn}$, $\mathbf{M}^{ee}$, $\mathbf{M}^{en}$, with metric-dependent coefficients $(\mathbf{M}^{nn})_{ll'} = \int_L \phi_l(x)\phi_{l'}(x)dx$, $(\mathbf{M}^{ee})_{mm'} = \int_L \chi_m(x)\chi_{m'}(x)dx$, $(\mathbf{M}^{en})_{lm'} = \int_L \phi_l(x)\chi_{m'}(x)dx$ (with $\widetilde{\mathbf{M}^{en}} = \{\mathbf{M}^{en}\text{in Or}, -\mathbf{M}^{en}\text{in -Or}\}$ for orientation Or of L and $\mathbf{M}^{en} = (\mathbf{M}^{ne})^T$ with T for the transposed matrix); and (ii) the stiffness matrix $\mathbf{D}^{en}$ with metric-independent coefficient $(\mathbf{D}^{en})_{lm'} = \int_L \frac{d\phi_l(x)}{dx}\chi_{m'}(x)dx$ (with $\mathbf{D}^{en} = (\mathbf{D}^{ne})^T$). We separate $\mathbf{M}^{ne} = \mathbf{P}^{ne}\,(\boldsymbol{\Delta}\mathbf{x}_e)^T$ into a metric-dependent $\boldsymbol{\Delta}\mathbf{x}_e$ and a metric-free part $\mathbf{P}^{ne}$, the latter is an averaging operator from e to n values (similarly for $\mathbf{M}^{en}$ and $\mathbf{P}^{en}$).

Moreover, $\mathbf{u}^1_e = \mathbf{M}^{ee}\mathbf{u}_e$ is a discrete 1-form associated to the vector array $\mathbf{u}_e = \{u_m(t)|m = 1,\dots N\}$ while $\tilde{\mathbf{h}}^1_e = \mathbf{M}^{ee}\tilde{\mathbf{h}}_\mathbf{e}$ (or $\tilde{\mathbf{v}}^1_e$) is a discrete 1-form with $\tilde{\mathbf{h}}_e = \{\widetilde{h}_m(t)|m = 1,\dots N\}$. The PV 1-form reads $\mathbf{q}^1_e = \widetilde{\mathbf{M}^{en}}\tilde{\mathbf{q}}^0_n = \widetilde{\mathbf{P}^{en}}\tilde{\mathbf{q}}^0_n\,(\boldsymbol{\Delta}\mathbf{x}_e)^T$ in agreement with the definition in (1). Discrete 0-forms read, e.g. $\mathbf{h}^0_n = \{h_l(t)|l = 1,\dots N\}$. The discrete mass fluxes are $\widetilde{\mathbf{F}_{\mathbf{u}_n}}^0 = \mathbf{h}^0_n \circ \tilde{\mathbf{u}}^0_n$ and $\mathbf{F}^0_{\mathbf{v}_n} = \mathbf{h}^0_n \circ \mathbf{v}^0_n$ and the discrete Bernoulli function reads $\mathbf{B}^0_n = \frac{1}{2}\tilde{\mathbf{u}}^0_n \circ \tilde{\mathbf{u}}^0_n + \frac{1}{2}\mathbf{v}^0_n \circ \mathbf{v}^0_n + g\mathbf{h}^0_n$. Finally, GP1$_u$, GP0$_u$, GP1$_h$, GP0$_h$ are the nonlinear GPs of (5) for P1 and P0 test functions.

2.3 A Structure-Preserving Approximation of Split RSW Schemes

Here we introduce a new, computationally more efficient split RSW scheme compared to those of [4]. We exploit the splitting of the topological and metric properties within the split FE framework to introduce structure-preserving approximations of the mass matrices used in the metric equations. Instead of using the full nontrivial Galerkin projections GP1$_h$, GP0$_h$ for height or GP1$_u$, GP0$_u$ for velocity u, v, we use the **averaged** versions:

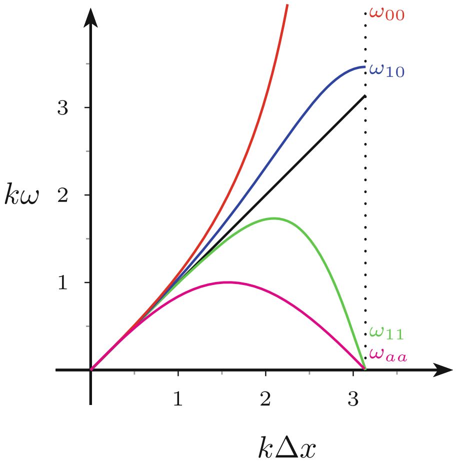
$$\text{AVG}_h : \mathbf{h}^0_n = \mathbf{P}^{ne}\tilde{\mathbf{h}}^1_e\,, \quad \text{AVG}_u : \tilde{\mathbf{u}}^0_n = \mathbf{P}^{ne}\mathbf{u}^1_e\,,$$

and denote the resulting scheme with AVG$_u$–AVG$_h$. Rather then solving linear systems in (8), we obtain values for $\mathbf{h}^0_n, \tilde{\mathbf{u}}^0_n, \mathbf{v}^0_n$ simply by averaging. This is computationally more efficient. In fact, already for this 1D problem we achieve a speedup by a factor of 2 (wall clock time) compared to the full GPs.

As stated in Sect. 2.1, such modification does not impact on the structure-preserving properties but will change the metric-dependent ones instead. Before we confirm this in Sect. 3 numerically, we first determine analytically the discrete dispersion relation related to this approximation. A similar calculation as done in [3] leads to the following discrete dispersion relation:

$$c_d = \frac{\omega_{aa}}{k} = \pm\sqrt{gH}\frac{1}{k\Delta x}\sin(k\Delta x)$$

Fig. 2 Dispersion relations: analytic (black) for $c = \sqrt{gH} = 1$, ω_{11} for GP1$_u$–GP1$_h$, ω_{10} for GP1$_u$–GP0$_h$ and GP0$_u$–GP1$_h$, ω_{00} for GP0$_u$–GP0$_h$ (cf. [4]), and ω_{aa} for AVG$_u$–AVG$_h$



with angular frequency $\omega_{aa} = \omega_{aa}(k)$ and discrete wave speed $c_d \to c = \sqrt{gH}$ (with mean height H) in case $k \to 0$ and with a spurious mode (second zero root) at shortest wave length $k = \frac{\pi}{\Delta x}$. As shown in Fig. 2 (with results relative to the nondimensional wave speed $c = \sqrt{gH} = 1$), this is similar to the dispersion relation of the GP1$_u$–GP1$_h$ scheme in the sense that both have a spurious mode at $k = \frac{\pi}{\Delta x}$, cf. [3]. For completeness, we added the dispersion relations for the other possible realizations of the metric equations (8) as introduced in [3, 4].

3 Numerical Results

We study the structure-preserving properties, as well as convergence, stability and dispersion relation for the averaged split scheme AVG$_u$– AVG$_h$ and compare it with the split schemes of [4]. We use test cases (TC) in the quasi-geostrophic regime such that effects of both gravity waves and compressibility are important.

The study of structure preservation (topological properties) will be performed with a flow in geostrophic balance in which the terms are linearly balanced while nonlinear effects are comparably small (Fig. 3). To illustrate the long term behaviour, we run the simulation in this TC 1 for about 10 cycles (meaning that the (analytical) wave solutions have traveled 10 times over the entire domain). To test convergence and stability (i.e. metric-dependent properties), we use in TC 2 a steady state solutions of Eqs. (1). To illustrate the metric dependency of the dispersion relations, we use in TC 3 an initial height distribution (as in Fig. 3, left) that is only partly in linear geostrophic balance such that shock waves with small scale oscillations develop that depend on the dispersion relation. More details on the TC can be found in [4].

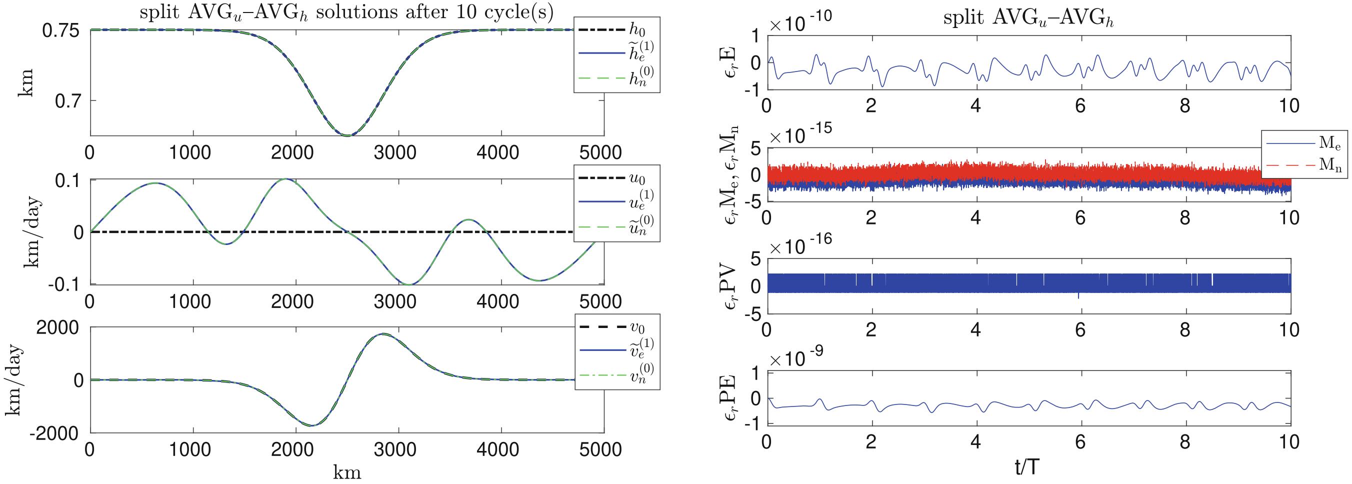


Fig. 3 Solutions for the split AVG_u–AVG_h scheme for a mesh with 512 elements (similar for other resolutions). Initial fields are shown as dashed-dotted lines, ϵ_r denotes the relative error. Left: flow in geostrophic balance after 10 cycles. Right: time series of the quantities of interest for 10 cycles

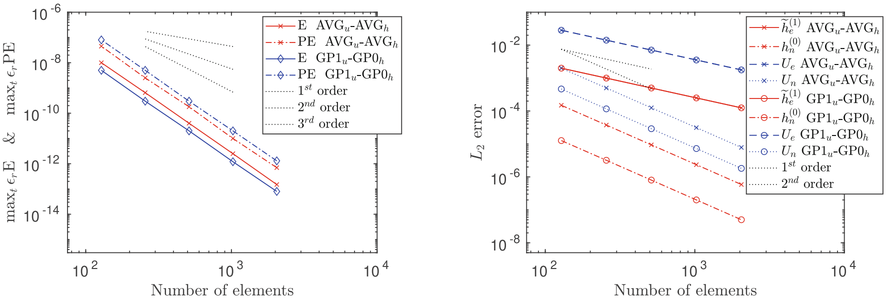


Fig. 4 Relative error values in dependence of N of AVG$_u$– AVG$_h$ compared to GP1$_u$– GP0$_h$ [4]. Left: errors for E and PE for TC 1. Right: errors for the steady state solution of TC 2 after 1 cycle

Topological Properties Figure 3 (right) shows for TC 1 the relative errors of the averaged split scheme for energy E, mass M_e or M_n, potential vorticity PV and enstrophy PE (see definitions in [4]). In all cases studied, these quantities exhibit no long term trend while M_e, M_n and PV are preserved at machine precision. The lower accuracy in E and PE result from using a Crank Nicolson time scheme. With increased resolution these errors decrease with third order rate (Fig. 4, left), cf. [4].

When compared to the split schemes of [4], these error values are very close to the results presented therein, underpinning the fact that modifications in the metric equations do not affect the quality of structure preservation of the schemes.

Metric-Dependent Properties Consider next the convergence behaviour of the averaged split scheme AVG$_u$– AVG$_h$ shown in Fig. 4 (right) for TC 2. To ease comparison, we include L_2 error values of the split scheme GP1$_u$– GP0$_h$ of [4] noting that the other split schemes presented therein share more or less the same error values for the corresponding fields. In all cases, the error values decrease as expected: all P1 fields show second order, all P0 fields first order convergences rates.

While the errors of the P0 fields of AVG$_u$– AVG$_h$ is close to the corresponding values of the split schemes of [4], the P1 fields of AVG$_u$– AVG$_h$ have error values that are about one order of magnitude large than the corresponding fields of e.g. GP1$_u$– GP0$_h$. This agrees well with the fact that we do not solve the full linear system in the metric equations to recover the P1 fields but use instead approximations, which slightly increases the P1 error values of AVG$_u$– AVG$_h$.

With TC 3 we illustrate numerically how the choice of metric equations determines the discrete dispersion relations. As derived in Sect. 3, the discrete dispersion relation of AVG$_u$– AVG$_h$ equals a sine wave, hence all waves of frequency k have wave speeds equal or slower than c (black curve in Fig. 2). In particular for wave numbers larger then $\frac{\pi}{2\Delta x}$, waves start to slow down until there is a standing wave at $k = \frac{\pi}{\Delta x}$. This is a similar behavior to the GP1$_u$– GP1$_h$ scheme of [4], but for AVG$_u$– AVG$_h$ this effect is stronger given the generally slower wave propagation. This behaviour is clearly visible in Fig. 5 where we observe in both fields lower

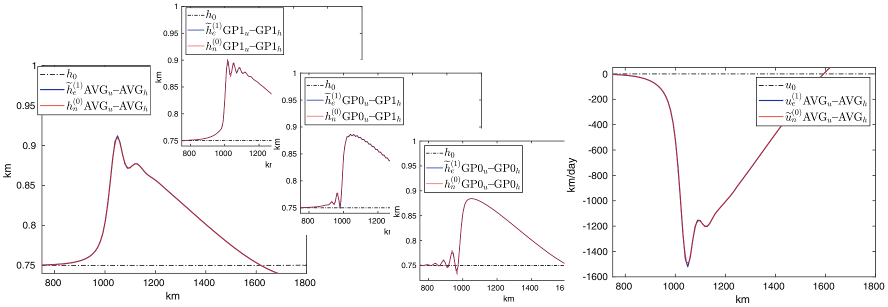


Fig. 5 Fields with oscillations at the wave fronts in dependency of the wave dispersion relations of Fig. 2 on a mesh with $N_e = 512$ elements and after a simulation time of 0.225 cycles

frequency oscillation behind the front when compared to GP1$_u$–GP1$_h$ (see inlet). This result agrees well with the discrete dispersion relations shown in Fig. 2.

4 Conclusions

We introduced a y-independent RSW slice-model in split Hamiltonian form and derived a family of lowest-order (P0–P1) structure-preserving split schemes. The splitting of the equations into topological and metric parts transfers also to schemes' properties. The framework allows for different realizations of metric equations which all preserve the Hamiltonian and the Casimirs of the Poisson bracket. This allowed us to introduce an approximation of the metric equations which is structure-preserving, achieving a speedup of a factor of 2 because no linear systems had to be solved.

References

1. Bauer, W. [2016], A new hierarchically-structured n-dimensional covariant form of rotating equations of geophysical fluid dynamics, *GEM - Intern. J. Geomathematics*, **7(1)**, 31–101.
2. Bauer, W., Cotter, C. J. [2018], Energy-enstrophy conserving compatible finite element schemes for the shallow water equations on rotating domains with boundaries, *J. Comput. Physics*, **373**, 171–187.
3. Bauer, W., Behrens, J. [2018], A structure-preserving split finite element discretization of the split wave equations, *Appl. Math. Comput.*, **325**, 375–400.
4. Bauer, W., Behrens, J., Cotter, C.J. [2019], A structure-preserving split finite element discretization of the rotating shallow water equations in split Hamiltonian form, preprint: https://hal.inria.fr/hal-02020379
5. Bauer, W., Gay-Balmaz, F. [2019]: Towards a variational discretization of compressible fluids: the rotating shallow water equations, *J. Comput. Dyn.*, **6(1)**, 1–37.

6. Bauer, W., Gay-Balmaz, F. [2019], Variational integrators for anelastic and pseudo-incompressible flows, *J. Geom. Mech.*, **11(4)**, 511–537.
7. Beirão Da Veiga, L., Lopez, L., Vacca, G. [2017], Mimetic finite difference methods for Hamiltonian wave equations in 2D, *Comput. Math. Appl.*, **74(5)**, 1123–1141.
8. Bochev, P., Hyman, J. [2006], Principles of mimetic discretizations of differential operators, Compatible Spatial Discretizations, *IMA Volumes in Math. and its Applications*, **142**, 89–119.
9. Cotter, C. J., Thuburn, J. [2012], A finite element exterior calculus framework for the rotating shallow-water equations, *J. Comput. Phys.*, **257**, 1506–1526.
10. Dubinkina, S., Frank, J. [2007], Statistical mechanics of Arakawa's discretizations, *J. Comput. Phys.*, **227**, 1286–1305.
11. McRae, A. T., Cotter, C. J. [2014], Energy- and enstrophy-conserving schemes for the shallow-water equations, based on mimetic finite elements, *Q. J. R. Meteorol. Soc.*, **140**, 2223–2234.
12. Natale, A, Cotter, C. J. [2017], Scale-selective dissipation in energy-conserving FE schemes for two-dimensional turbulence, *Q. J. R. Meteorol. Soc.*, **143**, 1734–1745.
13. Palha, A., Rebelo, P. P., Hiemstra, R., Kreeft, J. and Gerritsma, M. [2014], Physics-compatible discretization techniques on single and dual grids, with application to the Poisson equation of volume forms, *J. Comput. Phys.*, **257**, 1394–1422.
14. Pavlov, D., Mullen, P., Tong, Y., Kanso, E., Marsden, J.E., Desbrun, M. [2010] Structure-preserving discretization of incompressible fluids, *Physica D*, **240**, 443–458.
15. Staniforth, A., Thuburn, J. [2012], Horizontal grids for global weather and climate prediction models: A review, *Q. J. R. Meteorol. Soc.*, **138**, 1–26.

Iterative Coupling for Fully Dynamic Poroelasticity

Markus Bause, Jakub W. Both, and Florin A. Radu

Abstract We present an iterative coupling scheme for the numerical approximation of the mixed hyperbolic-parabolic system of fully dynamic poroelasticity. We prove its convergence in the Banach space setting for an abstract semi-discretization in time that allows the application of the family of diagonally implicit Runge–Kutta methods. Recasting the semi-discrete solution as the minimizer of a properly defined energy functional, the proof of convergence uses its alternating minimization. The scheme is closely related to the undrained split for the quasi-static Biot system.

1 Introduction

Information on flow in deformable porous media has become of increasing importance in various fields of natural sciences and technology. It offers an abundance of technical, geophysical, environmental and biomedical applications including modern material science polymers and metal foams, gaining significance particularly in lightweight design and aircraft industry, design of batteries or hydrogen fuel cells for green technologies, geothermal energy exploration or reservoir engineering as well as mechanism in the human body and food technology. Consequently, quantitative methods, based on numerical simulations, are desirable in analyzing experimental data and designing theories based on mathematical concepts. Recently, the quasi-static Biot system (cf., e.g., [12, 14]) has attracted researchers' interest and has been studied as a proper model for the numerical simulation of flow in deformable porous media. The design, analysis and optimization of approximation techniques that are based on an iterative coupling of the subproblems of fluid

M. Bause (✉)
Helmut Schmidt University, Hamburg, Germany
e-mail: bause@hsu-hh.de

J. W. Both · F. A. Radu
University of Bergen, Bergen, Norway
e-mail: Jakub.Both@uib.no; Florin.Radu@uib.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_10

flow and mechanical deformation were focused strongly. Iterative coupling offers the appreciable advantage over the fully coupled method that existing and highly developed discretizations and algebraic solver technologies can be reused. For the quasi-static Biot system, pioneering work is done in [10, 12]. Further research is presented in, e.g., [2, 4, 7–9, 13].

In the case of larger contrast coefficients that stand for the ratio between the intrinsical characteristic time and the characteristic domain time scale, the fully dynamic hyperbolic-parabolic system of poroelasticity has to be considered. In [11], this system (referred to as the Biot–Allard equations) is derived by asymptotic homogenization in the space and time variables. Here, to fix our ideas and carve out the key technique of proof, a simplified form of the system proposed in [11] is studied. However, its mixed hyperbolic-parabolic structure is preserved. Our modification of the fully dynamic poroelasticity model in [11] comes through a simplication of the solution's convolution with the dynamic permeability that is defined as the spatial average of pore system Stokes solutions on the unit cell (the periodic representative volume element of the porous medium). The fully dynamic system of poroelasticity to be analyzed here is given by (cf. also [14, p. 313])

$$\rho\, \partial_t^2 \boldsymbol{u} - \nabla \cdot (\mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}) - \boldsymbol{\alpha} p) = \boldsymbol{f}\,, \tag{1a}$$

$$\partial_t\, (c_0 p + \boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{u})) + \nabla \cdot \boldsymbol{q} = h\,, \tag{1b}$$

$$\boldsymbol{\kappa}^{-1}\boldsymbol{q} + \nabla p = \boldsymbol{g}\,. \tag{1c}$$

System (1) is equipped with appropriate initial and boundary conditions. In (1), the variable $\boldsymbol{u}$ is the unkown effective solid phase displacement and p is the unkown effective pressure. The quantity $\boldsymbol{\varepsilon}(\boldsymbol{u}) = (\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^\top)/2$ denotes the symmetrized gradient or strain tensor. Further, ρ is the effective mass density, $\mathbb{C}$ is Gassmann's fourth order effective elasticity tensor, $\boldsymbol{\alpha}$ is Biot's pressure-storage coupling tensor and c_0 is the specific storage coefficient. In the three field formulation (1), the vector field $\boldsymbol{q}$ is Darcy's velocity and $\boldsymbol{\kappa}$ is the permeability tensor. All tensors are assumed to be symmetric, bounded and uniformly positive definite, the constants ρ and c_0 are positive. By $\boldsymbol{A} : \boldsymbol{B}$ we denote the Frobenius inner product of $\boldsymbol{A}$ and $\boldsymbol{B}$. The functions on the right-hand side of (1) are supposed to be elements in dual spaces and, therefore, can include body forces and surface data (boundary conditions).

So far, the numerical simulation of the system (1) has been studied rarely in the literature despite its numerous applications in practice. This might be due to the mixed hyberbolic-parabolic character of the system and severe complexities involved in the construction of monolithic solver or iterative coupling schemes with guaranteed stability properties. Space-time finite element approximations of hyperbolic and parabolic problems and the quasi-static Biot system were recently proposed, analyzed and investigated numerically by the authors in [1–3]. Here, we propose an iterative coupling scheme for the system (1) and prove its convergence. This is done in Banach spaces for the semi-discretization in time of (1). An abstract setting is used for the time discretization such that the family of diagonally implicit

Runge–Kutta methods becomes applicable. The key ingredient of our proof of convergence is the observation that we can recast the semi-discrete approximation of (1) as the minimizer of an energy functional in the displacement and Darcy velocity fields. To solve the minimization problem, the general and abstract framework of alternating minimization (cf. [5, 6]) is applied. The resulting subproblems of this minimization are then reformulated as our final iterative coupling scheme. Thereby, the proof of convergence of the iterative scheme is traced back to the convergence of the alternating minimization approach. This shows that the latter provides an abstract and powerful tool of optimization for the design of iterative coupling schemes.

We use standard notation. In particular, we denote by $\langle \cdot, \cdot \rangle$ the standard inner product of $L^2(\Omega)$ and by $\|\cdot\|$ the norm of $L^2(\Omega)$.

2 Variational Formulation of a Semi-Discrete Approximation of the System of Dynamic Poroelasticity

Firstly, we discretize the continuous system of dynamic poroelasticity (1) in time by using arbitrary (diagonally implicit) Runge–Kutta methods and formulate the semi-discrete approximation as solution to a minimization problem, following the approach in [5]. For this, we consider an equidistant partition $0 = t_0 < t_1 < \ldots < t_N = T$ of the time interval of interest $[0, T]$ with time step size Δt. In the sequel, we use the following function spaces for displacement, pressure, and flux, respectively,

$$\begin{aligned}
\mathcal{V}^n &:= \left\{ \boldsymbol{v} \in H^1(\Omega)^d \,\middle|\, \text{satisfies prescribed BC at time } t_n \right\}, \\
\mathcal{Q}^n &:= L^2(\Omega), \\
\mathcal{W}^n &:= \left\{ \boldsymbol{w} \in H(\mathrm{div}; \Omega) \,\middle|\, \boldsymbol{w} \text{ satisfies prescribed BC at time } t_n \right\}.
\end{aligned}$$

Further, let $\mathcal{V}_0$, $\mathcal{Q}_0$, and $\mathcal{W}_0$ denote the corresponding natural test spaces, and $\mathcal{V}_0^\star$, $\mathcal{Q}_0^\star$, and $\mathcal{W}_0^\star$ their dual spaces.

Applying any diagonally implicit Runge–Kutta method for the temporal discretization of (1), eventually involves solving systems of the following structure.

Problem 1 In the n-th time step, find the displacement $\boldsymbol{u}^n \in \mathcal{V}^n$, pressure $p^n \in \mathcal{Q}^n$, and flux $\boldsymbol{q}^n \in \mathcal{W}^n$, satisfying for all $(\boldsymbol{v}, q, \boldsymbol{w}) \in \mathcal{V}_0 \times \mathcal{Q}_0 \times \mathcal{W}_0$ the equations

$$\frac{\rho}{\Delta t^2} \langle \boldsymbol{u}^n, \boldsymbol{v} \rangle + \theta_1 \langle \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}^n), \boldsymbol{\varepsilon}(\boldsymbol{v}) \rangle - \theta_1 \langle \boldsymbol{\alpha} p^n, \boldsymbol{\varepsilon}(\boldsymbol{v}) \rangle = \langle \boldsymbol{f}^n_{\theta,\Delta t}, \boldsymbol{v} \rangle , \tag{2a}$$

$$c_0 \langle p^n, q \rangle + \langle \boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{u}^n), q \rangle + \theta_2 \Delta t \langle \nabla \cdot \boldsymbol{q}^n, q \rangle = \langle h^n_{\theta,\Delta t}, q \rangle , \tag{2b}$$

$$\left\langle \boldsymbol{\kappa}^{-1} \boldsymbol{q}^n, \boldsymbol{w} \right\rangle - \langle p^n, \nabla \cdot \boldsymbol{w} \rangle = \langle \boldsymbol{g}^n_{\theta,\Delta t}, \boldsymbol{w} \rangle . \tag{2c}$$

In (2), the quantities $\theta_1, \theta_2 \in (0, 1]$ are discretization parameters, and the right-hand side functions $\boldsymbol{f}^n_{\theta,\Delta t} \in \mathcal{V}_0^\star$, $h^n_{\theta,\Delta t} \in \mathcal{Q}_0^\star$, $\boldsymbol{g}^n_{\theta,\Delta t} \in \mathcal{W}_0^\star$ include information on external volume and surface terms, as well as previous time steps depending on the choice of the implicit Runge–Kutta discretization.

Assuming positive compressibility, i.e., $c_0 > 0$ for the specific storage coefficient, the semi-discrete approximation satisfies equivalently the following variational problem; cf. [5] for the derivation of a similar equivalence in the framework of the quasi-static Biot system.

Problem 2 Find $(\boldsymbol{u}^n, \boldsymbol{q}^n) \in \mathcal{V}^n \times \mathcal{W}^n$, satisfying

$$(\boldsymbol{u}^n, \boldsymbol{q}^n) = \underset{(\boldsymbol{u},\boldsymbol{q}) \in \mathcal{V}^n \times \mathcal{W}^n}{\arg\min} \mathcal{E}(\boldsymbol{u}, \boldsymbol{q}) , \tag{3}$$

where the energy $\mathcal{E} : \mathcal{V}^n \times \mathcal{W}^n \to \mathbb{R}$ at time t_n is defined by $((\boldsymbol{u}, \boldsymbol{q}) \in \mathcal{V}^n \times \mathcal{W}^n)$

$$\begin{aligned}\mathcal{E}(\boldsymbol{u}, \boldsymbol{q}) := \frac{\rho}{2\Delta t^2}\|\boldsymbol{u}\|^2 + \frac{\theta_1}{2}\langle \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}), \boldsymbol{\varepsilon}(\boldsymbol{u})\rangle + \frac{\theta_1\theta_2\Delta t}{2}\left\langle \boldsymbol{\kappa}^{-1}\boldsymbol{q}, \boldsymbol{q}\right\rangle \\ + \frac{\theta_1}{2c_0}\left\|h^n_{\theta,\Delta t} - \boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{u}) - \theta_2\Delta t \nabla \cdot \boldsymbol{q}\right\|^2 - \left\langle \boldsymbol{f}^n_{\theta,\Delta t}, \boldsymbol{u}\right\rangle - \left\langle \boldsymbol{g}^n_{\theta,\Delta t}, \boldsymbol{q}\right\rangle .\end{aligned} \tag{4}$$

The semi-discrete pressure p^n may then be recovered by the post-processing step

$$p^n = c_0^{-1}\left(h^n_{\theta,\Delta t} - \boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{u}^n) - \theta_2\Delta t\, \nabla \cdot \boldsymbol{q}^n\right) . \tag{5}$$

3 Iterative Coupling for the System of Dynamic Poroelasticity

Following the philosophy of [5], we propose an iterative coupling of the semi-discrete equations (2) of dynamic poroelasticity by firstly applying the fundamental alternating minimization to the variational formulation (3); cf. Algorithm 1.

Algorithm 1: Single iteration of the alternating minimization

```
1 Input: $(\boldsymbol{u}^{n,k-1}, \boldsymbol{q}^{n,k-1}) \in \mathcal{V}^n \times \mathcal{W}^n$
2 Determine $\boldsymbol{u}^{n,k} := \arg\min_{\boldsymbol{u} \in \mathcal{V}^n} \mathcal{E}(\boldsymbol{u}, \boldsymbol{q}^{n,k-1})$
3 Determine $\boldsymbol{q}^{n,k} := \arg\min_{\boldsymbol{q} \in \mathcal{W}^n} \mathcal{E}(\boldsymbol{u}^{n,k}, \boldsymbol{q})$
```

Secondly, the resulting scheme is equivalently reformated in terms of a stabilized splitting scheme applied to the three-field formulation (2). For this, a pressure iterate $p^{n,k} = c_0^{-1}\left(h^n_{\theta,\Delta t} - \boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{u}^{n,k}) - \theta_2\Delta t\, \nabla \cdot \boldsymbol{q}^{n,k}\right) \in \mathcal{Q}^n$, $k \geq 0$, is introduced, consistent with (5), and the optimality conditions corresponding to the two steps of

Algorithm 1 are reformulated. The calculations are skipped here. We immediately present the resulting scheme, which in the end is closely related to the *undrained split* for the quasi-static Biot system [10].

Problem 3 Let $(\boldsymbol{u}^{n,0}, p^{n,0}) \in \mathcal{V}^n \times \mathcal{Q}^n$ be given and $k \geq 1$.

1. Step (Update of mechanical deformation): For given $(\boldsymbol{u}^{n,k-1}, p^{n,k-1}) \in \mathcal{V}^n \times \mathcal{Q}^n$, find $\boldsymbol{u}^{n,k} \in \mathcal{V}^n$ satisfying for all $\boldsymbol{v} \in \mathcal{V}_0$,

$$\frac{\rho}{\Delta t^2}\left\langle \boldsymbol{u}^{n,k}, \boldsymbol{v}\right\rangle + \theta_1 \left\langle \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{u}^{n,k}) + \frac{\boldsymbol{\alpha}\otimes\boldsymbol{\alpha}}{c_0}\boldsymbol{\varepsilon}(\boldsymbol{u}^{n,k} - \boldsymbol{u}^{n,k-1}), \boldsymbol{\varepsilon}(\boldsymbol{v})\right\rangle \qquad (6)$$

$$- \theta_1 \left\langle \boldsymbol{\alpha} p^{n,k-1}, \boldsymbol{\varepsilon}(\boldsymbol{v})\right\rangle = \left\langle \boldsymbol{f}^n_{\theta,\Delta t}, \boldsymbol{v}\right\rangle,$$

where $\otimes : \mathbb{R}^{d\times d} \times \mathbb{R}^{d\times d} \to \mathbb{R}^{d\times d\times d\times d}$ denotes the standard tensor product.

2. Step (Update of Darcy velocity and pressure): For given $(\boldsymbol{u}^{n,k}, p^{n,k-1}) \in \mathcal{V}^n \times \mathcal{Q}^n$ find $(p^{n,k}, \boldsymbol{q}^{n,k}) \in \mathcal{Q}^n \times \mathcal{W}^n$ satisfying for all $(q, \boldsymbol{w}) \in \mathcal{Q}_0 \times \mathcal{W}_0$,

$$c_0 \left\langle p^{n,k}, q\right\rangle + \left\langle \boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{u}^{n,k}), q\right\rangle + \theta_2 \Delta t \left\langle \nabla\cdot \boldsymbol{q}^{n,k}, q\right\rangle = \left\langle h^n_{\theta,\Delta t}, q\right\rangle, \qquad (7a)$$

$$\left\langle \boldsymbol{\kappa}^{-1}\boldsymbol{q}^{n,k}, \boldsymbol{w}\right\rangle - \left\langle p^{n,k}, \nabla\cdot\boldsymbol{w}\right\rangle = \left\langle \boldsymbol{g}^n_{\theta,\Delta t}, \boldsymbol{w}\right\rangle. \qquad (7b)$$

We note that the splitting scheme defined by (6) and (7) utilizes the identical stabilization as the *undrained split* for the quasi-static Biot equations [10].

4 Convergence of the Iterative Coupling Scheme

The identification of the undrained split approach (6) and (7) as the application of the alternating minimization, cf. Algorithm 1, to the variational problem (3) yields the basis for a simple convergence analysis. For this, we utilize the following abstract convergence result, that is rewritten here in terms of the specific formulation of Algorithm 1.

Theorem 1 (Convergence of the Alternating Minimization [6]) *Let* $|\cdot|$, $|\cdot|_{\mathrm{m}}$, *and* $|\cdot|_{\mathrm{f}}$ *denote semi-norms on* $\mathcal{V}_0 \times \mathcal{W}_0$, $\mathcal{V}_0$, *and* $\mathcal{W}_0$, *respectively. Let* $\beta_{\mathrm{m}}, \beta_{\mathrm{f}} > 0$ *satisfy the inequalities*

$$|(\boldsymbol{v}, \boldsymbol{w})|^2 \geq \beta_{\mathrm{m}} |\boldsymbol{v}|^2_{\mathrm{m}} \qquad \text{and} \qquad |(\boldsymbol{v}, \boldsymbol{w})|^2 \geq \beta_{\mathrm{f}} |\boldsymbol{w}|^2_{\mathrm{f}}$$

for all $(\boldsymbol{v}, \boldsymbol{w}) \in \mathcal{V}_0 \times \mathcal{W}_0$. *Furthermore, assume that the energy functional* $\mathcal{E}$ *of* (4) *satisfies the following conditions:*

- *The energy* $\mathcal{E}$ *is Frechét differentiable with* $D\mathcal{E}$ *denoting its derivative.*

- *The energy $\mathcal{E}$ is strongly convex wrt. $|\cdot|$ with modulus $\sigma > 0$, i.e., for all $\boldsymbol{u}, \bar{\boldsymbol{u}} \in \mathcal{V}^n$ and $\boldsymbol{q}, \bar{\boldsymbol{q}} \in \mathcal{W}^n$ it holds that*

$$\mathcal{E}(\bar{\boldsymbol{u}}, \bar{\boldsymbol{q}}) \geq \mathcal{E}(\boldsymbol{u}, \boldsymbol{q}) + \langle D\mathcal{E}(\boldsymbol{u}, \boldsymbol{q}), (\bar{\boldsymbol{u}} - \boldsymbol{u}, \bar{\boldsymbol{q}} - \boldsymbol{q})\rangle + \frac{\sigma}{2}|(\bar{\boldsymbol{u}} - \boldsymbol{u}, \bar{\boldsymbol{q}} - \boldsymbol{q})|^2 .$$

- *The partial functional derivatives $D_{\boldsymbol{u}}\mathcal{E}$ and $D_{\boldsymbol{q}}\mathcal{E}$ are uniformly Lipschitz continuous wrt. $|\cdot|_{\mathrm{m}}$ and $|\cdot|_{\mathrm{f}}$ with Lipschitz constants $L_{\mathrm{m}} > 0$ and $L_{\mathrm{f}} > 0$, respectively, i.e., for all $(\boldsymbol{u}, \boldsymbol{q}) \in \mathcal{V}^n \times \mathcal{W}^n$ and $(\boldsymbol{v}, \boldsymbol{w}) \in \mathcal{V}_0 \times \mathcal{W}_0$ it holds that*

$$\mathcal{E}(\boldsymbol{u} + \boldsymbol{v}, \boldsymbol{q}) \leq \mathcal{E}(\boldsymbol{u}, \boldsymbol{q}) + \langle D_{\boldsymbol{u}}\mathcal{E}(\boldsymbol{u}, \boldsymbol{q}), \boldsymbol{v}\rangle + \frac{L_{\mathrm{m}}}{2} \|\boldsymbol{v}\|_{\mathrm{m}}^2 ,$$

$$\mathcal{E}(\boldsymbol{u}, \boldsymbol{q} + \boldsymbol{w}) \leq \mathcal{E}(\boldsymbol{u}, \boldsymbol{q}) + \left\langle D_{\boldsymbol{q}}\mathcal{E}(\boldsymbol{u}, \boldsymbol{q}), \boldsymbol{w}\right\rangle + \frac{L_{\mathrm{f}}}{2} \|\boldsymbol{w}\|_{\mathrm{f}}^2 .$$

Let $(\boldsymbol{u}^n, \boldsymbol{q}^n) \in \mathcal{V}^n \times \mathcal{W}^n$ denote the solution to (3), and let $(\boldsymbol{u}^{n,k}, \boldsymbol{q}^{n,k})$ denote the corresponding approximation defined by Algorithm 1. Then, for all $k \geq 1$ it follows that

$$\begin{aligned} &\mathcal{E}(\boldsymbol{u}^{n,k}, \boldsymbol{q}^{n,k}) - \mathcal{E}(\boldsymbol{u}^n, \boldsymbol{q}^n) \\ &\quad\leq \left(1 - \frac{\beta_{\mathrm{m}}\sigma}{L_{\mathrm{m}}}\right)\left(1 - \frac{\beta_{\mathrm{f}}\sigma}{L_{\mathrm{f}}}\right)\left(\mathcal{E}(\boldsymbol{u}^{n,k-1}, \boldsymbol{q}^{n,k-1}) - \mathcal{E}(\boldsymbol{u}^n, \boldsymbol{q}^n)\right) . \end{aligned} \tag{8}$$

A simple application of Theorem 1 now yields the main result of the work, namely the global linear convergence of the undrained split (6), (7).

Corollary 1 (Linear Convergence of the Undrained Split) *Let $|\cdot|$ be defined by*

$$\begin{aligned} |(\boldsymbol{v}, \boldsymbol{w})|^2 := \frac{\rho}{\Delta t^2} \|\boldsymbol{v}\|^2 + \theta_1 \langle \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{v}), \boldsymbol{\varepsilon}(\boldsymbol{v})\rangle + \theta_1\theta_2\Delta t \left\langle \boldsymbol{\kappa}^{-1}\boldsymbol{w}, \boldsymbol{w}\right\rangle \\ + \frac{\theta_1}{c_0} \|\boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{v}) + \theta_2\Delta t\nabla \cdot \boldsymbol{w}\|^2 \end{aligned}$$

for all $(\boldsymbol{v}, \boldsymbol{w}) \in \mathcal{V}_0 \times \mathcal{W}_0$. Furthermore, let $(\boldsymbol{u}^n, \boldsymbol{q}^n) \in \mathcal{V}^n \times \mathcal{W}^n$ denote the solution to (3), and let $(\boldsymbol{u}^{n,k}, \boldsymbol{q}^{n,k}) \in \mathcal{V}^n \times \mathcal{W}^n$ denote the corresponding approximation defined by Algorithm 1. Then, for all $k \geq 1$ it holds that

$$\begin{aligned} \left|(\boldsymbol{u}^{n,k} - \boldsymbol{u}^n, \boldsymbol{q}^{n,k} - \boldsymbol{q}^n)\right|^2 \leq \left(\frac{\|\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha}\|_\infty}{c_0 + \|\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha}\|_\infty}\right)^2 \\ \cdot \left|(\boldsymbol{u}^{n,k-1} - \boldsymbol{u}^n, \boldsymbol{q}^{n,k-1} - \boldsymbol{q}^n)\right|^2 . \end{aligned}$$

Proof We first examine convexity and smoothness properties of $\mathcal{E}$ defined in (4) by analyzing the second functional derivative of $\mathcal{E}$. For this, let $(\boldsymbol{u}, \boldsymbol{q}) \in \mathcal{V}^n \times \mathcal{W}^n$

and $(\boldsymbol{v}, \boldsymbol{w}) \in \mathcal{V}_0 \times \mathcal{W}_0$ be arbitrary. Then, for the second functional derivative $D^2\mathcal{E}(\boldsymbol{u}, \boldsymbol{q}) : (\mathcal{V}_0^\star \times \mathcal{W}_0^\star)^2 \to \mathbb{R}$ of $\mathcal{E}$ it holds that

$$\left\langle D^2\mathcal{E}(\boldsymbol{u}, \boldsymbol{q})(\boldsymbol{v}, \boldsymbol{w}), (\boldsymbol{v}, \boldsymbol{w})\right\rangle = |(\boldsymbol{v}, \boldsymbol{w})|^2 . \tag{9}$$

Next, we define a norm $|\cdot|_{\mathrm{m}}$ on $\mathcal{V}_0$ by considering the partial second functional derivative of $\mathcal{E}$ with respect to the displacement field,

$$\left\langle D_{\boldsymbol{u}}^2\mathcal{E}(\boldsymbol{u}, \boldsymbol{q})\boldsymbol{v}, \boldsymbol{v}\right\rangle = \frac{\rho}{\Delta t^2}\|\boldsymbol{v}\|^2 + \theta_1 \left\langle \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{v}), \boldsymbol{\varepsilon}(\boldsymbol{v})\right\rangle + \frac{\theta_1}{c_0}\|\boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{v})\|^2 =: |\boldsymbol{v}|_{\mathrm{m}}^2 .$$

Similarly, we define a norm $|\cdot|_{\mathrm{f}}$ on $\mathcal{W}_0$ by means of

$$\left\langle D_{\boldsymbol{q}}^2\mathcal{E}(\boldsymbol{u}, \boldsymbol{q})\boldsymbol{w}, \boldsymbol{w}\right\rangle = \theta_1\theta_2\Delta t\left\langle \boldsymbol{\kappa}^{-1}\boldsymbol{w}, \boldsymbol{w}\right\rangle + \frac{\theta_1}{c_0}\|\theta_2\Delta t\nabla\cdot\boldsymbol{w}\|^2 =: |\boldsymbol{w}|_{\mathrm{f}}^2 .$$

It directly follows that $\mathcal{E}$ is strongly convex wrt. $|\cdot|$ with modulus $\sigma = 1$, and the partial functional derivatives $D_{\boldsymbol{u}}\mathcal{E}$ and $D_{\boldsymbol{q}}\mathcal{E}$ are uniformly Lipschitz continuous wrt. $|\cdot|_{\mathrm{m}}$ and $|\cdot|_{\mathrm{f}}$ with Lipschitz constants $L_{\mathrm{m}} = 1$ and $L_{\mathrm{f}} = 1$, respectively.

By the Hölder inequality we deduce that

$$\begin{aligned}\|\boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{v})\|^2 &= \int_\Omega |\boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{v})|^2 \,\mathrm{d}\boldsymbol{x} \leq \int_\Omega \left|\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha}\right| |\boldsymbol{\varepsilon}(\boldsymbol{v}) : \mathbb{C} : \boldsymbol{\varepsilon}(\boldsymbol{v})| \,\mathrm{d}\boldsymbol{x} \\ &\leq \left\|\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha}\right\|_\infty \left\langle \mathbb{C}\boldsymbol{\varepsilon}(\boldsymbol{v}), \boldsymbol{\varepsilon}(\boldsymbol{v})\right\rangle .\end{aligned} \tag{10}$$

Hence, it follows that

$$|\boldsymbol{v}|_{\mathrm{m}}^2 \leq \left(1 + \frac{\left\|\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha}\right\|_\infty}{c_0}\right) |(\boldsymbol{v}, \boldsymbol{w})|^2 .$$

On the other hand, applying the triangle inequality and Young's inequality, and balancing the arising constants properly yields that

$$\begin{aligned}\frac{\theta_1}{c_0}\|\theta_2\Delta t\nabla\cdot\boldsymbol{w}\|^2 \leq{}& \frac{\theta_1}{c_0}\left(1 + \frac{\left\|\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha}\right\|_\infty}{c_0}\right)\|\theta_2\Delta t\nabla\cdot\boldsymbol{w} + \boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{v})\|^2 \\ &+ \frac{\theta_1}{c_0}\left(1 + \frac{c_0}{\left\|\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha}\right\|_\infty}\right)\|\boldsymbol{\alpha} : \boldsymbol{\varepsilon}(\boldsymbol{v})\|^2 .\end{aligned}$$

Together with (10), we also conclude that

$$|\boldsymbol{w}|_{\mathrm{f}}^2 \leq \left(1 + \frac{\left\|\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha}\right\|_\infty}{c_0}\right) |(\boldsymbol{v}, \boldsymbol{w})|^2 \,.$$

Thereby, the assumptions of Theorem 1 are fulfilled and (8) is ensured with constants $\sigma = L_{\mathrm{m}} = L_{\mathrm{f}} = 1$ and $\beta_{\mathrm{m}} = \beta_{\mathrm{f}} = \left(1 + \frac{\|\boldsymbol{\alpha}:\mathbb{C}^{-1}:\boldsymbol{\alpha}\|_\infty}{c_0}\right)^{-1}$. Finally, the assertion follows directly, since $\mathcal{E}$ is quadratic and $(\boldsymbol{u}^n, \boldsymbol{q}^n)$ is a local minimum of $\mathcal{E}$ and $|\cdot|$ relates to the second functional derivative of $\mathcal{E}$ via (9). Therefore, we have that $\mathcal{E}(\boldsymbol{u}^{n,k}, \boldsymbol{q}^{n,k}) - \mathcal{E}(\boldsymbol{u}^n, \boldsymbol{q}^n) = 2\left|(\boldsymbol{u}^{n,k} - \boldsymbol{u}^n, \boldsymbol{q}^{n,k} - \boldsymbol{q}^n)\right|^2$ for all $k \geq 0$. □

Remark 1 (Convergence of $p^{n,k}$) The convergence of the sequence of pressures $\{p^{n,k}\}_k$ follows now immediately by a standard inf-sup argument.

Remark 2 (Comparison with Quasi-Static Case) The final convergence rate in Corollary 1 coincides with the one for the undrained split applied to the quasi-static Biot equations for a homogeneous and isotropic bulk; cf. [12]. In that case, the Biot tensor $\boldsymbol{\alpha}$ reduces to $\alpha\mathbf{I}$ for some constant $\alpha \in (0, 1]$, and $\mathbb{C}$ is defined by the Lamé parameters, such that $\boldsymbol{\alpha} : \mathbb{C}^{-1} : \boldsymbol{\alpha} = \frac{\alpha^2}{K_{\mathrm{dr}}}$, where K_{dr} is the drained bulk modulus.

Acknowledgments This work was supported by the German Academic Exchange Service (DAAD) under the grant ID 57458510 and by the Research Council of Norway (RCN) under the grant ID 294716.

References

1. M. Bause, U. Köcher, F. A. Radu, F. Schieweck, *Post-processed Galerkin approximation of improved order for wave equations*, Math. Comp., **89** (2020), pp. 595–627.
2. M. Bause, F. A. Radu, U. Köcher, *Space-time finite element approximation of the Biot poroelasticity system with iterative coupling*, Comput. Methods Appl. Mech. Engrg., **320** (2017), pp. 745–768.
3. M. Bause, F. A. Radu, U. Köcher, *Error analysis for discretizations of parabolic problems using continuous finite elements in time and mixed finite elements in space*, Numer. Math., **137** (2017), pp. 773–818.
4. J. W. Both, M. Borregales, J. M. Nordbotten, K. Kundan, F. A. Radu, *Robust fixed stress splitting for Biot's equations in heterogeneous media*, Appl. Math. Lett., **68**, pp. 101–108.
5. J. W. Both, K. Kundan, J. M. Nordbotten, F. A. Radu, *The gradient flow structures of thermo-poro-visco-elasticity*, arXiv:1907.03134.
6. J. W. Both, *On the rate of convergence of alternating minimization for non-smooth non-strongly convex optimization in Banach spaces*, arXiv:1911.00404.
7. N. Castelletto, J. A. White, H. A. Tchelepi, *Accuracy and convergence properties of the fixed-stress iterative solution of two-way coupled poromechanics*, Int. J. Num. Anal. Meth. Geomechanics, **39** (2015), pp. 1593–1618.
8. N. Castelletto, J. A. White, M. Ferronato, *Scalable algorithms for three-field mixed finite element coupled poromechanics*, J. Comp. Phys., **327** (2016), pp. 894–918.

9. Q. Hong, J. Kraus, M. Lymbery, F. Philo, *Conservative discretizations and parameter-robust preconditioners for Biot and multiple-network flux-based poroelasticity models*, Numer-. Linear Algebra Appl., **26** (2019), e2242, pp. 1–25.
10. J. Kim, H. A. Tchelepi, R. Juanes, *Stability and convergence of sequential methods for coupled flow and geomechanics: Drained and undrained splits*, Comput. Methods Appl. Mech. Engrg., **200** (2011), pp. 2094–2116.
11. A. Mikelić, M. F. Wheeler, *Theory of the dynamic Biot–Allard equations and their link to the quasi-static Biot system*, J. Math. Phys., **53** (2012), pp. 123702:1–15.
12. A. Mikelić, M. F. Wheeler, *Convergence of iterative coupling for coupled flow and geomechanics*, Comput. Geosci., **17** (2013), pp. 479–496.
13. A. Mikelić, B. Wang, M. F. Wheeler, *Numerical convergence study of iterative coupling for coupled flow and geomechanics*, Comput. Geosci., **18** (2014), pp. 325–341.
14. R. Showalter, *Diffusion in poro-elastic media*, J. Math. Anal. Appl., **251** (2000), pp. 310–340.

A Time-Dependent Parametrized Background Data-Weak Approach

Amina Benaceur

Abstract This paper addresses model reduction with data assimilation by elaborating on the Parametrized Background Data-Weak (PBDW) approach (Maday et al. Internat J Numer Methods Engrg 102(5):933–965, 2015) recently introduced to combine numerical models with experimental measurements. This approach is here extended to a time-dependent framework by means of a `POD-greedy` reduced basis construction.

1 Introduction

The Parameterized-Background Data-Weak (PBDW) formulation for variational data assimilation is a data-driven reduced order modeling approach that was initially devised in [6] so as to merge prediction by model with prediction by data. The PBDW approach has been developed in order to estimate the true state u^{true} of a physical system for several configurations. Supposing that the true state u^{true} depends on some unknown parameter ω in an unknown parameter set Θ that represents the unanticipated uncertainty, the goal is to account for the dependency of the true state $u^{\mathrm{true}}(\omega)$ on uncertain parameters by means of the sole knowledge of data. In this paper, whenever the context is unambiguous, the parameter ω is dropped.

The formulation combines a so-called 'best-knowledge' (`bk`) model represented by a parametrized partial differential equation (PDE) and experimentally observable measurements. The use of data in the PBDW approach is fundamental not only to

A. Benaceur (✉)
Massachusetts Institute of Technology, Cambridge, MA, USA

University Paris-Est, CERMICS (ENPC) and INRIA Paris, Paris, France

EDF Lab Les Renardières, Écuelles, France
e-mail: benaceur@mit.edu

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_11

reconstruct the quantities of interest, but also to correct the possible bias in the mathematical bk model.

The PBDW approach was devised in [6] for steady problems. It has been subject to active research in recent years and it has been used for several applications. Among others, we mention [2, 3, 5, 7, 8], and [9]. To the author's knowledge, the related research in the literature remains in the steady framework. In this paper, we propose, as initiated in [1], an extension of the PBDW approach to time-dependent state estimation. We build appropriate background spaces for the time-dependent setting using the POD-greedy algorithm [4].

This paper is organized as follows. Section 2 introduces the notation. Section 3 extends the PBDW approach to the time-dependent framework and discusses the offline stage. Section 4 assesses the method via numerical experiments.

2 Basic Notation and Best-Knowledge (bk) Models

We consider a spatial domain (open, bounded, connected subset) $\Omega \subset \mathbb{R}^d$, $d \geq 1$, with a Lipschitz boundary. We introduce a Hilbert space $\mathcal{U}$ composed of functions defined over Ω. The space $\mathcal{U}$ is endowed with an inner product $(\cdot, \cdot)$ and we denote by $\|\cdot\|$ the induced norm; $\mathcal{U}$ consists of functions $\{w : \Omega \to \mathbb{R} \mid \|w\| < \infty\}$. To fix the ideas, we assume that $H_0^1(\Omega) \subset \mathcal{U} \subset H^1(\Omega)$, and we denote the dual space of $\mathcal{U}$ by $\mathcal{U}'$. The Riesz operator $R_{\mathcal{U}} : \mathcal{U}' \to \mathcal{U}$ satisfies, for each $\ell \in \mathcal{U}'$, and for all $v \in \mathcal{U}$, the equality $(R_{\mathcal{U}}(\ell), v) = \ell(v)$. Finally, we introduce a parameter set $\mathcal{P} \subset \mathbb{R}^p$, $p \geq 1$, whose elements are generically denoted by $\mu \in \mathcal{P}$, and a discrete training subset $\mathcal{P}^{\text{tr}} \subset \mathcal{P}$.

The first source of information we shall afford ourselves in the PBDW approach is a so-called 'best-knowledge' (bk) mathematical model in the form of a parameterized PDE posed over the domain Ω. Given a parameter value μ in the parameter set $\mathcal{P}$, we denote the solution to the bk parameterized PDE as $u^{\text{bk}}(\mu) \in \mathcal{U}$. Then, the manifold associated with the solutions of the bk model is $\mathcal{M}^{\text{bk}} := \{u^{\text{bk}}(\mu) \mid \mu \in \mathcal{P}\} \subset \mathcal{U}$. In ideal situations, the true solution u^{true} is well approximated by the bk manifold, i.e., the model error $\epsilon_{\text{mod}}^{\text{bk}}(u^{\text{true}}) := \inf\limits_{z \in \mathcal{M}^{\text{bk}}} \|u^{\text{true}} - z\|$ is very small.

We introduce nested background subspaces $\mathcal{Z}_1 \subset \ldots \subset \mathcal{Z}_N \subset \ldots \subset \mathcal{U}$ that are generated to approximate the bk manifold $\mathcal{M}^{\text{bk}}$ to a certain accuracy. These subspaces can be built using various model-order reduction techniques, for instance, the Reduced Basis (RB) method. The indices of the subspaces conventionally indicate their dimensions. To measure how well the true solution is approximated by the background space $\mathcal{Z}_N$, we define the quantity $\epsilon_N^{\text{bk}}(u^{\text{true}}) := \inf_{z \in \mathcal{Z}_N} \|u^{\text{true}} - z\|$. Although N is large enough, $\epsilon_N^{\text{bk}}(u^{\text{true}})$ does not tend to zero since u^{true} rarely lies in $\mathcal{M}^{\text{bk}}$ in realistic engineering study cases.

3 Time-Dependent PBDW

Consider a finite time interval $I = [0, T]$, with $T > 0$. To discretize in time, we consider an integer $K \geq 1$, we define $0 = t^0 < \cdots < t^K = T$ as $(K + 1)$ distinct time nodes over I, and we set $\mathbb{K}^{\mathrm{tr}} = \{1, \ldots, K\}$, $\overline{\mathbb{K}}^{\mathrm{tr}} = \{0\} \cup \mathbb{K}^{\mathrm{tr}}$ and $I^{\mathrm{tr}} = \{t^k\}_{k \in \overline{\mathbb{K}}^{\mathrm{tr}}}$. We aim at deriving a state estimate for a time-dependent solution in the framework illustrated in Fig. 1.

3.1 Limited-Observations Statement

Assuming that $u^{\mathrm{true}} \in L^1(I; \mathcal{U})$, we introduce the time-integration intervals $\mathcal{I}^k = [t^k - \delta t^k, t^k + \delta t^k]$, for all $k \in \mathbb{K}^{\mathrm{tr}}$, where $\delta t^k > 0$ is a parameter related to the precision of the sensor (ideally, $\delta t^k < \min(t^{k+1} - t^k, t^k - t^{k-1})$ with obvious adaptation if k=K). Then, for any function $v \in L^1(I; \mathcal{U})$, we define the time-averaged snapshots

$$v^k(x) := \frac{1}{|\mathcal{I}^k|} \int_{\mathcal{I}^k} v(t, x)\, dt \in \mathcal{U}, \quad \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{1}$$

We consider observation functionals that render the behavior of given sensors. These functionals act on time-averaged snapshots of the true solution, i.e., we consider

$$\ell_m^{k,\mathrm{obs}}(u^{\mathrm{true}}) := \ell_m^{\mathrm{obs}}(u^{k,\mathrm{true}}), \quad \forall m \in \{1, \ldots, M\},\ \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{2}$$

We then introduce the time-independent observable space $\mathcal{U}_M = \mathrm{Span}\{q_1, \ldots, q_M\} \subset \mathcal{U}$. The observation functionals in $\mathcal{U}'$ are then defined as

$$\ell_m^{k,\mathrm{obs}}(u^{\mathrm{true}}) = (u^{k,\mathrm{true}}, q_m), \quad \forall m \in \{1, \ldots, M\}, \quad \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{3}$$

For fixed sensor locations, the computational effort to compute the Riesz representations of the observation functionals is time-independent and is incurred only once, so

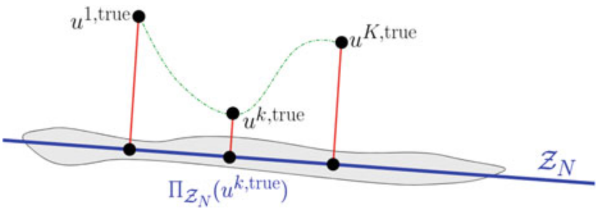


Fig. 1 Characterization of the bk model in a time-dependent context

that the experimental observations satisfy $\ell_m^{k,\mathrm{obs}}(u^{\mathrm{true}}) = \frac{1}{|\mathcal{I}_k|}\int_{\mathcal{I}^k} \ell_m^{\mathrm{obs}}(u^{\mathrm{true}}(t,\cdot))dt$, for all $m \in \{1,\ldots,M\}$ and $k \in \mathbb{K}^{\mathrm{tr}}$.

We are now ready to write the limited-observations PBDW statement: for each $k \in \mathbb{K}^{\mathrm{tr}}$, find $(u_{N,M}^{k,*}, z_{N,M}^{k,*}, \eta_{N,M}^{k,*}) \in \mathcal{U} \times \mathcal{Z}_N \times \mathcal{U}$ such that

$$(u_{N,M}^{k,*}, z_{N,M}^{k,*}, \eta_{N,M}^{k,*}) = \operatorname*{arginf}_{\substack{u_{N,M}\in\mathcal{U}\\ z_{N,M}\in\mathcal{Z}_N\\ \eta_{N,M}\in\mathcal{U}}} \|\eta_{N,M}\|, \tag{4}$$

subject to

$$(u_{N,M}, v) = (\eta_{N,M}, v) + (z_{N,M}, v), \qquad \forall v \in \mathcal{U}, \tag{5a}$$

$$(u_{N,M}, \phi) = (u^{k,\mathrm{true}}, \phi), \qquad \forall \phi \in \mathcal{U}_M. \tag{5b}$$

The limited-observations saddle-point problem associated with (4) reads: for each $k \in \mathbb{K}^{\mathrm{tr}}$, find $(z_{N,M}^{k,*}, \eta_{N,M}^{k,*}) \in \mathcal{Z}_N \times \mathcal{U}_M$ such that

$$(\eta_{N,M}^{k,*}, q) + (z_{N,M}^{k,*}, q) = (u^{k,\mathrm{true}}, q), \quad \forall q \in \mathcal{U}_M, \tag{6a}$$

$$(\eta_{N,M}^{k,*}, p) = 0, \qquad \forall p \in \mathcal{Z}_N, \tag{6b}$$

and the limited-observations state estimate is

$$u_{N,M}^{k,*} = z_{N,M}^{k,*} + \eta_{N,M}^{k,*}, \quad \forall k \in \mathbb{K}^{\mathrm{tr}}. \tag{7}$$

We use the following terminology. The PBDW statement (4) and (5) estimates the true state $u^{k,\mathrm{true}}$. Thus, the solution $u_{N,M}^{k,*}$ is called the '**state estimate**'. The first contribution $z_{N,M}^{k,*}$ in (7) lies in the background space $\mathcal{Z}_N$. Hence, $z_{N,M}^{k,*}$ is called the '**deduced background estimate**'. The second contribution $\eta_{N,M}^{k,*}$ in (7) is brought by the inclusion of the observations in the PBDW statement. The observations supplement the bk model. Thus, $\eta_N^{k,*}$ is called the '**update estimate**'. We highlight that the saddle-point problem (6) is well posed if and only if the stability constant $\beta_{N,M}$ satisfies

$$\beta_{N,M} := \inf_{w\in\mathcal{Z}_N} \sup_{v\in\mathcal{U}_M} \frac{(w,v)}{\|w\|\;\|v\|} \in (0,1]. \tag{8}$$

The deduced background estimate $z_N^{k,*}$ can only represent anticipated uncertainty. Since the bk model is often deficient, one cannot realistically assume that the state estimate $u_N^{k,*}$ of $u^{k,\mathrm{true}}$ lies completely in the bk manifold. Therefore, the update estimate $\eta_N^{k,*}$ is meant to cure the deficiency of the bk model by capturing unanticipated uncertainty. The key idea of the PBDW statement (4) and (5) is to search for the smallest correction to the `bk` manifold.

The saddle-point problem (6) is purely geometric and does not include any explicit reference to the `bk` model since the unique link to the `bk` model is through the background space $\mathcal{Z}_N$. This non-intrusiveness of (6) simplifies its implementation and makes the PBDW approach applicable to a wide class of engineering problems.

Remark 1 (Pointwise Measurements) For simplicity of implementation, assuming that $u^{\text{true}} \in C^0(I; \mathcal{U})$, one may consider pointwise measurements in time, i.e., $\left(u^{k,\text{true}}, q_m\right) = \ell_m^{\text{obs}}(u^{\text{true}}(t^k, \cdot))$, for all $m \in \{1, \ldots, M\}$ and $k \in \mathbb{K}^{\text{tr}}$. This assumption is typically reasonable for a sensor of small precision δt^k.

In algebraic form, the limited-observations PBDW statement reads: for each $k \in \mathbb{K}^{\text{tr}}$, find $(\mathbf{z}^{k,*}, \boldsymbol{\eta}^{k,*}) \in \mathbb{R}^N \times \mathbb{R}^M$ such that

$$\begin{pmatrix} \mathbf{A} & \mathbf{B} \\ \mathbf{B}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\eta}^{k,*} \\ \mathbf{z}^{k,*} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\ell}^{k,\text{obs}} \\ \mathbf{0} \end{pmatrix}, \tag{9}$$

with the matrices $\mathbf{A} = ((q_{m'}, q_m))_{m,m'} \in \mathbb{R}^{M \times M}$ and $\mathbf{B} = ((\zeta_n, q_m))_{m,n} \in \mathbb{R}^{M \times N}$, and the vector of observations $\boldsymbol{\ell}^{k,\text{obs}} = \left(\ell_m^{\text{obs}}(u^{k,\text{true}})\right)_m \in \mathbb{R}^M$. We solve (9) through an offline/online decomposed computational procedure whenever several realizations $u^{\text{true}}(\omega)$ of the true state are to be considered.

Remark 2 (PBDW Matrices) Notice that the PBDW matrices $\mathbf{A}$ and $\mathbf{B}$ are time-independent; only the right-hand side in (9) depends on k.

3.2 Offline Stage

The main goal is to address the construction of the background space $\mathcal{Z}_N$. Suppose that we have computed a set of High Fidelity (HF) trajectories $\mathcal{S} = (\mathcal{S}_k)_{k \in \mathbb{K}^{\text{tr}}} = \left((u^k(\mu))_{\mu \in \mathcal{P}^{\text{tr}}}\right)_{k \in \mathbb{K}^{\text{tr}}}$, where $u^k(\mu) := u(\mu)(t^k, \cdot)$, for all $k \in \mathbb{K}^{\text{tr}}$. If we were to consider the PBDW statement (4) and (5) for each $k \in \mathbb{K}^{\text{tr}}$ as an independent steady PBDW statement, we would be using the time-dependent background spaces $\mathcal{Z}^k_{N^k} = \texttt{POD}(\mathcal{S}_k, \epsilon_{\text{POD}})$, for all $k \in \mathbb{K}^{\text{tr}}$, where the procedure `POD` refers to the Proper Orthogonal Decomposition of the set $\mathcal{S}_k$ with a truncation threshold ϵ_{POD}. Yet, this strategy is not convenient since the sizes N^k of the background spaces $\mathcal{Z}^k_{N^k}$ would depend on k. Since the observable space $\mathcal{U}_M$ is fixed, the same non-homogeneity between time nodes would also arise in the stability constant $\beta_{N^k,M}$. Thus, we propose to apply a `POD-greedy` algorithm [4] to build a time-independent background space $\mathcal{Z}_N$ that will be used for all $k \in \mathbb{K}^{\text{tr}}$. The advantage is that the PBDW matrices $\mathbf{A}$ and $\mathbf{B}$ and the stability constant $\beta_{N,M}$ remain unchanged regardless of the discrete time node. The offline stage using the `POD-greedy` algorithm is summarized in Algorithm 1.

Algorithm 1 Offline stage via `POD-greedy`

```
INPUT : $\mathcal{S}$ and $\epsilon_{\mathrm{POD}}$.
        $Q^{\mathrm{init}}$: a set of Riesz representations for the observations.
Compute $\mathcal{Z}_N :=$ POD-greedy($\mathcal{S}, \epsilon_{\mathrm{POD}}$).
Set $\mathcal{U}_M := \mathrm{span}\{Q^{\mathrm{init}}\}$.
Compute the matrices $\mathbf{A}$ and $\mathbf{B}$ using $\mathcal{Z}_N$ and $\mathcal{U}_M$.
OUTPUT : $\mathcal{Z}_N$, $\mathcal{U}_M$, $\mathbf{A}$ and $\mathbf{B}$.
```

4 Numerical Results

In this section, we illustrate the above developments on a test case related to the heat equation. We consider a two-dimensional setting based on the plate illustrated in the left panel of Fig. 2 with $\Omega = (-2, 2)^2 \subset \mathbb{R}^2$. We use a finite element subspace $\mathcal{U}^{\mathcal{N}} \subset \mathcal{U} = H^1(\Omega)$ consisting of continuous, piecewise affine functions in order to generate HF trajectories. The FEM subspace $\mathcal{U}^{\mathcal{N}}$ is based on a mesh that contains $\mathcal{N} = 6561$ nodes. The experimental data is generated synthetically and the observation subsets $\{\mathcal{R}_m\}_{1\leq m\leq M}$ are uniformly selected over the plate as illustrated in the right panel of Fig. 2. Regarding implementation, the HF computations use the software `FreeFem++`, whereas the reduced-order modeling and the PBDW-related algorithms have been developed in `Python`. We address the following parabolic

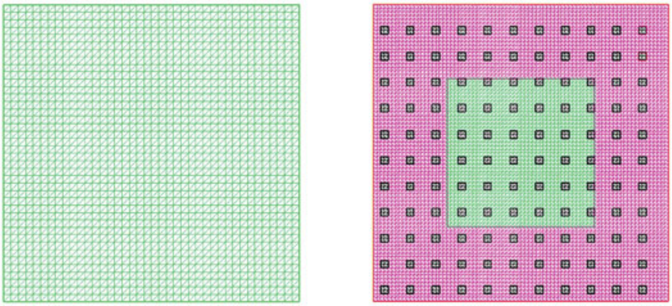


Fig. 2 Computational domain and mesh with $\mathcal{N} = 6561$. The little black squares are observation subsets $\{\mathcal{R}_m\}_{m=1}^{121}$. Left: Mono-material plate corresponding to the mathematical model. Right: Bi-material plate corresponding to the physical reality

PDE with nonlinear Stefan–Boltzmann boundary conditions: For many values of the parameter $\mu \in \mathcal{P}$, find $u(\mu) : I \times \Omega \to \mathbb{R}$ such that

$$\begin{cases} \dfrac{\partial u(\mu)}{\partial t} - \nabla \cdot (D(\mu)\nabla u(\mu)) = 0, & \text{in } I \times \Omega, \\ u(\mu)(t = 0, \cdot) = u_0, & \text{in } \Omega, \\ -D(\mu)\dfrac{\partial u}{\partial n} = \sigma\varepsilon(u^4 - u_r^4), & \text{on } I \times \partial\Omega, \end{cases} \tag{10}$$

where $u_0 = 293.15\,\text{K}$ (20 °C). The Stefan–Boltzmann boundary condition on $\partial\Omega$ is defined using an enclosure temperature $u_r = 303.15\,\text{K}$ (30 °C), the Stefan–Boltzmann constant $\sigma = 5.67 \times 10^{-8}\,\text{W.m}^{-2}.\text{K}^{-4}$, and an emissivity $\varepsilon = 3.10^{-3}$. Regarding time discretization, we consider the time interval $I = [0, 10]$s, the set of discrete times nodes $\mathbb{K}^{\text{tr}} = \{1, \dots, 200\}$, and a constant time step $\delta t^k = 0.1$s for all $k \in \mathbb{K}^{\text{tr}}$. We also define the parameter interval $\mathcal{P} = [0.1, 2]$ and the set $\mathcal{P}^{\text{tr}} = \{0.1i, 1 \leq i \leq 20\}$.

The background spaces $\mathcal{Z}_N$ will be generated by solving the nonlinear PDE (10) with a uniform diffusivity function $D(\mu)$ such that for all $x \in \Omega$, $D(\mu)(x) = D_{\text{uni}}(\mu)(x) := \mu \mathbf{1}_\Omega(x)$. The HF bk solution and the true solution are respectively displayed in the left and right panels of Fig. 3. The temperature profile for the true solution over the bi-material plate at the end of the simulation, i.e., at $t^K = 10$s, clearly shows a different behavior at the boundaries of the inner material.

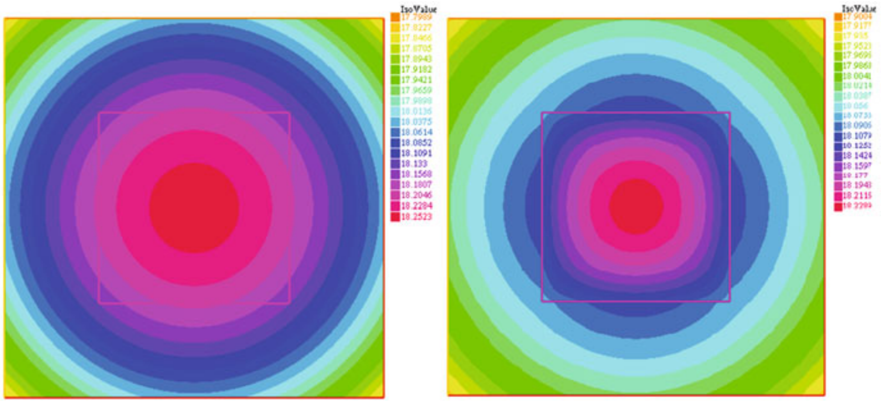


Fig. 3 Left: HF solution for the bk model, i.e $\mu = 1$ (values from 17.80 to 18.25 °C). Right: Synthetic true solution using a bi-material plate with $\mu = 2$ (values from 17.90 to 18.23 °C)

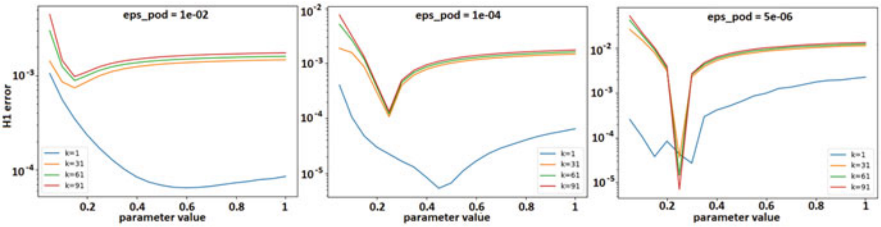


Fig. 4 Relative H^1-error $e^k(\mu)$ for some time nodes $k \in \mathbb{K}^{\mathrm{tr}}$ and $M = 121$. Left: $\epsilon_{\mathrm{POD}} = 10^{-2}$ ($N = 3$). Middle: $\epsilon_{\mathrm{POD}} = 10^{-4}$ ($N = 7$). Right: $\epsilon_{\mathrm{POD}} = 10^{-6}$ ($N = 11$)

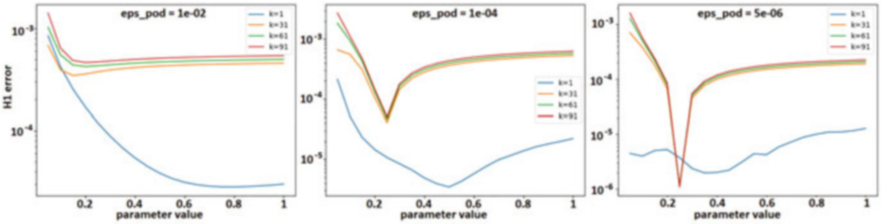


Fig. 5 Relative H^1-error $e^k(\mu)$ for some time nodes $k \in \mathbb{K}^{\mathrm{tr}}$ and $M = 676$. Left: $\epsilon_{\mathrm{POD}} = 10^{-2}$ ($N = 3$). Middle: $\epsilon_{\mathrm{POD}} = 10^{-4}$ ($N = 7$). Right: $\epsilon_{\mathrm{POD}} = 5.10^{-6}$ ($N = 11$)

Using the weighted H^1-norm, we define the state estimation relative H^1-error $e^k(\mu)$ as

$$e^k(\mu) := \frac{\|u^{k,\mathrm{true}}(\mu) - u^{k,*}_{N,M}(\mu)\|_{H^1(\Omega)}}{\|u^{k,\mathrm{true}}(\mu)\|_{H^1(\Omega)}}, \quad \forall \mu \in \mathcal{P}. \tag{11}$$

Figure 4 shows the relative H^1-error $e^k(\mu)$ defined in (11) using $M = 121$ observations to build the observable space $\mathcal{U}_M$. For $\epsilon_{\mathrm{POD}} = 10^{-4}$, $\mathcal{Z}_N$ is spanned by $N = 7$ vectors. Notice that the error vanishes for $\mu = 0.25$ since this configuration is equivalent to a perfect bk model, meaning that the mathematical model coincides with the physical reality. We notice that the relative H^1-error $e^k(\mu)$ increases on the right panel of Fig. 4 because the stability constant decreases. Figure 5 visualizes the relative H^1-error $e^k(\mu)$ for a higher number of observations $M = 676$. We observe that augmenting the dimension of the observable space $\mathcal{U}_M$ cures the stability issue. Also, the errors are lower owing to the higher number of observations. Finally, Fig. 6 shows the stability constant $\beta_{N,M}$ as a function of the number of observations M. The nonlinear character of the problem does not influence the overall features of the PBDW statement since previous linear tests in the literature have shown a similar behavior. This observation corroborates the independence of the saddle-point problem (6) with regard to the bk model.

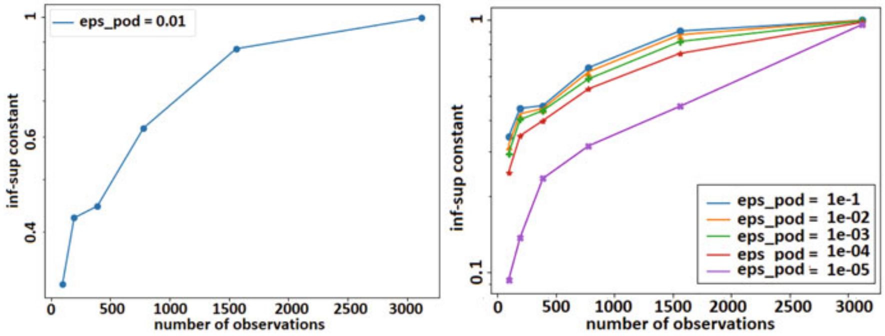


Fig. 6 Stability constant $\beta_{N,M}$. On the right panel, the values of N are respectively $2,\ 3,\ 5,\ 7,\ 11$ for the values of ϵ_{POD} in decreasing order

Acknowledgments This work is supported by EDF and ANRT. The author is thankful to A.T. Patera and A. Ern.

References

1. A. Benaceur. *Model reduction for nonlinear problems in thermics and mechanics*. Thesis, Université Paris-Est Marne la Vallée, 2018.
2. F. Galarce, J.-F. Gerbeau, D. Lombardi, and O. Mula. State estimation with nonlinear reduced models. Application to the reconstruction of blood flows with Doppler ultrasound images. *arXiv e-prints*, page arXiv:1904.13367, Apr 2019.
3. H. Gong, Y. Maday, O. Mula, and T. Taddei. PBDW method for state estimation: error analysis for noisy data and nonlinear formulation. *arXiv e-prints*, page arXiv:1906.00810, Jun 2019.
4. B. Haasdonk. Convergence rates of the POD-greedy method. *ESAIM Math. Model. Numer. Anal.*, 47(3):859–873, 2013.
5. J. K. Hammond, R. Chakir, F. Bourquin, and Y. Maday. PBDW: A non-intrusive Reduced Basis Data Assimilation method and its application to an urban dispersion modeling framework. *Appl. Math. Model.*, 76:1–25, 2019.
6. Y. Maday, A. T. Patera, J. D. Penn, and M. Yano. A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics. *Internat. J. Numer. Methods Engrg.*, 102(5):933–965, 2015.
7. Y. Maday and T. Taddei. Adaptive PBDW Approach to State Estimation: Noisy Observations; User-Defined Update Spaces. *SIAM J. Sci. Comput.*, 41(4):B669–B693, 2019.
8. T. Taddei and A. T. Patera. A localization strategy for data assimilation; application to state estimation and parameter estimation. *SIAM J. Sci. Comput.*, 40(2):B611–B636, 2018.
9. T. Taddei, J. D. Penn, and A. T. Patera. Validation by Monte Carlo sampling of experimental observation functionals. *Internat. J. Numer. Methods Engrg.*, 112(13):2135–2150, 2017.

Comparison of the Influence of Coniferous and Deciduous Trees on Dust Concentration Emitted from Low-Lying Highway by CFD

Luděk Beneš

Abstract Different types of vegetation barriers are frequently used for reduction of dust and noise levels. The effectivity of the measures depending on the type of used vegetation (decideous, coniferous) is studied in this article. The mathematical model is based on Reynolds—averaged Navier–Stokes (RANS) equations for turbulent fluid flow in Boussinesq approximation completed by the standard k-ϵ model. Pollutants, considered as passive scalar, were modelled by additional transport equation. An advanced vegetation model was used. The numerical method is based on finite volume formulation. Two fractions of pollutants, PM10 and PM75, emitted from a four–lane highway were numerically simulated. Forty-nine cases of coniferous and deciduous-type forest differing in density, width and height were studied. The main processes that play a role in modelled cases are described. The differences between the effects of coniferous and deciduous trees on pollutants deposition were studied.

1 Introduction

Increasing level of dustiness and noise pollution causes significant health problems in the populated areas. The inhabitants are negatively influenced by the increasing level of air pollution caused by the local heating, vehicular transport and industry. Vegetation plays an important role in the minimizing of these problems. Trees and forests can block or deflect the wind, improve thermal comfort and act as an filter for particulate matter. Deciduous and coniferous trees have different characteristics and affect the flow and sedimentation of particles in various ways. In nature, it is difficult to find the deciduous and coniferous forest of the same size under the same geometrical and meteorological conditions in order to compare their influence

L. Beneš (✉)
CTU in Prague, Faculty of Mechanical Engineering, Prague, Czech Republic
e-mail: ludek.benes@fs.cvut.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_12

precisely. The differences between both types of forests are studied and quantified in this contribution on a simple but important case of the road notch.

The widespread model for description of the Atmospheric Boundary Layer (ABL) flows are the RANS equations [1–4] but also LES simulations are used [5].

The effect of the vegetation on the pollutant dispersion and its filtration properties has been investigated in many studies. An overview regarding the aforementioned topics can be found in the reviews [6] and [7] on the vegetation in urban area, or a modeling study [8].

The model presented here is based on the work in [9], where the influence of the atmospheric conditions on the barrier efficiency was investigated.

The main aim of the article is to compare efficiency of the coniferous and deciduous vegetation for reduction of dustiness. Two fractions of pollutants, PM10 and PM75, emitted from a four–lane highway were numerically simulated. Forty-nine cases of conifer and deciduous-type forest differing in density, width and height were studied. The differences between the effects of coniferous and deciduous trees on pollutants deposition were studied.

2 Physical and Mathematical Model

2.1 Fluid Flow

The flow in ABL is described by the Reynolds-averaged Navier–Stokes (RANS) equations for viscous, incompressible, turbulent and stratified flow. This set of equations is simplified by the Boussinesq hypothesis.

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} + \nabla(p/\rho_*) = \nu_E \nabla^2 \boldsymbol{u} + \boldsymbol{g} + \boldsymbol{S_u}, \tag{2}$$

$$\frac{\partial \theta}{\partial t} + \nabla \cdot (\theta \boldsymbol{u}) = \frac{\nu_T}{\mathrm{Pr}} (\nabla \cdot (\nabla \theta)) \,. \tag{3}$$

Here vector $\boldsymbol{u}$ represents velocity, p, θ are fluctuations of pressure and potential temperature, ρ_* stands for the reference air density (usually near-ground concentration), $\nu_E = \nu + \nu_T$ is the effective kinematic viscosity which is composed of the laminar (molecular) and turbulent kinematic viscosity. The gravitational term is expressed by $\boldsymbol{g} = (0, 0, g\frac{\theta}{\theta_0})$ where g is the gravitational constant. Term $\boldsymbol{S_u}$ represents the momentum sink due to vegetation and $\mathrm{Pr} = 0.75$ is the turbulent Prandtl number.

2.2 Turbulence

The turbulence model is based on standard $k - \epsilon$ model modified by terms representing production and sink of the turbulence due to the vegetation.

$$\frac{\partial \rho k}{\partial t} + \nabla \cdot (\rho k \boldsymbol{u}) = \nabla \cdot \left(\left(\mu + \frac{\mu_T}{\sigma_k} \right) \nabla k \right) + P_k - \rho \epsilon + \rho S_k, \tag{4}$$

$$\frac{\partial \rho \epsilon}{\partial t} + \nabla \cdot (\rho \epsilon \boldsymbol{u}) = \nabla \cdot \left(\left(\mu + \frac{\mu_T}{\sigma_\epsilon} \right) \nabla \epsilon \right) + C_{\epsilon 1} \frac{\epsilon}{k} P_k - C_{\epsilon 2} \rho \frac{\epsilon^2}{k} + \rho S_\epsilon. \tag{5}$$

The production of turbulent kinetic energy caused by main stream interactions is denoted as P_k and μ stands for the laminar (molecular) dynamic viscosity. The model is completed with a constitutional relation for turbulent dynamic viscosity $\mu_T = C_\mu \rho \frac{k^2}{\epsilon}$. Source terms S_k and S_ϵ of k and ϵ respectively consist of two parts $S_k = S_k^r + S_k^v$ resp. $S_k = S_\epsilon^r + S_\epsilon^v$: a part expressing road traffic influence and a part expressing vegetation influence.

The terms S_k^r, S_ϵ^r modelling the road traffic sources are adopted from [11]. The sinks and sources due to vegetation influence will be described later.

Standard setting of the $k-\epsilon$ model with following constants was used: $\sigma_k = 1.0$, $\sigma_\epsilon = 1.167$, $C_{\epsilon_1} = 1.44$, $C_{\epsilon_2} = 1.92$ and $C_\mu = 0.09$. Wall functions from [16] are used.

2.3 Particles Transport

The dust in the air is assumed to be a passive scalar and its behaviour is modelled using the transport equation. The equation for each non dimensional mass fraction c is as follows:

$$\frac{\partial \rho c}{\partial t} + \nabla \cdot (\rho c \boldsymbol{u}) - \frac{\partial (\rho c u_s)}{\partial y} = \nabla \cdot \left(\frac{\nu_T}{\mathrm{Sc}} \nabla \rho c \right) + \rho F_c + S_c. \tag{6}$$

Here u_s is the settling velocity, F_c denotes the pollutant source term and S_c is the vegetation deposition term, $\mathrm{Sc} = 0.72$. The settling velocity u_s of a spherical particle with the diameter d and density ρ_p is given by the Stokes' equation see [12] with correction factor C_c:

$$u_s = (d^2 \rho_p g C_c)/(18\mu), \quad C_c = 1 + \frac{\lambda}{d} \left(2.34 + 1.05 \exp(-0.39 d/\lambda) \right).$$

2.4 Vegetation Model

Vegetation deforms the flow field, increases the level of turbulence and plays a significant role in the deposition processes. Therefore, an appropriate model is crucially important. The model from [13] for coniferous trees and from [14] for broadleaf trees are adopted in this work. Both models were used and validated in our previous studies [9, 10].

The vegetation barrier is modelled as a porous block described by a so called *Leaf Area Density* (LAD) profile which represents foliated surface area per unit volume. In our computation a horizontally homogeneous forest is assumed. The original LAD is multiplied by a coefficient representing the vegetation density. In our computations, the model of pine trees adopted from [15] and deciduous trees presented in [10] was used.

Three effects of vegetation are considered: the first one is the drag induced by the vegetation. It is modelled as momentum sink inside the vegetation in Eq. (2):

$$\boldsymbol{S}_u = -C_d \mathrm{LAD} |\boldsymbol{u}| \boldsymbol{u},$$

where $C_d = 0.3$ is the drag coefficient [1].
The second effect is the influence on the turbulent quantities. Following [1] the source terms in Eqs. (4) and (5) are written

$$S_k^v = C_d \mathrm{LAD}(\beta_p |\boldsymbol{u}|^3 - \beta_d |\boldsymbol{u}| k), \qquad S_\epsilon^v = C_{\epsilon 3} \frac{\epsilon}{k} S_k^v.$$

The constants are $\beta_p = 1.0\, m^{-1}$, $\beta_d = 5.1\, m^{-1}$ and dimensionless $C_{\epsilon_3} = 0.9$.
The particle deposition in the vegetation is the third process. According to [13], this effect is given by the term $S_c = -\mathrm{LAD} u_d \rho c$ in Eq. (6). The term is proportional to the deposition velocity u_d which reflects four main processes by which particles depose on the leaves (needles): Brownian diffusion, interception, impaction and gravitational settling. Its value generally depends on wind speed, particle size and vegetation properties.

2.5 Numerical Method

A finite volume method based on artificial compressibility method and AUSM$^+$up scheme with linear reconstruction is used. To prevent spurious oscilations the Venkatakrishnan limiter is utilized. The viscous terms are solved on a dual (diamond type) mesh. This discretization results in a set of ordinary differential equations (in time) solved using implicit BDF2 method.

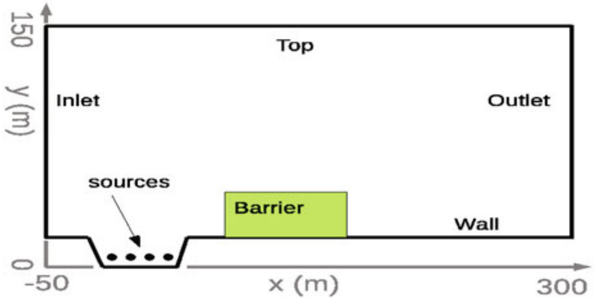


Fig. 1 Sketch of the domain (not to scale). Highway notch with dust sources and forest block on the plateau

Each of these nonlinear systems is solved by the JFNK method. Inner linear systems are solved using matrix-free GMRES solver. The linear systems are preconditioned by the ILU(3) preconditioner. Necessary evaluations of the Jacobians are done via finite differences.

3 The Numerical Experiment Setting

Figure 1 shows a sketch of a computational domain. Assuming the wind direction perpendicular to the highway notch, a simplified 2D case is solved. The domain dimensions are 350×150 m, the slopes of the notch are 4 m high. Four sources of pollutant are placed in the middle of each lane at height 0.8 m. Each source of the pollutant has the intensity 1 $\mu g/s$. A vegetation block is placed downstream from the road above the notch and starts at $x = 55$ m. Particles with diameter 10 μm and 75 μm and density 1000 kg/m^3 are modelled.

The ABL is considered as a weakly stable stratified layer ($\partial T/\partial y = 0$) K/m. Background temperature is set to $T_0 = 20\,°\mathrm{C}$ and the density is $\rho_* = 1.2\,\mathrm{kg/m^3}$. The logarithmic wind profile is prescribed with u_{ref} = 5 m/s at height y_{ref} = 10 m on the inlet.

All combinations of the following vegetative block geometrical parameters were tested: density (D): 0.25, 0.5, 1.0, 1.5, width (W): 50, 80, 110, and 140 m, height (H): 3, 7, and 11 m.

4 Results: Efficiency of the Barriers

Differences in responses of deciduous and coniferous vegetation can be studied from different angles. The flow structure inside and outside the forest changes depending on the different crown shapes, different distribution and properties of leaves and needles. A thorough description of these differences is beyond the scope of this article. We will focus on the efficiency of the barrier as one chosen aspect.

The basic question is what do we mean by efficiency. We investigate this question from two different points of view in this work. The first monitored parameter is the concentration in the given point $x = 250$ and 3 m above ground. The second one is the filtering capacity of the forest (total amount of the particles trapped in the forest).

The concentration of PM10 particles is shown in Fig. 2. The dependency on the conifer forest width is significant for lower vegetation (3, 7 m), the concentration monotonically decreases. Reduction of the concentration is up to 40% (compared to the value without vegetation). For high forest, the dependency on forest width is insignificant, the reduction is close to the 50%. Further enlarging the width of the forest gives no significant impact if the width exceeds 40 m. The situation for deciduous vegetation is only slightly different, the dependency on forest width is significant only for lower vegetation 3m height and reduction of concentration is close to the 30%. The dependency on the forest density is higher for conifer vegetation.

Similar behaviour can be observed in PM75 case see Fig. 3, only the dependency on the forest density is stronger.

An interesting question is how many particles will be captured on the needles, leaves, branches and twigs. These values are summarized in Figs. 4 and 5. For the lighter particles, the efficiency is significantly higher for the conifer trees compared to the deciduous ones. Leaves are surrounded by liquid, so the effective cross–section is relatively small. The situation in the case of PM75 particles is completely different. The effectiveness in particle filtering is similar both for conifer and deciduous trees, because the larger particles are not so affected by the flow. The main principle for the PM75 particles is gravitational settling which is not included in these graphs. In both cases we can see that longer forests have no significant effect.

5 Conclusions

The deciduous and pine type vegetations under the same geometrical and atmospheric conditions were numericaly modelled and studied. The effects of height, width and density of vegetation were examined for both types of vegetation.

The height of vegetation is the most important parameter of the forest, the density and width play minor roles. Extension of the forest has minimal effect on its filtering capacity. For the heavier particles, where the effect of the gravitational settling plays the dominant role, both conifer and deciduous vegetation are very efficient and comparable. In the case of lighter particles, the main effect is spreading of particles caused by the deflection of the flow and increasing of the turbulence. For this type of particles, the conifers may be more than three times as effective as deciduous vegetation.

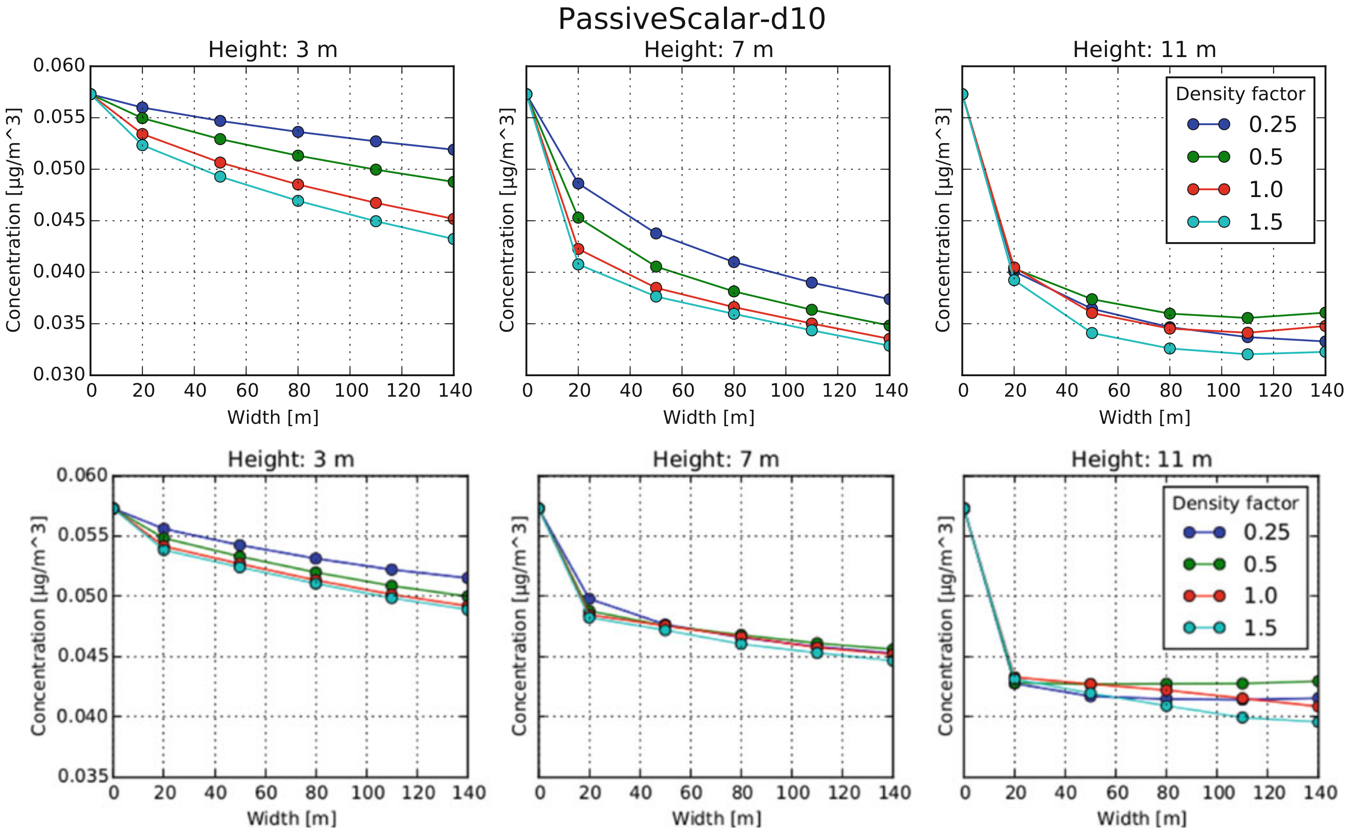


Fig. 2 Concentration of PM10 at $x = 250$, 3 m above the ground for different forest width and density. Conifer vegetation up, deciduous bottom

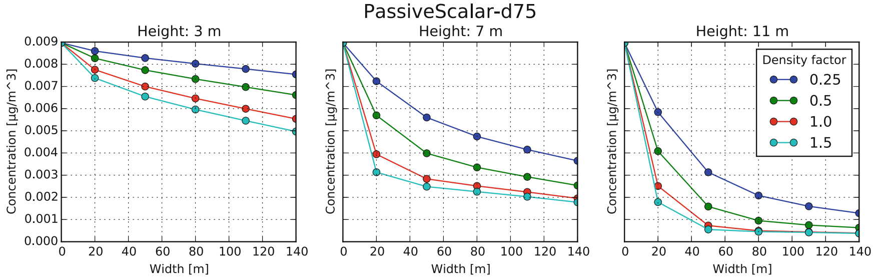


Fig. 3 Concentration of PM75 at $x = 250$, 3 m above the ground for conifer forest with different width and density

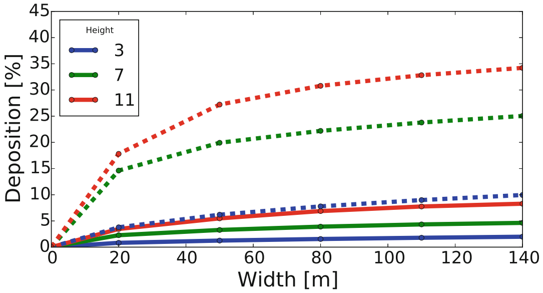


Fig. 4 Deposition of PM10 particles within barrier ($D = 1.0$) as % of the source. Dashes line for conifer

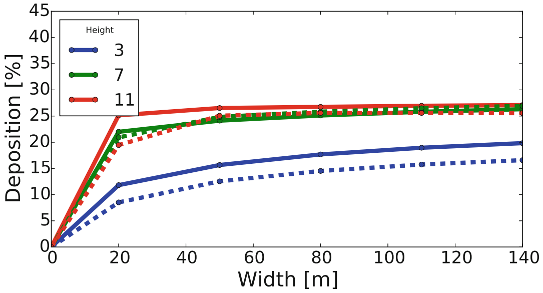


Fig. 5 Deposition of PM75 particles within barrier ($D = 1.0$) as % of the source. Dashes line for conifer

Acknowledgments This work was supported by project No. CZ.02.1.01/0.0/0.0/16_019/0000 778 (from Ministry of Education CR) and grant No. *SGS19/154/OHK2/3T/12* of Grant Agency of the CTU in Prague.

References

1. G. Katul, et al., One- and two-equation models for canopy turbulence, Bound. Layer Meteor. 113 (2004) 81–109. http://dx.doi.org/10.1023/B:BOUN.0000037333.48760.e5
2. C. Gromke, B. Blocken, Influence of avenue-trees on air quality at the urban neighborhood scale. part i:quality assurance studies and turbulent schmidf number analyzis for rans cfd simulations, Environ. Pollut. 196 (2015) 214–223. http://dx.doi.org/10.1016/j.envpol.2014.10.016
3. H. Řezníček, Modelling of the influence of vegetative barrier on particulate matter concentration using OpenFOAM, Proceding of the ENUMATH 2019 conference, Springer.
4. L. Beneš, T. Bodnár, K. Kozel., Comparison of two numerical methods for the stratified flow, COMPUTERS & FLUIDS 46(1) (2011) 148–154. http://dx.doi.org/10.1016/j.compfluid.2011.02.003
5. E. Muelner, W. Mell, Large eddy simulation of forest canopy flow for wildland fire modeling, Can. J. of Forest Research 44(12) (2014) 1534–1544. http://dx.doi.org/10.1139/cjfr-2014-0184
6. T. Litschke, W. Kuttler, On the reduction of urban particle concentration by vegetation - a review, Meteorol. Zeitsch. 17 (2008) 229–240. http://dx.doi.org/10.1127/0941-2948/2008/0284
7. S. Janhäll, Review on urban vegetation and particle air pollution - deposition and dispersion, Atmos. Environ. 105 (2015) 130–137. http://dx.doi.org/10.1016/j.atmosenv.2015.01.052
8. J. Steffens, Y. Wang, , K. Zhang, Exploration of effects of a vegetation barrier on particle size distributions in a near-road environment, Atmos. Environ. 50 (2012) 120–128. http://dx.doi.org/10.1016/j.atmosenv.2011.12.051
9. V. Šíp, L. Beneš, Modelling the effects of a vegetation barrier on road dust dispersion, Appl. Mech. Mater. 821 (2016) 105–112. http://dx.doi.org/10.4028/www.scientific.net/AMM.821.105
10. L. Beneš, V. Šíp, Numerical optimization near-road vegetation barriers, ECCOMASS congress 2016, https://eccomas2016.org/proceedings/.
11. D. Bäumer, B. Vogel, F. Fiedler, A new parameterisation of motorway-induced turbulence and its application in a numerical model, Atmos. Environ. 39 (2005) 5750–5759. http://dx.doi.org/10.1016/j.atmosenv.2004.10.046
12. W. C. Hinds, Aerosol technology: Properties, behavior, and measurement of airborne particles, Wiley, 1999.
13. A. Petroff, et al., Aerosol dry deposition on vegetative canopies. Part II: A new modelling approach and applications, Atmos. Environ. 42 (2008) 3654–3683. http://dx.doi.org/10.1016/j.atmosenv.2007.12.060
14. A. Petroff, L. Zhang, et al., An extended dry deposition model for aerosols onto broadleaf canopies, J. Aerosol Sci. 40 (2009) 218–240. http://dx.doi.org/10.1016/j.jaerosci.2008.11.006
15. B. Lalic, D. T. Mihailovic, An empirical relation describing leaf-area density inside the forest for environmental modeling, Applied Mathematics and Computation 55 (2004) 641–645. http://dx.doi.org/10.1175/1520-0450(2004)043<0641:AERDLD>2.0.CO;2
16. P. Richards, R. Hoxey, Appropriate boundary conditions for computational wind engineering models using the k-ϵ turbulence model, J. Wind Eng. Ind. Aerodyn. 46 & 47 (1993) 145–153. http://dx.doi.org/10.1016/0167-6105(93)90124-7

A Linear Domain Decomposition Method for Non-equilibrium Two-Phase Flow Models

Stephan Benjamin Lunowa, Iuliu Sorin Pop, and Barry Koren

Abstract We consider a model for two-phase flow in a porous medium posed in a domain consisting of two adjacent regions. The model includes dynamic capillarity and hysteresis. At the interface between adjacent subdomains, the continuity of the normal fluxes and pressures is assumed. For finding the semi-discrete solutions after temporal discretization by the θ-scheme, we proposed an iterative scheme. It combines a (fixed-point) linearization scheme and a non-overlapping domain decomposition method. This article describes the scheme, its convergence and a numerical study confirming this result. The convergence of the iteration towards the solution of the semi-discrete equations is proved independently of the initial guesses and of the spatial discretization, and under some mild constraints on the time step. Hence, this scheme is robust and can be easily implemented for realistic applications.

1 Introduction

Flow in porous media has become a significant field of research, as prominent applications such as CO_2 storage and enhanced oil recovery vitally depend on the understanding of the underlying phenomena. Since measurements below surface are costly, if feasible at all, mathematical modeling and simulation are crucial to predict such processes. These models usually consist of coupled nonlinear differential equations, which may degenerate and change type. Besides the increasing complexity of

S. B. Lunowa (✉) · I. S. Pop
Computational Mathematics, Hasselt University, Diepenbeek, Belgium
e-mail: stephan.lunowa@uhasselt.be; sorin.pop@uhasselt.be

B. Koren
Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: b.koren@tue.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_13

the models incorporating dynamic capillarity and hysteresis, another difficulty is caused by the largely varying or even discontinuous physical properties.

To solve the coupled nonlinear equations, discretization and linearization schemes are necessary. Since Newton based solvers suffer from severe constraints on the time step sizes to ensure convergence [19], a simple fixed point iteration, the L-type linearization, has been proposed. Its high robustness comes at the price of a slower, linear convergence. Additionally, this approach is typically independent of the spatial discretization, and has thus been combined e.g. with (M)FEM [13, 18] or a discontinuous Galerkin method [10].

In the situation of block-heterogeneous soils, the application of a domain decomposition method seems natural to decouple the different homogeneous blocks and speed up the convergence. This approach is used and optimized for a wide range of applications [4, 7–9]. In [15], a non-overlapping Schwarz waveform-relaxation was analyzed for nonlinear convection-diffusion equations in a time-continuous setting. Such methods can also be used after temporal discretization for porous media equations [1, 5]. In [21, 22], the domain decomposition was integrated in the linearization process for the Richards equation respectively two-phase flow.

Here, we propose such a linearization and domain decomposition scheme for two-phase flow in porous media, including dynamic and hysteretic effects in the capillary pressure. These methods are independent of the chosen spatial discretization and avoid the use of derivatives as in Newton based iterations.

2 Mathematical Model and Temporal Discretization

Below, $T > 0$ is a fixed, final time and $\Omega \subset \mathbb{R}^d$ ($d \in \mathbb{N}$) a Lipschitz domain. It is partitioned into two Lipschitz subdomains Ω_1 and Ω_2 separated by a $(d-1)$-dimensional manifold Γ. The outer normal vectors at $\partial\Omega_l$ for $l \in \{1, 2\}$ are denoted by $\boldsymbol{\nu}_l$. In each subdomain Ω_l, the flow of two immiscible, incompressible phases $\alpha \in \{n, w\}$ through a rigid porous medium is governed by the mass balance equations, the extended Darcy law and an extended, play-type capillary pressure model [2],

$$-\phi_l \partial_t s_l + \nabla\cdot \mathbf{u}_{n,l} = 0, \qquad \phi_l \partial_t s_l + \nabla\cdot \mathbf{u}_{w,l} = 0 \qquad \text{in } \Omega_l \times (0,T), \tag{1}$$

$$\mathbf{u}_{\alpha,l} = -\lambda_{\alpha,l}(s_l)\mathsf{K}_l \nabla p_{\alpha,l} \qquad \text{in } \Omega_l \times (0,T), \tag{2}$$

$$p_{n,l} - p_{w,l} = p_{c,l}(s_l) - \Phi_{\delta,l}(\partial_t s_l) - \partial_t T_l(s_l) \qquad \text{in } \Omega_l \times (0,T). \tag{3}$$

At Γ, the coupling conditions are the continuity of the normal fluxes and pressures

$$\mathbf{u}_{\alpha,1} \cdot \boldsymbol{\nu}_1 = -\mathbf{u}_{\alpha,2} \cdot \boldsymbol{\nu}_2, \qquad p_{\alpha,1} = p_{\alpha,2} \qquad \text{on } \Gamma \times (0,T). \tag{4}$$

Here, s_l denotes the saturation of the wetting phase, $u_{\alpha,l}$ the specific discharge of the α-phase and $p_{\alpha,l}$ its pressure. The parameters are the porosity $\phi_l \in (0,1)$, the intrinsic permeability $\mathsf{K}_l \in \mathbb{R}^{d\times d}$, which is symmetric, positive definite and bounded, the relative mobility $\lambda_{\alpha,l}$ and the capillary pressure p_c, while T_l and $\Phi_{\delta,l}$ model the dynamic respectively hysteretic effects. In contrast to equilibrium models, in which $T_l = \Phi_{\delta,l} = 0$, this model can reproduce experimental results such as fingering and saturation overshoots [17, 20]. Typically, (3) is a multi-valued relation $p_{n,l} - p_{w,l} \in p_{c,l}(s_l) - \gamma_l \operatorname{sign}(\partial_t s_l) - \partial_t T_l(s_l)$ involving a parameter $\gamma_l \geq 0$ and the sign graph. Here, we use a regularization $\Phi_{\delta,l}$ of sign; namely $\Phi_{\delta,l}(\xi) := \max\{-1, \min\{\delta^{-1}\xi, 1\}\}$ with $\delta > 0$ being a regularization parameter.

For simplicity, we only consider homogeneous Dirichlet boundary conditions for the pressures, i.e. $p_{w,l} \equiv p_{n,l} \equiv 0$ on $(\partial\Omega_l \cap \partial\Omega) \times (0, T)$. Together with an initial datum $s_l(0, \cdot) = s_l^0 \in L^\infty(\Omega)$, (1)–(4) form an initial-boundary-value problem in s, p_n and p_w.

Remark 1 For the existence of unique weak solutions to (1)–(3), we refer to [6, 11]. In particular, we mention [6] for the Hölder continuity of the pressure gradients $\nabla p_n, \nabla p_w$.

Notation 1 *We denote by $L^2(X)$, $H^1(X)$ and $H^{div}(X)$ the standard Hilbert spaces over $X \in \{\Omega, \Omega_1, \Omega_2\}$. $H^{1/2}(\Gamma)$ contains the traces $u|_\Gamma$ on Γ of functions $u \in H^1(\Omega)$. For the two subdomains Ω_l with $l \in \{1, 2\}$, the following spaces will be used*

$$\mathcal{W}_l := \{w \in H^1(\Omega_l) \,:\, w|_{\partial\Omega_l \cap \partial\Omega} \equiv 0\},$$

$$\mathcal{W} := L^2(\Omega) \times [\mathcal{W}_1 \times \mathcal{W}_2]^2, \quad \mathcal{V} := L^2(\Omega) \times [H_0^1(\Omega)]^2.$$

For any function $f \in L^2(\Omega)$, we denote by $f_l := f|_{\Omega_l}$ the restriction to Ω_l for $l \in \{1, 2\}$. Vice versa, we identify a pair of functions $(f_1, f_2) \in L^2(\Omega_1) \times L^2(\Omega_2)$ with f and consider f as the natural L^2-extension on the whole domain Ω. The L^2 inner product on Ω_1 or Ω_2 is denoted by $(\cdot,\ \cdot)$, while on Γ it is $(\cdot,\ \cdot)_\Gamma$.

Next, we summarize all assumptions on the coefficient functions, which are mostly also found in realistic physical systems. Note that the degeneration of the equations is excluded by requiring positive λ_α and Lipschitz continuous p_c. This can be enforced, if necessary, by a regularization like in [6, 16].

Assumption 1 For $l \in \{1, 2\}$ and $\alpha \in \{n, w\}$ we assume that

- $\lambda_{\alpha,l} : \mathbb{R} \to \mathbb{R}^+$ is Lipschitz continuous with Lipschitz constant $L_{\lambda_\alpha,l}$ and there exist $m_{\lambda_\alpha,l}, M_{\lambda_\alpha,l} \in \mathbb{R}^+$ such that $0 < m_{\lambda_\alpha,l} \leq \lambda_{\alpha,l}(s) < M_{\lambda_\alpha,l}$ for all $s \in \mathbb{R}$;
- $p_{c,l} : \mathbb{R} \to \mathbb{R}$ is strictly monotonically decreasing and there exist $m_{p_c,l}, L_{p_c,l} \in \mathbb{R}^+$ such that $m_{p_c,l}\,|r - s| \leq \left|p_{c,l}(r) - p_{c,l}(s)\right| \leq L_{p_c,l}\,|r - s|$ for all $r, s \in \mathbb{R}$;
- $T_l : \mathbb{R} \to \mathbb{R}$ is strictly monotonically increasing with Lipschitz constant $L_{T,l}$.

Remark 2 The extension of $\lambda_{\alpha,l}$, $p_{c,l}$ and T_l to any values $s \in \mathbb{R}$ can be constructed naturally. This is necessary since the solutions to the non-degenerated model need not to satisfy a maximum principle [16].

Furthermore, $\Phi_{\delta,l} : \mathbb{R} \to \mathbb{R}$ is monotonically increasing and Lipschitz continuous with Lipschitz constant $L_{\Phi_\delta,l} = \gamma_l/\delta$.

We discretize the equations in time by the implicit θ-scheme. Given $N \in \mathbb{N}$, let $\Delta t := \frac{T}{N}$ and $\theta \in (0, 1]$. The superscript $(\cdot)^k$ denotes the approximations of the quantities at time $t^k = k\Delta t$, in particular we have $\mathbf{u}^k_{\alpha,l} := -\lambda_{\alpha,l}(s^k_l)\mathsf{K}_l\nabla p^k_{\alpha,l}$ and $p^k_{c,l} := p_{c,l}(s^k_l)$. Time averaged quantities are given by $(\cdot)^{k,\theta} := \theta(\cdot)^k + (1 - \theta)(\cdot)^{k-1}$. After testing, partial integration and summation over $l = 1, 2$ using the continuity of the normal flux across Γ, we obtain the time-discrete counterparts of (1)–(4).

Problem 1 (Semi-Discrete Weak Formulation) Given $(s^{k-1}, p^{k-1}_n, p^{k-1}_w) \in \mathcal{V}$, find $(s^k, p^k_n, p^k_w) \in \mathcal{V}$ such that for all $(\psi_p, \psi_n, \psi_w) \in \mathcal{V}$ there holds

$$-\sum_{l=1}^{2} \phi_l \left(\tfrac{s^k_l - s^{k-1}_l}{\Delta t},\ \psi_{n,l}\right) = \sum_{l=1}^{2} \left(\mathbf{u}^{k,\theta}_{n,l},\ \nabla\psi_{n,l}\right), \tag{5}$$

$$\sum_{l=1}^{2} \phi_l \left(\tfrac{s^k_l - s^{k-1}_l}{\Delta t},\ \psi_{w,l}\right) = \sum_{l=1}^{2} \left(\mathbf{u}^{k,\theta}_{w,l},\ \nabla\psi_{w,l}\right), \tag{6}$$

$$\sum_{l=1}^{2} \left(p^{k,\theta}_{n,l} - p^{k,\theta}_{w,l},\ \psi_{p,l}\right) = \sum_{l=1}^{2} \left(p^{k,\theta}_{c,l} - \Phi_{\delta,l}\Big(\tfrac{s^k_l - s^{k-1}_l}{\Delta t}\Big) - \tfrac{T_l(s^k_l) - T_l(s^{k-1}_l)}{\Delta t},\ \psi_{p,l}\right). \tag{7}$$

Remark 3 (Well-Definedness) If $(s^k, p^k_n, p^k_w) \in \mathcal{V}$ is a solution to Problem 1, we have $p_{\alpha,1}|_\Gamma = p_{\alpha,2}|_\Gamma$ by the definition of $\mathcal{V}$. Since $s^k_l, s^{k-1}_l \in L^2(\Omega_l)$, testing (5) and (6) with arbitrary $\psi_{\alpha,l} \in C^\infty_0(\Omega_l)$ implies $\mathbf{u}^{k,\theta}_{\alpha,l} \in H^{\mathrm{div}}(\Omega_l)$. Therefore, the normal trace lemma [3, Lemma III.1.1] yields $\mathbf{u}^{k,\theta}_{\alpha,l} \cdot \boldsymbol{\nu}_l \in H^{1/2}(\partial\Omega_l)'$ and integration by parts in (5) and (6) implies $\mathbf{u}^{k,\theta}_{\alpha,1} \cdot \boldsymbol{\nu}_1 = -\mathbf{u}^{k,\theta}_{\alpha,2} \cdot \boldsymbol{\nu}_2$ in $H^{1/2}_{00}(\Gamma)'$.

Proving the existence of solutions to this problem lies out of the scope of this paper, but may be done analogously to the time-continuous case mentioned in Remark 1. By this, the time-discrete pressure gradients should be bounded.

3 Linearization and Domain Decomposition

To account for the possible discontinuities at the interface Γ, we decouple the problems in the subdomains. Following [12], we combine the interface conditions $\mathbf{u}^{k,\theta}_{\alpha,1} \cdot \boldsymbol{\nu}_1 = -\mathbf{u}^{k,\theta}_{\alpha,2} \cdot \boldsymbol{\nu}_2$ and $p^k_{\alpha,1} = p^k_{\alpha,2}$ by a parameter $\mathcal{L}_\Gamma \in (0, \infty)$ to obtain

$$g_{\alpha,3-l} = -2\mathcal{L}_\Gamma p^k_{\alpha,l} - g_{\alpha,l}, \qquad \text{where} \qquad g_{\alpha,l} := \mathbf{u}^{k,\theta}_{\alpha,l} \cdot \nu_l - \mathcal{L}_\Gamma p^k_{\alpha,l} \qquad \text{on } \Gamma.$$

This Robin-type formulation is equivalent to the original conditions for any $\mathcal{L}_\Gamma \neq 0$, cf. [22, Remark 1 & 2]. In the next step, we introduce a linearized, iterative scheme, where $i \in \mathbb{N}$ is the iteration index. Given the previous solution $(s^{k.i-1}, p_n^{k,i-1}, p_w^{k,i-1})$ and (g_n^{i-1}, g_w^{i-1}), we define the linearized fluxes and interface conditions as

$$\mathbf{u}_{\alpha,l}^{k,i} := -\theta\lambda_{\alpha,l}(s_l^{k,i-1})\mathsf{K}_l\nabla p_{\alpha,l}^{k,i} + (1-\theta)\mathbf{u}_{\alpha,l}^{k-1}, \quad g_{\alpha,l}^{i} := -2\mathcal{L}_\Gamma p_{\alpha,3-l}^{k,i-1} - g_{\alpha,3-l}^{i-1}.$$

In this way, (5) and (6) become linear and decouple into

$$-\phi_l\left(\frac{s_l^{k,i}-s_l^{k-1}}{\Delta t},\ \psi_{n,l}\right) = \left(\mathbf{u}_{n,l}^{k,i},\ \nabla\psi_{n,l}\right) - \left(\mathcal{L}_\Gamma p_{n,l}^{k,i} + g_{n,l}^{i},\ \psi_{n,l}\right)_\Gamma, \tag{8}$$

$$\phi_l\left(\frac{s_l^{k,i}-s_l^{k-1}}{\Delta t},\ \psi_{w,l}\right) = \left(\mathbf{u}_{w,l}^{k,i},\ \nabla\psi_{w,l}\right) - \left(\mathcal{L}_\Gamma p_{w,l}^{k,i} + g_{w,l}^{i},\ \psi_{w,l}\right)_\Gamma, \tag{9}$$

$$g_{\alpha,l}^{i} = -2\mathcal{L}_\Gamma p_{\alpha,3-l}^{k,i-1} - g_{\alpha,3-l}^{i-1} \quad \text{in } L^2(\Gamma). \tag{10}$$

Finally, we also linearize (7) by adding stabilization terms, which vanish in the limit if the iteration converges. For the latter, we use the parameters $\mathcal{L}_{p,l}, \mathcal{L}_{\Phi,l}, \mathcal{L}_{T,l} > 0$ to account for the nonlinearity of the functions $p_{c,l}$, $\Phi_{l,\delta}$ and T_l. They must satisfy some mild constraints to ensure the convergence of the scheme, as shown below. With this, the linearized and stabilized counterpart of (7) reads

$$\begin{aligned}\left(p_{n,l}^{k,\theta,i} - p_{w,l}^{k,\theta,i},\ \psi_{p,l}\right) &= \left(\theta p_{c,l}(s_l^{k,i-1}) + (1-\theta)p_{c,l}^{k-1} - \Phi_{\delta,l}\left(\frac{s_l^{k,i-1}-s_l^{k-1}}{\Delta t}\right),\ \psi_{p,l}\right)\\ &\quad - \left(\frac{T_l(s_l^{k,i-1})-T_l(s_l^{k-1})}{\Delta t} + \left(\mathcal{L}_{p,l} + \frac{\mathcal{L}_{T,l}+\mathcal{L}_{\Phi,l}}{\Delta t}\right)\left(s_l^{k,i} - s_l^{k,i-1}\right),\ \psi_{p,l}\right),\end{aligned} \tag{11}$$

where $p_{\alpha,l}^{k,\theta,i} := \theta p_{\alpha,l}^{k,i} + (1-\theta)p_{\alpha,l}^{k-1}$. The iteration reduces to solving

Problem 2 (Weak Formulation of the LDD-Scheme) Given $(s^{k-1}, p_n^{k-1}, p_w^{k-1}) \in \mathcal{V}$, $(s^{k,i-1}, p_n^{k,i-1}, p_w^{k,i-1}) \in \mathcal{W}$ and $(g_n^{i-1}, g_w^{i-1}) \in [L^2(\Gamma)]^4$, find $(s^{k,i}, p_n^{k,i}, p_w^{k,i}) \in \mathcal{W}$ and $(g_n^{i}, g_w^{i}) \in [L^2(\Gamma)]^4$ such that (8)–(11) hold for $l \in \{1, 2\}$ and all $(\psi_p, \psi_n, \psi_w) \in \mathcal{W}$.

3.1 Existence of Solutions and Convergence

Here, we summarize the theoretical results for the LDD iteration. This comprises the existence of unique solutions to Problem 2, and the convergence of the iterative sequence. The proofs are generalizations of the ones given in [14] and use ideas from [10, 12, 21, 22]. We omit the details here.

Lemma 1 (Existence) *Problem 2 has a unique solution.*

Theorem 1 (Convergence) *Assume that a solution* $(s^k, p_n^k, p_w^k) \in \mathcal{V}$ *of Problem 1 exists and satisfies* $\|\mathsf{K}_l^{1/2}\nabla p_{\alpha,l}^k\|_{L^\infty(\Omega_l)} \le M_{p_\alpha,l}$ *as well as* $\boldsymbol{u}_{\alpha,l}^k \cdot \boldsymbol{\nu}_l \in L^2(\Gamma)$. *Let Assumption 1 be fulfilled. If the stabilization parameters and time step fulfill for* $l \in \{1, 2\}$

$$\mathcal{L}_{p,l} \ge \theta L_{p_c,l}, \quad \mathcal{L}_{T,l} \ge \tfrac{L_{T,l}}{2}, \quad \mathcal{L}_{\Phi,l} \ge \tfrac{L_{\Phi_\delta,l}}{2} \quad \textit{and} \quad \Delta t < \frac{\phi_l m_{p_c,l}}{\sum\limits_{\alpha\in\{n,w\}} \frac{\theta L_{\lambda_\alpha,l}^2 M_{p_\alpha,l}^2}{m_{\lambda_\alpha,l}}},$$

the sequence of solutions of Problem 2 converges towards (s^k, p_n^k, p_w^k) *for any initial guess* $(s^{k,0}, p_n^{k,0}, p_w^{k,0}) \in \mathcal{W}$ *and* $(g_n^0, g_w^0) \in [L^2(\Gamma)]^4$, *i.e. for* $l \in \{1, 2\}$ *and* $\alpha \in \{n, w\}$

$$s_l^{k,i} \to s_l^k \ \textit{in}\ L^2(\Omega_l), \quad p_{\alpha,l}^{k,i} \to p_{\alpha,l}^k \ \textit{in}\ \mathcal{W}_l, \quad g_{\alpha,l}^i \rightharpoonup g_{\alpha,l} \ \textit{in}\ L^2(\Gamma) \qquad \textit{as}\ i \to \infty.$$

Remark 4 We have $L_{\Phi_\delta,l} = \gamma_l/\delta$, such that $\mathcal{L}_{\Phi,l} \ge \gamma_l/(2\delta)$, while the other parameters and the time step are independent of the regularization.

4 Numerical Experiment

For the validation of the theoretical results, we present a numerical study in a rectangular domain $\Omega = (-1,\ 1) \times (0, 1)$ split into subdomains at the interface $\Gamma = \{0\} \times (0, 1)$. We use a standard finite element method (Q_2) with a uniform mesh with mesh size Δx matching at the interface Γ. We choose the final time $T = 1$ and the Crank-Nicolson method ($\theta = 1/2$) in time, so that we expect errors of the order $O(\Delta t^2 + \Delta x^2)$. Furthermore, we take the same linearization parameters on both subdomains, i.e. $\mathcal{L}_f := \mathcal{L}_{f,1} = \mathcal{L}_{f,2}$ for $f \in \{p, T, \Phi\}$.

We consider an analytically solvable example with isotropic and constant absolute permeability $K_1 = K_2 = I$, and constant porosity $\phi_1 = \phi_2 = 1$ to explicitly compute the experimental order of convergence (EOC). We choose linear coefficient functions, but no hysteresis, i.e. $\lambda_n(s) = 1 - s$, $\lambda_w(s) = s$, $p_c(s) = 0.2 - s$, $T(s) = s$, and $\gamma \equiv 0$. The boundary conditions and right-hand side are selected such that the solution is

$$p_n(x,t) = \tfrac{(1-x_1)(1+x_1)^2}{2(1+t)^2}, \quad p_w(x,t) = \tfrac{(1-x_1)(1+x_1)^2}{2(1+t)}, \quad s(x,t) = \tfrac{(1-x_1)(1+x_1)^2}{2(1+t)} + 0.2.$$

Table 1 The LDD-scheme with the parameters $\mathcal{L}_p = 0.5$, $\mathcal{L}_T = 1$ and $\mathcal{L}_\Gamma = 0.375$ ($\mathcal{L}_\Phi = 0$) achieves experimentally second order convergence (EOC) in pressure (p) and saturation (s). The average number of iterations per time step stays almost constant

$\Delta t = \Delta x$	$\|e_p\|_{L^2H^1}$	EOC$_p$	$\|e_s\|_{L^2H^1}$	EOC$_s$	Avg. # iter.
0.2	$5.352 \cdot 10^{-3}$		$5.824 \cdot 10^{-3}$		13
0.1	$1.394 \cdot 10^{-3}$	1.94	$1.463 \cdot 10^{-3}$	1.993	12.3
0.05	$3.564 \cdot 10^{-4}$	1.968	$3.670 \cdot 10^{-4}$	1.995	12
0.025	$9.013 \cdot 10^{-5}$	1.983	$9.192 \cdot 10^{-5}$	1.997	11.5
0.0125	$2.273 \cdot 10^{-5}$	1.987	$2.312 \cdot 10^{-5}$	1.991	15.5

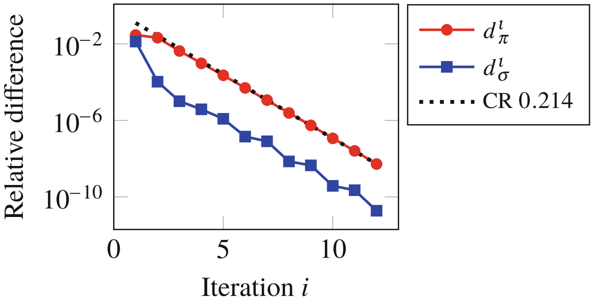


Fig. 1 Error reduction within the last time step of the LDD-scheme for $\Delta t = 0.05$ and $\Delta x = 0.05$. The relative L^2-differences d^i_p and d^i_s in pressure and saturation decrease fast, and the fitted convergence rate (CR) is low

First, we study the behavior of the method with respect to the time step and mesh size. The results in Table 1 clearly confirm the second order convergence in Δt and Δx and indicate that the LDD-iteration is discretization independent, since the average number of iterations per time step stays almost constant.

Next, we study the convergence properties of the method within one time step. For fixed discretization, we study the error reduction and convergence rate in the last time step. The results in Fig. 1 indicate a fast, linear convergence. Moreover, a proper choice of the LDD parameters is crucial for the fast convergence, which can be seen in Fig. 2. Finding the optimum is an open problem, but the lower bounds from our analysis ($\mathcal{L}_p \geq 1/2$ and $\mathcal{L}_T \geq 1/2$) are reasonable indicators.

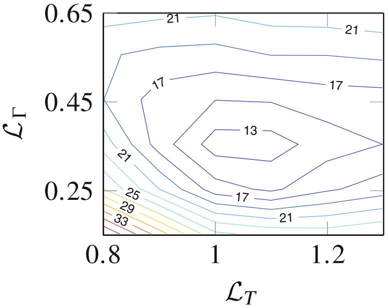


Fig. 2 Parameter dependence of the average number of iterations per time step for fixed $\Delta t = \Delta x = 0.05$ (For simplicity $\mathcal{L}_p = 0$). Deviations from the optimal parameter set drastically increase the convergence rate

5 Conclusion

We proposed an iterative LDD-scheme for finding the semi-discrete solutions of a non-equilibrium two-phase model in a block-heterogeneous domain. We summarized the existence and convergence of the solutions of this LDD-scheme, which holds under a mild restriction for the time step, independently of the initial guesses or of the used spatial discretization. Therefore, the scheme is robust and can be easily adapted for realistic applications.

We will provide a detailed analysis and further numerical studies in a follow-up article. Further investigation is necessary to generalize the method for the degenerated cases. Moreover, an a-posteriori error analysis might lead to estimates for efficient and adaptive stopping criteria.

Acknowledgments This work was supported by Eindhoven University of Technology, Hasselt University (Project BOF17NI01) and the Research Foundation Flanders (FWO, Project G051418N).

References

1. Ahmed, E.: Splitting-based domain decomposition methods for two-phase flow with different rock types. Adv. Water Resour. **134**, 103431 (2019)
2. Beliaev, A.Y., Hassanizadeh, S.M.: A theoretical model of hysteresis and dynamic effects in the capillary relation for two-phase flow in porous media. Transp. Porous Media **43**, 487–510 (2001)
3. Brezzi, F., Fortin, M.: Mixed and Hybrid Finite Element Methods. Springer New York (1991)
4. Caetano, F., et al.: Schwarz waveform relaxation algorithms for semilinear reaction-diffusion equations. Netw. Heterog. Media **5**, 487–505 (2010)
5. Calugaru, D.G., Tromeur-Dervout, D.: Non-overlapping DDMs to solve flow in heterogeneous porous media. In: Domain Decomposition Methods in Science and Engineering, pp. 529–536. Springer-Verlag (2005)
6. Cao, X., Pop, I.S.: Two-phase porous media flows with dynamic capillary effects and hysteresis: Uniqueness of weak solutions. Comput. Math. Appl. **69**, 688–695 (2015)

7. Gander, M.J., Dubois, O.: Optimized schwarz methods for a diffusion problem with discontinuous coefficient. Numer. Algor. **69**, 109–144 (2014)
8. Gander, M.J., Halpern, L., Nataf, F.: Optimal schwarz waveform relaxation for the one dimensional wave equation. SIAM J. Numer. Anal. **41**, 1643–1681 (2003)
9. Gander, M.J., Rohde, C.: Overlapping schwarz waveform relaxation for convection-dominated nonlinear conservation laws. SIAM J. Sci. Comput. **27**, 415–439 (2005)
10. Karpinski, S., Pop, I.S., Radu, F.A.: Analysis of a linearization scheme for an interior penalty discontinuous galerkin method for two-phase flow in porous media with dynamic capillarity effects. Int. J. Numer. Meth. Engng. **112**, 553–577 (2017)
11. Koch, J., Rätz, A., Schweizer, B.: Two-phase flow equations with a dynamic capillary pressure. Eur. J. Appl. Math. **24**, 49–75 (2012)
12. Lions, P.L.: On the Schwarz alternating method III: A variant for nonoverlapping subdomains. In: Third International Symposium on Domain Decomposition Methods for Partial Dierential Equations, pp. 202–223. SIAM (1990)
13. List, F., Radu, F.A.: A study on iterative methods for solving richards' equation. Comput. Geosci. **20**, 341–353 (2016)
14. Lunowa, S.B.: Linearization and domain decomposition methods for two-phase flow in porous media. Master's thesis, Eindhoven University of Technology (2018). URL research.tue.nl/en/studentTheses/linearization-and-domain-decomposition-methods
15. Lunowa, S.B., Rohde, C., Gander, M.J.: Non-overlapping Schwarz waveform-relaxation for quasi-linear convection-diffusion equations (2020). In preparation.
16. Mikelić, A.: A global existence result for the equations describing unsaturated flow in porous media with dynamic capillary pressure. J. Differ. Equ. **248**, 1561–1577 (2010)
17. Mitra, K., van Duijn, C.J.: Wetting fronts in unsaturated porous media: The combined case of hysteresis and dynamic capillary pressure. Nonlinear Anal. Real World Appl. **50**, 316–341 (2019)
18. Pop, I.S., Radu, F., Knabner, P.: Mixed finite elements for the richards' equation: Linearization procedure. J. Comput. Appl. Math. **168**, 365–373 (2004)
19. Radu, F.A., Pop, I.S., Knabner, P.: Newton-type methods for the mixed finite element discretization of some degenerate parabolic equations. In: Numerical Mathematics and Advanced Applications ENUMATH 2005, pp. 1192–1200. Springer Berlin Heidelberg (2006)
20. Schweizer, B.: Instability of gravity wetting fronts for Richards equations with hysteresis. Interfaces Free Bound. **14**, 37–64 (2012)
21. Seus, D., Radu, F.A., Rohde, C.: A linear domain decomposition method for two-phase flow in porous media. In: Numerical Mathematics and Advanced Applications ENUMATH 2017, pp. 603–614. Springer International Publishing (2019)
22. Seus, D., et al.: A linear domain decomposition method for partially saturated flow in porous media. Comput. Methods Appl. Mech. Eng. **333**, 331–355 (2018)

An Adaptive Penalty Method for Inequality Constrained Minimization Problems

W. M. Boon and J. M. Nordbotten

Abstract The primal-dual active set method is observed to be the limit of a sequence of penalty formulations. Using this perspective, we propose a penalty method that adaptively becomes the active set method as the residual of the iterate decreases. The adaptive penalty method (APM) therewith combines the main advantages of both methods, namely the ease of implementation of penalty methods and the exact imposition of inequality constraints inherent to the active set method. The scheme can be considered a quasi-Newton method in which the Jacobian is approximated using a penalty parameter. This spatially varying parameter is chosen at each iteration by solving an auxiliary problem.

1 Introduction

Inequality constrained minimization problems arise in a variety of applications, most prominently in contact problems in mechanics. To solve these problems, written as variational inequalities, a vast number of numerical methods exist and we refer the reader to [7–10], and references therein, for thorough expositions of such methods. This work concerns two seemingly unrelated families of numerical schemes, namely penalty methods (see e.g. [1, 4]) and the primal-dual active set method (see e.g. [5, 6]).

One of the main advantages of penalty methods is the ease of implementation. The penalty term can generally be incorporated as an addition to the original minimization problem in existing numerical software. Strictly speaking, however, the penalty term slightly alters the problem and the obtained solution may not

W. M. Boon (✉)
Department of Mathematics, KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: wietse@kth.se

J. M. Nordbotten
Department of Mathematics, University of Bergen, Bergen, Norway

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_14

satisfy the original constraints exactly. The active set method therefore forms an attractive alternative, as it does explicitly ensure that the solution complies to these constraints. Its disadvantage, however, is that the method typically requires an intrusive implementation in existing software and is prone to slow convergence.

This work forms a link between these two families by proposing a penalty method that adaptively evolves to the primal-dual active set method. Depending on its interpretation, the scheme therefore belongs to both families. In particular, the scheme can be implemented as a penalty method and converges to the same solution as the active set method.

Our starting point is the observation from [5], in which the primal-dual active set method is identified as a semi-smooth Newton method. We expand on this result by considering a regularization of the minimization problem to which the conventional Newton method can be applied. Instead of iterating until convergence, we introduce an adaptive removal of the regularization based on the residual in each iterative step. Thus, as the residual becomes smaller, the regularization decreases and the method is expected to convergence to the solution of the original problem.

The article proceeds as follows. Section 2 introduces the family of constrained minimization problems of interest, the notational conventions, and a concise introduction to the primal-dual active set method and a specific class of penalty methods. The main contribution of this work is presented in Sect. 3, namely an iterative scheme that adaptively combines the advantages of penalty and active set methods. Finally, Sect. 4 presents the numerical performance of the proposed scheme for a synthetic test case corresponding to a one-dimensional obstacle problem.

2 Problem Formulation and Solution Methods

On a given, open domain $\Omega \subset \mathbb{R}^n$, we consider the function space V. We assume V is a reflexive Banach space with norm $\|\cdot\|$ and let V^* denote its dual. Let $f \in V^*$ be a bounded linear functional and $A : V \to V^*$ a continuous, V-elliptic operator, i.e.

$$\langle f, v\rangle \lesssim \|v\|, \qquad \langle Au, v\rangle \lesssim \|u\|\|v\|, \qquad \langle Av, v\rangle \gtrsim \|v\|^2, \qquad \forall u, v \in V.$$

Here, $\langle\cdot,\cdot\rangle$ denotes the $V^* \times V$ duality pairing and the notation $a \lesssim b$ implies that a constant $C > 0$ exists such that $a \leq Cb$. For given $g \in V$, we consider the following constrained minimization problem:

$$\min_{v\in V} J(v) = \min_{v\in V} \frac{1}{2}\langle Av, v\rangle - \langle f, v\rangle \tag{1a}$$

$$\text{subject to } v \leq g \tag{1b}$$

Finding the minimizer $u \in V$ of problem (1) is equivalent to solving either of the following two problems:
Primal formulation:
Find $u \in V$ such that

$$Au - f \leq 0, \tag{2a}$$

$$u - g \leq 0, \tag{2b}$$

$$\langle Au - f, u - g \rangle = 0. \tag{2c}$$

Dual formulation:
Find $(u, \lambda) \in V \times V^*$ such that

$$Au - f + \lambda = 0, \tag{3a}$$

$$\lambda \geq 0, \tag{3b}$$

$$u - g \leq 0, \tag{3c}$$

$$\langle \lambda, u - g \rangle = 0. \tag{3d}$$

For both formulations, we can simplify the inequalities as well as the final equation into a single equation. For that purpose, we introduce the function $M : V^* \times V \to V^*$ given by

$$M(\phi, \varphi) := \phi - [\phi + c\varphi]_+, \tag{4}$$

with $[\psi]_+ = \max\{0, \psi\}$ in the appropriate sense of elements of V^*. Moreover, $c : V \to V^*$ is the inverse Riesz map and we allow c to include a scaling with a positive distribution. Clearly, we have

$$M(\phi, \varphi) = 0 \qquad \Leftrightarrow \qquad \phi \geq 0,\ \varphi \leq 0,\ \langle \phi, \varphi \rangle = 0. \tag{5}$$

Thus, we can equivalently describe the primal formulation (2) by

$$M(f - Au, u - g) = 0, \tag{6}$$

and the dual formulation (3) by

$$Au + \lambda = f, \tag{7a}$$

$$M(\lambda, u - g) = 0. \tag{7b}$$

To solve such problems numerically, we consider two families of iterative schemes, namely the active set method and penalty methods. We continue with a concise expsoition of these methods, presented in the following subsections, respectively.

2.1 Primal-Dual Active Set Method

The primal-dual active set method uses the dual formulation (7) and iteratively updates the set on which the constraint $u = g$ is imposed. For the general problem (1), we define this *active set* at iterate k as

$$\mathcal{A}^k := \{x \in \Omega : \lambda^k(x) + c(u^k(x) - g(x)) > 0\}. \tag{8a}$$

In the case that V is a piecewise linear finite element space defined by nodal evaluations at coordinates x_i, the active set $\mathcal{A}^k$ is defined by

$$\mathcal{A}^k := \{i : \lambda^k(x_i) + c(u^k(x_i) - g(x_i)) > 0\}. \tag{8b}$$

Its complement on Ω is referred to as the *inactive set*, denoted by $\mathcal{I}^k$. For brevity of notation, we introduce the indicator function $\mathbb{1}^k_{\mathcal{A}}$ which is identity in $\mathcal{A}^k$ and zero otherwise. The indicator function $\mathbb{1}^k_{\mathcal{I}}$ is defined analogously. For a given active set $\mathcal{A}^k$, the primal-dual active set method then solves the following system of equations

$$\begin{bmatrix} A & I \\ -\mathbb{1}^k_{\mathcal{A}} c & \mathbb{1}^k_{\mathcal{I}} \end{bmatrix} \begin{bmatrix} u^{k+1} \\ \lambda^{k+1} \end{bmatrix} = \begin{bmatrix} f \\ -\mathbb{1}^k_{\mathcal{A}} cg \end{bmatrix}$$

We simplify this system by substituting $\lambda^{k+1} = f - Au^{k+1}$ from the first row into the second, giving us Algorithm 1.

Algorithm 1 Active set method

```
(i) Set $k = 0$ and initialize $u^0$.
(ii) Compute $\mathcal{A}^k$ using (8).
(iii) Solve for $u^{k+1}$:

    $(\mathbb{1}^k_{\mathcal{I}} A + \mathbb{1}^k_{\mathcal{A}} c)u^{k+1} = \mathbb{1}^k_{\mathcal{I}} f + \mathbb{1}^k_{\mathcal{A}} cg.$    (9)

(iv) Stop if converged, else increment $k$ and return to (ii).
```

2.2 Penalty Method

The defining attribute of penalty methods is the modification of the formulation by introducing a term which penalizes the solution u if it is outside the admissible set [3]. To be precise, we introduce a penalty parameter $\rho \geq 0$ and an operator

$\Pi_\rho : V \to V^*$ to modify the primal formulation (2) to:
Find $u \in V$ such that

$$Au - f + \Pi_\rho u = 0. \tag{10}$$

We use the convention that a smaller value of ρ corresponds to a stricter penalization.

It is advantageous to choose the penalty operator Π_ρ sufficiently smooth in order to apply the Newton method. We consider a particular choice of Π_ρ obtained from a regularization of the problem (6). For that purpose we use [2] and let $[\cdot]_\rho$ be the smooth approximation of $[\cdot]_+$ given by

$$[\phi]_\rho := \phi + \rho \log(1 + \exp(-\phi/\rho)), \qquad [\phi]'_\rho = (1 + \exp(-\phi/\rho))^{-1}.$$

It is important to note that this function and its derivative have the following properties for all $\phi \in V^*$:

$$\lim_{\rho \downarrow 0} [\phi]_\rho = [\phi]_+, \qquad \lim_{\rho \downarrow 0} [\phi]'_\rho = \lim_{\rho \downarrow 0} \frac{d}{d\phi} [\phi]_\rho = \mathbb{1}_{\phi > 0}. \tag{11}$$

Using this operator, we define the regularization of M as

$$M_\rho(\phi, \varphi) := \phi - [\phi + c\varphi]_\rho.$$

In turn, a regularization of the primal formulation (6) arises:

$$-M_\rho(f - Au, u - g) = Au - f + [f - Au + c(u - g)]_\rho = 0 \tag{12}$$

Note that this corresponds to setting $\Pi_\rho u := [f - Au + c(u - g)]_\rho$ in Eq. (10) and we conclude that the regularized formulation (12) has the structure of a penalty method.

Applying this regularization to the dual formulation (7), we similarly obtain

$$Au + \lambda = f, \tag{13a}$$

$$M_\rho(\lambda, u - g) = 0. \tag{13b}$$

Due to the smoothness of M_ρ, the Newton method becomes an attractive solution strategy and we therefore apply this method to the regularized primal problem (12). This leads us to the penalty method presented as Algorithm 2 below. We remark that α_ρ^k is interpreted as a diagonal operator here.

We make two observations concerning Algorithm 2, presented as two lemmas. First, we show an equivalent derivation using the dual formulation (13) and secondly, we note the behavior of the scheme as the penalty parameter tends to zero.

Algorithm 2 Penalty method

```
(i)   Set $k = 0$ and initialize $u^0$.
(ii)  Compute $\alpha_\rho^k = [f - Au^k + c(u^k - g)]'_\rho$.
(iii) Solve for $\delta u$:
          $$((I - \alpha_\rho^k)A + \alpha_\rho^k c)\delta u = M_\rho(f - Au^k, u^k - g). \qquad (14)$$
      and set $u^{k+1} = u^k + \delta u$.
(iv)  Stop if converged, else increment $k$ and return to (ii).
```

Lemma 1 *Applying the Newton method to the regularized dual formulation (13) equivalently leads to Algorithm 2.*

Proof Let us linearize the dual formulation (13) around the previous iterate (u^k, λ^k). Applying the Newton method leads to

$$\begin{bmatrix} A & I \\ \frac{\partial}{\partial u} M_\rho(\lambda^k, u^k - g) & \frac{\partial}{\partial \lambda} M_\rho(\lambda^k, u^k - g) \end{bmatrix} \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix} = - \begin{bmatrix} Au^k + \lambda^k - f \\ M_\rho(\lambda^k, u^k - g) \end{bmatrix}$$

By introducing $\alpha_\rho^k = [\lambda^k + c(u^k - g)]'_\rho$, we specify the derivatives and rewrite:

$$\begin{bmatrix} A & I \\ -\alpha_\rho^k c & I - \alpha_\rho^k \end{bmatrix} \begin{bmatrix} \delta u \\ \delta \lambda \end{bmatrix} = - \begin{bmatrix} Au^k + \lambda^k - f \\ M_\rho(\lambda^k, u^k - g) \end{bmatrix}. \tag{15}$$

Next, we note that $\lambda^k = f - Au^k$ for $k > 0$, giving us $\delta \lambda = -A\delta u$ from the first row. Substituting this into the second row gives us

$$(-\alpha_\rho^k c - (I - \alpha_\rho^k)A)\delta u = -M_\rho(f - Au^k, u^k - g).$$

Negation of this equation gives us (14), thereby concluding the proof. □

Lemma 2 *Algorithm 2 is equivalent to Algorithm 1 in the limit $\rho \downarrow 0$.*

Proof By (11), the limit $\rho \downarrow 0$ gives us $\alpha_\rho^k \to \mathbb{1}_{\mathcal{A}}^k$, i.e. the indicator function of $\mathcal{A}^k$. Moreover, the operator $M_\rho(\cdot, \cdot)$ on the right-hand side becomes $M(\cdot, \cdot)$. Equation (14) then becomes

$$(\mathbb{1}_{\mathcal{I}}^k A + \mathbb{1}_{\mathcal{A}}^k c)\delta u = M(f - Au^k, u^k - g) = \mathbb{1}_{\mathcal{I}}^k (f - Au^k) - \mathbb{1}_{\mathcal{A}}^k c(u^k - g) \tag{16}$$

Addition of $(\mathbb{1}_{\mathcal{I}}^k A + \mathbb{1}_{\mathcal{A}}^k c)u^k$ to both sides of the equation gives us (9). □

3 The Adaptive Penalty Method

In the previous section, we have made two observations. First, introducing a penalty parameter ρ leads to a regularized problem on which the Newton method can be applied. This method is known to be converge (locally) to the solution of the regularized problem. Secondly, as ρ tends to zero, the penalty method becomes equivalent to the active set method, which respects the inequality constraint of (1) exactly. The next step is to combine these two advantages into a single iterative method.

With this goal in mind, we modify the penalty method by letting ρ be a spatially varying function on Ω. This allows us to adaptively remove the penalization in regions where the solution is sufficiently accurate. We achieve this by constructing the penalty function ρ as a regularization of the residual. Let us therefore introduce the following differential equation for ρ:

$$\rho - \epsilon\Delta\rho = \gamma|M(f - Au, u - g)| \qquad \text{in } \Omega, \tag{17a}$$

$$\boldsymbol{n} \cdot \nabla\rho = 0 \qquad \text{on } \partial\Omega. \tag{17b}$$

Here, $|\cdot|$ denotes the absolute value, $\boldsymbol{n}$ is the outward unit normal vector on $\partial\Omega$ and ϵ, γ are chosen, nonnegative constant parameters. For simplicity, we limit our exposition to these two tuning parameters.

By elliptic regularity of (17), the penalization ρ will tend to zero as the residual becomes smaller. We exploit this property and propose Algorithm 3, which we refer to as the Adaptive Penalty Method (APM).

Algorithm 3 Adaptive penalty method

```
(i) Set $k = 0$ and initialize $u^0$.
(ii) Solve (17) for the regularization parameter $\rho$ with data $u = u^k$.
(iii) Compute $\alpha_\rho^k = [f - Au^k + c(u^k - g)]'_\rho$.
(iv) Solve for $\delta u$:
        $((I - \alpha_\rho^k)A + \alpha_\rho^k c)\delta u = M(f - Au^k, u^k - g).$    (18)
     and set $u^{k+1} = u^k + \delta u$.
(v) Stop if converged, else increment $k$ and return to (ii).
```

It is important to note that the exact solution to the auxiliary problem (17) is not our main priority. Thus, in order to reduce computational cost, it will suffice to use an approximate solution in step (ii) with the use of a coarse solve or multi-grid cycle.

Algorithm 3 can be interpreted in a variety of ways. First, the scheme is a quasi-Newton method on (6) in which the Jacobian gets approximated more accurately as the solution converges. The accuracy of the Jacobian adaptively depends on the residual, hence the chosen name.

Alternatively, the algorithm can be considered a warm-start that gradually behaves like the active set method in convergence. The advantage in this context is that no invasive implementations are necessary to switch from the warm-start to the active set method.

Thirdly, the choice of $\gamma = 0$ results in $\rho = 0$ in all iterations and the scheme is effectively reduced to Algorithm 1. In that sense, this construction serves as a generalization of primal-dual active set method. This is an advantage in case optimal parameter values are difficult to find, since the scheme can easily be reduced to the active set method without requiring additional, numerical implementation.

Other extreme choices of the parameters lead to different behaviors of the proposed scheme. A large value of γ, for example, results in a slower decrease of the regularization parameter and therewith, a slower convergence to the solution. On the other hand, setting $\epsilon = 0$ removes the diffusion in (17) which typically results in sporadic behavior of the scheme and possibly, loss of convergence. However, choosing a too large value for ϵ makes the diffusion term dominate which results in a spatially uniform penalty parameter. This is disadvantageous since it leads to unnecessarily poor approximations of the Jacobian in regions where the solution is close to exact.

4 Numerical Results

In this section, we test the numerical performance of the adaptive penalty method using a synthetic test case. Let us consider an obstacle problem on $\Omega = (0, 1)$. We aim to find $u \in H_0^1(\Omega)$ that weakly satisfies

$$-\Delta u \leq f, \quad f(x) := 10, \tag{19a}$$

$$u \leq g, \quad g(x) := 0.2(1 + \mathbb{1}_{x>0.25} + \mathbb{1}_{x>0.5} + \mathbb{1}_{x>0.75}), \tag{19b}$$

$$\langle \Delta u + f, u - g \rangle = 0, \quad \text{in } \Omega, \tag{19c}$$

$$u = 0, \quad \text{on } \partial\Omega. \tag{19d}$$

We set the scaling in the Riesz operator c to unity and iterate until the Euclidean norm of the residual is below a tolerance level of 1e-10. In the numerical experiments, we have not observed significant sensitivities of the scheme with respect to ϵ and therefore limit this exposition to $\epsilon = 1$.

As remarked in the previous section, an interesting variant of the method arises if the penalty parameter is approximated, instead of solving (17) exactly. To explore this variant, we perform a solve on a coarse mesh of 16 elements and interpolate back to the original mesh. We compare three methods, namely the primal-dual active

Table 1 Number of iterations necessary to obtain the desired accuracy for the active set method (ASM), the adaptive penalty method with the penalization ρ solved exactly (APM), and on a coarse mesh (C-APM). The proposed schemes obtain the same solution as the active set method in fewer iterations

	ASM	APM				C-APM			
$1/h$	$\gamma = 0$	$\gamma = 0.1$	$\gamma = 1$	$\gamma = 10$	$\gamma = 100$	$\gamma = 0.1$	$\gamma = 1$	$\gamma = 10$	$\gamma = 100$
256	52	18	9	13	29	18	9	13	31
512	103	33	10	12	33	32	10	13	32
1024	206	69	33	16	45	68	32	14	47
2048	411	162	52	18	50	160	50	18	50
4096	820	382	73	22	53	378	69	26	56
8192	1639	876	107	31	48	864	99	36	54

set method (Algorithm 1), the adaptive penalty method (Algorithm 3) introduced in Sect. 3, and its variant with a coarse solve. The results are shown in Table 1.

From the numerical experiment, we observe that the Adaptive Penalty Method requires significantly fewer iterations than the primal-dual active set method for this problem. As discussed, small values of γ cause the scheme to behave like the active set method and this can be observed in the iteration numbers. Moreover, the number of iterations appear robust with respect to the grid size for the largest choices of γ.

The results from C-APM indicate that the exact evaluation of ρ can be avoided, in practice. This makes the scheme attractive for larger linear systems in terms of computational cost, since there is no need to solve an additional linear system during each iteration.

To conclude, the proposed Adaptive Penalty Method rapidly converges to the same solution as the primal-dual active set method, which satisfies the constraints of the original problem exactly. The scheme is easily implementable as a penalty method or as a quasi-Newton scheme in existing software. To reduce computational cost, the penalty parameter can be approximated using a coarse solve, without significantly affecting the convergence of the method.

Acknowledgments This work was partially supported by Norwegian Research Council grant 233736.

References

1. Carstensen, C., Scherf, O., Wriggers, P.: Adaptive finite elements for elastic bodies in contact. SIAM Journal on Scientific Computing **20**(5), 1605–1626 (1999)
2. Chen, C., Mangasarian, O.L.: Smoothing methods for convex inequalities and linear complementarity problems. Mathematical programming **71**(1), 51–69 (1995)
3. Grossmann, C., Roos, H.G., Stynes, M.: Numerical Treatment of Partial Differential Equations. Universitext. Springer Berlin Heidelberg (2007)

4. Hansbo, P., Johnson, C.: Adaptive finite element methods for elastostatic contact problems. In: Grid Generation and Adaptive Algorithms, pp. 135–149. Springer (1999)
5. Hintermüller, M., Ito, K., Kunisch, K.: The primal-dual active set strategy as a semismooth newton method. SIAM Journal on Optimization **13**(3), 865–888 (2002)
6. Hüeber, S., Wohlmuth, B.I.: A primal–dual active set strategy for non-linear multibody contact problems. Computer Methods in Applied Mechanics and Engineering **194**(27–29), 3147–3166 (2005)
7. Kikuchi, N., Oden, J.T.: Contact Problems in Elasticity: A Study of Variational Inequalities and Finite Element Methods. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics (1988)
8. Suttmeier, F.T.: Numerical Solution of Variational Inequalities by Adaptive Finite Elements. Advances in Numerical Mathematics. Vieweg+Teubner Verlag (2009)
9. Trémolières, R., Lions, J.L., Glowinski, R.: Numerical Analysis of Variational Inequalities. Studies in Mathematics and its Applications. Elsevier Science (2011)
10. Wohlmuth, B.: Variationally consistent discretization schemes and numerical algorithms for contact problems. Acta Numerica **20**, 569–734 (2011)

Multipreconditioning with Application to Two-Phase Incompressible Navier–Stokes Flow

Niall Bootland and Andrew Wathen

Abstract We consider the use of multipreconditioning to solve linear systems when more than one preconditioner is available but the optimal choice is not known. In particular, we consider a selective multipreconditioned GMRES algorithm where we incorporate a weighting that allows us to prefer one preconditioner over another. Our target application lies in the simulation of incompressible two-phase flow. Since it is not always known if a preconditioner will perform well within all regimes found in a simulation, we also consider robustness of the multipreconditioning to a poorly performing preconditioner. Overall, we obtain promising results with the approach.

1 Introduction

In challenging fluid flow simulations used to model hydraulic processes it is often not clear what the best choice of preconditioner might be for solving a given linear system $\mathcal{A}\mathbf{x} = \mathbf{b}$. Further, disparate flow regimes can be encountered in a simulation and the optimal preconditioner may change throughout. One can imagine trying to adaptively change the preconditioner based on tracking the current flow regime. However, this requires knowing *a priori* which preconditioner is likely best in any given regime as well as a suitable evaluation of the current flow, which may well vary within the domain. The required sophistication and good prior knowledge of the preconditioners' performance makes such an adaptive approach less appealing.

Instead we consider using multiple preconditioners simultaneously, aiming to get the best of each. If we can combine the preconditioners then we would like to know whether we can achieve performance similar to the (unknown) best preconditioner

N. Bootland (✉)
University of Strathclyde, Department of Mathematics and Statistics, Glasgow, UK
e-mail: niall.bootland@strath.ac.uk

A. Wathen
University of Oxford, Mathematical Institute, Oxford, UK
e-mail: andy.wathen@maths.ox.ac.uk

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_15

and, further, if together they provide an improvement over any individual approach. Another key question to ask would be that of robustness: whether inclusion of a poorly performing preconditioner significantly affects the overall performance.

These ideas are encompassed within multipreconditioning strategies, where either the iterative method or preconditioning incorporates more than one preconditioner. There are several ways in which multipreconditioning can be employed but it is salient to consider the computational cost incurred weighed against the performance improvements that might be gained. Note, however, that such a strategy might not simply be aiming to give the optimal performance for solving a given system but to provide an overall robustness during a simulation spanning differing regimes.

A simple way to incorporate multiple preconditioners into an iterative method is to change the preconditioner at each iteration, in which case a flexible solver such as FGMRES [9] is required. This is exemplified in *cycling*, where the preconditioner choice changes in a prescribed cyclic order [8]. However, results show convergence never better than the best choice of preconditioner on its own; though such a choice is unknown in advance. While only observed empirically, it stands to reason that this is unlikely to provide improvement over the best preconditioner for any given linear system, though it may help provide robustness over a sequence of problems.

Another strategy is to form a single preconditioner from the options available. This is employed in *combination* preconditioning, in which the action of the inverse of the preconditioner is a linear combination of other preconditioner inverses. The term was introduced in [10] and pursued further in [7], however, their main focus is on maintaining symmetry or positive definiteness (in some nonstandard inner product) so more efficient iterative methods can be used. Nonetheless, combination preconditioning could equally be applied to nonsymmetric cases with less restriction on requiring certain parameter choices or need for a nonstandard inner product.

A similar idea, using linear combinations of preconditioned operators, is found in the earlier *multi-splitting* method [6]. The idea is to utilise multiple different splitting methods to solve the linear system. The approach can be thought of as a stationary iteration with each splitting providing a preconditioner. Yet, as with combination preconditioning, fixed weights for the contributions must be chosen in advance.

Except for cycling, these approaches allow for parallelism in the application of multiple preconditioners. However, the performance of the underlying iterative method will depend on the overall effectiveness of the preconditioners and how they are combined. Instead, we consider a multipreconditioned GMRES method [5] that retains the parallelisable application of preconditioners but computes weights as part of the algorithm which are, in some sense, optimal. It considers not just one new search direction at each iteration but several, given by each preconditioner. We note that the idea was first applied to the conjugate gradient method for symmetric positive definite systems in [3]. However, with multiple preconditioners the search space grows exponentially fast as we continue to iterate. Thus, a selective variant of the algorithm which restricts this growth to be linear is typically necessary.

2 Multipreconditioned GMRES (MPGMRES)

In the standard preconditioned GMRES (or FGMRES) method, at each iteration a new search direction, based on the preconditioned operator, is added to the search space and then a least-squares problem is solved to find a solution with minimum residual norm. The key idea behind *multipreconditioned GMRES* (MPGMRES) [5] is to add *multiple* new search directions at each iteration coming from the different preconditioners available. In fact, the method adds all new search directions from combinations of the preconditioned operators applied to vectors in the current search space, making the search space very rich. An Arnoldi-type block procedure is then used to obtain an orthonormal basis of the search space. MPGMRES then computes the optimal new iterate from this space in the minimum residual least-squares sense. Hence, note that the weights defining the contributions from each preconditioned operator are computed as part of the procedure, unlike in other approaches.

To understand how this *complete* MPGMRES algorithm works, suppose we have $\ell \geq 2$ preconditioners $\mathcal{P}_i$, $i = 1, \ldots, \ell$. We start with an initial residual vector $\mathbf{r}^{(0)}$, which we normalise to give the first basis vector $V^{(1)} = \beta^{-1}\mathbf{r}^{(0)}$, with $\beta = \|\mathbf{r}^{(0)}\|_2$, and collect together the preconditioned (normalised) residuals

$$Z^{(1)} = \beta^{-1}\left[\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \ldots, \mathcal{P}_\ell^{-1}\mathbf{r}^{(0)}\right] \in \mathbb{R}^{n\times\ell}. \tag{1}$$

Using an Arnoldi-type block procedure we orthogonalise columns of $W = \mathcal{A}Z^{(1)}$ with respect to our current basis $V^{(1)}$ and amongst themselves by using a reduced QR factorisation. Normalising then provides new basis vectors $V^{(2)} \in \mathbb{R}^{n\times\ell}$.

At each iteration, k, we increase the MPGMRES search space by applying each of the preconditioners to our newest basis vectors $V^{(k)}$, computing

$$Z^{(k)} = \left[\mathcal{P}_1^{-1}V^{(k)}, \ldots, \mathcal{P}_\ell^{-1}V^{(k)}\right] \in \mathbb{R}^{n\times\ell^k}. \tag{2}$$

The Arnoldi-type block procedure is then used to orthogonalise $W = \mathcal{A}Z^{(k)}$ with respect to the current basis $\widetilde{V}_k = \left[V^{(1)} \ \ldots \ V^{(k)}\right]$ and within itself. This yields new basis vectors $V^{(k+1)} \in \mathbb{R}^{n\times\ell^k}$ and, by storing the coefficients from the Arnoldi-type step in an upper Hessenberg matrix $\widetilde{H}_k$, we obtain an Arnoldi-type decomposition

$$\mathcal{A}\widetilde{Z}_k = \widetilde{V}_{k+1}\widetilde{H}_k, \tag{3}$$

where $\widetilde{Z}_k = \left[Z^{(1)} \ \ldots \ Z^{(k)}\right]$. Note that any linear dependency in columns of $\widetilde{Z}_k$, due to redundancy in the user-provided preconditioners, can be avoided using deflation; see [5, §3]. Now that we have a search space then, similarly to FGMRES, we solve a linear least-squares problem for the minimum residual solution to

$$\min_{\mathbf{x}\in\mathbf{x}^{(0)}+\mathrm{range}(\widetilde{Z}_k)} \|\mathbf{b} - \mathcal{A}\mathbf{x}\|_2 = \min_{\mathbf{y}} \left\| \|\mathbf{r}^{(0)}\|_2\,\mathbf{e}_1 - \widetilde{H}_k\mathbf{y}, \right\|_2, \tag{4}$$

where $\mathbf{x} = \mathbf{x}^{(0)} + \widetilde{Z}_k\mathbf{y}$. Note that there is a natural generalisation of the standard GMRES polynomial minimisation property, as detailed in [5].

While the search space for complete MPGMRES is very rich, we note that it grows exponentially at each iteration, and thus becomes prohibitive in practice. As such, a variant which selects only some of the potential search directions, ideally ensuring only linear growth, is natural to consider as a more practical alternative.

3 Selective MPGMRES (sMPGMRES)

To balance the benefits gained by adding multiple search directions with the storage and compute costs, we might wish to fix the number of preconditioner applications and matrix–vector products independent of the iteration, allowing for parallelisation of these operations via use of a fixed number of processors. To do so, we consider limiting the growth of the search space to be linear with respect to the iteration number k by using a *selective MPGMRES* (sMPGMRES) algorithm outlined in [5].

The search directions in MPGMRES are given by a collection of column vectors Z. To limit the growth of the search space we limit the size of Z, in particular to be proportional to the number of preconditioners, independent of k. To do this we select only certain search directions from the span of the columns of Z, giving a selective MPGMRES algorithm. There are many strategies to choose these directions, for instance, instead of applying the preconditioners to all columns of $V^{(k)}$, as in (2), we might apply them to just a single vector from $V^{(k)}$, selecting this vector differently for each preconditioner. This selection choice need not be the same at each iteration and could incorporate randomness if desired. The corresponding $Z^{(k)}$ is then

$$Z^{(k)} = \left[\mathcal{P}_1^{-1} V_{:,s_1}^{(k)}, \ldots, \mathcal{P}_\ell^{-1} V_{:,s_\ell}^{(k)}\right], \tag{5}$$

where $V_{:,s_i}^{(k)}$ is the s_ith column of $V^{(k)}$ and s_i might change with k.

An alternative to applying each preconditioner to just one vector from $V^{(k)}$ is to apply them all to a linear combination of these vectors, namely to $V^{(k)}\boldsymbol{\alpha}^{(k)}$ for some vector $\boldsymbol{\alpha}^{(k)}$ of appropriate size detailing the contribution from each column of $V^{(k)}$. The corresponding $Z^{(k)}$ is then

$$Z^{(k)} = \left[\mathcal{P}_1^{-1} V^{(k)}\boldsymbol{\alpha}^{(k)}, \ldots, \mathcal{P}_\ell^{-1} V^{(k)}\boldsymbol{\alpha}^{(k)}\right] \in \mathbb{R}^{n\times\ell}. \tag{6}$$

Note that a natural choice for $\boldsymbol{\alpha}^{(k)}$ is the vector $\mathbf{1}$, of all ones. All of these selection methods result in choosing a lower dimensional subspace of the full space and then minimising over this subspace. With these selection strategies, where we limit $Z^{(k)}$ to ℓ new directions each iteration, $\widetilde{V}_{k+1}$ has $k\ell + 1$ basis vectors while the number of columns of $\widetilde{Z}_k$ is $k\ell$. Hence, the storage is proportional to k, as in FGMRES, as opposed to exponential in k, like complete MPGMRES.

Now suppose we have reason to favour one preconditioner over another and, for simplicity, that there are just two candidate preconditioners $\mathcal{P}_1$ and $\mathcal{P}_2$. We would like our selective approach to incorporate knowledge of which preconditioner to favour. As such, we might choose an $\boldsymbol{\alpha}^{(k)} = \boldsymbol{\alpha}$ to weight more the contributions coming from one of the preconditioners. Consider the initial steps in sMPGMRES: we start with new search directions $Z^{(1)}$ and orthogonalise them to be $V^{(2)}$

$$Z^{(1)} = \beta^{-1}\left[\mathcal{P}_1^{-1}\mathbf{r}^{(0)}, \mathcal{P}_2^{-1}\mathbf{r}^{(0)}\right] \overset{\text{orthog.}}{\longrightarrow} V^{(2)}, \tag{7}$$

then add search directions $Z^{(2)}$ which are orthogonalised to be $V^{(3)}$

$$Z^{(2)} = \left[\mathcal{P}_1^{-1}V^{(2)}\boldsymbol{\alpha}, \mathcal{P}_2^{-1}V^{(2)}\boldsymbol{\alpha}\right] \overset{\text{orthog.}}{\longrightarrow} V^{(3)}. \tag{8}$$

So $\boldsymbol{\alpha} = (\alpha_1, \alpha_2)^T$ weighs the contributions from each of the two preconditioners as $V^{(2)}\boldsymbol{\alpha} = \alpha_1 V^{(2)}_{:,1} + \alpha_2 V^{(2)}_{:,2}$ and the two columns of $V^{(2)}$ come from the two different preconditioned residuals. If we let $\boldsymbol{\alpha} = (\alpha, 1-\alpha)^T$, for some $\alpha \in (0, 1)$, then the parameter α states how much we favour the first preconditioner, with $\alpha = \frac{1}{2}$ giving equal weighting and being equivalent to using the vectors of all ones ($\boldsymbol{\alpha} = \mathbf{1}$), as suggested above. Similar strategies could be used to weight contributions from more than two preconditioners.

In this weighted version of sMPGMRES the ordering of the preconditioners $\mathcal{P}_1, \ldots, \mathcal{P}_\ell$ is important as we weight them differently. However, even with equal weighting (that is, $\boldsymbol{\alpha} = \mathbf{1}$) ordering is important. This more nuanced asymmetry within sMPGMRES is an aspect not mentioned in [5]. The asymmetry comes about from the need to orthogonalise the new search directions in $Z^{(k)}$ within themselves. The contribution from the first preconditioner is allowed to be in any new direction but this direction is taken out of the contribution from subsequent preconditioners, and so on as we orthogonalise *in order* the contributions from all preconditioners. This means that if the direction from the last preconditioner is mostly within the span of the preceding directions it may well contribute very little of value, despite coming from a good preconditioner when applied by itself. As a general rule then, we might value less these final search directions as the useful components may have already been taken out. This suggests taking a weighting $\boldsymbol{\alpha}$ which decreases in the components, instead of being equal, might be preferred. Nonetheless, in practice with a small number of good preconditioners, $\boldsymbol{\alpha} = \mathbf{1}$ might suffice to be as good. We will see that when we favour a preconditioner the ordering will matter, even if we are weighting the preconditioners in the same way. Further, ordering can still have a significant impact even when just two preconditioners are used and they are weighted equally, especially when one of the preconditioners is poorer.

4 Numerical Results for sMPGMRES

Here we apply sMPGMRES within a two-phase incompressible Navier–Stokes flow problem. That is, to solve linear systems associated with discretisation of

$$\rho \frac{\partial \mathbf{u}}{\partial t} + \rho \, \mathbf{u} \cdot \nabla \mathbf{u} - \nabla \cdot \left(\mu \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right) \right) + \nabla p = \rho \, \mathbf{f}, \tag{9a}$$

$$\nabla \cdot \mathbf{u} = 0, \tag{9b}$$

for velocity $\mathbf{u}$ and pressure p where density ρ and dynamic viscosity μ are piecewise constant, representing the two phases. An important dimensionless quantity that appears is the dominating Reynolds number Re over the two phases, a parameter which quantifies the ratio of inertial to viscous forces within a fluid. Our results will also exhibit how performance depends on Re. An auxiliary equation to describe how ρ and μ vary in time with the flow is required, such as a level set equation; for the full model see [2]. We consider seeking the $\boldsymbol{Q}_2$– $\boldsymbol{Q}_1$ finite element solution using Newton iteration to treat the nonlinearity. We utilise block preconditioners, in particular those introduced in [2]. These are two-phase versions of the *pressure convection–diffusion* (PCD) and *least-squares commutator* (LSC) approaches [4]. To answer questions of robustness we further use a SIMPLE-type preconditioner, also discussed in [2]. We restrict our results to focus on the two preconditioner case ($\ell = 2$) using (6) with $\boldsymbol{\alpha}^{(k)} = (\alpha, 1 - \alpha)^T$ for some $\alpha \in (0, 1)$. We follow exactly the simplified problem of a lid-driven cavity used in [2] along with the same implementations, as such we omit the details for brevity. The only difference is we now use sMPGMRES to solve the Newton systems via the MATLAB implementation[1] which accompanies [5].

We focus on iteration counts, as opposed to timings, since our implementation runs in serial and so does not take advantage of the inherent parallelism. Note that, when we tabulate our results using sMPGMRES, the iteration counts given in bold emphasise the best choice of weighting parameter α which provides the minimum number of iterations for a given pair of preconditioners. The preconditioner given on the left of a set of results is used as the first preconditioner in sMPGMRES.

Table 1 displays results for combining two-phase PCD and LSC. We see that the best iteration counts are seen towards the centre of the table, that is with a weighting parameter α closer to $\frac{1}{2}$, though we see some bias towards larger α for both orderings as the asymmetry of ordering might suggest. In this example most choices of α will provide some improvement over either of PCD or LSC individually while the best choice can allow convergence using up to 32% fewer iterations. Note that the choice $\alpha = \frac{1}{2}$ typically gives iterations counts close to optimum. Given that it is not clear that we necessarily should do any better than the best preconditioner by itself, these results are quite promising and show that sMPGMRES can improve performance

[1] www.mathworks.com/matlabcentral/fileexchange/34562-multi-preconditioned-gmres.

Table 1 Average preconditioned sMPGMRES iterations upon Newton linearisation using weighted combinations of PCD and LSC with density ratio 1.2×10^{-3}, viscosity ratio 1.8×10^{-2} (values for air-water flow), $h = 1/64$, and varying Reynolds number Re and time-step Δt

			α in PCD–LSC						α in LSC–PCD					
Δt	Re	PCD	0.9	0.7	0.5	0.3	0.1	LSC	0.9	0.7	0.5	0.3	0.1	PCD
10^{-1}	10	16	**14**	**14**	**14**	16	18	18	15	**14**	15	15	21	16
	$10^{1.5}$	16	14	**13**	**13**	14	16	17	14	**13**	14	15	18	16
	100	15	14	**13**	14	16	17	26	15	14	**13**	14	16	15
	$10^{2.5}$	16	14	**12**	**12**	**12**	14	14	12	**11**	12	14	24	16
	1000	19	15	**13**	14	15	15	19	**13**	**13**	**13**	32	38	19
1	10	19	18	**17**	18	21	23	24	19	**18**	**18**	**18**	21	19
	$10^{1.5}$	21	19	**18**	**18**	20	22	23	19	**18**	**18**	19	22	21
	100	25	21	**18**	20	21	23	32	21	19	**18**	21	26	25
	$10^{2.5}$	27	24	19	**18**	**18**	20	22	18	**17**	19	25	37	27
	1000	31	27	**24**	**24**	**24**	26	34	25	**23**	26	37	63	31
10	10	20	19	**18**	19	22	23	25	20	**19**	**19**	**19**	22	20
	$10^{1.5}$	24	21	**20**	21	23	25	26	22	**20**	21	22	27	24
	100	30	26	**23**	26	27	28	42	27	25	**23**	26	30	30
	$10^{2.5}$	35	31	29	**27**	30	35	38	32	29	**28**	32	41	35
	1000	44	47	**40**	44	41	49	58	43	**38**	39	46	85	44

in terms of the number of iterations required. Furthermore, we see in this case that the performance is not particularly sensitive to α. To examine robustness, we now include the SIMPLE-type preconditioner, a method which performs poorly here.

Table 2 combines the LSC and SIMPLE-type preconditioners. We see that, when LSC is used as the first preconditioner, primarily there is relatively little gained from including the SIMPLE-type approach with the best choice either being to simply use LSC or else a large α favouring LSC, though the best reduction in iteration counts does reach to 15%. However, if we change the ordering to have the SIMPLE-type approach first, the picture looks slightly different. While the best iteration counts are very similar, this time any $\alpha \leq \frac{1}{2}$ gives results comparable to LSC. This suggests that, while we do not gain much in the way of improved performance, the algorithm is still fairly robust to varying α so long as we do not favour the poorly performing preconditioner too strongly. This example also provides a case where, with equal weighting ($\alpha = \frac{1}{2}$), the ordering of the preconditioners can substantially matter, with one choice giving iteration counts that are similar or better than LSC and the other giving results that are somewhat worse than LSC. Furthermore, it is by putting the worst preconditioner first (which by the asymmetry is subtly favoured) that we obtain the better results. While at first this may sound counter-intuitive, we can make sense of this observation by considering what the selection in sMPGMRES is doing. If the good preconditioner is used first then we take this contribution away from that of the second preconditioner, likely making it even worse, then by equally weighting these we are allowing a large component of this much worse contribution to prevail. On the other hand, if the worse preconditioner is first, we remove this

Table 2 Average preconditioned sMPGMRES iterations upon Newton linearisation using weighted combinations of LSC and SIMPLE with density ratio 1.2×10^{-3}, viscosity ratio 1.8×10^{-2} (values for air-water flow), $h = 1/64$, and varying Reynolds number Re and time-step Δt

			α in LSC–SIMPLE						α in SIMPLE–LSC					
Δt	Re	LSC	0.9	0.7	0.5	0.3	0.1	SIMPLE	0.9	0.7	0.5	0.3	0.1	LSC
10^{-1}	10	**18**	**18**	19	24	48	97	164	48	24	19	**18**	19	**18**
	$10^{1.5}$	17	**16**	17	21	32	85	154	41	21	17	**16**	18	17
	100	26	23	**22**	27	35	51	131	38	26	23	**22**	24	26
	$10^{2.5}$	**14**	**14**	15	18	27	35	116	33	19	15	15	15	**14**
	1000	**19**	**19**	20	22	24	39	109	31	25	21	21	21	**19**
1	10	24	**22**	24	32	60	93	177	40	27	23	**22**	24	24
	$10^{1.5}$	23	**21**	24	31	60	95	185	40	27	23	**22**	25	23
	100	32	**30**	31	40	70	103	188	60	36	31	**30**	31	32
	$10^{2.5}$	22	**20**	21	26	50	109	190	46	25	21	**20**	22	22
	1000	34	**31**	33	43	51	78	190	62	38	35	**32**	**32**	34
10	10	25	**22**	25	33	60	96	179	40	28	**23**	**23**	24	25
	$10^{1.5}$	26	**24**	26	34	62	98	192	41	29	**24**	25	27	26
	100	42	**38**	39	47	69	104	207	64	42	37	**36**	39	42
	$10^{2.5}$	38	**33**	35	42	82	125	233	54	38	**33**	34	37	38
	1000	58	**49**	51	64	96	150	294	85	57	51	**49**	52	58

component from the contribution of the better preconditioner, which is unlikely to make this contribution worse and may possibly make it even better. Thus we see this latter combination is more favourable than the former, though we may not expect it to provide significantly better results than the best preconditioner by itself. We note that, in results not shown, a somewhat similar scenario occurs when combining PCD and the SIMPLE-type approach; see also [1] for further numerical results.

Our study show promise that sMPGMRES can combine multiple preconditioners to reduce overall iteration counts and, additionally, provide robustness in situations when one preconditioner is performing poorly. Further, weights can be incorporated to favour preconditioners and results are not particularly sensitive to any sensible choice of weights, though ordering can be important. It remains to confirm how much speed-up can be gained from sMPGMRES but initial results in [5] are positive.

Acknowledgments This publication is based on work supported by the EPSRC Centre for Doctoral Training in Industrially Focused Mathematical Modelling (EP/L015803/1) in collaboration with the US Army Coastal and Hydraulics Laboratory and HR Wallingford. In particular, we would like to thank Dr Chris Kees for his expert guidance in this project.

References

1. Bootland, N.: Scalable two-phase flow solvers. D.Phil. thesis, University of Oxford (2018)
2. Bootland, N., Bentley, A., Kees, C., Wathen, A.: Preconditioners for two-phase incompressible Navier–Stokes flow. SIAM J. Sci. Comput. **41**(4), B843–B869 (2019)
3. Bridson, R., Greif, C.: A multipreconditioned conjugate gradient algorithm. SIAM J. Matrix Anal. Appl. **27**(4), 1056–1068 (2006)
4. Elman, H.C., Silvester, D.J., Wathen, A.J.: Finite Elements and Fast Iterative Solvers: with Applications in Incompressible Fluid Dynamics, second edn. Oxford University Press (2014)
5. Greif, C., Rees, T., Szyld, D.B.: GMRES with multiple preconditioners. SeMA **74**(2), 213–231 (2017)
6. O'Leary, D.P., White, R.E.: Multi-splittings of matrices and parallel solution of linear systems. SIAM J. Algebraic Discret. Methods **6**(4), 630–640 (1985)
7. Pestana, J., Wathen, A.J.: Combination preconditioning of saddle point systems for positive definiteness. Numer. Linear Algebra Appl. **20**(5), 785–808 (2013)
8. Rui, P.L., Yong, H., Chen, R.S.: Multipreconditioned GMRES method for electromagnetic wave scattering problems. Microw. Opt. Technol. Lett. **50**(1), 150–152 (2007)
9. Saad, Y.: A flexible inner-outer preconditioned GMRES algorithm. SIAM J. Sci. Comput. **14**(2), 461–469 (1993)
10. Stoll, M., Wathen, A.: Combination preconditioning and the Bramble–Pasciak$^+$ preconditioner. SIAM J. Matrix Anal. Appl. **30**(2), 582–608 (2008)

On the Dirichlet-to-Neumann Coarse Space for Solving the Helmholtz Problem Using Domain Decomposition

Niall Bootland and Victorita Dolean

Abstract We examine the use of the Dirichlet-to-Neumann coarse space within an additive Schwarz method to solve the Helmholtz equation in 2D. In particular, we focus on the selection of how many eigenfunctions should go into the coarse space. We find that wave number independent convergence of a preconditioned iterative method can be achieved in certain special cases with an appropriate and novel choice of threshold in the selection criteria. However, this property is lost in a more general setting, including the heterogeneous problem. Nonetheless, the approach converges in a small number of iterations for the homogeneous problem even for relatively large wave numbers and is robust to the number of subdomains used.

1 Introduction

Within domain decomposition methods, the use of a coarse space as a second level is typically required to provide scalability with respect to the number of subdomains used [4]. More recently, coarse spaces have also been designed to provide robustness to model parameters, especially for large contrasts in heterogeneous problems. For example, the GenEO coarse space has been successfully employed for the robust solution of highly heterogeneous elliptic problems [8]. One way in which a coarse space can be derived is via solving local eigenvalue problems on subdomains, as is the case for the GenEO method. An earlier approach, having many similarities, is the Dirichlet-to-Neumann (DtN) coarse space [6]. We focus on this method which

N. Bootland (✉)
University of Strathclyde, Department of Mathematics and Statistics, Glasgow, UK
e-mail: niall.bootland@strath.ac.uk

V. Dolean
University of Strathclyde, Department of Mathematics and Statistics, Glasgow, UK

Université Côte d'Azur, CNRS, Laboratoire J.A. Dieudonné, Nice, France
e-mail: work@victoritadolean.com

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_16

solves eigenvalue problems on the boundary of subdomains related to a Dirichlet-to-Neumann map.

We are interested in using domain decomposition methodology to solve wave propagation problems. In particular, we consider the Helmholtz problem[1]

$$-\Delta u - k^2 u = f \qquad \text{in } \Omega, \tag{1a}$$

$$u = 0 \qquad \text{on } \Gamma_D, \tag{1b}$$

$$\frac{\partial u}{\partial n} + iku = 0 \qquad \text{on } \Gamma_R, \tag{1c}$$

with wave number $k > 0$, where $\partial\Omega = \Gamma_D \cup \Gamma_R$ and $\Gamma_D \cap \Gamma_R = \emptyset$. Such problems arise in many wave propagation and scattering problems in science and engineering, for instance, acoustic and seismic imaging problems. Furthermore, we also consider the heterogeneous problem, in which case $k(\mathbf{x})$ varies in the domain Ω. We suppose the variation in k stems from the wave speed $c(\mathbf{x})$ depending on the heterogeneous media, with the wave number being given by $k = \omega/c$ for angular frequency ω.

The wave number k is the key parameter within the Helmholtz equation and as k increases the problem becomes more challenging. We are interested in the case when k becomes large and so solutions are highly oscillatory. The numerical method employed needs to be able to capture this behaviour, often through an increasing number of grid points, such as a fixed number of points per wavelength. However, typically the number of grid points needs to grow faster than linearly in k if accuracy is to be maintained due to the pollution effect [1]. For instance, when using P1 finite elements for the numerical solution of (1), the mesh spacing h should decrease proportional to $k^{-3/2}$. This means very large linear systems must be solved when k is large and, since these systems are sparse, iterative methods are most often employed for their solution. However, efficiently solving large discrete Helmholtz systems is challenging since classical iterative methods fail to be effective [5]. As such, we require a more robust iterative solver. Here we consider a restricted additive Schwarz (RAS) method with a Dirichlet-to-Neumann coarse space [3] and will be interested in the performance of this solver methodology as k increases. We now review the underlying numerical methods we use.

2 Discretisation and Solver Methodology

To discretise we use finite element methodology, in particular using piecewise linear (P1) finite elements on simplicial meshes. Given a simplicial mesh T^h on a bounded polygonal domain Ω, let $V^h \subset \left\{H^1(\Omega) \colon u = 0 \text{ on } \Gamma_D\right\}$ be the space of piecewise

[1] Note that if $\Gamma_R = \emptyset$ then the problem will be ill-posed for certain choices of k corresponding to Dirichlet eigenvalues of the corresponding Laplace problem.

linear functions on T^h. The P1 finite element solution $u_h \in V^h$ satisfies the weak formulation $a(u_h, v_h) = F(v_h)\ \forall v_h \in V^h$, where

$$a(u, v) = \int_\Omega \left(\nabla u \cdot \nabla \bar{v} - k^2 u \bar{v}\right) \mathrm{d}\mathbf{x} + \int_{\Gamma_R} iku\bar{v}\,\mathrm{d}s, \quad \text{and} \quad F(v) = \int_\Omega f\bar{v}\,\mathrm{d}\mathbf{x}. \tag{2}$$

Using the standard nodal basis for V^h we can represent the solution u_h through its basis coefficients $\mathbf{u}$ and reduce the problem to solving the complex symmetric linear system $A\mathbf{u} = \mathbf{f}$ where A comes from the bilinear form $a(\cdot, \cdot)$ and $\mathbf{f}$ the linear functional $F(\cdot)$; see, for example, [3].

To solve the discrete Helmholtz system $A\mathbf{u} = \mathbf{f}$ we utilise a two-level domain decomposition preconditioner within an iterative Krylov method. Since A is only complex symmetric rather than Hermitian, we use GMRES as the iterative Krylov method [7]. For the domain decomposition, given an overlapping partition $\left\{\Omega_j\right\}_{j=1}^N$ of Ω, let R_j represent the matrix form of the restriction onto subdomain Ω_j. Then the restricted additive Schwarz (RAS) domain decomposition preconditioner is given by

$$M_{\mathrm{RAS}}^{-1} = \sum_{j=1}^{N} R_j^T D_j A_j^{-1} R_j, \tag{3}$$

where $A_j = R_j A R_j^T$ is the local Dirichlet matrix on Ω_j and the diagonal matrices D_j are a discrete representation of a partition of unity (see [4]); this removes "double counting" in regions of overlap. Note that each subdomain contribution from the sum in (3) can be computed locally in parallel. Using the one-level preconditioner (3) is not sufficient to provide robustness with respect to the number of subdomains N used and also becomes much worse when k increases. To this end we incorporate a coarse space as a second level within the method.

A coarse space provides a more efficient way to transfer information globally between subdomains, rather than relying solely on local solutions, as in (3). The coarse space constitutes a collection of column vectors Z, having full column rank. We then utilise the coarse correction operator $Q = ZE^{-1}Z^\dagger$, where $E = Z^\dagger AZ$ is the coarse space operator, which provides a coarse solution in the space spanned by the columns of Z. To incorporate the coarse correction we use an adapted deflation (AD) approach given by the two-level preconditioner

$$M_{AD}^{-1} = M_{\mathrm{RAS}}^{-1}(I - AQ) + Q. \tag{4}$$

To complete the specification, we must choose which vectors go into the coarse space matrix Z.

3 The Dirichlet-to-Neumann Coarse Space

We now introduce the Dirichlet-to-Neumann coarse space. The construction is based on solving local eigenvalue problems on subdomain boundaries related to the DtN map. To define this map we first require the Helmholtz extension operator from the boundary of a subdomain Ω_j.

Let $\Gamma_j = \partial\Omega_j \setminus \partial\Omega$ and suppose we have Dirichlet data v_{Γ_j} on Γ_j, then the Helmholtz extension v in Ω_j is defined as the solution of

$$-\Delta v - k^2 v = 0 \qquad \text{in } \Omega_j, \tag{5a}$$

$$v = v_{\Gamma_j} \qquad \text{on } \Gamma_j, \tag{5b}$$

$$\mathcal{C}(v) = 0 \qquad \text{on } \partial\Omega_j \cap \partial\Omega, \tag{5c}$$

where $\mathcal{C}(v) = 0$ represents the original problem boundary conditions (1b) and (1c). The DtN map takes Dirichlet data v_{Γ_j} on Γ_j to the corresponding Neumann data, that is

$$\mathrm{DtN}_{\Omega_j}(v_{\Gamma_j}) = \left.\frac{\partial v}{\partial n}\right|_{\Gamma_j} \tag{6}$$

where v is the Helmholtz extension defined by (5).

We now seek eigenfunctions of the DtN map locally on each subdomain Ω_j, given by solving

$$\mathrm{DtN}_{\Omega_j}(u_{\Gamma_j}) = \lambda u_{\Gamma_j} \tag{7}$$

for eigenfunctions u_{Γ_j} and eigenvalues $\lambda \in \mathbb{C}$. To provide functions to go into the coarse space, we take the Helmholtz extension of u_{Γ_j} in Ω_j and then extend by zero into the whole domain Ω using the partition of unity. For further details and motivation, as well as the discrete formulation of the eigenproblems, see [3].

It remains to determine which eigenfunctions of (7) should be included in the coarse space. Several selection criteria were investigated in [3] and it was clear that the best choice was to select eigenvectors corresponding to eigenvalues with the smallest real part. That is, we use a threshold on the abscissa $\eta = \mathrm{Re}(\lambda)$ given by

$$\eta < \widehat{\eta}_j \tag{8}$$

where $\widehat{\eta}_j$ depends on $k_j = \max_{\mathbf{x}\in\Omega_j} k(\mathbf{x})$. In particular, [3] advocates the choice $\widehat{\eta}_j = k_j$. Clearly, the larger $\widehat{\eta}_j$ is taken, the more eigenfunctions we include in the coarse space, increasing its size and the associated computational cost. However, it is not clear that $\widehat{\eta}_j = k_j$ is necessarily the best choice. We investigate the utility of choosing $\widehat{\eta}_j$ larger than k_j and will see that, in some cases, taking a slightly larger coarse space can give improved behaviour of the iteration counts as k increases.

4 Numerical Results

To investigate the dependence on $\widehat{\eta}_j$ we use a 2D wave guide problem on the unit square $\Omega = (0, 1)^2$ as a model test problem. The Dirichlet condition (1b) is imposed on the left and right boundaries $\Gamma_D = \{0, 1\} \times [0, 1]$ while the Sommerfeld radiation condition (1c) is prescribed for the top and bottom boundaries $\Gamma_R = [0, 1] \times \{0, 1\}$. The right-hand side f models a point source at the centre $(\frac{1}{2}, \frac{1}{2})$. The wave number k is either constant throughout Ω for the homogeneous problem or else $k = \omega/c$ where ω is constant and $c(\mathbf{x})$ is piecewise constant as illustrated in Fig. 1 for a contrast parameter $\rho > 1$. These heterogeneous problems model layered media.

To discretise we use a uniform square grid with n_{glob} points in each direction and triangulate with alternating diagonals to form the P1 elements. As we increase k we choose $n_{\text{glob}} \propto k^{3/2}$ in order to ameliorate the pollution effect. To begin with, we use a uniform decomposition into N square subdomains and throughout use minimal overlap (non-overlapping subdomains are extended by having adjoining elements added). All computations are performed using FreeFem (http://freefem.org/), in particular using the `ffddm` framework. When solving the linear systems we use preconditioned GMRES with the two-level preconditioner (4) incorporating the DtN coarse space with threshold $\widehat{\eta}_j$ to reach a relative residual tolerance of 10^{-6}.

In Table 1 we vary the threshold $\widehat{\eta}_j = \widehat{\eta}$ as powers of k for the homogeneous problem using a fixed 5×5 square decomposition. The best choice advocated in [3], namely $\widehat{\eta} = k$, succeeds in requiring relatively low iteration counts in order to reach convergence with a modest size of coarse space. However, we observe that as the wave number k increases the number of iterations required also increases, suggesting the approach will begin to struggle if k becomes too large. We see from other choices of $\widehat{\eta}$ that taking a larger coarse space reduces the iteration counts. For instance, with the largest wave number tested when $\widehat{\eta} = k^{1.2}$ the size of the coarse space doubles while the iteration count it cut almost by a factor of three compared to $\widehat{\eta} = k$. In fact, there is a qualitative change in behaviour with respect to the wave number k, namely independence of the iteration counts to k, once $\widehat{\eta}$ becomes large enough, this point being approximately given by $\widehat{\eta} = k^{4/3}$. We note that the size of the coarse space is approximately proportional to $\widehat{\eta}$ in the results of Table 1 (see

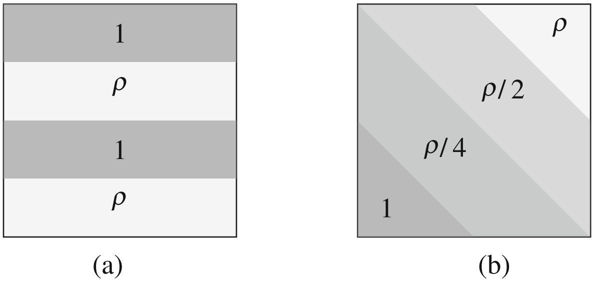


Fig. 1 Different layered configurations for the heterogeneous wave speed $c(\mathbf{x})$ within the wave guide problem, where $\rho > 1$ is a contrast parameter. (**a**) Alternating layers. (**b**) Diagonal layers

Table 1 Preconditioned GMRES iteration counts using the two-level method while varying the threshold parameter $\widehat{\eta}$ for the DtN coarse space. The size of the coarse space is given in brackets. A uniform decomposition into 5 × 5 square subdomains is used

n_{glob}	k	$\widehat{\eta} = k$	$\widehat{\eta} = k^{1.1}$	$\widehat{\eta} = k^{1.2}$	$\widehat{\eta} = k^{1.3}$	$\widehat{\eta} = k^{1.4}$	$\widehat{\eta} = k^{1.5}$
100	18.5	12 (144)	9 (160)	8 (200)	7 (240)	6 (320)	5 (400)
200	29.3	16 (215)	11 (240)	9 (320)	7 (434)	6 (560)	5 (760)
400	46.5	18 (299)	13 (393)	10 (545)	7 (784)	6 (1074)	4 (1480)
800	73.8	27 (499)	18 (674)	10 (960)	8 (1376)	6 (2025)	4 (2928)

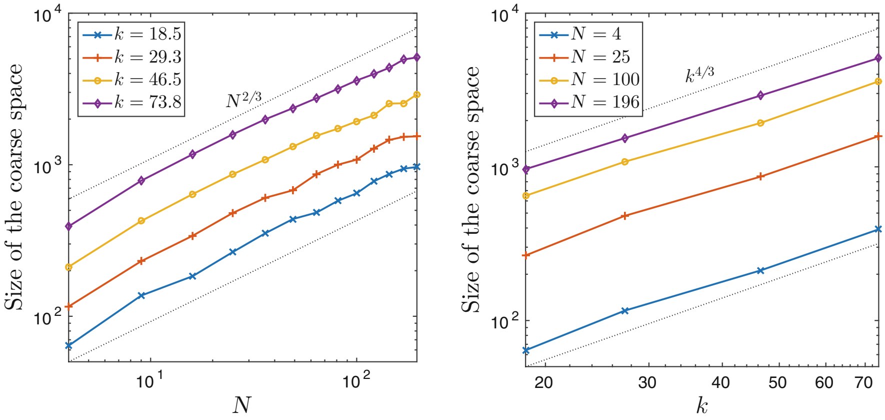


Fig. 2 The size of the DtN coarse space as a function of the number of subdomains N (left) and wave number k (right) for the homogeneous problem with threshold $\widehat{\eta} = k^{4/3}$

also Fig. 2). As such, we see that the coarse space should grow faster than linearly in k in order to achieve wave number independent iteration counts for this problem.

We now verify that the DtN coarse space provides an approach which is scalable with respect to the number of subdomains N. Table 2 details results for a varying number of square subdomains when using a threshold $\widehat{\eta} = k^{4/3}$. As well as seeing the iteration counts staying predominantly constant as we increase k, they do also as we increase the number of subdomains N (aside from a small number of slightly larger outliers). Note that, while the size of the coarse space increases as we increase N, approximately at a rate proportional to $N^{2/3}$ as shown in Fig. 2 (in fact, independent of our choice of $\widehat{\eta}$), the number of eigenfunctions required per subdomain decreases with N. This means the solution of each eigenproblem is much cheaper for large N as they are of smaller size and we require fewer eigenfunctions.

Table 2 Preconditioned GMRES iteration counts when using the two-level method with threshold parameter $\widehat{\eta} = k^{4/3}$ for the DtN coarse space and varying the number of subdomains N. A uniform decomposition into $\sqrt{N} \times \sqrt{N}$ square subdomains is used

		N												
n_{glob}	k	4	9	16	25	36	49	64	81	100	121	144	169	196
100	18.5	6	6	8	6	6	6	6	6	6	6	6	7	7
200	29.3	6	13	8	6	6	17	7	7	7	7	7	7	7
400	46.5	7	11	7	7	7	7	7	7	10	20	7	7	7
800	73.8	7	9	9	7	7	7	7	7	7	7	8	7	7

We now turn our attention to the heterogeneous case. Table 3 (left) gives results for the alternating layers wave guide problem (see Fig. 1a) for varying angular frequency ω, contrast in wave speed ρ, and number of subdomains N when using $\widehat{\eta}_j = {k_j}^{4/3}$ in subdomain Ω_j. The picture painted is now rather different from the homogeneous case. While for some choices of N iteration counts remain robust to wave number, in general they degrade as ω increases. The best results are for $N = 4$, 16, and 64 (powers of 2) while the poorest are with large N. More generally, if the subdomains are close to being aligned with the jumps in k we obtain better results, otherwise robustness is lost. We note, however, that iteration counts are robust to large contrasts ρ. We confirm that the disparate trends observed for the alternating layers problem are due to the geometrical aspects of the problem by considering instead the diagonal layers problem (see Fig. 1b). Results for the diagonal layers problem are given in Table 3 (right) and now show that any robustness to the wave number is, in general, lost for the heterogeneous problem. We note that increasing the threshold to $\widehat{\eta}_j = {k_j}^{3/2}$ does not improve this assessment. Nonetheless, the DtN approach remains robust to increasing the number of subdomains N.

We now show that the sensitivity of the DtN approach is not solely due to the heterogeneity of the media by reconsidering the homogeneous problem but using non-uniform subdomains, which we compute using METIS. Results for this case are given in Table 4 where we see a slow but definite increase in iteration counts as k increases. Again, we see robustness to the number of subdomains but lose robustness to the wave number. Note that this persists even for $\widehat{\eta} = k^{3/2}$. Nonetheless, in our DtN approach we still have rather few GMRES iterations required to compute the solution when k is relatively large (in this case up to $k = 117.2$).

Table 3 Preconditioned GMRES iteration counts for the heterogeneous layers problem when using the two-level method with threshold $\widehat{\eta}_j = k_j{}^{4/3}$ for the DtN coarse space and varying the number of subdomains N. A uniform decomposition into $\sqrt{N} \times \sqrt{N}$ square subdomains is used

			Alternating layers problem											Diagonal layers problem										
			N											N										
n_{glob}	ω	ρ	4	9	16	25	36	49	64	81	100	121	144	4	9	16	25	36	49	64	81	100	121	144
100	18.5	10	6	6	7	6	6	14	6	16	13	17	13	7	12	15	18	18	18	18	18	17	17	18
		100	6	6	7	6	6	14	6	16	13	17	13	11	20	21	20	19	19	18	18	17	16	16
		1000	6	6	7	6	6	14	6	16	13	17	13	11	20	21	20	19	19	18	18	17	16	16
200	29.3	10	8	8	9	8	9	29	9	29	25	32	22	9	17	20	19	19	23	21	28	28	28	28
		100	8	7	9	8	8	28	8	28	23	30	20	16	28	30	30	28	29	27	27	25	25	24
		1000	8	7	9	8	8	28	8	28	23	30	20	16	28	30	29	28	29	27	27	25	25	24
400	46.5	10	9	7	7	7	7	25	7	29	25	45	22	10	18	20	25	26	26	25	26	26	32	29
		100	9	7	7	7	7	24	7	28	24	44	22	22	39	43	43	40	40	39	37	37	39	35
		1000	9	7	7	7	7	24	7	28	24	44	22	22	38	43	42	40	40	39	37	37	39	34
800	73.8	10	8	10	8	11	9	30	8	34	27	36	33	11	19	24	29	28	30	31	34	37	40	41
		100	8	10	8	11	9	38	8	43	33	39	31	32	52	62	61	60	58	58	54	53	53	51
		1000	8	10	8	11	9	38	8	43	33	39	31	32	55	60	60	59	56	56	54	52	51	49

Table 4 Preconditioned GMRES iteration counts (above) and size of the coarse space (below) when using the two-level method with threshold parameter $\widehat{\eta} = k^{4/3}$ for the DtN coarse space and varying the number of subdomains N. A non-uniform decomposition into N subdomains is used

		N												
n_{glob}	k	4	9	16	25	36	49	64	81	100	121	144	169	196
Preconditioned GMRES iteration counts														
100	18.5	7	7	11	7	7	7	7	7	7	7	7	7	7
200	29.3	8	9	10	7	11	9	7	7	7	7	7	7	7
400	46.5	8	10	10	13	14	16	9	14	15	9	8	8	8
800	73.8	8	10	12	12	15	15	13	17	11	14	10	12	17
1600	117.2	10	12	12	15	16	16	17	17	17	16	16	16	16
Size of the coarse space														
100	18.5	75	158	219	303	397	476	558	644	740	829	923	1024	1118
200	29.3	135	282	418	558	677	860	1003	1123	1275	1435	1588	1731	1867
400	46.5	241	516	751	1001	1291	1569	1818	2048	2294	2596	2850	3145	3366
800	73.8	481	979	1446	1919	2378	2844	3261	3753	4291	4651	5246	5720	6126
1600	117.2	925	1857	2639	3566	4408	5244	6201	7008	7909	8770	9563	10448	11402

5 Conclusions

In this work we have investigated a two-level domain decomposition approach to solving the heterogeneous Helmholtz equation. Our focus has been on the Dirichlet-to-Neumann coarse space and how the approach depends on the threshold to select which eigenfunctions go into the coarse space. We have seen that the threshold in [3] can be improved in order to give wave number independent convergence with only moderate added cost due to the larger coarse space. However, this is only true for the homogeneous problem with sufficiently uniform subdomains. In particular, convergence depends on the wave number for a general heterogeneous problem.

In order to obtain fully wave number independent convergence for Helmholtz problems, a stronger coarse space is needed. A recent approach that achieves this, based on a related GenEO-type method, can be found in [2].

References

1. Babuška, I.M., Sauter, S.A.: Is the pollution effect of the FEM avoidable for the Helmholtz equation considering high wave numbers? SIAM J. Numer. Anal. **34**(6), 2392–2423 (1997)
2. Bootland, N.: Coarse Spaces for Helmholtz. Scottish Numerical Methods Network Workshop on Iterative Methods for Partial Differential Equations. 2019, http://personal.strath.ac.uk/jennifer.pestana/lms19/Bootland.pdf

   Bootland, N., Dolean, V., Jolivet, P.: A GenEO-type coarse space for heterogeneous Helmholtz problems (2019). In preparation
3. Conen, L., Dolean, V., Krause, R., Nataf, F.: A coarse space for heterogeneous Helmholtz problems based on the Dirichlet-to-Neumann operator. J. Comput. Appl. Math. **271**, 83–99 (2014)
4. Dolean, V., Jolivet, P., Nataf, F.: An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation, vol. 144. SIAM (2015)
5. Ernst, O.G., Gander, M.J.: Why it is difficult to solve Helmholtz problems with classical iterative methods. In: I.G. Graham, T.Y. Hou, O. Lakkis, R. Scheichl (eds.) Numerical Analysis of Multiscale Problems, pp. 325–363. Springer (2012)
6. Nataf, F., Xiang, H., Dolean, V., Spillane, N.: A coarse space construction based on local Dirichlet-to-Neumann maps. SIAM J. Sci. Comput. **33**(4), 1623–1642 (2011)
7. Saad, Y., Schultz, M.H.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**(3), 856–869 (1986)
8. Spillane, N., Dolean, V., Hauret, P., Nataf, F., Pechstein, C., Scheichl, R.: Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps. Numer. Math. **126**(4), 741–770 (2014)

A Comparison of Boundary Element and Spectral Collocation Approaches to the Thermally Coupled MHD Problem

Canan Bozkaya and Önder Türk

Abstract The thermally coupled full magnetohydrodynamic (MHD) flow is numerically investigated in a square cavity subject to an externally applied uniform magnetic field. The governing equations given in terms of stream function, vorticity, temperature, magnetic stream function, and current density, are discretized spatially using both the dual reciprocity boundary element method (DRBEM) and the Chebyshev spectral collocation method (CSCM) while an unconditionally stable backward difference scheme is employed for the time integration. Apart from the novelty of the methodology that allows the use of two different methods, the work aims to accommodate various characteristics related to the application of approaches differ in nature and origin. The qualitative and quantitative comparison of the methods are conducted in several test cases. The numerical simulations indicate that the effect of the physical controlling parameters of the MHD problem on the flow and heat transfer can be monitored equally well by both proposed schemes.

1 Introduction

Magnetohydrodynamics investigates the dynamics of electrically conducting fluids under the effect of magnetic fields. The MHD flow and heat transfer studies have attracted many researchers due to their wide range of engineering applications such as cooling systems, crystal growth, MHD generators, nuclear reactors, and electromagnetic pumps. The incompressible full MHD and energy equations involve the coupling of the Navier-Stokes equations of fluid dynamics with Maxwell's equations of electromagnetism through Ohm's law while the thermal coupling is performed by Boussinesq approximation. The resulting governing equations are

C. Bozkaya (✉)
Department of Mathematics, Middle East Technical University, Ankara, Turkey
e-mail: bcanan@metu.edu.tr

Ö. Türk
Department of Mathematics, Gebze Technical University, Kocaeli, Turkey

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_17

highly nonlinear due to the additional terms with the existence of Lorentz force, which allows the availability of analytical solutions only in some restricted circumstances. Hence, an extensive research is ongoing in establishing and developing effective numerical techniques which are applicable to the full MHD flow and heat transfer models. In many studies available in the literature, the magnetic Reynolds number is assumed to be so small that the induced magnetic field is neglected, [1–3]. However, it is well known that the magnetic induction should be taken into account in the mathematical model especially for large values of Hartmann number, [4]. There are several studies investigating the full MHD model in which the existence of external and internal magnetic fields is taken into account, [5–8]. On the other hand, one of the main difficulty in solving the full MHD flow numerically at the discrete level is to satisfy the divergence-free constraints for both the velocity and magnetic fields as well as the existence of pressure terms in the equations. Thus, various numerical models for the full MHD flow have been developed (see, e.g., [9] and the references therein).

The aim of this work is to present a comparative numerical analysis for the solution of thermally coupled full MHD flow in a square cavity by the use of two widely used methods, namely DRBEM and CSCM. An iterative approach that accommodates both techniques to discretize the full MHD flow given in a special mathematical model has been proposed. The governing equations are considered in the form of stream function-vorticity-magnetic induction-current density-temperature, so that the pressure gradient can be eliminated, and the divergence-free conditions for the velocity and the magnetic field are automatically satisfied through the application of the numerical methods. The qualitative and quantitative comparisons of the results obtained by DRBEM and the ones by CSCM (which are given in [8]) are conducted for several cases to investigate the effects of the problem physical parameters on the flow field and the temperature distribution.

2 Physical Problem and Mathematical Formulation

The unsteady, two-dimensional full MHD flow and heat transfer in a square cavity of width ℓ filled with an electrically conducting fluid is considered. A transverse uniform magnetic field of intensity B_0 in the positive y-direction is externally applied. The vertical walls of the cavity are assumed to be adiabatic while the horizontal upper and bottom walls are maintained at constant hot (T_h) and cold (T_c) temperatures, respectively. The flow generated inside the cavity obeys Boussinesq approximation, and the induced magnetic field is taken into account while the effects of Joule heating, viscous dissipation, displacement and convection currents are neglected. Thus, the unsteady non-dimensional governing equations of the full MHD flow in stream function ψ, magnetic induction A, current density j, temperature T and vorticity w are given as [8, 9]

$$
\begin{aligned}
&\Delta\psi = -w, \\
&\Delta A = -j, \\
&\Delta j = Re_m \frac{\partial j}{\partial t} - Re_m \Delta(u\frac{\partial A}{\partial x} + v\frac{\partial A}{\partial y}), \\
&\Delta T = PrRe\frac{\partial T}{\partial t} + PrRe(u\frac{\partial T}{\partial x} + v\frac{\partial T}{\partial y}), \\
&\Delta w = Re\left[\frac{\partial w}{\partial t} + u\frac{\partial w}{\partial x} + v\frac{\partial w}{\partial y} - \frac{Ra}{PrRe^2}\frac{\partial T}{\partial x} - \frac{Ha^2}{ReRe_m}(\frac{\partial A}{\partial y}\frac{\partial j}{\partial x} - \frac{\partial A}{\partial x}\frac{\partial j}{\partial y})\right],
\end{aligned}
\tag{1}
$$

by introducing the stream function and vorticity with $u = \partial\psi/\partial y$, $v = -\partial\psi/\partial x$ and $w = \partial v/\partial x - \partial u/\partial y$, (u, v) being the velocity field, and the magnetic stream function and the current density with $B_x = \partial A/\partial y$, $B_y = -\partial A/\partial x$, and $j = \partial B_y/\partial x - \partial B_x/\partial y$, (B_x, B_y) being the magnetic field. The dimensionless parameters are the Reynolds number ($Re = \ell u_0/\nu$), Prandtl number ($Pr = \nu/\alpha$), magnetic Reynolds number ($Re_m = \mu_m\sigma\ell u_0$), Rayleigh number ($Ra = g\beta\ell^3(T_h - T_c)/\alpha\nu$), and Hartmann number ($Ha = B_0\ell\sqrt{\sigma/\mu}$). Here, $\alpha, \nu, \mu, \mu_m, \beta, \sigma, u_0$, and g are the fluid thermal diffusivity, kinematic viscosity, dynamic viscosity, magnetic permeability, volume expansion coefficient, electrical conductivity, characteristic velocity, and the gravitational acceleration, respectively. Homogeneous initial conditions are imposed for all the unknowns at $t = 0$. The velocity at the upper wall is given by $(\hat{u}, 0)$ for a prescribed $\hat{u}$ while the other walls have zero velocity conditions. The temperature of the top and bottom walls are taken as $T = 0.5$ and $T = -0.5$, respectively, and on vertical walls the condition $\partial T/\partial n = 0$ is imposed. Since $B_x = 0$ and $B_y = 1$, the magnetic stream function is taken as $A = -x$ on all walls. On the other hand, the unknown boundary conditions of the vorticity and current density are calculated numerically with the use of the stream function and magnetic stream function equations, respectively, through the application of the numerical methods.

3 Numerical Methods

As already mentioned, the thermally coupled full MHD flow equations are discretized spatially by using two methods, namely the Chebyshev spectral collocation and dual reciprocity boundary element methods, both combined with an unconditionally stable backward difference scheme given by

$$\frac{\partial S}{\partial t}|^{n+1} = \frac{S^{n+1} - S^n}{\delta t} \,, \tag{2}$$

for the time integration, where n and δt are the time level and time step, respectively.

3.1 Application of CSCM

The CSCM discretization of the equations in (1) is based on requiring the numerical approximation of each unknown to be exactly satisfied on the abscissae of the extreme points of the Chebyshev polynomials defined as $x_i = \cos(i\pi/N), i = 0, 1, \ldots, N$. The method is of global nature; each function spans the whole domain under consideration and thus, the derivatives of the function depend on the entire discretization. The interpolating polynomials are differentiated analytically by means of the so-called Chebyshev spectral differentiation matrices. Utilization of these matrices in combination with the time integration scheme (2) results in the following CSCM and time discretized form of (1):

$$\begin{aligned}
&\hat{K}\psi^{n+1} = -w^n, \\
&\hat{K}A^{n+1} = -j^n, \\
&\left[I - \frac{\delta t}{Re_m}\hat{K}\right] j^{n+1} = j^n - \delta t \hat{K}\mathscr{P}(\psi^{n+1}, A^{n+1}), \\
&\left[I + \delta t \left(\mathscr{P}(\psi^{n+1}, \iota) - \frac{1}{Pr\,Ra}\hat{K}\right)\right] T^{n+1} = T^n, \\
&\left[I + \delta t \left(\mathscr{P}(\psi^{n+1}, \iota) - \frac{1}{Re}\hat{K}\right)\right] w^{n+1} = w^n + \delta t \frac{Ra}{Pr\,Re^2}\hat{D}^{(1)}T^{n+1} \\
&\qquad\qquad\qquad\qquad\qquad\qquad + \delta t \frac{Ha^2}{Re\,Re_m}\mathscr{P}(A^{n+1}, j^{n+1}).
\end{aligned} \tag{3}$$

In these equations, the $(N+1)^2 \times (N+1)^2$ matrix $\hat{K}$ is given as $\hat{K} = \hat{D}^{(2)} + \hat{E}^{(2)}$, where $\hat{D}^{(i)}$ and $\hat{E}^{(i)}$ are the Chebyshev differentiation matrices in $x-$ and $y-$ directions, respectively, and are defined with the use of the Kronecker product as $\hat{D}^{(i)} = I \otimes D^{(i)}$ and $\hat{E}^{(i)} = E^{(i)} \otimes I$, $i = 1, 2$, being the order. I is the identity matrix of order $(N+1)^2$, and ι is the vector of order $(N+1)^2$ whose all entries are 1. $\mathscr{P}(\hat{\phi}, \hat{\varphi})$ denotes the vector formed by multiplication of the approximations to the first partial derivatives of its argument vectors, and is defined as

$$\mathscr{P}(\hat{\phi}, \hat{\varphi}) = \mathscr{D}(\hat{E}^{(1)}\hat{\phi})\hat{D}^{(1)}\hat{\varphi} - \mathscr{D}(\hat{D}^{(1)}\hat{\phi})\hat{E}^{(1)}\hat{\varphi}.$$

$\mathscr{D}(\hat{\phi})$ denotes the diagonal matrix with the entries of a vector $\hat{\phi}$ on its diagonal. The resulting fully coupled nonlinear system of equations is solved iteratively incorporating the unknown boundary conditions of the vorticity and current density by means of the velocity and magnetic field components, respectively. The iterative steps are repeated until preassigned convergence criteria are met for a given tolerance for all the unknowns on the whole problem domain. For further details regarding the method and calculations, we refer to [8].

3.2 Application of DRBEM

The DRBEM aims to transform the governing equations (1) into boundary integral equations by using the fundamental solution of the Laplace equation, $u^* = 1/2\pi \ln(1/r)$, and treating the terms on the right hand sides (rhs) of these equations as the non-homogeneity. Thus, Eqs. (1) are weighted by u^* and the application of Green's second identity results in, [10],

$$c_i S_i + \int_{\Gamma} (q^* S - u^* \frac{\partial S}{\partial n}) d\Gamma = - \int_{\Omega} b_s u^* d\Omega, \tag{4}$$

where S is used for each unknown ψ, A, j, T, w. Here, $q^* = \partial u^*/\partial n$, Γ is the boundary of the domain Ω, and the constant $c_i = \phi_i/2\pi$ with the internal angle ϕ_i at the source point i. All the terms on the rhs of Eqs. (1) denoted by b_S, are approximated by a set of radial basis functions $f_j (= 1 + r_j)$ linked with the particular solutions $\hat{u}_j$ of $\Delta \hat{u}_j = f_j$, [10]. That is, these approximations are given by $b_S \approx \sum_{j=1}^{N+L} \alpha_{S_j} f_j = \sum_{j=1}^{N+L} \alpha_{S_j} \Delta \hat{u}_j$ where α_{S_j} are undetermined coefficients, N and L are the number of boundary and interior nodes, respectively. When Green's identity is applied to the rhs as well, and the boundary is discretized with constant elements, the matrix-vector form of Eq. (4) can be expressed as

$$HS - G\frac{\partial S}{\partial n} = (H\hat{U} - G\hat{Q})F^{-1}b_S, \tag{5}$$

where the components of matrices H and G are calculated by integrating q^* and u^*, respectively, over each boundary element. The matrices $\hat{U}$, $\hat{Q}$ and F take the vectors $\hat{u}_j$, $\hat{q}_j$ and f_j of sizes $(N + L)$ as their columns, respectively. When the backward finite difference given in Eq. (2) is applied to approximate the time derivatives in Eq. (5), the DRBEM system of algebraic equations takes the form

$$H\psi^{n+1} - G\psi_q^{n+1} = -Cw^n \ , \tag{6}$$

$$HA^{n+1} - GA_q^{n+1} = -Cj^n \ , \tag{7}$$

$$(H - \frac{PrRe}{\delta t}C - PrReCK)T^{n+1} - GT_q^{n+1} = -\frac{1}{\delta t}PrReCT^n \tag{8}$$

$$(H - \frac{Re_m}{\delta t}C)j^{n+1} - Gj_q^{n+1} = -\frac{1}{\delta t}Re_mCj^n - Re_mC\Delta KA^{n+1} \tag{9}$$

$$\begin{aligned}(H - \frac{Re}{\delta t}C - ReCK)w^{n+1} - Gw_q^{n+1} &= -\frac{Re}{\delta t}Cw^n - \frac{Ra}{PrRe}CD_xT^{n+1} \\ &- \frac{Ha^2}{Re_m}C(D_yA^{n+1}D_xj^{n+1} - D_xA^{n+1}D_yj^{n+1}),\end{aligned} \tag{10}$$

where $C = (H\hat{U} - G\hat{Q})F^{-1}$, $K = u^{n+1}D_x + v^{n+1}D_y$, $D_x = \partial F/\partial x F^{-1}$ and $D_y = \partial F/\partial y F^{-1}$. The resulting system of coupled equations is solved iteratively with the initial estimates of w, j, T. In each time level, the required space derivatives of S, the boundary conditions of the vorticity and current density are obtained by using F as

$$\frac{\partial S}{\partial x} = D_xS,\ \frac{\partial S}{\partial y} = D_yS,\ w = -(\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2})F^{-1}\psi,\ j = -(\frac{\partial^2 F}{\partial x^2} + \frac{\partial^2 F}{\partial y^2})F^{-1}A\ .$$

4 Results and Discussions

The thermally coupled full MHD flow in a square cavity is investigated under the effect of a vertically applied uniform magnetic field. The numerical simulations with the CSCM and DRBEM are carried out to investigate the effect of various combinations of problem parameters Re, Ha and Re_m at a moderate $Ra = 10^4$ and $Pr = 0.1$. The boundaries of the cavity with side length $\ell = 1$ are discretized by using $N = 50$ nodes and constant boundary elements along one side of the cavity, respectively, in CSCM and DRBEM, while a constant time step $\delta t = 0.25$ is used in both methods. The stopping criteria of the iterative schemes is set to be 10^{-5} for all the unknowns, and the solutions in regard to this criteria are referred as the steady-state solutions. We specifically consider the regularized lid-driven cavity flow with a moving upper wall whose velocity is given as $\hat{u} = 4x^2(1 - x^2)$.

First, the validation of the present methods is performed by solving the full MHD flow in a regularized lid-driven cavity subject to a transverse magnetic field in the absence of heat sources. Table 1 shows that the results obtained by DRBEM and CSCM are quantitatively in good agreement with the ones given in [5, 9] in terms of the values of ψ, location of primary vortex, and extrema of magnetic field intensity.

Table 1 Characteristics of the primary vortex and the magnetic field: $Re = Re_m = 100$, $Ha = 10$

	Primary vortex			Magnetic field intensity			
	ψ	x	y	$\min(B_x)$	$\max(B_x)$	$\min(B_y)$	$\max(B_y)$
Bozkaya [5]	−0.07346	0.6719	0.7656	−0.9067	1.8729	−0.1130	2.0792
Yu [9]	−0.07354	0.6641	0.7656	−0.9092	1.8989	−0.1093	2.0789
CSCM [8]	−0.07324	0.6545	0.7679	−0.8988	1.9093	−0.1120	2.0751
DRBEM	−0.07197	0.6643	0.7643	−0.9094	1.9190	−0.1292	2.0817

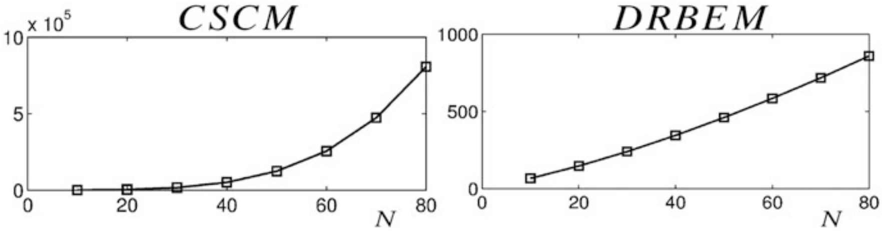


Fig. 1 Condition number in 2-norm versus the number of nodes N in CSCM and DRBEM along one side of the cavity

Figure 1 displays the variation of the condition number of the resulting CSCM and DRBEM coefficient matrices for the discretized system of stream function equation with respect to the number of nodes N when $Ha = 50$, $Re = 400$ and $Re_m = 100$. Although the matrices are dense in both methods, the condition number is very large giving ill-conditioned matrices in spectral method when compared to the one in DRBEM for large values of N, which is a well-known characteristics of collocation methods. Moreover, for larger N, the condition number in CSCM increases faster than it does in DRBEM. However, in this problem we obtain systems of sizes that remain in solvable ranges which can be handled by both CSCM and DRBEM.

The effects of Re on the flow, vorticity and temperature distribution are visualized in Fig. 2 when $Ha = 25$ and $Re_m = 1$. It is well-observed that the results of CSCM and DRBEM are quite compatible. A circular vortex formed at the upper right corner of the cavity due to the motion of upper lid, moves towards the center of cavity with an increasing magnitude as Re increases from 100 to 1000. Vorticity contours are concentrated mainly close to the upper and right walls, and form a boundary layer at $Re = 1000$. As Re increases, the isotherms change their profiles due the strong temperature gradients indicating that the heat transfer is dominated by convection.

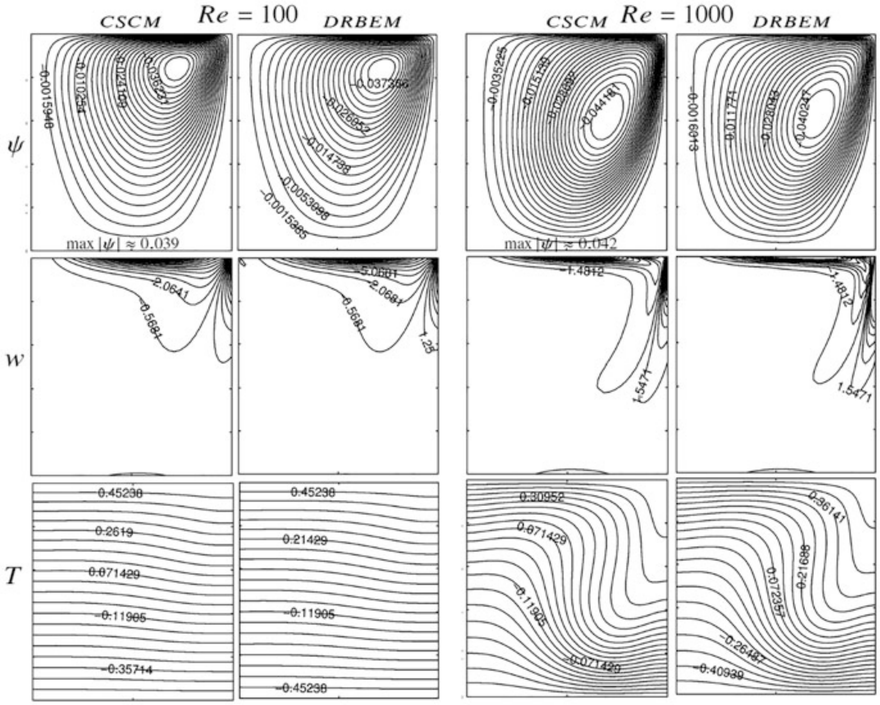


Fig. 2 Effect of $Re(= 100, 1000)$ on ψ, w and T: $Ra = 10^4$, $Ha = 25$, $Re_m = 1$

Finally, the magnetic streamlines and the contours of current density obtained by CSCM and DRBEM are drawn in Fig. 3 to analyze the effect of magnetic Reynolds number $Re_m(= 1, 100)$ when $Ra = 10^4$, $Ha = 25$, $Re = 100$. Both methods give similar results in each case. The magnetic streamlines extended vertically in the same direction of the applied magnetic field at low Re_m, are distorted by forming a prominent circulation in the region close to the upper right corner at $Re_m = 100$. On the other hand, an increase in Re_m results in a rise in the magnitude of the current density although they have similar profiles for each Re_m.

5 Conclusion

A numerical model which is divergence-free of magnetic field is proposed for solving thermally coupled unsteady incompressible full MHD equations. Two different techniques, namely, CSCM and DRBEM, coupled with a backward difference time integration, have been shown to accurately represent the solution of the physical model. The numerical simulations have demonstrated that both of

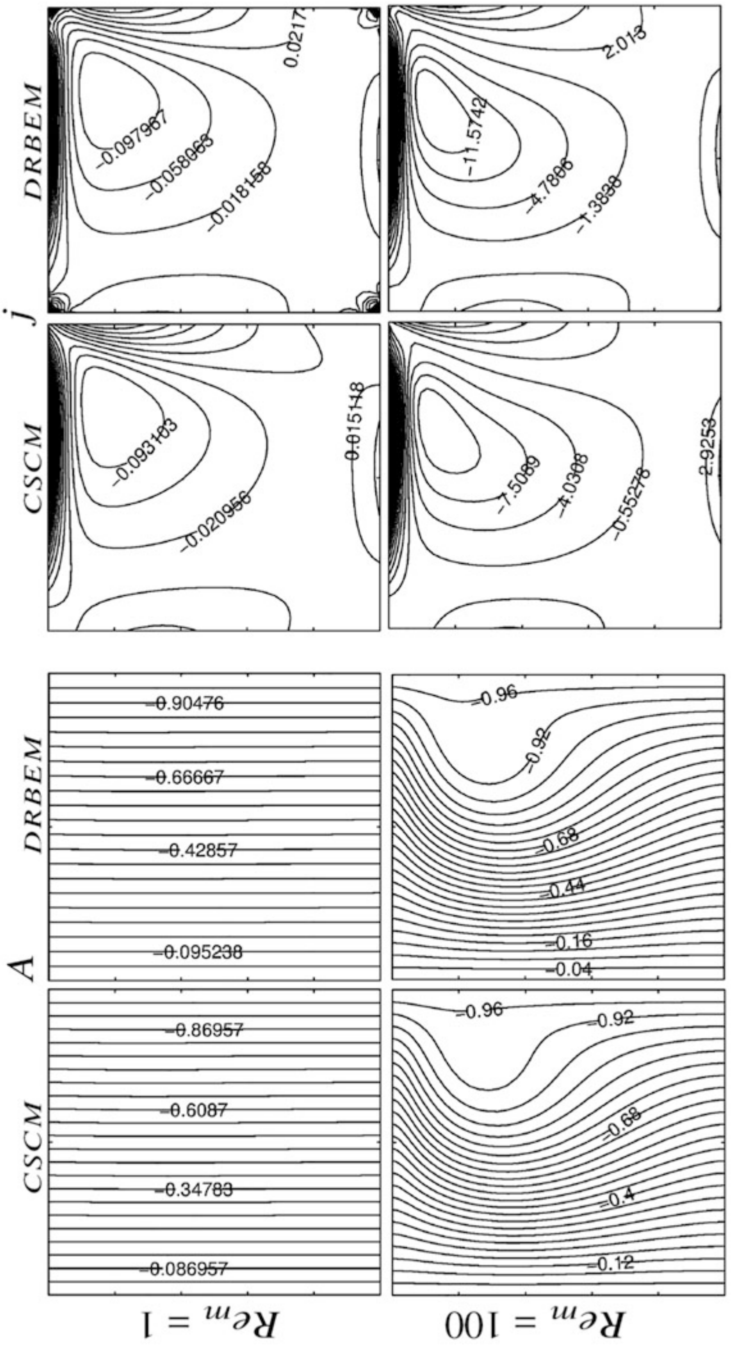


Fig. 3 Effect of $Re_m(= 1, 100)$ on A and j: $Ra = 10^4$, $Ha = 25$, $Re = 100$

the present approaches are accurate and reliable, and have the ability to solve the full MHD problems in a reasonably wide range of the problem parameters.

References

1. Oztop, H. F., Al-Salem, K., Pop, I.: MHD mixed convection in a lid-driven cavity with corner heater. Int J Heat Mass Tran, **54**, 3494–3504 (2011)
2. Mramor, K., Vertnik, R., Sărler, B.: Simulation of natural convection influenced by magnetic field with explicit local radial basis function collocation method. CMES- Comp Model Eng, **92**, 327–352 (2013)
3. Türk, Ö., Tezer-Sezgin, M.: Natural convection flow under a magnetic field in an inclined square enclosure differentially heated on adjacent walls. Int J Numer Method H, **23**, 844–866 (2013)
4. Sarris, I.E., Zikos, G.K., Grecos, A.P., Vlachos, N.S.: On the limits of validity of the low magnetic Reynolds number approximation in MHD natural convection heat transfer. Numer Heat Tr B-Fund, **50**, 157–180 (2006)
5. Bozkaya, N., Tezer-Sezgin, M.: The DRBEM solution of incompressible MHD flow equations. Int J Numer Meth Fl, **67**, 1264–1282 (2011)
6. Codina, R., Hernández, N.: Approximation of the thermally coupled MHD problem using a stabilized finite element method. J Comput Phys, **230**, 1281–1303 (2011)
7. Sivakumar, R., Vimala S., Sekhar, T. V. S.: Influence of induced magnetic field on thermal MHD flow. Numer Heat Tr A-Appl, **68**, 797–811 (2015)
8. Türk, Ö.: Chebyshev spectral collocation method approximation to thermally coupled MHD equations. SDU J Nat Appl Sci, **22**, 355–366 (2018)
9. Yu, P.X., Tian, Z.F., Ying A.Y., Abdou, M.A.: Stream function-velocity-magnetic induction compact difference method for the 2D steady incompressible full magnetohydrodynamic equations. Comput Phys Commun, **219**, 45–69 (2017)
10. Brebbia C.A., Partridge P.W., Wrobel L.C.: The Dual Reciprocity Boundary Element Method. Computational Mechanics Publications, Southampton, Boston (1992)

Minimal Sets of Unisolvent Weights for High Order Whitney Forms on Simplices

Ana Alonso Rodríguez, Ludovico Bruni Bruno, and Francesca Rapetti

Abstract Whitney forms—degree one *trimmed* polynomials—are a crucial tool for finite element analysis of electromagnetic problem. They not only induce several finite element methods, but they also bear interesting geometrical features. If, on the one hand, features of degree one elements are well understood, when it comes to higher degree elements one is forced to choose between an analytical approach and a geometric one, that is, the duality that holds for the lower degree gets lost. Using tools of finite element exterior calculus, we show a correspondence between the usual basis of a high order Whitney forms space and a subset of the *weights*, that is, degrees of freedom obtained by integration over subsimplices of the mesh.

1 High Order Whitney Forms

In what follows $T \doteq [\mathbf{x}_0, \ldots, \mathbf{x}_n]$ denotes a non degenerate oriented n-simplex. Its (oriented) k-subsimplices are in a bijective correspondence with the subsets of $k+1$ elements so they are $\binom{n+1}{k+1}$. Any ordered listing of $k+1$ vertices of T yields a bijective map $\sigma \mapsto f_\sigma \doteq [\mathbf{x}_{\sigma(0)}, \ldots, \mathbf{x}_{\sigma(k)}]$. We denote by $\Delta_k(T)$ the set of (oriented) k-subsimplices of T.

With each point $\mathbf{P} \in T$ we may associate a $(n+1)$-uple $(\lambda_0, \lambda_1, \ldots, \lambda_n)$ such that $\mathbf{P} = \sum_{i=0}^{n} \lambda_i \mathbf{x}_i$, with the constraints $\sum_{i=0}^{n} \lambda_i = 1$ and $\lambda_i \geq 0$. We call such functions *barycentric coordinates* for $\mathbf{P}$. Similarly (see [3]) we may associate with

A. A. Rodríguez · L. B. Bruno (✉)
Dipartimento di Matematica, Università degli Studi di Trento, Povo, Trento, Italy
e-mail: ana.alonso@unitn.it; ludovico.brunibruno@unitn.it

F. Rapetti
Départment de Mathématiques, Université Côte d'Azur, Nice, France
e-mail: francesca.rapetti@univ-cotedazur.fr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_18

each subsimplex its *Whitney form*

$$\omega_{f_\sigma} \doteq \sum_{i=0}^{k} (-1)^i \lambda_{\sigma(i)} d\lambda_{\sigma(0)} \wedge \ldots \wedge \widehat{d\lambda_{\sigma(i)}} \wedge \ldots \wedge d\lambda_{\sigma(k)},$$

where we take σ as an increasing permutation. Consequently, we have a map

$$f_\sigma \mapsto \omega_{f_\sigma},$$

which is known as the Whitney map (see [8]), and provides a relationship between the simplicial homology of a simplicial complex and the de Rham cohomology of that complex. Observe that if $f_\sigma,\ f_{\sigma'} \in \Delta_k(T)$ and $\sigma \neq \sigma'$ then

$$\omega_{f_\sigma}\Big|_{f_{\sigma'}} = 0, \tag{1}$$

since if $i \in \{0, \ldots, k\}$ is such that $\mathbf{x}_{\sigma(i)} \notin f_{\sigma'}$, both $\lambda_{\sigma(i)}$ and $d\lambda_{\sigma(i)}$ vanish on $f_{\sigma'}$. Moreover it is known (see [7]) that such forms are closed when restricted to the subsimplex they are generated by, that is, $d\omega_{f_\sigma}\Big|_{f_\sigma} = 0$.

Whitney forms of order k form a vector space, which may be characterized in terms of the *Koszul differential* (see [1]), which acts as the contraction of a differential form $\omega \in \Lambda^k(\mathbb{R}^n)$ with the identity vector field $\mathbf{X}$, that is

$$\kappa\omega(\mathbf{v}_1, \ldots, \mathbf{v}_{k-1}) \doteq \omega(\mathbf{X},\ \mathbf{v}_1, \ldots, \mathbf{v}_{k-1}),$$

and which is assumed to map smooth functions to 0.

For $r \geq 0$ and $k \in \{0, \ldots, n\}$ let us denote by $\mathcal{P}_r\Lambda^k(\mathbb{R}^n)$ the space of polynomial differential k-forms of degree r. For $k > 0$ the spaces of *trimmed* polynomial differential k-forms of degree r are defined

$$\mathcal{P}_r^-\Lambda^k(\mathbb{R}^n) \doteq \{\omega \in \mathcal{P}_r\Lambda^k(\mathbb{R}^n) \mid \kappa\omega \in \mathcal{P}_r\Lambda^{k-1}(\mathbb{R}^n)\}, \tag{2}$$

while $\mathcal{P}_r^-\Lambda^0(\mathbb{R}^n) \doteq \mathcal{P}_r\Lambda^0(\mathbb{R}^n)$. Their elements are the so called *Whitney forms of higher degree*. One has (see [1]) $\dim \mathcal{P}_r^-\Lambda^k(\mathbb{R}^n) = \binom{r+k-1}{k}\binom{n+r}{n-k}$.

For $r > 0$ the following decomposition holds (see [2]):

$$\mathcal{P}_r^-\Lambda^k(\mathbb{R}^n) = \mathcal{P}_{r-1}\Lambda^k(\mathbb{R}^n) \oplus \kappa\mathcal{H}_{r-1}\Lambda^k(\mathbb{R}^n), \tag{3}$$

being $\mathcal{H}_{r-1}\Lambda^k(\mathbb{R}^n)$ the space of homogeneous polynomial differential k-forms of degree $r-1$. Spaces $\mathcal{P}_r^-\Lambda^k(T)$ are then defined by pulling back with respect to the inclusion map $T \hookrightarrow \mathbb{R}^n$, and from now on we will consider just such spaces.

It is easy to check that (3) implies the following result.

Lemma 1 *If $\omega \in \mathcal{P}_r^-\Lambda^k(T)$ is a closed form, then $\omega \in \mathcal{P}_{r-1}\Lambda^k(T)$.*

A crucial aspect of Whitney forms is that they provide a basis for the case $r = 1$ of the above spaces (see [1]).

Theorem 1 *Whitney forms associated with the k-subsimplices of T are a basis for $\mathcal{P}_1^- \Lambda^k(T)$.*

Property (1) suggests as degrees of freedom for Whitney forms the *weights*

$$\omega_{f_\sigma} \mapsto \int_{f_\sigma} \omega_{f_\sigma} . \tag{4}$$

Concerning high order Whitney forms, in [4] it has been proved that for $r > 0$

$$\mathcal{P}_{r+1}^- \Lambda^k(T) = \mathcal{P}_r(T) \cdot \mathcal{P}_1^- \Lambda^k(T) . \tag{5}$$

Denote by $\mathcal{I}(n+1, r)$ the collection of multi-indices $\boldsymbol{\alpha} = (\alpha_0, \ldots, \alpha_n)^T$ of weight r, and by $\lambda^{\boldsymbol{\alpha}} \doteq \Pi_{i=0}^n \lambda_i^{\alpha_i}$. In view of (5), the set $\{\lambda^{\boldsymbol{\alpha}} \omega_{f_\sigma} \ : \ \boldsymbol{\alpha} \in \mathcal{I}(n+1, r)$ and $f_\sigma \in \Delta_k(T)\}$ provides a system of generators for $\mathcal{P}_{r+1}^- \Lambda^k(T)$ but not a basis. A wise way to get rid of redundant objects consists in considering, for each $f_\sigma \in \Delta_k(T)$, a subset of multi-indices $\widetilde{\mathcal{I}}_\sigma(n+1, r) \doteq \{\boldsymbol{\alpha} \in \mathcal{I}(n+1, r) \ : \ \lambda_i = 0 \ \forall i < \sigma(0)\}$. A mnemonic rule to visualize this is the following: if f_σ is the subsimplex associated with the permutation σ, then we have to discard all the possible $\boldsymbol{\alpha}$ that have a non-zero entry in a position smaller than $\sigma(0)$. For instance, if $f_\sigma = [1, 2]$, then the multi-indices $\boldsymbol{\alpha}$ whose first entry is not empty are to be discarded, as the associated $\lambda^{\boldsymbol{\alpha}} \omega_{f_\sigma}$ turn out to be linearly dependent from other elements of the basis. We have in fact the following (see [1]):

Theorem 2 *The set $\{\lambda^{\boldsymbol{\alpha}} \omega_{f_\sigma} \ : \ f_\sigma \in \Delta_k(T)$ and $\boldsymbol{\alpha} \in \widetilde{\mathcal{I}}_\sigma(n+1, r)\}$ is a basis for $\mathcal{P}_{r+1}^- \Lambda^k(T)$.*

A dual question is to investigate how degrees of freedom change for high order Whitney forms. If on the one hand moments, in the classical sense of [6], have been proved to be unisolvent (see [5]), on the other they lack of geometrical meaning. Thus, we aim to adapt the definition of weights, which for obvious dimensional reasons does not follow from the case of $r = 1$. The solution proposed in [7] and [4] consists in introducing of a subdivision of T into *small simplices*, which basically provide a particular subtriangulation of the simplex T one deals with.

2 Small Simplices

The construction of the small simplices needs some auxiliary results.

Definition 1 (Principal Lattice $\Sigma_r(T)$) Let $T \subseteq \mathbb{R}^n$ be a simplex and $r \in \mathbb{N}$. Let $\{\lambda_i\}$ be the set of barycentric coordinates for T. We define the principal lattice of order r as the set of points

$$\Sigma_r(T) \doteq \left\{ \mathbf{x} \in T \mid \lambda_i(\mathbf{x}) \in \left\{ 0, \frac{1}{r+1}, \dots, \frac{r}{r+1}, 1 \right\} \text{ for each } i \in \{0, \dots, n\} \right\}. \tag{6}$$

For each r, $\Sigma_r(T)$ consists of $\binom{n+r+1}{n}$ points, where n is the dimension of the simplex T. Thus, an enumeration of those points yields a bijective association with a basis of $\mathcal{P}_{r+1}(T)$ and it is a well known matter of linear algebra to conclude that a polynomial $p(\mathbf{x})$ of degree $r+1$ that vanishes for each $\mathbf{x} \in \Sigma_r(T)$ for some n-simplex T is identically zero.

Small simplices may be characterized in terms of their principal lattice (see [7]).

Definition 2 (Small Simplices $\Sigma_r^k(T)$) Let $T \subseteq \mathbb{R}^n$ be a simplex and $\Sigma_r(T)$ be its principal lattice. We define the set of small simplices of order n and degree r as the collection of all the n-simplices that are $\frac{1}{r+1}$-homothetic to T and whose vertices belong to $\Sigma_r(T)$. We denote such a set by $\Sigma_r^n(T)$. Formally, $\Sigma_r^n(T) \doteq \{s_{(\boldsymbol{\alpha},T)} : \boldsymbol{\alpha} \in \mathcal{I}(n+1,r)\}$, being

$$s_{(\boldsymbol{\alpha},T)} \doteq \left\{ \frac{1}{r+1}\mathbf{x} + [\mathbf{x}_0, \dots, \mathbf{x}_n]\boldsymbol{\alpha} \ : \ \mathbf{x} \in T \right\}.$$

For $k \in \{0, 1, \dots, n-1\}$ the set of small simplices $\Sigma_r^k(T)$ of order k and degree r is the collection of all the k-subsimplices of all the elements of $\Sigma_r^n(T)$.

From the perspective of degrees of freedom the unisolvence of the set $\Sigma_r^k(T)$ for $\mathcal{P}_r^- \Lambda^k(T)$ has been proved for each $k = 0, 1, \dots, n$. We have in fact the following.

Proposition 1 *Let $\omega \in \mathcal{P}_r^- \Lambda^k(T)$. If $\int_{s_{(\boldsymbol{\alpha},f_\sigma)}} \omega = 0$ for each $s_{(\boldsymbol{\alpha},f_\sigma)} \in \Sigma_r^k(T)$, then $\omega = 0$.*

For a complete proof, which is a bit technical, we address the interested reader to [4]. We here recall just the main ingredients.

Observe that every $\omega \in \mathcal{P}_{r+1}^- \Lambda^n(T)$ is closed, hence by Lemma 1 it in fact belongs to $\mathcal{P}_r \Lambda^n(T)$.

Let $\tau_{\boldsymbol{\xi}}$ denote the translation by the vector $\boldsymbol{\xi}$: $(\tau_{\boldsymbol{\xi}} u)(\mathbf{x}) = u(\mathbf{x} - \boldsymbol{\xi})$. Since the map $\boldsymbol{\xi} \to \int_T \tau_{\boldsymbol{\xi}} \omega$ is a polynomial of degree r which is zero at the points of the principal lattice $\Sigma_{r-1}(T')$ of the n-simplex T' in Fig. 1, it is zero everywhere. By Lemma 3.12 of [4] $\omega = 0$.

We also remark that for $k = 0$ equality can be restored by avoiding double counting of points. In this case integrals assume the meaning of evaluations and we thus fall into the preceding case.

Intermediate cases are trickier, as the closedness of ω is not a priori granted. However, one may work inductively in a descending way. For $k = n-1$, it follows from Stokes' Theorem that $\int_{\partial S} \omega = \int_S d\omega$ and hence, being able to write the boundary of each $S \in \Sigma_r^k(T)$, one gets that $d\omega \in \mathcal{P}_r^- \Lambda^k(T)$ is in fact closed and it is now possible to reproduce more or less the same proof as before. Then one works by induction. This makes clear that one is not able to leave Stokes' Theorem and some hypotheses on the boundaries of the subtriangulation out of consideration.

3 A Minimal Set of Unisolvent Weights

As one may readily check, the set $\Sigma_r^k(T)$ bears some redundancy, since

$$\#\Sigma_r^k(T) \geq \dim \mathcal{P}_r^- \Lambda^k(T), \tag{7}$$

where equality holds just when $k = n$. It is natural to investigate whether it is possible to extract a unisolvent subfamily of $\Sigma_r^k(T)$ or to build a new one which is minimal in the sense that (7) becomes an equality for each k. The answer to both questions is yes: in the following we prove the second fact and show how the first can be deduced.

We build a new family of "subsimplices" of T (in the sense that such subsimplices are topologically contained in T) as follows. For each $\boldsymbol{\alpha} \in \mathcal{I}(n+1,r)$ we define the n-simplex

$$\tilde{s}_{(\boldsymbol{\alpha},T)} \doteq \left\{ \frac{1+\alpha_0}{r+1}\mathbf{x} + [\,\mathbf{x}_1,\ldots,\mathbf{x}_n]R(\boldsymbol{\alpha}) \;:\; \mathbf{x} \in T \right\}.$$

with $R(\boldsymbol{\alpha}) = (\alpha_1,\ldots,\alpha_n)^T$.

They are homothetic to T (with different ratios, from $\frac{1}{r+1}$ to 1) and their vertices belong to $\Sigma_r(T)$. Moreover they have n vertices on the principal lattice of F_0, the face opposite to the vertex $\mathbf{x}_0$ (see Fig. 1).

We denote

$$\widetilde{\Sigma}_r^n(T) \doteq \{\tilde{s}_{(\boldsymbol{\alpha},T)} \;:\; \boldsymbol{\alpha} \in \mathcal{I}(n+1,r)\}\,.$$

For $k \in \{0,1,\ldots,n\}$ the set $\widetilde{\Sigma}_r^k(T)$ is the collection of all the k-subsimplices of all the elements of $\widetilde{\Sigma}_r^n(T)$, namely, if $f_\sigma \in \Delta_k(T)$ and

$$\tilde{s}_{(\boldsymbol{\alpha},f_\sigma)} \doteq \left\{ \frac{1+\alpha_0}{r+1}\,\mathbf{x} + [\mathbf{x}_1,\ldots,\mathbf{x}_n]R(\boldsymbol{\alpha}) \;:\; \mathbf{x} \in f_\sigma \right\},$$

then $\widetilde{\Sigma}_r^k(T) = \{\tilde{s}_{(\boldsymbol{\alpha},f_\sigma)} \;:\; f_\sigma \in \Delta_k(T) \text{ and } \boldsymbol{\alpha} \in \mathcal{I}(n+1,r)\}$.

Remark 1 If $f_\sigma \subset F_0$ then $\{\tilde{s}_{(\boldsymbol{\alpha},f_\sigma)} \;:\; \boldsymbol{\alpha} \in \mathcal{I}(n+1,r)\} \subset F_0$.

Observe that since $\boldsymbol{\alpha}$ ranges over $\mathcal{I}(n+1,r)$ we have not removed any redundancy. Therefore, it is just a matter of adapting Proposition 1 to see that also the following result holds true.

Proposition 2 *Let $\omega \in \mathcal{P}_r^- \Lambda^k(T)$ be such that $\int_{\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)}} \omega = 0$ for each $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}_r^k(T)$. Then $\omega = 0$.*

It is worth noting that Lemma 3.12 of [4] can be generalized to a mapping $\rho_{\boldsymbol{\xi}}$ which is not just a translation but also contains a scaling:

$$(\rho_{\boldsymbol{\xi}} u)(\mathbf{x}) = u \left(\frac{1 + \xi_0}{r + 1} \mathbf{x} + (0, \xi_1, \dots, \xi_n)^T \right).$$

Since the map $\boldsymbol{\xi} \to \int_T \rho_{\boldsymbol{\xi}} \omega$ is a polynomial of degree r which is zero at the points of the principal lattice $\Sigma_{r-1}(T'')$ of the n-simplex T'' in Fig. 1, it is zero everywhere and then by the extension of Lemma 3.12 of [4] $\omega = 0$.
The rest of the proof can be carried out similarly to that of Proposition 1.

In order to restore the cardinality equality (7), for $0 < k < n$ we define $\widetilde{\Sigma}_{r,min}^k(T) \subsetneq \widetilde{\Sigma}_r^k(T)$ in the following way:

$$\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}_{r,min}^k(T) \Leftrightarrow \alpha_i = 0 \text{ for all } i \text{ such that } 1 \le i \le \sigma(0)\,.$$

A dimension count shows that

$$\#\widetilde{\Sigma}_{r,min}^k(T) = \dim \mathcal{P}_r^- \Lambda^k(T).$$

Roughly speaking, $\widetilde{\Sigma}_{r,min}^n(T)$ coincides with $\widetilde{\Sigma}_r^n(T)$ and contains the collection of n-simplices that are $\frac{j}{r+1}$ homothetic to T (for $j = 1, \dots, r + 1$) that have n vertices belonging to F_0, the face opposite to the vertex $\mathbf{x}_0$. On the other hand, consider the face $e_1 \subset F_0$ which is opposite to $\mathbf{x}_1$. One may think of $\widetilde{\Sigma}_{r,min}^{n-1}(T)$, as the set that contains two kind of elements (see Fig. 2):

- $f \in \partial \widetilde{\Sigma}_r^n(T)$ such that $\text{int}(f) \cap \text{int}(F_0) = \emptyset$;
- $(n - 1)$-simplices in F_0 that have $n - 1$ vertices on $\Sigma_r(e_1)$ and that are $\frac{l}{r+1}$-homothetic to F_0 for some $l = 1, \dots, r + 1$.

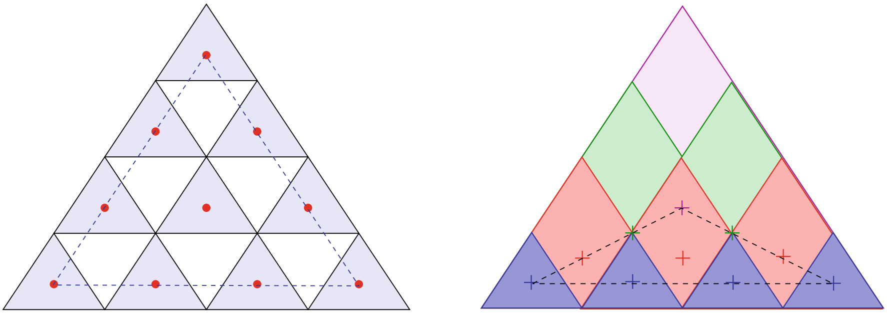


Fig. 1 On the left the elements of $\Sigma_3^2(T)$ and the vertices of the principal lattice of T'. On the right the elements of $\widetilde{\Sigma}_3^2(T)$ and the vertices of the principal lattice of T''. In this example $n = 2$

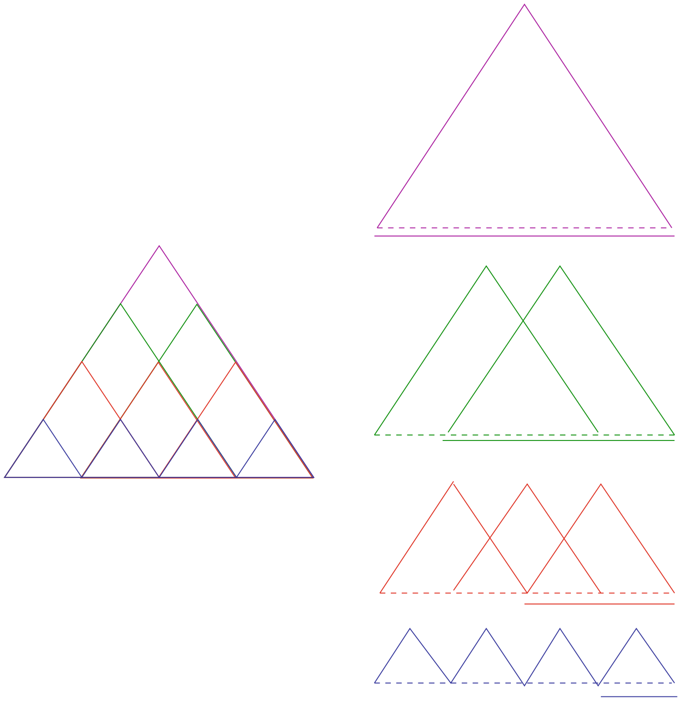


Fig. 2 On the left the elements of $\Sigma^1_3(T)$, on the right the elements of $\widetilde{\Sigma}^1_{3,min}(T)$. In this example $n = 2$

Note that $\widetilde{\Sigma}^{n-1}_r(T)$ also contains two kinds of elements: $f \in \partial\widetilde{\Sigma}^n_r(T)$ such that $\mathrm{int}(f) \cap \mathrm{int}(F_0) = \emptyset$, and $(n-1)$-simplices in F_0 with vertices on $\Sigma_r(F_0)$ and that are $\frac{l}{r+1}$-homothetic to F_0 for some $l = 1, \ldots, r+1$. Hence they differ on those $(n-1)$-simplices in F_0 that have not $n-1$ vertices on $\Sigma_r(e_1)$.

The generalization to each $k = n-2, \ldots, 1$ follows by recursion, whereas for $k = 0$ one just takes the remaining 0-simplices.

We need two auxiliary results: for any $f_\sigma \in \Delta_k(T)$ and $\boldsymbol{\beta} \in \mathcal{I}(k+1, r)$ we put

$$\tilde{s}_{\ell,(\boldsymbol{\beta}, f_\sigma)} \doteq \left\{ \frac{1+\beta_0}{r+1}\,\mathbf{x} + [\mathbf{x}_{\sigma(1)}, \ldots, \mathbf{x}_{\sigma(k)}]R(\boldsymbol{\beta}) \;:\; \mathbf{x} \in f_\sigma \right\}.$$

Let us also define $\widetilde{\Sigma}^k_r(f_\sigma) \doteq \{\tilde{s}_{\ell,(\boldsymbol{\beta}, f_\sigma)} \;:\; \boldsymbol{\beta} \in \mathcal{I}(k+1, r)\}$. The following result is a corollary of Proposition 2.

Lemma 2 *Let* $\omega \in \mathcal{P}^-_r\Lambda^k(f_\sigma)$ *be such that* $\int_{\tilde{s}_{\ell,(\boldsymbol{\beta}, f_\sigma)}} \omega = 0$ *for each* $\tilde{s}_{\ell,(\boldsymbol{\beta}, f_\sigma)} \in \widetilde{\Sigma}^k_r(f_\sigma)$*. Then* $\omega = 0$.

It is then easy to prove the following.

Lemma 3 *Let $\omega \in \mathcal{P}_r^- \Lambda^k(T)$ be such that $\int_{\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)}} \omega = 0$ for each $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}^k_{r,min}(T)$. Then $\omega\big|_{f_\sigma} = 0$ for all $f_\sigma \in \Delta_k(T)$.*

Proof It is enough to prove that $\widetilde{\Sigma}^k_r(f_\sigma) \subset \widetilde{\Sigma}^k_{r,min}(T)$ for all $f_\sigma \in \Delta_k(T)$. In fact, given $f_\sigma \in \Delta_k(T)$ and $\boldsymbol{\beta} \in \mathcal{I}(k+1, r)$, we consider the following $\boldsymbol{\alpha} \in \mathcal{I}(r+1, n)$

$$\alpha_j = \begin{cases} \beta_0 \text{ if } j = 0 \\ \beta_i \text{ if } j = \sigma(i)\; 1 \le i \le k \\ 0 \quad \text{otherwise.} \end{cases}$$

Then $\tilde{s}_{\ell,(\boldsymbol{\beta}, f_\sigma)} = \tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}^k_{r,min}(T)$ because $\alpha_i = 0$ for all i such that $1 \le i \le \sigma(0)$. □

We are now in a position to prove that the set of weights on the elements of $\widetilde{\Sigma}^k_{r,min}(T)$ is a minimal set of unisolvent degrees of freedom for $\mathcal{P}^-_{r+1}\Lambda^k(T)$.

Theorem 3 *If $\omega \in \mathcal{P}_r^- \Lambda^k(T)$ is such that $\int_{\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)}} \omega = 0$ for each $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}^k_{r,min}(T)$, then $\omega = 0$.*

Proof We will prove that if $\int_{\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)}} \omega = 0$ for each $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}^k_{r,min}(T)$, then $\int_{\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)}} \omega = 0$ for each $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}^k_r(T)$.

First we notice that $\widetilde{\Sigma}^k_r(T) \setminus \widetilde{\Sigma}^k_{r,min}(T) \subset \{f_\sigma \in \Delta_k(T) : \sigma(0) > 0\}$. In fact, $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}^k_r(T) \setminus \widetilde{\Sigma}^k_{r,min}(T)$ if and only if there exists i such that $1 \le i \le \sigma(0)$ and $\alpha_i \ne 0$. If $\sigma(0) = 0$ such an i does not exist. Hence $f_\sigma \subset F_0$ and by Remark 1 $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \subset F_0$.

From Proposition 3, $\omega\big|_{f_\sigma} = 0$ for all $f_\sigma \in \Delta_k(T)$ hence in particular $\omega\big|_{\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)}} = 0$ for all $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}^k_r(T) \setminus \widetilde{\Sigma}^k_{r,min}(T)$ and $\int_{\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)}} \omega = 0$ for each $\tilde{s}_{(\boldsymbol{\alpha}, f_\sigma)} \in \widetilde{\Sigma}^k_r(T)$. □

This minimal set of unisolvent weights allows to show that also the weights on the natural subset of small simplices

$$\Sigma^k_{r,min}(T) = \{s_{(\boldsymbol{\alpha}, f_\sigma)} \in \Sigma^k_r(T) \;:\; f_\sigma \in \Delta_k(T) \text{ and } \boldsymbol{\alpha} \in \widetilde{\mathcal{I}}_\sigma(n+1, r)\},$$

that are clearly in a one to one correspondence with the elements of the basis of $\mathcal{P}^-_{r+1}\Lambda^k(T)$ introduced in Theorem 2, are a minimal set of unisolvent degrees of freedom. We have in fact the following.

Theorem 4 *If $\omega \in \mathcal{P}_r^- \Lambda^k(T)$ is such that $\int_{s_{(\boldsymbol{\alpha}, f_\sigma)}} \omega = 0$ for each $s_{(\boldsymbol{\alpha}, f_\sigma)} \in \Sigma^k_{r,min}(T)$, then $\omega = 0$.*

Acknowledgments This research was partially supported by the program INdAM-GNCS Projects 2019.

References

1. Arnold, D. N., Falk, R. S., Winther, R.: Finite element exterior calculus, homological techniques, and applications. Acta Numer. **15**, (2006).
2. Arnold, D. N., Falk, R. S., Winther, R.: Differential complexes and stability of finite element methods I: The De Rham complex. IMA Vol. Math. Appl. **142**, 24–46 (2006).
3. Bossavit, A.: Whitney forms: A class of finite elements for three-dimensional computations in electromagnetism. IEE Proceedings A, science measurement and technology **135**, 493–500 (1988).
4. Christiansen, S. H., Rapetti, F.: On high order finite element spaces of differential forms. Math. Comp. **85**, 517–548 (2016).
5. Hiptmair, R.: Canonical construction of finite elements. Math. Comp. **68**, 1325–1346 (1999).
6. Nédélec, J.C.: Mixed finite elements in $\mathbb{R}^3$. Numer. Math. **35**, 315–341 (1980).
7. Rapetti, F., Bossavit, A.: Whitney forms of higher degree. Siam J. Numer. Anal. **47**, 2369–2386 (2009).
8. Whitney, H.: Geometric Integration Theory. Princeton Mathematical Series, Princeton, New Jersey (1957).

Experimental Comparison of Symplectic and Non-symplectic Model Order Reduction on an Uncertainty Quantification Problem

Patrick Buchfink and Bernard Haasdonk

Abstract Uncertainty Quantification (UQ) is an important field to quantify the propagation of uncertainties, analyze sensitivities or realize statistical inversion of a mathematical model. Sampling-based estimation techniques evaluate the model for many different parameter samples. For computationally intensive models, this might require long runtimes or even be infeasible. This so-called multi-query problem can be speeded up or even be enabled with surrogate models from model order reduction (MOR) techniques. For accurate and physically consistent MOR, structure-preserving reduction is essential.

We investigate numerically how so-called symplectic model reduction techniques can improve the UQ results for Hamiltonian systems compared to conventional (non-symplectic) approaches. We conclude that the symplectic methods give better results and more robustness with respect to the size of the reduced model.

1 Introduction

Sampling-based Uncertainty Quantification (UQ) is known to benefit from surrogate modelling which includes model order reduction (MOR) methods [1, 4]. The problem with conventional MOR techniques is that the reduced model might lose the original system structure and might thus produce unphysical results. A solution are structure-preserving MOR techniques [2, 6, 9, 10].

A popular example for structured, mathematical models are Hamiltonian systems. These are known for the characteristic property to preserve the Hamiltonian function which can in many contexts be interpreted as the energy of the system.

As structure-preserving reduction technique for parametric Hamiltonian systems, we consider symplectic MOR with the Proper Symplectic Decomposition (PSD) basis generation technique [2, 9, 10].

P. Buchfink (✉) · B. Haasdonk
University of Stuttgart, Stuttgart, Germany
e-mail: patrick.buchfink@ians.uni-stuttgart.de; bernard.haasdonk@ians.uni-stuttgart.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_19

Table 1 MOR techniques used in the experiments. Classified by orthogonality and symplecticity

MOR technique	Abbreviation	Ortho.	Sympl.	Ref.
POD of complete state	POD complete	✓	✗	[1]
Block structure preserving POD, i.e. $\boldsymbol{V} = \mathrm{blkdiag}(\boldsymbol{V_q}, \boldsymbol{V_p})$	POD separate	✓	✗	[14]
PSD complex SVD	PSD cSVD	✓	✓	[10]
PSD SVD-like decomposition	PSD SVD-like	✗	✓	[2]

Based on a two-dimensional, linear elasticity problem, we compare two symplectic and two non-symplectic MOR techniques (see Table 1) numerically on an UQ experiment. We observe a significantly higher stability and more accuracy for the symplectic techniques in comparison to the established, non-symplectic methods.

2 Symplectic Model Order Reduction for Parametrized Hamiltonian Systems

Symplectic model order reduction (MOR) is a structure-preserving MOR technique for parametrized, finite-dimensional, canonical, autonomous Hamiltonian systems. These systems are formulated in terms of an even-dimensional state $\boldsymbol{x}(t, \boldsymbol{\mu}) \in \mathbb{R}^{2n}$ as

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}(t, \boldsymbol{\mu}) = \mathbb{J}_{2n}\nabla_{\boldsymbol{x}}\mathcal{H}(\boldsymbol{x}(t, \boldsymbol{\mu}), \boldsymbol{\mu}), \qquad \mathbb{J}_{2n} := \begin{pmatrix} \boldsymbol{0}_n & \boldsymbol{I}_n \\ -\boldsymbol{I}_n & \boldsymbol{0}_n \end{pmatrix}, \qquad \boldsymbol{x}(t_0, \boldsymbol{\mu}) = \boldsymbol{x}_0(\boldsymbol{\mu}) \tag{1}$$

where $(\boldsymbol{x}_0, t_0) \in \mathbb{R}^{2n} \times \mathbb{R}$ is the initial condition, $t \in I := [t_0, \infty)$ is the time, $\boldsymbol{\mu} \in \mathcal{P} \subset \mathbb{R}^p$ is the parameter vector from a parameter space $\mathcal{P}$, $\mathbb{J}_{2n}$ is the so-called canonical Poisson matrix, $\nabla_{\boldsymbol{x}}(\cdot)$ is the gradient, $\mathcal{H} : \mathbb{R}^{2n} \times \mathcal{P} \to \mathbb{R}$ is the Hamiltonian (function). For a detailed introduction to Hamiltonian systems, we refer e.g. to [8].

The underlying geometry of a Hamiltonian system is the symplectic geometry. In the following, we give a concise presentation of symplectic geometry in finite dimensional vector spaces as a background for the motivation of symplectic MOR. For more details, we refer e.g. to [3].

The symplectic geometry over finite dimensional vector spaces is based on symplectic forms $\omega_{2n} : \mathbb{R}^{2n} \times \mathbb{R}^{2n} \to \mathbb{R}$ which are special (skew symmetric and non-degenerate) bilinear maps. The canonical form is defined by

$$\omega_{2n}(\boldsymbol{v}, \ \boldsymbol{w}) := \boldsymbol{v}^{\mathsf{T}}\mathbb{J}_{2n}\boldsymbol{w}, \qquad \boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^{2n}.$$

A linear map $A : \mathbb{R}^{2k} \to \mathbb{R}^{2n}, \boldsymbol{x} \mapsto \boldsymbol{A}\boldsymbol{x}$ with the coefficient matrix $\boldsymbol{A} \in \mathbb{R}^{2n\times 2k}$ with $k \leq n$ is called symplectic if it preserves the symplectic structure in the sense that it holds for all $\boldsymbol{v}, \boldsymbol{w} \in \mathbb{R}^{2k}$

$$\omega_{2n}\left(\boldsymbol{A},\ \boldsymbol{v}\right)\boldsymbol{A}\boldsymbol{w} = \boldsymbol{v}^{\mathsf{T}}(\boldsymbol{A}^{\mathsf{T}}\mathbb{J}_{2n}\boldsymbol{A})\boldsymbol{w} = \boldsymbol{v}^{\mathsf{T}}\mathbb{J}_{2k}\boldsymbol{w} = \omega_{2k}\left(\boldsymbol{v},\ \boldsymbol{w}\right)$$

which is equivalent to $\boldsymbol{A}^{\mathsf{T}}\mathbb{J}_{2n}\boldsymbol{A} = \mathbb{J}_{2k}$. We call a matrix $\boldsymbol{A}$ symplectic (with respect to ω_{2n} and ω_{2k}) if it fulfills this condition.

Most common MOR methods are projection-based techniques [1] which proceed in two steps: firstly, the original state is approximated in a k-dimensional linear subspace with a reduced-order basis (ROB) by

$$\boldsymbol{x}(t, \boldsymbol{\mu}) \approx \boldsymbol{V}_k\boldsymbol{x}_k(t, \boldsymbol{\mu}), \qquad \text{ROB: } \boldsymbol{V}_k \in \mathbb{R}^{2n\times k}, \qquad \text{reduced state: } \boldsymbol{x}_k(t, \boldsymbol{\mu}) \in \mathbb{R}^k. \tag{2}$$

As a second step, the residual of this approximation is projected in a k-dimensional space with a projection matrix $\boldsymbol{W}_k \in \mathbb{R}^{2n\times k}$ in order to get a well-posed problem.

The symplectic MOR follows this standard procedure with the requirements that (i) $\boldsymbol{V}_{2k} \in \mathbb{R}^{2n\times 2k}$ is a symplectic matrix with (necessarily even) reduced order $2k$ and (ii) the projection matrix is chosen to be the so-called symplectic inverse $\boldsymbol{V}_{2k}^{+}$, i.e.

$$\text{(i)}\quad \boldsymbol{V}_{2k}^{\mathsf{T}}\mathbb{J}_{2n}\boldsymbol{V}_{2k} = \mathbb{J}_{2k} \qquad \text{and} \qquad \text{(ii)}\quad \boldsymbol{W}_{2k} = \boldsymbol{V}_{2k}^{+} := {\mathbb{J}_{2k}}^{\mathsf{T}}\boldsymbol{V}_{2k}^{\mathsf{T}}\mathbb{J}_{2n}.$$

This choice ensures that the reduced system is a $2k$-dimensional Hamiltonian system

$$\frac{\mathrm{d}}{\mathrm{d}t}\boldsymbol{x}_{2k}(t, \boldsymbol{\mu}) = \mathbb{J}_{2k}\nabla_{\boldsymbol{x}_{\mathrm{r}}}\mathcal{H}_{2k}(\boldsymbol{x}_{2k}(t, \boldsymbol{\mu}), \boldsymbol{\mu}), \qquad \mathcal{H}_{2k}(\boldsymbol{x}_{2k}, \boldsymbol{\mu}) := \mathcal{H}(\boldsymbol{V}_{2k}\boldsymbol{x}_{2k}, \boldsymbol{\mu}),$$
$$\boldsymbol{x}_{2k}(t_0, \boldsymbol{\mu}) = \boldsymbol{W}_{2k}^{\mathsf{T}}\boldsymbol{x}_0(\boldsymbol{\mu}).$$

For further details on symplectic MOR, we refer to [2, 9, 10].

The projection-based MOR (2) leaves a high flexibility in the choice of a ROB. To this end, basis generation techniques are required. One common class are snapshot-based methods [11]. In [10], this idea is adapted to generate a symplectic ROB via an optimization problem which is labelled Proper Symplectic Decomposition (PSD). A general solution procedure for the PSD is yet unknown since it is highly non-convex. If in addition, it is assumed that the ROB has orthonormal columns, it is shown in [2] that a solution can be computed with the Complex Singular Value Decomposition (complex SVD) [10]. Up to our knowledge, the only basis generation technique that generates a non-orthogonal ROB is introduced in [2] which is based on a SVD-like decomposition (see [13]). We will compare both of these methods together with two non-symplectic MOR techniques in the numerical experiments and investigate the acceleration of an UQ experiment.

3 Model Order Reduction and Uncertainty Quantification

The connection of MOR and UQ is based on the following principle: We call the repetitive evaluation of a mathematical model a so-called multi-query scenario. For sampling-based UQ methods like the Monte Carlo method [12], this setting applies since the model is evaluated for multiple samples $\boldsymbol{\mu}_i \in \mathcal{P}$ to estimate the statistics of a quantity of interest. If a reduced model of reduced order $2k \ll 2n$ is trained, the approximation (2) is used afterwards for the evaluations instead of the original full-order model with $\boldsymbol{x}(t, \boldsymbol{\mu}) \in \mathbb{R}^{2n}$. The crucial point then is that the computation of the state $\boldsymbol{x}_{2k}(t, \boldsymbol{\mu}_i) \in \mathbb{R}^{2k}$ (and the approximation $\boldsymbol{V}_{2k}\boldsymbol{x}_{\mathrm{r}}(\cdot)$) is much faster than the original model since the state dimension typically directly correlates with the runtime of the simulation.

4 Numerical Experiment

To compare our approach with conventional MOR techniques in an UQ experiment, we consider an oversimplified muscle model as a linear elastic problem with a two-dimensional fusiform geometry (see Fig. 1). The specimen is loaded in axial direction with a force homogeneously distributed on the right boundary of the domain and an amplitude $F(t) = F_{\max}\sigma(t - 0.1)$ where $F_{\max} \geq 0$ is a parameter and $\sigma(\cdot)$ is the step function. The parameter vector $\boldsymbol{\mu}^{\mathsf{T}} = [\lambda_{\mathrm{L}}, \mu_{\mathrm{L}}, F_{\max}] \in \mathcal{P} := \mathbb{R}^2_{>0} \times [0.39, 4.71]$ of the system consists of the Lamé constants $\lambda_{\mathrm{L}}, \mu_{\mathrm{L}} > 0$ and the maximal amplitude $0.49 \leq F_{\max} \leq 5.89$ of the force $F(t)$. The simulation time is $t \in I := [0, 0.5]$.

We derive a semi-discretized system with the Finite Element Method with piecewise linear Lagrangian ansatz functions on a triangular mesh with 910 nodes resulting in $2n = 3640$ degrees of freedom. This comparably small example is already sufficient to display the advantages of our method. The Hamiltonian function of the underlying system for the state $\boldsymbol{x} = [\boldsymbol{q}, \boldsymbol{p}]^{\mathsf{T}}$ with displacement $\boldsymbol{q} \in \mathbb{R}^n$ and linear momentum $\boldsymbol{p} \in \mathbb{R}^n$ is

$$\mathcal{H}(\boldsymbol{x}, t, \boldsymbol{\mu}) = \frac{1}{2}\left(\boldsymbol{p}^{\mathsf{T}}\boldsymbol{M}\boldsymbol{p} + \boldsymbol{q}^{\mathsf{T}}\boldsymbol{K}(\boldsymbol{\mu})\boldsymbol{q}\right) - \boldsymbol{q}^{\mathsf{T}}\boldsymbol{f}(t, \boldsymbol{\mu}), \quad \begin{aligned} \boldsymbol{M} &\in \mathbb{R}^{n \times n} : \text{mass matrix}, \\ \boldsymbol{K} &\in \mathbb{R}^{n \times n} : \text{stiffness matrix}, \\ \boldsymbol{f} &\in \mathbb{R}^{n} : \quad \text{force vector} \quad . \end{aligned}$$

Fig. 1 The discretization of our spatial domain $\Omega \subset \mathbb{R}^2$ simulating a fusiform muscle

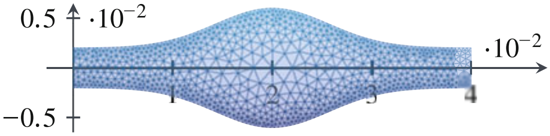


Table 2 Model parameters and respective probability distributions. The mean values $\overline{\lambda_{\mathrm{L}}}$, $\overline{\mu_{\mathrm{L}}}$ of the Lamé constants λ_{L}, μ_{L} are obtained from [7]

Model parameter	Distribution
Lamé constant λ_{L}	Log-normal, $\log(\lambda_{\mathrm{L}}) \sim \mathcal{N}(\log(\overline{\lambda_{\mathrm{L}}}), \sigma^2)$, $\overline{\lambda_{\mathrm{L}}} = 80{,}069$, $\sigma^2 = 0.01$
Lamé constant μ_{L}	Log-normal, $\log(\mu_{\mathrm{L}}) \sim \mathcal{N}(\log(\overline{\mu_{\mathrm{L}}}), \sigma^2)$, $\overline{\mu_{\mathrm{L}}} = 8966$, $\sigma^2 = 0.01$
Maximal force $F_{\max}$	Uniformly distributed on the interval [0.39, 4.71] N

The time is discretized with the implicit midpoint rule [5] with $n_t = 1501$ equidistant time steps. This choice ensures that, in our example, the Hamiltonian is conserved since (1) this integrator conserves quadratic invariants and (2) the Hamiltonian is quadratic. The use of a symplectic integrator like the implicit midpoint method is essential for structure-preserving MOR.

We consider two time-dependent quantities of interest (QoIs): this includes the displacement $s_1(\cdot)$ of one of the nodes on the right boundary and the potential part of the Hamiltonian $s_2(\boldsymbol{x}, t, \boldsymbol{\mu}) = \frac{1}{2}\boldsymbol{q}^{\mathsf{T}}\boldsymbol{K}(\boldsymbol{\mu})\boldsymbol{q} - \boldsymbol{q}^{\mathsf{T}}\boldsymbol{f}(t, \boldsymbol{\mu})$. As point evaluation, $s_1(\cdot)$ is a linear and local functional. In contrast to that, $s_2(\cdot)$ is a global and quadratic functional. We may suppress the explicit dependence of s_2 on t and $\boldsymbol{\mu}$ in the following.

As estimator for the distribution of the QoIs, we use the classical Monte Carlo algorithm [12] with 1000 sample parameter vectors which follow the probability distributions listed in Table 2. Let ν denote the associated probability measure.

The simulations are conducted in RBmatlab[1] which is an MATLAB®-based open-source library that implements multiple state-of-the-art MOR techniques. The versions used in the experiments are RBmatlab 1.16.09 and MATLAB® 2019b.

As described in Sect. 3, MOR techniques can be used in the multi-query context posed by the UQ problem. Thus, we approximate $s_i(\boldsymbol{x}(t, \boldsymbol{\mu})) \approx s_i(\boldsymbol{V}_{2k}\boldsymbol{x}_{2k}(t, \boldsymbol{\mu}))$ with a ROB of logarithmically spaced ROB size $2k \in \{2^i \mid 2 \leq i \leq 10\}$ to investigate the reduced model for nine different sizes. The four MOR techniques we compare are based on the Proper Orthogonal Decomposition (POD) and the Proper Symplectic Decomposition (PSD). We either compute the ROB with a POD of the complete state (POD complete) or separately for the displacement $\boldsymbol{q}$ and linear momentum $\boldsymbol{p}$ (POD separate). The PSD methods are based on the complex SVD (PSD cSVD) and the SVD-like decomposition (PSD SVD-like). The techniques are summarized in Table 1 classified by orthogonality and symplecticity of the associated ROB.

The training was performed on a regular $3 \times 3 \times 3$ grid in the parameter space resulting in $3^3 = 27$ sample parameter vectors. For each parameter vector, a full dynamic simulation was calculated and every ninth time step was included in the snapshot set which gives in total $n_{\mathrm{s}} = 4509$ snapshots. To analyze the performance

[1] https://www.morepas.org/software/rbmatlab/, last accessed: 30. Oct. 2019.

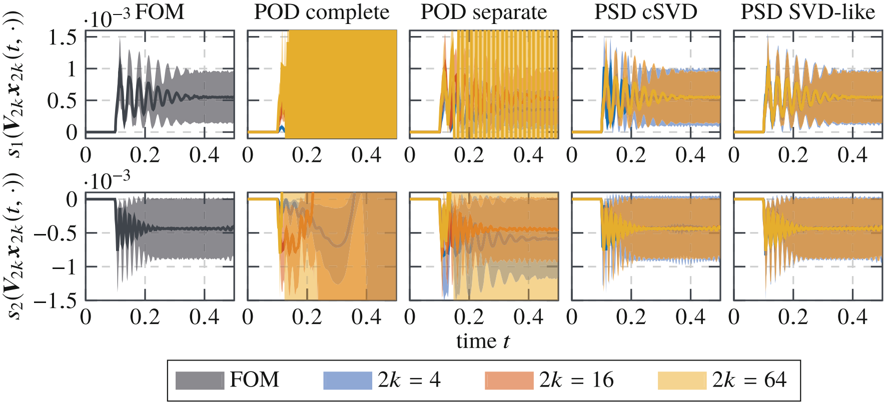


Fig. 2 The evolution of the mean value (bold line) and the standard deviation (shade) of the two QoIs $s_i(V_{2k}x_{2k}(t,\cdot))$ displayed for all times $t \in I$. The different colors visualize the full-order model (FOM) solution (black) or reduced solutions with different ROB sizes $2k \in \{4, 16, 64\}$

in the prediction phase for the QoIs, we depicted in Fig. 2 the estimated mean value and the standard deviation for times $t \in I$ approximated by the different MOR techniques for selected basis sizes $2k \in \{4, 16, 64\}$. The estimation with the full-order model (FOM) is plotted in black as reference. For $2k = 64$ and $t > 0.15$, the mean value of the POD methods shows heavy oscillations which are not present in the FOM and thus are an unphysical artefact of the reduction. The best solutions for the POD methods are obtained for a small time horizon $t < 0.15$ or a medium basis size $2k = 16$. In contrast, the PSD methods show very robust results considering the ROB size $2k$. Comparing the two PSD methods, we see that the PSD cSVD struggles with small basis sizes $2k = 4$ whereas PSD SVD-like, visually, yields very accurate solutions for any basis size—even $2k = 4$. This shows that the additional requirement for the ROB to be orthonormal in PSD cSVD requires more basis vectors to yield as accurate results as the PSD SVD-like method.

In order to quantify the error of the reduction more precisely, we plot the relative $L_{2,\nu}(\mathcal{P} \times I)$ error $e_{L_{2,\nu}}[s_i]$ of the i-th QoI $s_i(\cdot)$ for different ROB sizes $2k$ in Fig. 3. We again clearly see that a practical application of the POD methods is not possible in our example since a relative error less than 60% is not obtained. The robustness of the two PSD methods with respect to the ROB size is expressed by the monotonic and actually exponential decrease for increasing ROB sizes. The figure quantifies the supremacy of PSD SVD-like in comparison to PSD cSVD between 2–30% improvement in the relative error for a fixed basis size $2k$. This superiority can especially be observed for small and medium ROB sizes $2k < 10^2$ which supports the conjecture that the additional requirement of orthogonality in PSD cSVD hampers the approximation for such ROB sizes.

The ultimate goal is to reduce the runtime of the multi-query setting posed by the UQ problem with model reduction without introducing a too big error. To this end, we inspect in Table 3 the runtime of 1000 model evaluations relative to the

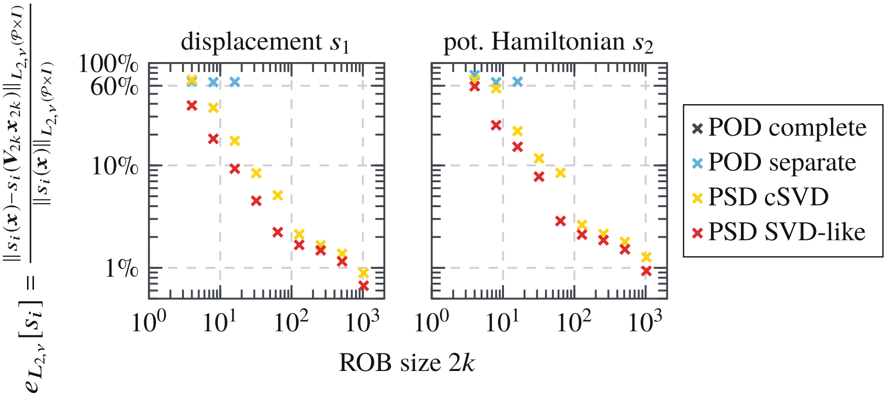


Fig. 3 The relative $L_{2,\nu}(\mathcal{P} \times I)$ reduction error $e_{L_{2,\nu}}[s_i]$ of the two QoIs $s_i(\cdot)$ depicted for the ROB size $2k$. For $\|\cdot\|_{L_{2,\nu}(\mathcal{P}\times I)}$, the integral over I is approximated with the trapezoidal rule and the integral over $\mathcal{P}$ with measure ν with Monte Carlo. Errors bigger than 100% are excluded

Table 3 The relative runtime for the offline part, 1000 online simulations ($2k = 16$) and in total

	FOM	POD compl.	POD sep.	PSD cSVD	PSD SVD-like
Offline	–	2.7%	2.5%	2.5%	17.8%
1000· online	–	8.9%	8.7%	8.6%	8.6%
Total	25 min. $\widehat{=}$ 100%	11.6%	11.2%	11.1%	26.4%

FOM. As typically done for MOR, we split the runtime in the offline and the online part. The offline part consists mainly of computing the ROB. In the online part, we evaluate the reduced model for the 1000 different parameter vectors. We see that all investigated methods require a similar amount of runtime in the offline phase except for the PSD SVD-like. The reason is that the PSD SVD-like method is implemented in native MATLAB® whereas the other methods use internal functions which run in Fortran. In terms of complexity orders, the computation of the SVD-like decomposition should be comparable to the computation of the underlying matrix decompositions of the other methods. The runtime for the online part is displayed for $2k = 16$. Since all methods are projection-based MOR techniques, the computational cost in the online part is equal which is expressed by near equal runtimes of roughly 8.7%.

The user can trade accuracy for runtime by adjusting the basis size $2k$. We visualize this trade-off in Fig. 4 with the reduction error (from Fig. 3) in relation to the relative runtime (from Table 3). Due to the instability of the POD methods in our example, only the PSD methods are interesting for this purpose. Both PSD methods are able to speed up the simulation by a factor between 2.2 and 11.6 while introducing a relative reduction error between 1.3 and 68.5% for both QoIs $s_i(\cdot)$. Considering only the slightly better PSD SVD-like, the error improves to numbers between 0.9 and 59.3%. There is a plateau in the trade-off curve at a runtime of

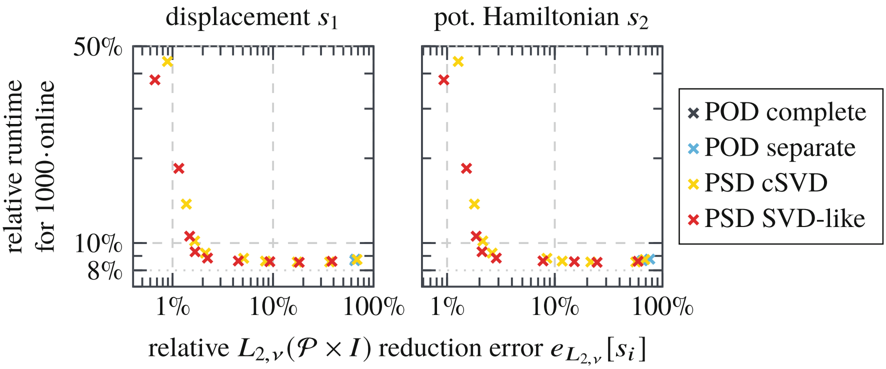


Fig. 4 The relative runtime for 1000 online simulations illustrated in relation to the relative $L_{2,\nu}(\mathcal{P} \times I)$ reduction error $e_{L_{2,\nu}}[s_i]$ of the two QoIs $s_i(\cdot)$. Errors bigger than 100% are excluded

8.5%. The reason is that a major part of the runtime is spent in iterations over the time steps n_t and the samples n_s which cannot be further reduced with (2).

5 Summary and Outlook

We presented a framework to conduct Uncertainty Quantification (UQ) experiments in combination with structure-preserving model order reduction (MOR) for Hamiltonian systems. The numerical experiments showed that the preservation of the symplectic structure improves the stability and the accuracy of the reduction with MOR and thus, also of the accelerated UQ framework. This enabled us to provide a broad spectrum of reduced models ranging from a speed up of factor 2.2 up to 11.6 while introducing an error between 0.9 and 59.3%. The best results are achieved with the symplectic, non-orthogonal basis generation technique PSD SVD-like decomposition.

In future work, a more realistic muscle model should be considered since the assumption of small strains is too restricting for muscle models. Furthermore, mathematical stability analysis for non-orthogonal but symplectic bases is required.

Acknowledgments We highly acknowledge the funding within the Soft Tissue Project (GRK 2198/1) and by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC-2075—390740016. The valuable input of the anonymous reviewers is acknowledged.

References

1. Benner, P., Ohlberger, M., Cohen, A., Willcox, K.: Model Reduction and Approximation. Society Industrial Appl. Math., Philadelphia, PA (2017). https://doi.org/10.1137/1.9781611974829
2. Buchfink, P., Bhatt, A., Haasdonk, B.: Symplectic Model Order Reduction with Non-Orthonormal Bases. Math. Comput. Appl. **24**(2) (2019). https://doi.org/10.3390/mca24020043
3. da Silva, A.C.: Introduction to Symplectic and Hamiltonian Geometry (2007). https://people.math.ethz.ch/~acannas/Papers/impa.pdf. Notes for a Short Course at IMPA.
4. Galbally, D., Fidkowski, K., Willcox, K., Ghattas, O.: Non-linear model reduction for uncertainty quantification in large-scale inverse problems. Int. J. Numer. Methods Eng. **81**(12), 1581–1608 (2010). https://doi.org/10.1002/nme.2746
5. Hairer, E., Wanner, G., Lubich, C.: Geometric Numerical Integration. Springer, Berlin, Heidelberg (2006). https://doi.org/10.1007/3-540-30666-8
6. Hesthaven, J.S., Pagliantini, C.: Structure-Preserving Reduced Basis Methods for Hamiltonian Systems with a Nonlinear Poisson Structure (2018). Preprint, Infoscience EPFL sci. publ.
7. Kajee, Y., Pelteret, J.P.V., Reddy, B.D.: The biomechanics of the human tongue. Int. J. Numer. Methods Biomed. Eng. **29**, 492–514 (2013). https://doi.org/10.1002/cnm.2531
8. Leimkuhler, B., Reich, S.: Simulating Hamiltonian Dynamics. Cambridge Monogr. Appl. Comput. Math. Cambridge University Press (2005). https://doi.org/10.1017/CBO9780511614118
9. Maboudi Afkham, B., Hesthaven, J.: Structure Preserving Model Reduction of Parametric Hamiltonian Systems. SIAM SISC **39**(6), A2616–A2644 (2017). https://doi.org/10.1137/17M1111991
10. Peng, L., Mohseni, K.: Symplectic Model Reduction of Hamiltonian Systems. SIAM SISC **38**(1), A1–A27 (2016). https://doi.org/10.1137/140978922
11. Sirovich, L.: Turbulence the dynamics of coherent structures. Part I: coherent structures. Q. Appl. Math. **45**(3), 561–571 (1987)
12. Sullivan, T.J.: Introduction to Uncertainty Quantification. Springer Int. Publ. (2015)
13. Xu, H.: An SVD-like matrix decomposition and its applications. Linear Algebr. Appl. **368**, 1–24 (2003). https://doi.org/10.1016/S0024-3795(03)00370-7
14. Yu, H., He, L., Tar, S.X.D.: Block structure preserving model order reduction. In: BMAS. Proc. IEEE Int. Behav. Model. Simul. Work., pp. 1–6 (2005). https://doi.org/10.1109/BMAS.2005.1518178

3D-2D Stokes-Darcy Coupling for the Modelling of Seepage with an Application to Fluid-Structure Interaction with Contact

Erik Burman, Miguel A. Fernández, Stefan Frei, and Fannie M. Gerosa

Abstract In this note we introduce a mixed dimensional Stokes-Darcy coupling where a d dimensional Stokes' flow is coupled to a Darcy model on the $d-1$ dimensional boundary of the domain. The porous layer introduces tangential creeping flow along the boundary and allows for the modelling of boundary flow due to surface roughness. This leads to a new model of flow in fracture networks with reservoirs in an impenetrable bulk matrix. Exploiting this modelling capability, we then formulate a fluid-structure interaction method with contact, where the porous layer allows for mechanically consistent contact and release. Physical seepage in the contact zone due to rough surfaces is modelled by the porous layer. Some numerical examples are reported, both on the Stokes'-Darcy coupling alone and on the fluid-structure interaction with contact in the porous boundary layer.

1 Introduction

In numerous environmental or biomedical applications there is a need to model the coupling between a flow in a reservoir and flow in a surrounding porous medium. This is particularly challenging if the porous medium is fractured and the bulk matrix has very low permeability. Typically the fractures are modelled as $d-1$

E. Burman (✉)
Department of Mathematics, University College London, London, UK
e-mail: e.burman@ucl.ac.uk

M. A. Fernández · F. M. Gerosa
Inria, Sorbonne Université, Paris, France

M. A. Fernández · F. M. Gerosa
CNRS, UMR 7598, LJLL, Paris, France
e-mail: miguel.fernandez@inria.fr; fannie.gerosa@inria.fr

S. Frei
Department of Mathematics & Statistics, University of Konstanz, Konstanz, Germany
e-mail: stefan.frei@uni-konstanz.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_20

dimensional manifolds, embedded in a d dimensional porous bulk matrix. For the modelling of the fractured porous medium we refer to [3]. Observe however that if the bulk permeability is negligible the fluid in the reservoir cannot penetrate into the fractures since the $d-1$ dimensional manifolds have an intersection of the reservoir boundary of $d-1$ measure zero. This means that such a model cannot be used for the fluid flow between two reservoirs connected by a fracture in an impenetrable medium. Here we propose to introduce a Darcy equation for the tangential flow on the boundary of the reservoir. Since this equation is set on a $d-1$ dimensional manifold it can provide an interface allowing for flow from the reservoir to the cracks. The flow on the boundary communicates with the flow in the cracks through continuity of pressure and conservation expressed by Kirchhoff's law. This gives a cheap and flexible model for flow in reservoirs connected by fractures.

Our original motivation for this model is the particular case of fluid structure interaction with contact where the situation described above occurs when two boundaries enter in contact provoking a change of topology of the fluid domain. It has recently been observed by several authors [1, 4] that the consistent modelling of fluid-structure interaction with contact requires a fluid model, in particular a pressure, also in the contact zone. Indeed, some seepage is expected to occur due to permeability of the contacting bodies or their surface roughness. Otherwise there is no continuous mechanism for the release of contact and non-physical voids can occur. For instance, it was argued in [1] that a consistent modelling of FSI with contact requires a complete modelling of the FSI-poroelastic coupling. Similar ideas were introduced in [4], but for computational reasons. Indeed, in the latter reference an elastic body immersed in a fluid enters in contact with a rigid wall and to allow for a consistent numerical modelling the permeability of the wall is relaxed. This motivates the introduction of an artificial porous medium whose permeability goes to zero with the mesh-size. Both approaches allow for the seepage that appears to be necessary for physical contact and release. However, in case the contacting solids are (modelled as) impenetrable, this seepage must be due to porous media flow in a thin layer in the contact zone due to surface roughness. The complete modelling of the poroelastic interaction of [1] or the bulk porous medium flow of [4] then appears artificial and unnecessarily expensive. For such situations the mixed dimensional modelling suggested above can offer an attractive compromise between model detail and computational cost.

In this note, we will focus exclusively on the modelling aspect. The coupled Stokes-Darcy model is introduced in Sect. 2. Then, in Sect. 3, we show how the ideas of [4] can be used to model FSI with contact together with the mixed-dimensional fluid system. Finally, we illustrate the two model situations numerically in Sect. 4. First, the Stokes'-Darcy reservoir coupling (Sect. 4.1) and then the full FSI with contact (Sect. 4.2). In the latter case, we also give comparisons with the results from [4]. The numerical analysis of the resulting methods will be the subject of future work.

2 The Coupled Stokes-Darcy System

We consider the coupling of a Darcy system in a thin-walled domain $\Omega_l = \Sigma_l \times (-\frac{\epsilon}{2}, \frac{\epsilon}{2}) \in \mathbb{R}^d$ for $d = 2, 3$ with a Stokes equation in the bulk domain Ω_f. The Darcy problem on Ω_l writes

$$\begin{cases} u_l + K\nabla p_l = 0 \\ \nabla \cdot u_l = 0 \end{cases} \quad \text{in} \quad \Omega_l, \tag{1}$$

where u_l denotes the Darcy velocity, p_l the Darcy pressure and K is a $d \times d$ matrix that allows for the decomposition

$$K\nabla p_l = K_\tau \nabla_\tau p_l + K_n \partial_n p_l.$$

We denote the upper boundary of Ω_l which couples to Ω_f by γ_f and the outer boundary by γ_o. The normal vector n of the middle surface Σ_l of Ω_l is chosen in such a way that it points towards γ_o.

By averaging across the thickness ϵ, Martin, Jaffré and Roberts derived in [3] an effective equation for the averaged pressure across the thickness

$$P_l := \frac{1}{\epsilon}\int_{-\frac{\epsilon}{2}}^{\frac{\epsilon}{2}} p_l.$$

Under the modelling assumption that the average pressure is equal to the mean of the pressures on the upper and lower boundary

$$P_l = \frac{1}{2}\left(p_l|_{\gamma_f} + p_l|_{\gamma_o}\right) \quad \text{in} \quad \Sigma_l, \tag{2}$$

the authors derived the system

$$\begin{cases} -\nabla_\tau \cdot (\epsilon K_\tau \nabla_\tau P_l) = u_{l,n}|_{\gamma_f} - u_{l,n}|_{\gamma_o} \\ p_l|_{\gamma_f} = P_l + \dfrac{\epsilon K_n^{-1}}{4}\left(u_{l,n}|_{\gamma_o} + u_{l,n}|_{\gamma_f}\right) \end{cases} \quad \text{in} \quad \Sigma_l. \tag{3}$$

Here, $u_{l,n} = u_l \cdot n$ denotes the normal component of the velocity and τ is a tangential vector of Σ_l. We will couple (3) to Stokes flow in Ω_f

$$\begin{cases} \rho_f \partial_t u_f - \nabla \cdot \sigma_f(u_f, p_f) = 0 \\ \nabla \cdot u_f = 0 \end{cases} \quad \text{in} \quad \Omega_f, \tag{4}$$

where u_f denotes the fluid velocity, p_f the pressure, ρ_{f} the fluid density,

$$\sigma_f(u_f, p_f) := \mu(\nabla u_f + \nabla u_f^T) - p_f I,$$

the fluid Cauchy stress tensor and μ the dynamic viscosity. We assume that the coupling to the Darcy system (1) on γ_f takes place via the interface conditions

$$\begin{cases} \sigma_{f,nn} = -p_l \\ \tau^T \sigma_f n = 0 \quad \text{on} \quad \gamma_f, \\ u_{f,n} = u_{l,n} \end{cases} \tag{5}$$

where $\sigma_f = \nabla u_f - p_f I$ and $\sigma_{f,nn} = n^T \sigma_f n$. In the lower porous wall γ_o we assume for simplicity that $u_{l,n} = 0$. Then, the relations (3) can be written as

$$\begin{cases} -\nabla_\tau \cdot (\epsilon K_\tau \nabla_\tau P_l) = u_{f,n} \\ \sigma_{f,nn} = -P_l - \dfrac{\epsilon K_n^{-1}}{4} u_{f,n} \end{cases} \quad \text{in} \quad \Sigma_l.$$

Note that the only remaining porous medium variable is the averaged pressure P_l. In the limit of permeability $K_n \to 0$, the system converges to a pure Stokes system with slip conditions on γ_f with an extension of the fluid forces into the porous medium pressure P_l.

We have the following coupled variational problem for (u_f, p_f, P_l):

$$\begin{cases} \rho_f(\partial_t u_f, v_f)_{\Omega_f} + (\sigma_f(u_f, p_f), \nabla v_f)_{\Omega_f} + (q_f, \nabla \cdot u_f)_{\Omega_f} \\ \qquad + \big(P_l, v_{f,n}\big)_{\Sigma_l} + \dfrac{\epsilon K_n^{-1}}{4}\big(u_{f,n}, v_{f,n}\big)_{\Sigma_l} = 0, \\ \qquad (\epsilon K_\tau \nabla_\tau P_l, \nabla_\tau q_l)_{\Sigma_l} - \big(u_{f,n}, q_l\big)_{\Sigma_l} = 0, \end{cases} \tag{6}$$

for all v_f, q_f, q_l, where $n = n_f$ is the outer normal of the fluid domain Ω_f.

3 The Fluid-Structure-Poroelastic-Contact Interaction System

Now, we consider a fluid-structure-contact interaction system with a thin porous layer on the part of the exterior boundary, where contact might take place. The moving boundary of the solid is denoted by $\Sigma(t)$ and the porous layer by Σ_l. In absence of contact, we have the following system of equations

$$\begin{cases} \rho_f \partial_t u_f - \nabla \cdot \sigma_f(u_f, p_f) = 0 \\ \nabla \cdot u_f = 0 \end{cases} \quad \text{in} \quad \Omega_f(t),$$

$$\rho_s \partial_t \dot{d} - \nabla \cdot \sigma_s(d) = 0 \quad \text{in} \quad \Omega_s(t),$$

$$u_f = \dot{d}, \quad \sigma_s n = \sigma_f n \quad \text{in} \quad \Sigma(t),$$

$$\begin{cases} -\nabla_\tau \cdot (\epsilon K_\tau \nabla_\tau P_l) = u_{l,n}|_{\gamma_f} \\ \sigma_{f,nn} = \underbrace{-P_l - \frac{\epsilon K_n^{-1}}{4} u_{l,n}|_{\gamma_f}}_{\sigma_p} \\ \tau^T \sigma_f n = 0 \end{cases} \quad \text{in} \quad \Sigma_l, \tag{7}$$

where, in addition to the quantities introduced above, ρ_s denotes the solid density, d stands for the solid displacement and σ_s denotes the tensor of linear elasticity

$$\sigma_s = \frac{\lambda_s}{2}\left(\nabla d + \nabla d^T\right) + \frac{\mu_s}{2}\text{tr}\left(\nabla d + \nabla d^T\right).$$

In addition, we impose that the solid Ω_s cannot penetrate into the porous medium Σ_l

$$d_n - g \leq 0, \quad \lambda \leq 0, \quad \lambda(d_n - g) = 0 \quad \text{on } \Sigma(t). \tag{8}$$

Here, g denotes the gap function to Σ_l and λ is a Lagrange multiplier for the no-penetration condition defined by

$$\lambda = \sigma_{s,nn} - \sigma_{f,nn} \qquad \text{on } \Sigma(t) \setminus \Sigma_l,$$
$$\lambda = \sigma_{s,nn} - \sigma_p \qquad \text{on } \Sigma(t) \cap \Sigma_l.$$

The "switch" on the right-hand side occurs, as the solid on one side of $\Sigma(t)$ couples either to the fluid Ω_f or the porous medium Σ_l on the other side of $\Sigma(t)$. The conditions (8) can equivalently be written as

$$\lambda = -\gamma_C \big[\underbrace{d_n - g - \gamma_C^{-1}\lambda}_{P_\gamma}\big]_+ \quad \text{on } \Sigma(t)$$

for arbitrary $\gamma_C > 0$. Using this notation, we can characterise the zone of "active" contact as follows

$$\Sigma_c(t) = \left\{x \in \Sigma(t) \mid P_\gamma > 0\right\}.$$

To summarise, we have the following interface conditions:

- Contact condition on $\Sigma(t)$:

$$d_n - g \leq 0, \quad \lambda \leq 0, \quad \lambda(d_n - g) = 0 \quad \text{on} \quad \Sigma(t).$$

- Kinematic coupling on $\Sigma_{fsi}(t) = \Sigma(t) \backslash \Sigma_l$

$$u_f = \dot{d} \quad \text{on} \quad \Sigma_{fsi}(t).$$

- Dynamic coupling on $\Sigma(t)$:

$$\begin{aligned} \sigma_s n &= -\lambda n + \sigma_p n = -\gamma_C [P_\gamma]_+ n + \sigma_p n \quad \text{on } \Sigma(t) \cap \Sigma_l, \\ \sigma_s n &= -\lambda n + \sigma_f n = -\gamma_C [P_\gamma]_+ n + \sigma_f n \quad \text{on } \Sigma(t) \setminus \Sigma_l. \end{aligned}$$

We have the following Nitsche-based variational formulation: *Find* $u_f \in \mathcal{V}_f$, $p_f \in \mathcal{L}_f$, $d \in \mathcal{V}_s$, $P_l \in \mathcal{V}_l$ such that

$$\begin{aligned} &(\partial_t u_f, v)_{\Omega_f} + (\partial_t \dot{d}, w)_{\Omega_s} + (\sigma_f(u_f, p_f), \nabla v_f)_{\Omega_f} + (\sigma_s(d), \nabla w)_{\Omega_s} \\ &- (\sigma_f n, v - w)_{\Sigma(t) \backslash \Sigma_l} - (u_f - \dot{d}, \sigma_f(v, -q))_{\Sigma(t) \backslash \Sigma_l} + \frac{\gamma_{\text{fsi}}}{h}(u_f - \dot{d}, v - w)_{\Sigma(t) \backslash \Sigma_l} \\ &- (\sigma_p, v \cdot n)_{\Sigma_l \backslash \Sigma(t)} - (\sigma_p, w \cdot n)_{\Sigma_l \cap \Sigma(t)} + \big(\gamma_C [P_\gamma]_+, w \cdot n\big)_{\Sigma(t)} \\ &+ (\epsilon K_\tau \nabla_\tau P_l, \nabla_\tau q_l)_{\Sigma_l} - \big(u_{f,n}, q_l\big)_{\Sigma_l \backslash \Sigma(t)} - \big(\dot{d}_n, q_l\big)_{\Sigma_l \cap \Sigma(t)} = 0 \\ &\qquad \forall v \in \mathcal{V}_f, q \in \mathcal{L}_f, w \in \mathcal{V}_s, q_l \in \mathcal{V}_l, \end{aligned}$$

where $\mathcal{V}_f$, $\mathcal{L}_f$, $\mathcal{V}_s$ and $\mathcal{V}_l$ are suitable finite element spaces. The porous stress σ_p is given by

$$\sigma_p = -P_l + \frac{\epsilon K_n^{-1}}{4} u_{l,n}|_{\gamma_f} = \begin{cases} -P_l + \frac{\epsilon K_n^{-1}}{4} u_{f,n} & \text{on } \Sigma_l \setminus \Sigma(t) \\ -P_l + \frac{\epsilon K_n^{-1}}{4} \dot{d}_n & \text{on } \Sigma_l \cap \Sigma(t). \end{cases} \tag{9}$$

4 Numerical Experiments

Here we will report on some numerical experiments using the above models. First we consider the mixed dimensional Stokes'-Darcy system and then the fluid-structure interaction system with contact and porous layer in the contact zone.

4.1 Stokes-Darcy Example

In this example, we consider two disconnected fluid reservoirs, the domain Ω_f, connected through a thin-walled porous media located on the bottom wall Σ_l, as shown in Fig. 1. The physical parameters are $\mu = 0.03$, $\rho_f = 1$, $\epsilon = 0.01$ and $K_\tau = K_n = 1$. We impose a pressure drop across the two parts of the boundary Γ_f^N. The purpose of this example is to illustrate how the porous model is able to connect the fluid flow between the two containers. This can be clearly inferred from the results reported in Fig. 2, which respectively show a snapshot of the fluid velocity, the elevation of the fluid pressure and the associated porous pressure.

4.2 Fluid-Structure Interaction with Contact

To test the FSI-contact model, we consider flow in a two-dimensional pipe, where the upper wall is elastic, see Fig. 3. Due to the application of a large pressure $\overline{P}$ on the left and right boundary, the upper wall is deflected downwards until it reaches the bottom. Note that when contact occurs, the configuration is topologically equivalent to the situation in Sect. 4.1. Shortly before the time of impact we set $\overline{P}$ to zero, such that contact is released again after a certain time. This model problem is taken

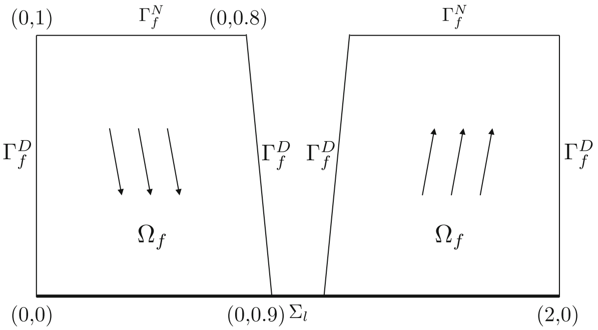


Fig. 1 Geometrical configuration for the Stokes model with a thin-walled porous medium on the bottom wall

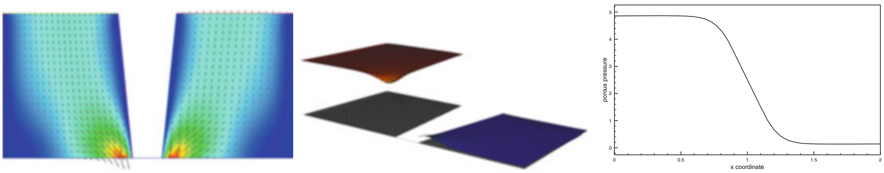


Fig. 2 Left: Snapshot of the fluid velocity. Middle: Elevation of the fluid pressure. Right: Porous pressure

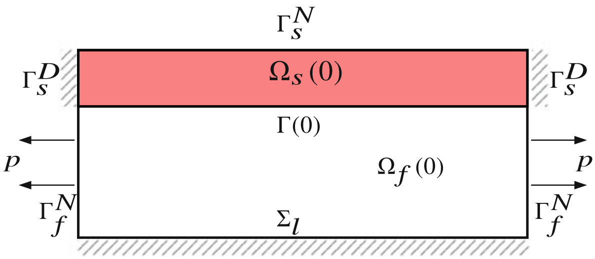


Fig. 3 Geometrical configuration for the FSI-contact model. We apply a porous medium model on the (rigid) lower wall, where contact might take place

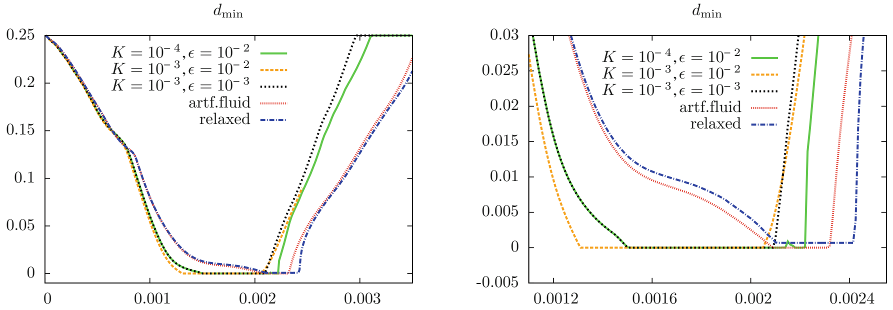


Fig. 4 Minimal distance of Ω_s to the lower wall Σ_p over time. Right: zoom-in around the contact interval. We compare the new approach presented in Sect. 3 for different parameters with the *artificial fluid* and the *relaxed* contact approach studied in [4]

from [4], where further details on the configuration and the discretisation can be found. To deal with the topology change in the fluid domain at the impact, we apply a *Fully Eulerian* approach for the FSI problem [2]. In order to obtain a continuous and physically relevant transition from FSI to solid-solid contact, we use the FSI-contact model derived in Sect. 3 and place a thin porous domain Σ_l on the lower boundary.

In Fig. 4 we compare this model for different parameters $K = K_\tau = K_n$ and ϵ with the approaches for FSI-contact problems introduced in [4] in terms of the minimal distance of the solid to Σ_p over time. In [4] two approaches were presented in order to extend the fluid stresses to the contact region during solid-solid contact, namely a so-called *relaxed* and an *artificial fluid* approach. It was observed that for the *artificial fluid* approach contact happens earlier, as penetration of the fluid flow into the artificial region is prevented only asymptotically, i.e. $u_{f,n} \to 0$ $(h \to 0)$ on Σ_p, in contrast to $u_{f,n} = 0$ for the relaxed approach. In the model presented here, we have similarly from (7) and $u_{l,n} = u_{f,n}$ on Σ_p

$$u_{f,n} = -\nabla_\tau \cdot (\epsilon K_\tau \partial_\tau P_l) \to 0 \quad (\epsilon K_\tau \to 0).$$

For this reason we observe in Fig. 4 that the impact happens earlier for a larger value of ϵK_τ. The time of the release seems to depend also on ϵK_n^{-1}, which appears in the definition of σ_p (9). A detailed investigation of this dependence and the investigation of stability and convergence of the numerical method are subject to future work.

Acknowledgments Erik Burman was partially supported by the EPSRC grant: EP/P01576X/1. Stefan Frei acknowledges support by the DFG Research Scholarship FR3935/1-1.

References

1. C Ager, B Schott, AT Vuong, A Popp, and WA Wall. A consistent approach for fluid-structure-contact interaction based on a porous flow model for rough surface contact. *Int J Numer Methods Eng*, 119(13):1345–1378, 2019.
2. S Frei. *Eulerian finite element methods for interface problems and fluid-structure interactions*. PhD thesis, Heidelberg University, 2016. http://www.ub.uni-heidelberg.de/archiv/21590.
3. V Martin, J Jaffré, and JE Roberts. Modeling fractures and barriers as interfaces for flow in porous media. *SIAM J. Sci. Comput.*, 26(5):1667–1691, 2005.
4. E Burman S Frei and MA Fernández. Nitsche-based formulation for fluid-structure interactions with contact. *ESAIM: M2AN (published online)*. https://doi.org/10.1051/m2an/2019072.

A Second Order Time Integration Method for the Approximation of a Parabolic 2D Monge-Ampère Equation

Alexandre Caboussat and Dimitrios Gourzoulidis

Abstract Parabolic fully nonlinear equations may be found in various applications, for instance in optimal portfolio management strategy. A numerical method for the approximation of a canonical parabolic Monge-Ampère equation is investigated in this work. A second order semi-implicit time-stepping method is presented, coupled to safeguarded Newton iterations A low order finite element method is used for space discretization. Numerical experiments exhibit appropriate convergence orders and a robust behavior.

1 Introduction

Fully nonlinear equations, and among them the elliptic Monge-Ampère equation, have raised a lot of interest from the theoretical and numerical communities [1, 7, 9, 10], and also from the authors [4, 6]. We focus here on a time-evolutive, parabolic, Monge-Ampère equation that has raised much less attention from a computational perspective. Some known applications of interest arise, e.g., in finance [12], or in mesh adaptation techniques [2, 3]. Numerical results for parabolic fully nonlinear equations, including the equation that we study here, are given, e.g., in [8].

The purpose of this work is to introduce a second-order semi-implicit numerical scheme for the approximation of the time-evolutive Monge-Ampère equation. It extends the Newton-based approaches in [1, 10] to the non-stationary case by means

A. Caboussat (✉)
Geneva School of Business Administration, University of Applied Sciences and Arts Western Switzerland (HES-SO), Geneva, Switzerland
e-mail: alexandre.caboussat@hesge.ch

D. Gourzoulidis
Geneva School of Business Administration, University of Applied Sciences and Arts Western Switzerland (HES-SO), Geneva, Switzerland

Institute of Mathematics, Ecole Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
e-mail: dimitrios.gourzoulidis@hesge.ch; dimitrios.gourzoulidis@epfl.ch

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_21

of a midpoint time-stepping algorithm. Continuous, low order, finite elements are used for the space discretization. Numerical validation is achieved with simple examples, and appropriate convergence results are obtained from a computational perspective.

2 Model Problem

Let Ω be a smooth bounded convex domain of $\mathbb{R}^2$, and $T > 0$ a fixed time horizon. We consider a time evolutive two-dimensional Monge-Ampère equation, with Dirichlet boundary conditions, which reads as follows: find $u : \Omega \times (0, T) \to \mathbb{R}$ satisfying

$$\begin{cases} \dfrac{\partial u}{\partial t} - \det \mathbf{D}^2 u = f & \text{in } \Omega \times (0, T), \\ u = g & \text{in } \partial\Omega \times (0, T), \\ u(0) = u_0 & \text{in } \Omega. \end{cases} \tag{1}$$

Here $f = f(\mathbf{x}, t)$, $g = g(\mathbf{x}, t)$ and $u_0 = u_0(\mathbf{x})$ are given functions with the required regularity, and $\mathbf{D}^2 u (:= \mathbf{D}^2_{\mathbf{x}} u)$ is the Hessian of the unknown function u (with respect to the space variable $\mathbf{x}$), defined by $\mathbf{D}^2 u = (D^2_{ij} u)_{1 \le i,j \le 2}$, and $D^2_{ij} u = \dfrac{\partial^2 u}{\partial x_i \partial x_j}$.

We assume in the sequel that u_0 is convex, in order to favor the regularity of a smooth transient. A constraint on the time step may have to be enforced to make sure that the numerical solution remains convex at all times. Numerical results will show that the right-hand side f may change sign, as long as the numerical solution remains convex and the operator in the parabolic Monge-Ampère equation remains coercive. Following [9], the Monge-Ampère operator can be rewritten under a divergence form, namely

$$\det \mathbf{D}^2 u = \frac{1}{2} \nabla \cdot \left(\operatorname{cof}(\mathbf{D}^2 u) \nabla u \right).$$

The differential operator of (1) can thus be written as

$$\frac{\partial u}{\partial t} - \frac{1}{2} \nabla \cdot \left(\operatorname{cof}(\mathbf{D}^2 u) \nabla u \right) = f \quad \text{in } \Omega \times (0, T), \tag{2}$$

meaning that (1) can be interpreted as a, strongly nonlinear, parabolic equation reminiscent of a nonlinear heat equation. When looking for a convex solution, if the nonlinearity $\operatorname{cof}(\mathbf{D}^2 u)$ remains positive definite, then the operator is well-posed. The challenge becomes thus to capture convex solutions, and to derive numerical methods that take into account accurately the strongly nonlinear diffusion and guarantee the coercivity of the diffusion operator at all times.

Remark 1 In [9], an alternative formulation is considered, which consists in augmenting the differential equation into a differential system. This approach has proved to be very efficient in capturing a stationary solution. However, numerical experiments have shown that it is not efficient to approximate the whole transient trajectory of the evolutive problem.

In the sequel, we thus propose a second-order numerical method for the numerical approximation of the solution of (1), which relies on an implicit time-stepping scheme and a Newton's method.

3 Numerical Algorithm

Let $\Delta t > 0$ be a constant given time step, $t^n = n\Delta t$, $n = 1, 2, \ldots$, to define the approximations $u^n \simeq u(t^n)$. The numerical algorithm proposed hereafter relies on a discretization of the formulation (1). In order to handle the stiff behavior of the Monge-Ampère equation, a semi-implicit time discretization of (1) is considered. In this case, we advocate a *midpoint rule* and, u^n being known, we look for the next time step approximation u^{n+1} satisfying

$$\frac{u^{n+1} - u^n}{\Delta t} - \det\left(\mathbf{D}^2 u^{n+1/2}\right) = f^{n+1/2} \quad n = 0, 1, \ldots, \tag{3}$$

where $u^{n+1/2} := \dfrac{u^{n+1} + u^n}{2}$ and $f^{n+1/2} := f\left(\dfrac{t^{n+1} + t^n}{2}\right)$. Then (3) can be written as

$$u^{n+1/2} - \frac{1}{2}\Delta t \det \mathbf{D}^2 u^{n+1/2} = u^n + \frac{1}{2}\Delta t f^{n+1/2}, \tag{4}$$

and

$$u^{n+1} = 2u^{n+1/2} - u^n. \tag{5}$$

Let us define $b^n := u^n + \frac{1}{2}\Delta t f^{n+1/2}$. Relationship (4) is rewritten at each time step as

$$F(u^{n+1/2}) := u^{n+1/2} - \frac{\Delta t}{2}\det(\mathbf{D}^2 u^{n+1/2}) - b^n = 0.$$

This nonlinear problem is solved with a safeguarded Newton method at each time step. For the ease of notation, we denote $u^{n+1/2}$ by v. Starting from the initial guess $v^0 = u^n$, the increments δv^k of the Newton method are obtained by solving

$$DF(v^k)\delta v^k = -F(v^k), \quad k = 0, 1, 2, \ldots, \tag{6}$$

then, the next iterate is given by $v^{k+1} = v^k + \delta v^k$, until some stopping criterion is satisfied at step M, and set $u^{n+1/2} := v^M$. At the end of the Newton loop, the approximation of the solution at the next time step is given by (5). In order to write the variational formulation corresponding to (6) we use the following identity which holds for 2×2 symmetric matrices (see, e.g., [1]):

$$\det \mathbf{D}^2(a+b) = \det(\mathbf{D}^2 a) + \det(\mathbf{D}^2 b) + \mathrm{tr}(A^* \mathbf{D}^2 b), \tag{7}$$

where $A^* = \mathrm{cof}(\mathbf{D}^2 a) = \det(\mathbf{D}^2 a)(\mathbf{D}^2 a)^{-1}$. This yields

$$\mathrm{tr}(A^* \mathbf{D}^2 b) = \mathrm{cof}(\mathbf{D}^2 a)\mathbf{:}\mathbf{D}^2 b = \nabla \cdot (\mathrm{cof}(\mathbf{D}^2 a)\nabla b),$$

where $\mathbf{A} : \mathbf{B} := \mathrm{tr}(\mathbf{A}^T \mathbf{B})$ is the Frobenius inner product for $\mathbf{A}$, $\mathbf{B} \in \mathbb{R}^{2\times 2}$. Equation (7) becomes,

$$\det \mathbf{D}^2(a+b) = \det(\mathbf{D}^2 a) + \nabla \cdot (\mathrm{cof}(\mathbf{D}^2 a)\nabla b) + \det(\mathbf{D}^2 b). \tag{8}$$

We thus have, for $s \in \mathbb{R}$,

$$F(v^k + s\delta v) = v^k + s\delta v - \frac{\Delta t}{2}\Big(\det(\mathbf{D}^2 v^k) + \nabla \cdot (\mathrm{cof}(\mathbf{D}^2 v^k) s \nabla \delta v) + s^2 \det(\mathbf{D}^2 \delta v)\Big) - b^n.$$

We thus compute $DF(v^k)$ as follows:

$$DF(v^k)\delta v = \lim_{s\to 0} \frac{F(v^k + s\delta v) - F(v^k)}{s} = \delta v - \frac{\Delta t}{2}\nabla \cdot \Big(\mathrm{cof}(\mathbf{D}^2 v^k)\nabla \delta v\Big). \tag{9}$$

In order to incorporate (9) in the variational formulation corresponding to (6), let us define $V_g = \big\{w \in H^1(\Omega) : w|_{\partial\Omega} = g\big\}$, and $V_0 = H_0^1(\Omega)$. Using (9), the variational formulation corresponding to the Newton system (6) can be explicited into : find $\delta v^k \in V_0$, for $k = 0, 1, 2, \ldots$, such that

$$\int_\Omega \delta v^k w d\mathbf{x} + \frac{\Delta t}{2}\int_\Omega \mathrm{cof}(\mathbf{D}^2 v^k)\nabla(\delta v^k) \cdot \nabla w d\mathbf{x} = \\ -\int_\Omega \left(v^k - \frac{\Delta t}{2}\det(\mathbf{D}^2 v^k) - b^n\right) w d\mathbf{x}, \tag{10}$$

for all $w \in V_0$. This Newton's variational problem is coupled with a safeguarding strategy (Armijo's rule) when needed. In addition, the method guarantees that the matrix $\mathrm{cof}(\mathbf{D}^2 v^k)$ remains positive definite. This procedure is achieved by computing the SVD of this matrix, and truncating its negative eigenvalues to zero.

4 Finite Element Discretization

In order to avoid the construction of finite element sub-spaces of $H^2(\Omega)$ and to handle arbitrary shaped domains, we consider a mixed low order finite element method for the approximation of (10) see, e.g., [4, 6]. Let us thus denote by $\mathcal{T}_h$ a regular finite element discretization of $\Omega \subset \mathbb{R}^2$ in triangles. From $\mathcal{T}_h$, we approximate the spaces $L^2(\Omega)$, $H^1(\Omega)$ and $H^2(\Omega)$, respectively $H_0^1(\Omega)$ and

$H^2(\Omega) \cap H_0^1(\Omega)$, by the finite dimensional space V_h, respectively V_{0h}, defined by:

$$V_h = \left\{ v \in C^0\left(\overline{\Omega}\right),\ v|_K \in \mathbb{P}_1,\ \forall K \in \mathcal{T}_h \right\}, \qquad V_{0,h} = V_h \cap H_0^1(\Omega), \tag{11}$$

with $\mathbb{P}_1$ the space of the two-variables polynomials of degree one. Moreover, let us define $V_{g,h} = \left\{ v \in C^0\left(\overline{\Omega}\right),\ v|_K \in \mathbb{P}_1,\ \forall K \in \mathcal{T}_h,\ v|_{\partial\Omega} = g \right\}$. As in [6], for a function φ being given in $H^1(\Omega)$, we approximate the differential operators D^2_{ij} by D^2_{hij}, for $1 \leq i, j \leq 2$, defined by $D^2_{hij}(\varphi) \in V_{0h}$ and

$$\int_\Omega D^2_{hij}(\varphi) v d\mathbf{x} = -\frac{1}{2} \int_\Omega \left[\frac{\partial \varphi}{\partial x_i} \frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j} \frac{\partial v}{\partial x_i} \right] d\mathbf{x},\ \forall v \in V_{0h}. \tag{12}$$

As emphasized in [11], the a priori estimates for the error on the second derivatives of the solution φ are, in general, $O(1)$ in the L^2-norm when using piecewise linear mixed finite elements. Therefore the convergence properties of the solution method depend strongly on the type of triangulations one employs. To cure the non-convergence properties associated with the approximations of $D^2_{hij}(\varphi)$, we use a regularization procedure as in [6], and we replace (12) by: find $D^2_{hij}(\varphi) \in V_{0h}$, $1 \leq i, j \leq 2$, such that

$$\begin{aligned} \int_\Omega D^2_{hij}(\varphi) v d\mathbf{x} + C \sum_{K \in \mathcal{T}_h} |K| \int_K \nabla D^2_{hij}(\varphi) \cdot \nabla v d\mathbf{x} = \\ -\frac{1}{2} \int_\Omega \left[\frac{\partial \varphi}{\partial x_i} \frac{\partial v}{\partial x_j} + \frac{\partial \varphi}{\partial x_j} \frac{\partial v}{\partial x_i} \right] d\mathbf{x}, \end{aligned}$$

where $C \geq 0$ and $|K| = \text{meas}(K)$. Set u_h^0 be an approximation of u^0 in $V_{g,h}$. At each time step, the numerical approximation of (10) is computed as follows: let $v_h^0 := u_h^n$ at each time iteration; then, for $k = 0, 1, 2, \ldots$, we search for $\delta v_h^k \in V_{0,h}$ such that:

$$\begin{aligned} \int_{\Omega_h} \delta v_h^k w_h d\mathbf{x} + \frac{\Delta t}{2} \int_{\Omega_h} \text{cof}(\mathbf{D}^2 v_h^k) \nabla(\delta v_h^k) \cdot \nabla w_h d\mathbf{x} = \\ -\int_\Omega \left(v_h^k - \frac{\Delta t}{2} \det(\mathbf{D}^2 v_h^k) - b_h^n \right) w_h d\mathbf{x}, \end{aligned} \tag{13}$$

for all $w_h \in V_{0,h}$. Then we set $v_h^{k+1} := v_h^k + \delta v_h^k$; when some stopping criterion is satisfied at step M, we set $u_h^{n+1/2} := v_h^M$. To progress to the next time step, we compute $u_h^{n+1} = 2u_h^{n+1/2} - u_h^n$.

5 Numerical Experiments

Numerical results are presented to validate the method for convex solutions. In the following examples, $\Omega = (0, 1)^2$ and $T = 1$. Both a triangular structured asymmetric mesh and an unstructured isotropic mesh are used. The mesh size h and the time step Δt vary together. The stopping criterion for the Newton method is $||v_h^{k+1} - v_h^k||_{L_2(\Omega)} \leq 10^{-12}$, with a maximal number of 200 Newton iterations. The Newton method typically needs 9–12 iterations to converge, depending on the mesh size and the time step. The parameter C is set to 1 (unless specified otherwise). The convergence of the error $e = u - u_h$ is quantified by the following quantities

$$||e||_{L^2(L^2)} := \int_0^T \|u - u_h\|_{L^2}\, dt, \quad ||e||_{L^2(H^1)} := \int_0^T \|\nabla u - \nabla u_h\|_{L^2}\, dt,$$

In the tables below, those norms are approximated using the trapezoidal rule in time, and quadrature formulas in space (see [5]).

5.1 A Polynomial Example

Let us consider $T = 1$, and the exact solution:

$$u(x, y, t) = 0.5\,(0.5 + t)\,(x^2 + 5y^2), \qquad (x, y) \in \Omega, \quad t \in (0, T)\,. \tag{14}$$

This function is the solution of (1) with the data $f(x, y, t) := 0.5\left(x^2 + 5y^2\right) - 5\,(0.5 + t)^2$, $g(x, y, t) := 0.5\,(0.5 + t)\,(x^2+5y^2)$, and $u_0(x, y) := 0.25(x^2+5y^2)$. The solution (14) is convex for all $t \in (0, T)$. Note that the eigenvalues of the Hessian $\mathbf{D}^2 u$ are $\lambda_1 = (0.5 + t)^2$ and $\lambda_2 = 5\,(0.5 + t)^2$, and are both positive for all $t \in (0, T)$. Figure 1 illustrates $u_{0,h}(x, y)$ (left) and $u_h(x, y, T)$ (right), while Table 1 shows that the solution method exhibits appropriate convergence orders (for the discrete version of the norms $||u - u_h||_{L^2(0,T;H^1(\Omega))}$ and $||u - u_h||_{L^2(0,T;L^2(\Omega))}$).

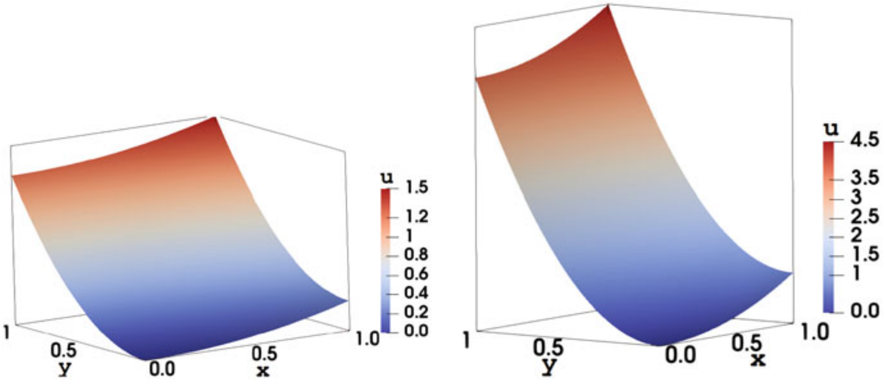


Fig. 1 A polynomial example corresponding to the exact solution (14). Numerical approximation of the solution for $h = 1/80$ and $\Delta t = 0.25 \cdot 10^{-3}$. Left: initial condition at time $t = 0$. Right: final solution at time $t = 1$

Table 1 A polynomial example. Estimated errors of $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt. Left: structured meshes (with $C = 0$), right: unstructured meshes

h	Δt	$\|\|e\|\|_{L^2(L^2)}$		$\|\|e\|\|_{L^2(H^1)}$		h	Δt	$\|\|e\|\|_{L^2(L^2)}$		$\|\|e\|\|_{L^2(H^1)}$	
1/20	1.00e−03	1.55e−03	–	7.37e−02	–	0.062	2.00e−03	2.10e−02	–	3.19e−01	–
1/40	0.50e−03	3.81e−04	2.02	3.68e−02	1.00	0.031	1.00e−03	7.28e−03	1.52	1.51e−01	1.07
1/80	0.25e−03	9.01e−05	2.08	1.84e−02	1.00	0.015	0.50e−03	1.90e−03	1.93	6.14e−02	1.29
1/160	0.125e−03	1.99e−05	2.17	9.20e−03	1.00	0.010	0.33e−03	8.29e−04	2.04	3.49e−02	1.39

5.2 An Exponential Example

Let us consider $T = 1$, and the exact solution

$$u(x, y, t) = e^{-t} e^{\frac{1}{2}(x^2+y^2)}, \qquad (x, y) \in \Omega, \;\; t \in (0, T)\,. \tag{15}$$

This function is the solution of (1) with the data

$$f(x, y, t) := -e^{-t} e^{\frac{1}{2}(x^2+y^2)} \left(1 + e^{-t}\left(x^2 + y^2 + 1\right) e^{\frac{1}{2}(x^2+y^2)}\right),$$

together with $g(x, y, t) := e^{-t} e^{\frac{1}{2}(x^2+y^2)}$, and $u_0(x, y) := e^{\frac{1}{2}(x^2+y^2)}$. The solution (15) is convex for all time $t \in (0, T)$, since the eigenvalues of $\mathbf{D}^2 u$ are $\lambda_1 = e^{-t}\mathrm{e}^{\frac{1}{2}(x^2+y^2)}$, and $\lambda_2 = e^{-t}\mathrm{e}^{\frac{1}{2}(x^2+y^2)}\left(x^2 + y^2 + 1\right)$, which are both positive for all times $t \in (0, T)$. Figure 2 illustrates $u_{0,h}(x, y)$ (left) and $u_h(x, y, T)$ (right), while Table 2 shows that the solution method exhibits nearly optimal convergence orders (for structured and unstructured mesh we have $\mathcal{O}(h)$ and $\mathcal{O}(h^{1.5})$ for the discrete version of the norm $||e||_{L^2(H^1)}$ and $\mathcal{O}(h^{1.8})$ and $\mathcal{O}(h^2)$ for $||e||_{L^2(L^2)}$, respectively).

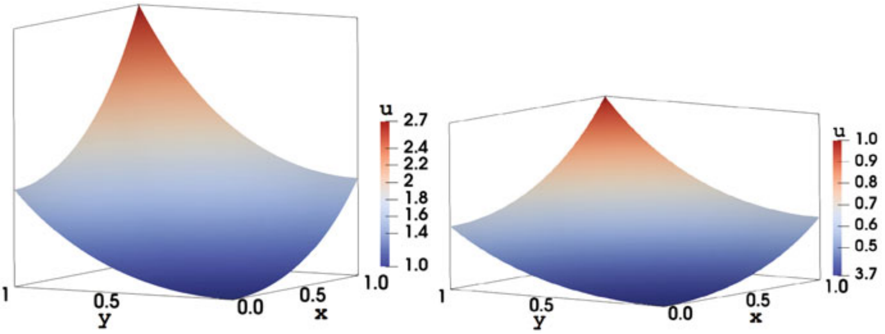


Fig. 2 Exponential example corresponding to the exact solution (15). Numerical approximation of the solution for $h = 1/80$ and $\Delta t = 0.25 \cdot 10^{-3}$. Left: initial condition at time $t = 0$. Right: the final solution at time $t = 1$

Table 2 Exponential example. Estimated errors of $u - u_h$ in corresponding norms, and related convergence orders for various h and Δt. Left: structured meshes (with $C = 0$ when $h \geq 1/80$, and $C = 0.1$ when $h = 1/160$), right: unstructured meshes

h	Δt	$\|\|e\|\|_{L^2(L^2)}$		$\|\|e\|\|_{L^2(H^1)}$		h	Δt	$\|\|e\|\|_{L^2(L^2)}$		$\|\|e\|\|_{L^2(H^1)}$	
1/20	1.00e−03	8.96e−04	–	3.58e−02	–	0.062	2.00e−03	1.49e−02	–	2.02e−01	–
1/40	0.50e−03	2.40e−04	1.90	1.79e−02	1.00	0.031	1.00e−03	5.31e−03	1.48	8.93e−02	1.17
1/80	0.25e−03	6.69e−05	1.80	8.96e−03	0.99	0.015	0.50e−03	1.25e−03	2.08	3.27e−02	1.44
1/160	0.125e−03	9.97e−06	2.74	4.44e−03	1.01	0.010	0.33e−03	5.26e−04	2.13	1.81e−02	1.45

Acknowledgments This work was partially supported by the Swiss National Science Foundation (Grant #165785). The authors thank Prof. Marco Picasso (EPFL), and Prof. R. Glowinski (Univ. of Houston/Hong Kong Baptist Univ.) for helpful comments and discussions.

References

1. S. C. Brenner and M. Neilan. Finite element approximations of the three dimensional Monge-Ampère equation. *ESAIM: M2AN*, 46(5):979–1001, 2012.
2. C. J. Budd, M. J. P. Cullen, and E. J. Walsh. Monge-Ampère based moving mesh methods for numerical weather prediction, with applications to the Eady problem. *J. Comput. Phys.*, 236:247–270, 2013.
3. C.J. Budd and J. F. Williams. Moving mesh generation using the parabolic Monge-Ampère equation. *SIAM J. Sci. Comput.*, 31:3438–3465, 2009.
4. A. Caboussat, R. Glowinski, and D. Gourzoulidis. A least-squares/relaxation method for the numerical solution of the three-dimensional elliptic Monge-Ampère equation. *J. Sci. Comp.*, 77:53–78, 2018.
5. A. Caboussat, R. Glowinski, D. Gourzoulidis, and M. Picasso. Numerical approximation of orthogonal maps. *SIAM J. Sci. Comput.*, 41:B1341–B1367, 2019.
6. A. Caboussat, R. Glowinski, and D. C. Sorensen. A least-squares method for the numerical solution of the Dirichlet problem for the elliptic Monge-Ampère equation in dimension two. *ESAIM: Control, Optimization and Calculus of Variations*, 19(3):780–810, 2013.
7. X. Feng, R. Glowinski, and M. Neilan. Recent developments in numerical methods for fully nonlinear second order partial differential equations. *SIAM Review*, 55(2):205–267, 2013.
8. X. Feng and Th. Lewis. Nonstandard local discontinuous Galerkin methods for fully nonlinear second order elliptic and parabolic equations in high dimensions. *J. Sci. Comp.*, 77(3):1534–1565, Dec 2018.
9. R. Glowinski, H. Liu, S. Leung, and J. Qian. A finite element/operator-splitting method for the numerical solution of the two dimensional elliptic Monge-Ampère equation. *J. Sci. Comp.*, 79:1–47, 2018.
10. G. Loeper and F. Rapetti. Numerical solution of the Monge-Ampère equation by a Newton's algorithm. *C. R. Math. Acad. Sci. Paris*, 340(4):319–324, 2005.
11. M. Picasso, F. Alauzet, H. Borouchaki, and P.-L. George. A numerical study of some Hessian recovery techniques on isotropic and anisotropic meshes. *SIAM J. Sci. Comp.*, 33(3):1058–1076, 2011.
12. S. Stojanovic. Optimal momentum hedging via Monge-Ampère PDEs and a new paradigm for pricing options. *SIAM J. Control and Optimization*, 43:1151–1173, 2004.

Local Flux Reconstruction for a Frictionless Unilateral Contact Problem

Daniela Capatina and Robert Luce

Abstract We are interested in the a posteriori error analysis based on locally reconstructed fluxes for the 2D Signorini problem. We start from a P^1-conforming approximation where the contact condition is treated by means of a Nitsche method. We propose an extension of a general approach previously developed for the Laplace operator, allowing to obtain $H(div)$-conforming conservative fluxes by a local post-process. The reconstructed flux yields an a posteriori error indicator, which is completed by two additional terms taking into account the non-linear contact condition. We then prove the reliability of the indicator, without any additional assumption.

1 Introduction

We are interested in the numerical approximation of the 2D frictionless unilateral contact problem, modelled by Signorini's equations. Different formulations exist in the literature (mixed/hybrid, stabilized, penalty methods etc.), most of them treating the contact condition by means of a variational inequality. In general, they are suboptimal or need additional assumptions to reach optimality. In this paper, we consider the Nitsche-type formulation introduced in [2] and its P^1-continuous finite element approximation, for which the authors proved an optimal a priori error estimate.

As regards the a posteriori analysis, residual-based error estimators for the previous Nitsche formulation were proposed in [3]. However, the error analysis is carried out under a saturation assumption. Our goal is twofold: on the one hand, reconstruct locally a conservative flux and on the other hand, define a reliable a posteriori error estimator based on this flux. This kind of approach is widely studied in the literature, see [1] and references therein for the Laplace problem. It has

D. Capatina (✉) · R. Luce
Université de Pau et des Pays de l'Adour, E2S UPPA, CNRS, LMAP, Pau, France
e-mail: daniela.capatina@univ-pau.fr; robert.luce@univ-pau.fr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_22

also been applied to the contact problem, see for instance [4, 5] where a mixed formulation with variational inequality is considered; however, the error estimator contains a higher order term depending on the unknown solution.

Here, we first extend the framework proposed in [1] to the non-linear Signorini problem, by treating the contact boundary as a Neumann one. The flux computation is achieved by local post-processing of the finite element solution, without solving any mixed problem. Then we use the reconstructed flux to define a standard a posteriori indicator, to which we add two more terms on the contact boundary. This allows us to establish the reliability of the error indicator without any saturation assumption.

2 Model Problem and Discrete Formulation

We consider here the scalar Signorini problem in a polygonal bounded domain $\Omega \subset \mathbb{R}^2$ of boundary $\partial\Omega = \Gamma^{\mathrm{D}} \cup \Gamma^{\mathrm{N}} \cup \Gamma^{\mathrm{C}}$, with Γ^{D}, Γ^{N} and the contact boundary Γ^{C} disjoint, and with $|\Gamma^{\mathrm{D}}| > 0$, $|\Gamma^{\mathrm{C}}| > 0$. The boundary value problem is given by:

$$-\Delta u = f \quad \text{in } \Omega, \qquad u = u^{\mathrm{D}} \quad \text{on } \Gamma^{\mathrm{D}}, \qquad \partial_n u = g \quad \text{on } \Gamma^{\mathrm{N}}$$

$$u \le 0, \quad \partial_n u \le 0, \quad u\,\partial_n u = 0 \quad \text{on } \Gamma^{\mathrm{C}} \tag{1}$$

where (1) are the frictionless unilateral contact conditions. We take $f \in L^2(\Omega)$, $g \in L^2(\Gamma^{\mathrm{N}})$ and we assume that the Dirichlet data u^{D} is continuous, piecewise linear on Γ^{D} and vanishes at $\overline{\Gamma^{\mathrm{D}}} \cap \overline{\Gamma^{\mathrm{C}}}$ (if non-empty).

It is important to note that conditions (1) are equivalent to $(u - \alpha\partial_n u)_- = u$ or to $(u - \alpha\partial_n u)_+ = -\alpha\partial_n u$ for a given $\alpha > 0$, where a_+ and a_- stand for the positive and the negative part of $a \in \mathbb{R}$, respectively. This remark is used in the derivation of the Nitsche's formulation introduced in [2] and considered here.

For the discretization, we use a regular family of triangular meshes. We denote by $\mathcal{K}_h$ the set of cells and by $\mathcal{S}_h^{\mathrm{int}}$, $\mathcal{S}_h^{\mathrm{D}}$, $\mathcal{S}_h^{\mathrm{N}}$ and $\mathcal{S}_h^{\mathrm{C}}$ the interior, Dirichlet, Neumann and contact sides, respectively. We put $\mathcal{S}_h = \mathcal{S}_h^{\mathrm{int}} \cup \mathcal{S}_h^{\mathrm{D}}$, $\mathcal{S}_h^{\partial} = \mathcal{S}_h^{\mathrm{N}} \cup \mathcal{S}_h^{\mathrm{C}}$ and we use similar notation for the nodes: $\mathcal{N}_h = \mathcal{N}_h^{\mathrm{int}} \cup \mathcal{N}_h^{\mathrm{D}}$ and $\mathcal{N}_h^{\partial}$ (for the nodes lying on $\overline{\Gamma^{\mathrm{N}}} \cup \overline{\Gamma^{\mathrm{C}}}$). We denote by π_ω^l the $L^2(\omega)$ orthogonal projection on $P^l(\omega)$. For $S = \partial K^{\mathrm{in}} \cap \partial K^{\mathrm{ex}}$, n_S is a fixed, arbitrary unit normal vector, oriented from K^{in} towards K^{ex}. For a discontinuous function v, we define its jump and mean on $S \in \mathcal{S}_h^{\mathrm{int}}$ by $[v] = v^{\mathrm{in}} - v^{\mathrm{ex}}$ and $\{v\} = \frac{1}{2}\left(v^{\mathrm{in}} + v^{\mathrm{ex}}\right)$; on a boundary side, we set $[u] = \{u\} = u_S^{\mathrm{in}}$.

The approximation of the contact problem is achieved by means of continuous, piecewise linear finite elements. In order to focus on the contact condition, the Dirichlet one is treated strongly. Let

$$V_h^{u^{\mathrm{D}}} = \left\{ v_h \in C^0(\bar{\Omega}) : \; v_h|_K \in P^1(K) \; \forall K \in \mathcal{K}_h, \; v_h = u^{\mathrm{D}} \text{ on } \Gamma^{\mathrm{D}} \right\},$$

$$A_h(u_h, v_h) = \int_{\mathcal{K}_h} \nabla u_h \cdot \nabla v_h - \int_{\mathcal{S}_h^{\mathrm{C}}} \frac{|S|}{\gamma} \partial_n u_h \partial_n v_h + \int_{\mathcal{S}_h^{\mathrm{C}}} \frac{|S|}{\gamma} P(u_h)_+ P(v_h),$$

$$L_h(v_h) = \int_{\Omega} f v_h + \int_{\Gamma^{\mathrm{N}}} g v_h,$$

where $\gamma > 0$ is a stabilisation parameter independent of the mesh size h and where

$$P(v) = \frac{\gamma}{|S|} v - \partial_n v, \quad \forall S \in \mathcal{S}_h^{\mathrm{C}}. \tag{2}$$

The discrete problem introduced in [2] reads: Find $u_h \in V_h^{u^{\mathrm{D}}}$ such that

$$A_h(u_h, v_h) = L_h(v_h), \quad \forall v_h \in V_h^0. \tag{3}$$

The authors proved the consistency and well-posedness of (3) for γ sufficiently large, as well as an optimal $O(h^{\frac{1}{2}+\nu})$ error estimate, for $u \in H^{\frac{3}{2}+\nu}(\Omega)$ with $0 < \nu \leq \frac{1}{2}$.

3 Definition of Locally Reconstructed Flux

We are interested in defining a discrete conservative flux $\sigma_h \in H(\mathrm{div}, \Omega)$ for problem (3), which can be computed patch-wise. For this purpose, we first write an equivalent mixed formulation and then we construct σ_h by using the Lagrange multiplier.

3.1 Equivalent Mixed Formulation

The general idea is inspired by the hybridisation of classical finite element methods. We dualize the continuity of u_h across the sides of $\mathcal{S}_h$ by means of Lagrange multipliers, and we thus obtain a mixed formulation where the primal unknown belongs to a completely discontinuous finite element space. Let

$$D_h = \left\{ v_h \in L^2(\Omega);\, v_h|_K \in P^1 \; \forall K \in \mathcal{K}_h \right\}, \; M_h = \left\{ v_h \in L^2(\mathcal{S}_h);\, v_h|_S \in P^1 \; \forall S \in \mathcal{S}_h \right\}$$

and the mixed problem: Find $U_h \in D_h$ and $\theta_h \in X_h \subset M_h$ such that

$$\begin{aligned} a_h(U_h, v_h) + b_h(\theta_h, v_h) &= l_h(v_h), \qquad \forall v_h \in D_h \\ b_h(\mu_h, U_h) &= j_h(\mu_h), \qquad \forall \mu_h \in X_h, \end{aligned} \tag{4}$$

where $a_h(u_h, v_h) = A_h(u_h, v_h) - \int_{\mathcal{S}_h} \{\partial_n u_h\}[v_h] - \int_{\mathcal{S}_h} [u_h]\{\partial_n v_h\}$ and $l_h(v_h) = L_h(v_h) - \int_{\mathcal{S}_h^{\mathrm{D}}} u^{\mathrm{D}} \partial_n v_h$. The forms $b_h(\cdot, \cdot)$ and $j_h(\cdot)$ approximate $\int_{\mathcal{S}_h} \mu_h[v_h]$ and $\int_{\Gamma^{\mathrm{D}}} u^{\mathrm{D}} \mu_h$ by means of the trapeze integration formula, which allow to locally compute θ_h. Thus, we define

$$b_h(\mu_h, v_h) = \sum_{S \in \mathcal{S}_h} \frac{|S|}{2} \sum_{i=1}^{2} (\mu_h[v_h])(N_S^i), \quad j_h(\mu_h) = \sum_{S \in \mathcal{S}_h^D} \frac{|S|}{2} \sum_{i=1}^{2} (u^{\mathrm{D}} \mu_h)(N_S^i)$$

with $(N_S^i)_{1 \le i \le 2}$ the vertices of a side $S \in \mathcal{S}_h$. The simplest choice $X_h = M_h$ does not ensure uniqueness of the multiplier θ_h. Guided by a node-wise identity satisfied by the jump of a function of D_h, we are led to introduce

$$X_h = \left\{ \mu_h \in M_h : \sum_{S \in \mathcal{S}_N} \alpha_{N,S} |S| \mu_h(N) = 0 \quad \forall N \in \mathcal{N}_h \right\}, \tag{5}$$

where $\mathcal{S}_N$ is the set of sides containing N and $\alpha_{N,S}$ is equal to 1 if n_S is oriented clock-wise around N, and to -1 otherwise.

It is important to note that Ker b_h coincides with the P^1 conforming space V_h^0, which implies that U_h satisfies the weak formulation (3), and hence $U_h = u_h$. The key-point for the stability of the mixed formulation is the uniform inf-sup condition; its proof closely follows the one of [1] for the Laplace problem. It allows to obtain existence and uniqueness of θ_h, as well as an optimal $O(h^{\frac{1}{2}+\nu})$ error estimate for θ_h.

3.2 Local Computation of the Multiplier

The main interest of the mixed formulation (4) is that θ_h can be computed locally, as the sum of local contributions θ_N for $N \in \mathcal{N}_h$. Each θ_N is defined on the support ω_N of the P^1 shape function φ_N associated to the node N, vanishes on $\partial\omega_N$, lives on the sides $S \in \mathcal{S}_N$ and belongs to $P^1(S)$. Let the residual and local bilinear form

$$r_h(\cdot) = l_h(\cdot) - a_h(u_h, \cdot), \quad b_S(\theta, \varphi) = \frac{|S|}{2} \sum_{i=1}^{2} (\theta\varphi)(N_i^S), \quad \forall S \in \mathcal{S}_h.$$

As in [1], we impose that θ_N satisfies the following system, for any $K \subset \omega_N$, $S \in \mathcal{S}_N$:

$$\sum_{S\subset\partial K} b_S(\theta_N, [\varphi_N]) = r_h(\varphi_N \chi_K), \qquad b_S(\theta_N, \varphi_M) = \rho_S\, r_h(\varphi_M \chi_{K^{\text{in}}}), \tag{6}$$

where M denotes the other vertex of S and where ρ_S is a coefficient equal to 0, $\frac{1}{2}$ or 1 which takes into account the overlapping of the patches ω_N, see [1] for more details. The linear system (6) is compatible thanks to the following relation:

$$r_h(\varphi_N) = \sum_{K\subset\omega_N} r_h(\varphi_N \chi_K) = 0, \quad \forall N \in \mathcal{N}_h^{\text{int}} \cup \mathcal{N}_h^{\partial},$$

which holds true because u_h is solution to (3). However, (6) has a one-dimensional kernel $\mathcal{K}_N$. In order to obtain a unique solution θ_N, we impose in addition that $\sum_{S\in\mathcal{S}_N} \alpha_{N,S}|S|\theta_N(N) = 0$, which characterizes the orthogonal of $\mathcal{K}_N$ and which ensures that $\sum_{N\in\mathcal{N}_h} \theta_N$ belongs to X_h. It was shown in [1] that it also satisfies the first equation of (4), so by uniqueness of its solution we get that $\theta_h = \sum_{N\in\mathcal{N}_h} \theta_N$.

3.3 Conservative Locally Reconstructed Flux

We now use θ_h to define a local flux $\sigma_h \in H(\text{div}, \Omega)$. We employ the Raviart-Thomas space RT_h^m with $m = 1$ or $m = 0$. We impose the degrees of freedom of σ_h on the edges as below. On the Neumann and contact boundaries, we set respectively

$$\sigma_h \cdot n_S = \pi_S^m g \quad \forall S \in \mathcal{S}_h^{\text{N}}, \qquad \sigma_h \cdot n_S = -\pi_S^m\,(P(u_h)_+) \qquad \forall S \in \mathcal{S}_h^{\text{C}}, \tag{7}$$

whereas on the interior or the Dirichlet sides we impose:

$$\int_S \sigma_h \cdot n_S \varphi = \int_S \{\partial_n u_h\}\varphi - b_S(\theta_h, \varphi), \quad \forall \varphi \in P^m(S), \quad \forall S \in \mathcal{S}_h. \tag{8}$$

Note that for $m = 0$, (8) is simply equivalent to $\sigma_h \cdot n_S = \{\partial_n u_h\} - \pi_S^0 \theta_h$.

The normal trace $\sigma_h \cdot n_S$ is thus well-defined in $P^m(S)$. In order to define σ_h in RT_h^m, we also prescribe interior degrees of freedom when $m = 1$ as follows:

$$\int_K \sigma_h \cdot r = \int_K \nabla u_h \cdot r - \int_{\partial K\cap\mathcal{S}_h^{\text{C}}} \frac{|S|}{\gamma}\,(\partial_n u_h + P(u_h)_+)\, r \cdot n_S, \quad \forall r \in P^0(K)^2. \tag{9}$$

By taking now in (4) the test-function $v\chi_K$ with $v \in P^m(K)$ and by integrating by parts, we immediately obtain the conservation property of the flux:

$$\mathrm{div}\sigma_h|_K = -\pi_K^m f, \quad \forall K \in \mathcal{K}_h. \tag{10}$$

4 A Posteriori Error Analysis

In [3], the authors studied residual a posteriori error estimators for (3). In addition to the Laplace operator, they consider the term $|S|^{1/2}\|\partial_n u_h + P(u_h)_+\|_{0,S}$ on each contact side. However, the error analysis is carried out under a saturation assumption.

We propose an error estimator based on the correction of the flux $\tau_h = \sigma_h - \nabla u_h$, leading to the local/global indicators $\eta_K = \|\tau_h\|_{0,K}$ and $\eta_0 = \|\tau_h\|_{0,\Omega}$.

In the sequel, we focus on the additional terms (with respect to the Laplace problem) in the error estimator, which are related to the contact condition. We will establish the reliability of the estimator without any saturation assumption.

4.1 A Posteriori Error Estimator

We define the following local error estimators on a contact side $S \in \mathcal{S}_h^{\mathrm{C}}$:

$$\eta_{1,S} = |S|^{1/2}\|\partial_n u_h + P(u_h)_+\|_{0,S}, \qquad \eta_{2,S} = |S|^{-1/2}\|\frac{|S|}{\gamma}P(u_h)_- - \mathcal{L}_h u_h\|_{0,S}. \tag{11}$$

Here above, $\mathcal{L}_h u_h$ is P^1-continuous on Γ^{C}, defined by $\mathcal{L}_h u_h(N) = \frac{1}{\gamma}\{|S|P(u_h)_-\}_N$ at any node $N \in \overline{\Gamma^{\mathrm{C}}}$. The notation $\{\cdot\}_N$ stands for the mean along Γ^{C} if N is interior to Γ^{C}; if $N \in \overline{\Gamma^{\mathrm{C}}} \cap \overline{\Gamma^{\mathrm{D}}}$ then we set $\{A\}_N = 0$ and if $N \in \overline{\Gamma^{\mathrm{C}}} \cap \overline{\Gamma^{\mathrm{N}}}$ then $\{A\}_N = A$.

Remark 1 Recall that $u = \frac{|S|}{\gamma}P(u)_-$ on Γ^{C}. Since u_h is globally continuous, the estimator $\eta_{2,S}$ measures the lack of continuity of $\frac{|S|}{\gamma}P(u_h)_-$ along the side S.

Furthermore, we consider the global error indicators:

$$\eta_1^2 = \sum_{S\in\mathcal{S}_h^{\mathrm{C}}} \eta_{1,S}^2, \quad \eta_2^2 = \sum_{S\in\mathcal{S}_h^{\mathrm{C}}} \eta_{2,S}^2, \quad \eta^2 = \eta_0^2 + \eta_1^2 + \eta_2^2,$$

as well as the usual higher-order term $\varepsilon_{\text{data}}$ related to the data approximation, $\varepsilon_{\text{data}}^2 \simeq \sum_{K\in\mathcal{K}_h} h_K^2 \|f - \pi_K^m f\|_{0,K}^2 + \sum_{S\in\mathcal{S}_h^{\text{N}}} |S| \|g - \pi_S^m g\|_{0,S}^2$. In the next subsection we prove:

Theorem 1 *One has:*

$$|u - u_h|_{1,\Omega} \le c\,(\eta + \varepsilon_{\text{data}})\,.$$

4.2 Upper Error Bound

As usually in a posteriori error analysis with reconstructed fluxes, we evaluate $|u - u_h|_{1,\Omega}^2$ by means of an integration by parts, with $\nabla u = \sigma$ and $\nabla u_h = \sigma_h - \tau_h$. By using the properties (7), (10) of $\sigma_h \in H(\text{div}, \Omega)$ and the Dirichlet condition, we get:

$$|u - u_h|_{1,\Omega}^2 \le \left(\eta_0^2 + \varepsilon_{\text{data}}^2\right)^{1/2} |u - u_h|_{1,\Omega} + \int_{\Gamma^{\text{C}}} (\sigma - \sigma_h)\cdot n(u - u_h)ds. \tag{12}$$

Next, we focus on the integral of (12), which we decompose as $\mathcal{T}_I + \mathcal{T}_2 + \mathcal{T}_3$, with

$$\mathcal{T}_1 = \int_{\Gamma^{\text{C}}} (\sigma - \sigma_h)\cdot n(\mathcal{L}_h u_h - u_h),\;\; \mathcal{T}_2 = \int_{\Gamma^{\text{C}}} (\sigma\cdot n - \sigma_h\cdot n - P(u_h)_+)(u - \mathcal{L}_h u_h),$$
$$\mathcal{T}_3 = \int_{\Gamma^{\text{C}}} P(u_h)_+ (u - \mathcal{L}_h u_h) = \sum_{S\in\mathcal{S}_h^{\text{C}}} \mathcal{T}_3^S.$$

In order to bound $\mathcal{T}_1$, we introduce a lifting $\mathcal{R}_h u_h \in V_h^{u^{\text{D}}}$ of $\mathcal{L}_h u_h$, defined by: $\mathcal{R}_h u_h(N) = \mathcal{L}_h u_h(N)$ if $N \in \overline{\Gamma^{\text{C}}}$, and $\mathcal{R}_h u_h(N) = u_h(N)$ at the other vertices. Thus, $\mathcal{R}_h u_h - u_h \in V_h^0$ and its (weighted) discrete H^1-norm can be bounded by that of $\mathcal{L}_h u_h - u_h$ on Γ^{C}, which yields:

$$\begin{aligned}|\mathcal{T}_1| =& \left|\int_\Omega \nabla(u - u_h)\cdot\nabla(\mathcal{R}_h u_h - u_h) - \int_\Omega \tau_h\cdot\nabla(\mathcal{R}_h u_h - u_h)\right.\\ &\left. - \int_{\mathcal{K}_h} (f - \pi_K^m f)(\mathcal{R}_h u_h - u_h) - \int_{\mathcal{S}_h^{\text{N}}} (g - \pi_S^m g)(\mathcal{R}_h u_h - u_h)\right|\\ \le& c\left(|u - u_h|_{1,\Omega}^2 + \eta_0^2 + \varepsilon_{\text{data}}^2\right)^{1/2} \times \left(\sum_{S\in\mathcal{S}_h^{\text{C}}} |S|^{-1} \|\mathcal{L}_h u_h - u_h\|_{0,S}^2\right)^{1/2}.\end{aligned}$$

By using the definition of $P(u_h)$ and the relation $a = a_+ + a_-$, we can write that:

$$\|\mathcal{L}_h u_h - u_h\|_{0,S} \leq \frac{|S|}{\gamma}\|\partial_n u_h + P(u_h)_+\|_{0,S} + \|\frac{|S|}{\gamma}P(u_h)_- - \mathcal{L}_h u_h\|_{0,S}$$

so for $\gamma \geq 1$ we obtain $|S|^{-1/2}\|\mathcal{L}_h u_h - u_h\|_{0,S} \leq \eta_{1,S} + \eta_{2,S}$ and hence,

$$|\mathcal{T}_1| \leq c\left(|u - u_h|_{1,\Omega} + \eta + \varepsilon_{\text{data}}\right)\eta. \tag{13}$$

As regards $\mathcal{T}_2$, we use $(\sigma \cdot n)u = 0$, $\sigma \cdot n \leq 0$, $\mathcal{L}_h u_h \leq 0$ and (7) to get:

$$\begin{aligned}\mathcal{T}_2 &\leq -\int_{\Gamma^{\mathrm{C}}}(\sigma_h \cdot n + P(u_h)_+)(u - \mathcal{L}_h u_h)\\ &= \int_{\mathcal{S}_h^{\mathrm{C}}}(\sigma_h \cdot n + P(u_h)_+)(u - u_h - \pi_S^m(u - u_h)) + (\sigma_h \cdot n + P(u_h)_+)(u_h - \mathcal{L}_h u_h).\end{aligned}$$

The Cauchy-Schwarz inequality together with a standard scaling argument gives

$$|S|^{1/2}\|\sigma_h \cdot n + P(u_h)_+\|_{0,S} \leq \eta_{1,S} + |S|^{1/2}\|\tau_h \cdot n\|_{0,S} \leq \eta_{1,S} + c\eta_K.$$

Thanks to the discrete trace inequality on $S \in \mathcal{S}_h^{\mathrm{C}}$, we next get that

$$\mathcal{T}_2 \leq c\eta(|u - u_h|_{1,\Omega} + \eta). \tag{14}$$

Finally, we consider $\mathcal{T}_3$. We only have to bound $\mathcal{T}_3^S$ on the contact sides S where $P(u_h)$ is non-negative. By using that $u \leq 0$ and $P(u_h)_+ \geq 0$, we first have that $\mathcal{T}_3^S \leq -\int_S P(u_h)_+\mathcal{L}_h u_h$. Thanks to the property $a_+a_- = 0$, we further get:

$$\mathcal{T}_3^S \leq \int_S \left(\partial_n u_h + P(u_h)_+\right)\left(\frac{|S|}{\gamma}P(u_h)_- - \mathcal{L}_h u_h\right) - \int_S \partial_n u_h\left(\frac{|S|}{\gamma}P(u_h)_- - \mathcal{L}_h u_h\right).$$

The first integral is bounded by $\eta_{1,S}\eta_{2,S}$. For the second one, the linear function $P(u_h)$ either changes its sign in a point $M \in S$ or is strictly positive.

In the first case, by means of the exact Simpson formula on the segment (of length d) where $P(u_h) \geq 0$ we obtain that:

$$\int_S(\partial_n u_h + P(u_h)_+)^2 ds \geq \frac{d}{6}(\partial_n u_h + P(u_h))^2(M) + (|S| - d)(\partial_n u_h)^2 \geq \frac{|S|}{6}(\partial_n u_h)^2,$$

so $|S|^{1/2}\|\partial_n u_h\|_{0,S} \leq c\eta_{1,S}$ with c independent of d. Thus, $\mathcal{T}_3^S \leq c\eta_{1,S}\eta_{2,S}$.

In the second case, $P(u_h)_- = 0$ so we have to bound $|\int_S \partial_n u_h \mathcal{L}_h u_h|$. We evaluate it by the exact trapeze formula and we bound it thanks to the triangular inequality by

$$\frac{|S|}{2}\sum_{i=1}^{2}|(\partial_n u_h + P(u_h)_+)(N_i)\mathcal{L}_h u_h(N_i)| + \frac{|S|}{2}\sum_{i=1}^{2}|P(u_h)_+(N_i)\mathcal{L}_h u_h(N_i)|.$$

The first sum is clearly bounded by $c\eta_{1,S}\eta_{2,S}$. Concerning the second sum, let us first note that only the interior nodes of Γ^C contribute to it. We then use the definition of $\mathcal{L}_h u_h$, the fact that $(P(u_h)_-)_{|S}(N_i) = 0$, the relation $a_+ a_- = 0$ and we obtain:

$$|S||P(u_h)_+(N_i)\mathcal{L}_h u_h(N_i)| = \left|[|S|P(u_h)_+]_{N_i}\right| \times |\mathcal{L}_h u_h(N_i)| \le c\left|[|S|P(u_h)_+]_{N_i}\right|\eta_{2,S}.$$

For the jump term, we need to consider the adjacent contact side $\tilde{S}$ containing N_i. If $P(u_h)_{|\tilde{S}}(N_i) \ge 0$, then $[|S|P(u_h)_+]_{N_i} = [|S|P(u_h)]_{N_i} = -[|S|\partial_n u_h]_{N_i}$. By combining an estimate established in [5] with the fact that $\sigma_h \in H(\mathrm{div}, \Omega)$, and hence $[\partial_n u_h] = -[\tau_h \cdot n]$ on any interior side, we next obtain that

$$[|S|\partial_n u_h]_N^2 \le c \sum_{S_j \in \mathcal{S}_N^{int}} |S_j|\|[\partial_n u_h]\|_{0,S_j}^2 \le c \sum_{K \subset \omega_N} \eta_K^2. \tag{15}$$

If $P(u_h)_{|\tilde{S}}(N_i) < 0$, then we write by means of the triangular inequality that:

$$\left|[|S|P(u_h)_+]_{N_i}\right| \le \left|[|S|(P(u_h)_+ + \partial_n u_h)]_{N_i}\right| + \left|[|S|\partial_n u_h]_{N_i}\right|.$$

The second right-hand side term is bounded in (15); the first one is bounded by $\left||S|(P(u_h) + \partial_n u_h)_{|S}(N_i)\right| + \left||\tilde{S}|(\partial_n u_h)_{|\tilde{S}}(N_i)\right| \le c(\eta_{1,S} + \eta_{1,\tilde{S}})$, where we have used on $\tilde{S}$ the estimate for $\partial_n u_h$ previously established in the case where $P(u_h)$ changes its sign (otherwise, $P(u_h)_+ = 0$ on $\tilde{S}$ so the estimate is obvious).

So finally, $\mathcal{T}_3 \le c\eta^2$; together with (13) and (14), it ends the proof of Theorem 1.

Acknowledgments The authors would like to thank F. Chouly for suggesting the application to the contact problem and for the interesting discussions on its Nitsche formulation.

References

1. R. Becker, D. Capatina, R. Luce, SIAM J. Numer. Anal., https://doi.org/10.1137/16M1064817
2. F. Chouly, P. Hild, Y. Renard, Math. Comput., https://doi.org/10.1090/S0025-5718-2014-02913-X
3. F. Chouly, M. Fabre, P. Hild, J. Pousin, Y. Renard, IMA J. Numer. Anal., https://doi.org/10.1093/imanum/drx02
4. B.I. Wohlmuth, J. Sci. Comput., https://doi.org/10.1007/s10915-007-9139-7
5. A. Weiss, B.I. Wohlmuth, Math. Comput., https://doi.org/10.1090/S0025-5718-09-02235-2

Study on an Adaptive Finite Element Solver for the Cahn–Hilliard Equation

G. Fabian Castelli and Willy Dörfler

Abstract In this work we present an adaptive matrix-free finite element solver for the Cahn–Hilliard equation modelling phase separation in electrode particles of lithium ion batteries during lithium insertion. We employ an error controlled variable-step, variable-order time integrator and a regularity estimator for the adaptive mesh refinement. In particular, we propose a matrix-free applicable preconditioner. Numerical experiments demonstrate the importance of adaptive methods and show for our preconditioner practically no dependence of the number of GMRES iterations on the mesh size, even for locally refined meshes.

1 Phase Separation in Electrode Particles

Lithium ion batteries have become a promising energy storage technology for mobile power devices. For the better understanding of the cyclability and the loss of capacity we want to investigate the degradation behaviour of single electrode particles. For example in electrode materials like lithium manganese oxide (LMO) or lithium iron phosphate (LFP) the occurrence of a phase transition between lithium poor and lithium rich phases can lead to high stresses [6, 11–13], which in the end can also cause particle fracture.

However, simulating such a complex multi-physical problem is a very challenging task. So neglecting the mechanics for the moment we focus in this work on the efficient numerical simulation of phase separation in electrode particles during lithium insertion. Main challenges of this problem like the almost sharp moving phase transition as well as the varying time scales over several orders of magnitude give rise to use adaptive methods in space and time. In particular, to be able to use the high performance parallel matrix-free framework within the open-source

G. F. Castelli (✉) · W. Dörfler
Institute of Applied and Numerical Mathematics, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
e-mail: fabian.castelli@kit.edu; willy.doerfler@kit.edu

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_23

finite element library *deal.II* [2, 7] in future an appropriate preconditioner is needed. Whereas preconditioning the Cahn–Hilliard equation is already a research topic itself, see for example [3, 4].

In the rest of this section we review the model equations for phase separation in electrode particles. Following in Sect. 2 we explain our numerical solution algorithm and propose our matrix-free applicable preconditioner. Numerical results will be discussed in Sect. 3 and a conclusion is given in Sect. 4.

Model Equations In contrast to the sharp interface model we previously used in [5], we follow the phase field modelling in [6] and consider the resulting dimensionless mixed formulation of the Cahn–Hilliard equation from a mathematical point of view: Let $T > 0$ and $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$ be a bounded domain. Find the normalised lithium concentration $c\colon [0, T] \times \overline{\Omega} \to [0, 1]$ and the chemical potential $\mu\colon [0, T] \times \overline{\Omega} \to \mathbb{R}$ satisfying the initial boundary value problem:

$$\begin{cases} \partial_t c = \nabla\cdot\big(m(c)\nabla\mu\big) & \text{in } (0, T) \times \Omega, \\ \mu = \partial_z \psi(c, \nabla c) - \nabla\cdot\partial_{\boldsymbol{p}}\psi(c, \nabla c) & \text{on } (0, T) \times \Omega, \\ \nabla c \cdot \boldsymbol{n} = 0 & \text{on } (0, T) \times \partial\Omega, \\ m(c)\nabla\mu \cdot \boldsymbol{n} = -J_{\text{ext}} & \text{on } (0, T) \times \partial\Omega, \\ c(0, \cdot) = c_0 & \text{in } \Omega. \end{cases} \tag{1}$$

The nonlinear mobility is given by $m(c) = Dc(1 - c)$ with the diffusion coefficient $D > 0$. The system's free energy density $\psi(z, \boldsymbol{p}) = \psi_{\text{dwp}}(z) + \psi_{\text{gd}}(\boldsymbol{p})$ is decomposed into the homogeneous chemical free energy density $\psi_{\text{dwp}}\colon [0, 1] \to \mathbb{R}$,

$$\psi_{\text{dwp}}(z) = \alpha_1 z + \frac{1}{2}\alpha_2 z^2 + z\log(z) + (1 - z)\log(1 - z),$$

for which the two material parameters $\alpha_1, \alpha_2 \in \mathbb{R}$ control the shape of this function, and the gradient energy density $\psi_{\text{gd}} : \mathbb{R}^d \to \mathbb{R}$,

$$\psi_{\text{gd}}(\boldsymbol{p}) = \frac{1}{2}\kappa|\boldsymbol{p}|^2,$$

with the parameter $\kappa > 0$ controlling the width of the phase transition. Note, that phase separation can only occur if α_1 and α_2 are chosen, such that ψ_{dwp} has a double well shape.

To model the lithium insertion we use an inhomogeneous boundary condition of Neumann-type for μ, corresponding to a given particle surface flux $J_{\text{ext}}\colon [0, T] \times \partial\Omega \to \mathbb{R}$, while a homogeneous Neumann-type boundary condition for c ensures that the phase transition stays always orthogonal to the particle surface.

As initial condition for the simulation of lithium insertion we assume an approximately zero given initial distribution of concentration $c_0\colon \overline{\Omega} \to (0, 1)$ consistent with the boundary conditions.

Parameters to specify the electrode material will be given in Sect. 3.

2 The Numerical Method

Semi-discretisation in Space We start with the weak formulation of the system (1): Find $c\colon [0,T] \to \{H^1(\Omega) : c \in [0,1]\}$ and $\mu\colon [0,T] \to H^1(\Omega)$ satisfying the weak form that we get by multiplication with test functions v, $w \in V := H^1(\Omega)$. We assume $V_h \subset V$ to be a finite dimensional space with basis $\{\varphi_i : i = 1,\dots,N\}$. Now we seek $c_h\colon [0,T] \to \{V_h : c_h \in [0,1]\}$ and $\mu_h\colon [0,T] \to V_h$ to be solutions of the discrete system

$$\begin{aligned}\big(\varphi_i, \partial_t c_h\big)_\Omega &= -\big(\nabla\varphi_i, m(c_h)\nabla\mu_h\big)_\Omega - \big(\varphi_i, J_{\text{ext}}\big)_{\partial\Omega},\\ 0 &= -\big(\varphi_i, \mu_h\big)_\Omega + \big(\varphi_i, \psi'_{\text{dwp}}(c_h)\big)_\Omega + \kappa\big(\nabla\varphi_i, \nabla c_h\big)_\Omega,\end{aligned}$$

for $i = 1,\dots,N$. In this set of equations we insert the basis representation for the discrete functions $c_h(t,x) = \sum_{j=1}^N c_j(t)\varphi_j(x)$ and $\mu_h(t,x) = \sum_{j=1}^N \mu_j(t)\varphi_j(x)$. Now we identify the spatially discrete function $c_h(t,\cdot)$ with the vector-valued function $\boldsymbol{c}(t) = [c_j(t)]_j \in [0,1]^N$ and $\mu_h(t,\cdot)$ with $\boldsymbol{\mu}(t) = [\mu_j(t)]_j \in \mathbb{R}^N$. Gathering the solution variables for the concentration and the chemical potential in a vector-valued solution variable $\boldsymbol{y}\colon [0,T] \to \mathbb{R}^{2N}$, $t \mapsto \boldsymbol{y}(t) = [\boldsymbol{c}(t), \boldsymbol{\mu}(t)]^\top$, we arrive at the nonlinear differential algebraic equation (DAE) for the spatially discretised system: Find $\boldsymbol{y}\colon [0,T] \to \mathbb{R}^{2N}$ satisfying:

$$\begin{pmatrix} M & 0\\ 0 & 0\end{pmatrix} \partial_t \boldsymbol{y} = \boldsymbol{f}(t,\boldsymbol{y}) \quad \text{for } t > 0, \qquad \boldsymbol{y}(0) = \boldsymbol{y}^0. \tag{2}$$

The system mass matrix on the left hand side of the DAE is singular and its entry $M = [\big(\varphi_i, \varphi_j\big)_\Omega]_{ij}$ denotes the mass matrix of our finite element space. The right hand side function is defined according to the weak formulation: If $\boldsymbol{y}$ is related to c_h, μ_h as described, we have $\boldsymbol{f}\colon [0,T] \times \mathbb{R}^{2N} \to \mathbb{R}^{2N}$ with

$$(t,\boldsymbol{y}) \mapsto \boldsymbol{f}(t,\boldsymbol{y}) := \begin{pmatrix} -\big(\nabla\varphi_i, m(c_h)\nabla\mu_h\big)_\Omega - \big(\varphi_i, J_{\text{ext}}\big)_{\partial\Omega}\\ -\big(\varphi_i, \mu_h\big)_\Omega + \big(\varphi_i, \psi'_{\text{dwp}}(c_h)\big)_\Omega + \kappa\big(\nabla\varphi_i, \nabla c_h\big)_\Omega \end{pmatrix}_{i=1,\dots,N}.$$

Defining the matrices $K_1 = [\big(\nabla\varphi_i, \nabla\varphi_j\big)_\Omega]_{ij}$, $K_m(\boldsymbol{y}_1) = [\big(\nabla\varphi_i, m(c_h)\nabla\varphi_j\big)_\Omega]_{ij}$ and the vectors $\boldsymbol{\Psi}(\boldsymbol{y}_1) = [\big(\varphi_i, \psi'_{\text{dwp}}(c_h)\big)_\Omega]_i$, $\boldsymbol{J} = [\big(\varphi_i, J_{\text{ext}}\big)_{\partial\Omega}]_i$ we can rewrite $\boldsymbol{f}$ as

$$\boldsymbol{f}(t,\boldsymbol{y}) = \begin{pmatrix} \boldsymbol{f}_1(t,\boldsymbol{y}_1,\boldsymbol{y}_2)\\ \boldsymbol{f}_2(t,\boldsymbol{y}_1,\boldsymbol{y}_2)\end{pmatrix} := \begin{pmatrix} -K_m(\boldsymbol{y}_1)\boldsymbol{y}_2 - \boldsymbol{J}\\ -M\boldsymbol{y}_2 + \boldsymbol{\Psi}(\boldsymbol{y}_1) + \kappa K_1\boldsymbol{y}_1\end{pmatrix}.$$

Note, that an explicit dependence of $\boldsymbol{f}$ on t will only occur via J_{ext}.

The Time Integration Method As a robust solver for the arising DAE we use the family of NDF multistep methods in a variable-step, variable-order algorithm, in

Matlab known as `ode15s` [8–10]. This incorporates an error controlled adaptive time step size and adapted orders. We implemented this as a `C++` code with the functionalities of the *deal.II* library.

In summary, the resulting fully discrete problem for the approximate solution $\boldsymbol{y}^{n+1}$ to the solution $\boldsymbol{y}(t_{n+1})$ of (2) at the time step $t_{n+1} = t_n + \tau_n$ involves a nonlinear algebraic set of equations to solve. According to the k-th order NDF time integrator these nonlinear equations have the form [8, Sect. 2.3]:

$$\begin{aligned} \xi_k \tau_n^{-1} M(\boldsymbol{\Phi}_1 + \boldsymbol{y}_1^{n+1}) - \boldsymbol{f}_1(t_{n+1}, \boldsymbol{y}_1^{n+1}, \boldsymbol{y}_2^{n+1}) &= \mathbf{0}, \\ -\boldsymbol{f}_2(t_{n+1}, \boldsymbol{y}_1^{n+1}, \boldsymbol{y}_2^{n+1}) &= \mathbf{0}, \end{aligned}$$

with some constant $\xi_k > 0$ for the chosen order k and the fixed term $\boldsymbol{\Phi}_1$ depending on the solution at some previous time steps $\boldsymbol{y}_1^n, \boldsymbol{y}_1^{n-1}, \ldots$

Solving the System of Equations Our approach follows the work [4]: We introduce the scaling (for c) $\boldsymbol{y}_1^{n+1} = \tau^{1/4}\boldsymbol{z}_1^{n+1}$, $\boldsymbol{\Phi}_1 = \tau^{1/4}\widetilde{\boldsymbol{\Phi}}_1$ and (for μ) $\boldsymbol{y}_2^{n+1} = \tau^{-1/4}\boldsymbol{z}_2^{n+1}$ and multiply the first equation with $\tau^{3/4}$ and the second equation with $\tau^{1/4}$ to get:

$$\begin{aligned} \xi_k M\big(\widetilde{\boldsymbol{\Phi}}_1 + \boldsymbol{z}_1^{n+1}\big) + \tau^{1/2} K_m\big(\tau^{1/4}\boldsymbol{z}_1^{n+1}\big)\boldsymbol{z}_2^{n+1} + \tau^{3/4}\boldsymbol{J} &= \mathbf{0}, \\ M\boldsymbol{z}_2^{n+1} - \tau^{1/4}\boldsymbol{\Psi}\big(\tau^{1/4}\boldsymbol{z}_1^{n+1}\big) - \tau^{1/2}\kappa K_1 \boldsymbol{z}_1^{n+1} &= \mathbf{0}. \end{aligned}$$

To solve this problem for $\boldsymbol{z}^{n+1}$, Newton's method is applied and the essential work is to solve the linearised problem. This is done by a GMRES method with a right preconditioner. As in [4] we compute the Jacobian and swapping columns yields a generalisation of the preconditioner obtained in [4] depending on the Newton step $\boldsymbol{z}^{(i)}$:

$$P(\boldsymbol{z}^{(i)}) = \begin{pmatrix} \tau^{1/2} K_m\big(\tau^{1/4}\boldsymbol{z}_1^{(i)}\big) + M & 0 \\ 0 & \tau^{1/2}\kappa K_1 + M \end{pmatrix}.$$

The advantage of the proposed preconditioner is that it respects the nonlinear mobility and is applicable for matrix-free computations [7]. In particular, as in [4], for the efficient application of the preconditioner, the action of the inverse blocks can be computed by a CG method with a suitable preconditioner. Furthermore, we benefit from the implementation of the matrix-free framework in *deal.II*, since we are able to parallelise the code directly with MPI for future simulations.

The Space-Time Adaptive Algorithm We will first present the algorithm and will then explain some details.

1. Given $\boldsymbol{c}^n$ and $\boldsymbol{\mu}^n$ (and previous ones), $\tau_n, \mathcal{T}_n$, order k for time stepping.
2. Solve for $\boldsymbol{y}^{n+1}$ (defining $\boldsymbol{c}^{n+1}$ and $\boldsymbol{\mu}^{n+1}$).
3. Estimate time error err_t and compute spatial regularity estimates est_x.

4. (b) If $\text{err}_t > \text{RelTol}_t$: Decrease time step size and/or reduce order.
   (b) If $\text{est}_x > \text{RelTol}_x$: Mark elements and refine mesh.
5. If both err_t and est_x were OK, accept $\boldsymbol{y}^{n+1}$ as solution at time t_{n+1}.
   Else go to Step 2.
6. If a sufficient number of time steps were accepted with the same time step size

   (b) Adapt time step size and order according to error estimate err_t.
   (b) Mark all cells with $\eta_T^2 < 10^{-10}$ but $h_{\max} = 2^{-5}$ and coarsen mesh.

7. Advance time step.

The method to obtain $\boldsymbol{y}^{n+1}$ in Step 2 has been explained before. For Step 3 the estimation of the time error is part of the NDF method [8–10]. For the spatial error we use a method to judge the regularity of a finite element approximation u_h (here c_h or μ_h), see [1, Chap. 4]. For $u_h \in V_h$ we compute a recovered gradient $G_h(u_h) \in (V_h)^d$, here via an approximated L^2-projection, and define for each cell T of our spatial partition

$$\eta_T^2(u_h) := \int_T |G_h(u_h) - \nabla u_h|^2 \,\mathrm{d}x$$

and $\text{est}_x := \left(\sum_{T\in\mathcal{T}_n} \eta_T^2\right)^{1/2}$ with $\eta_T^2 := \eta_T^2(c_h) + \eta_T^2(\mu_h)$. For the adaptive change of the time step size and the order in Steps 4a and 6a, compare [8–10] and the references cited therein. Step 4b is done with fixed energy marking up to a minimal mesh width $h_{\min} = 2^{-20}$ and an absolute tolerance $\eta_T^2 > \text{AbsTol}_x$.

3 Numerical Experiments

For the numerical experiments we consider the model equations from Sect. 1 for a spherical shaped electrode particle of LFP. The parameters, taken from [13], are $\alpha_1 = 4.5$, $\alpha_2 = -9$, $\kappa = 3.91 \times 10^{-4}$ for the free energy density and $D = 1.6 \times 10^3$ for the diffusion coefficient. Assuming a radial symmetric solution we can reduce the computational domain to the one-dimensional unit interval $\Omega = (0, 1)$ representing the radial line from the particle centre at $\Gamma_0 = \{0\}$ to the particle surface at $\Gamma_{\text{ext}} = \{1\}$. As boundary condition we apply a constant insertion rate $J_{\text{ext}} = -1/3$ at the particle surface such that the particle would get fully charged within 1 h $T = 1$. To preserve the symmetry we impose homogeneous boundary conditions of Neumann-type at the artificial boundary in the particle centre

$$\nabla c \cdot \boldsymbol{n} = m(c)\nabla\mu \cdot \boldsymbol{n} = 0 \quad \text{on } (0, T) \times \Gamma_0.$$

At initial time $t = 0$ we assume a constant concentration of $c_0 = 0.01$.

First we solved the model equations numerically with linear finite elements and the variable-step, variable-order time integrator, as explained in Sect. 2. We used a uniform mesh with mesh width $h = 2^{-10}$, according to the rule of thumb that the phase transition should be resolved with at least ten cells or unknowns. As the width of the phase transition behaves like $\sqrt{\kappa}$ it is related to the uniform mesh width $h = 2^{-n}$ by $2^{-n} < \sqrt{\kappa}/10 \Leftrightarrow n > \log\left(\sqrt{\kappa}/10\right)/\log(2)$, with n the number of uniform refinements. For the time integration the relative tolerance was set to $\mathrm{RelTol}_t = 10^{-4}$. In Fig. 1 we see snapshots of the numerical solution for the concentration and the chemical potential at three characteristic time steps: (1) Initiation of phase separation, (2) Migration of the phase transition through the particle, (3) Vanishing of the phase transition.

The importance of adaptive methods for this problem becomes immediately clear when we look into Fig. 2. At the times when the phase separation is initiated and when the phase transition vanishes, the time step size jumps over several orders

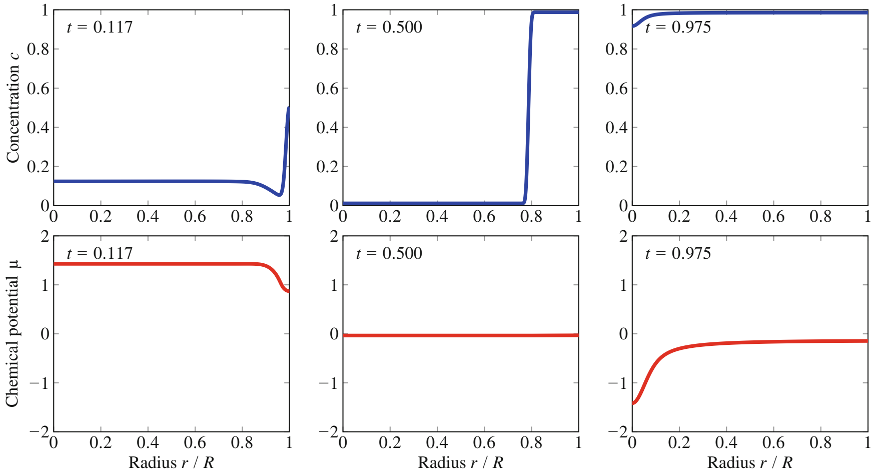


Fig. 1 Temporal evolution of the concentration (top) and the chemical potential (bottom)

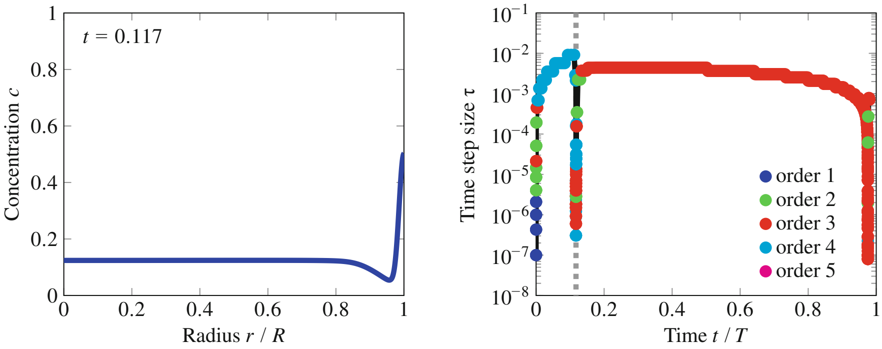


Fig. 2 Concentration at initiation of phase separation (left) and used time step size over time with markers for used order (right)

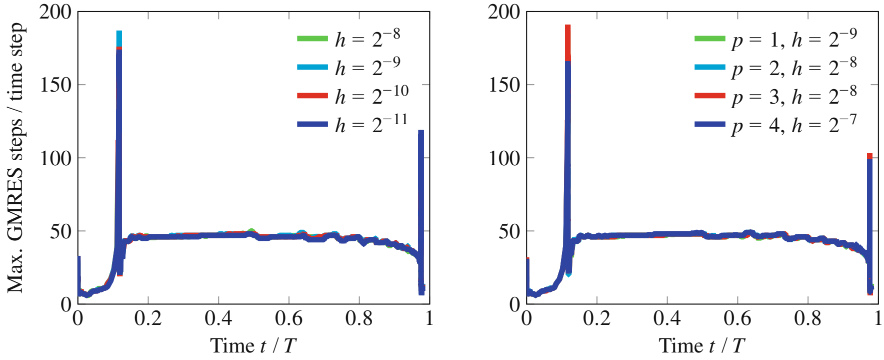


Fig. 3 Maximum number of GMRES steps per time step for different uniform refinements (left) and for different polynomial degrees with approximately equal number of DoFs (right)

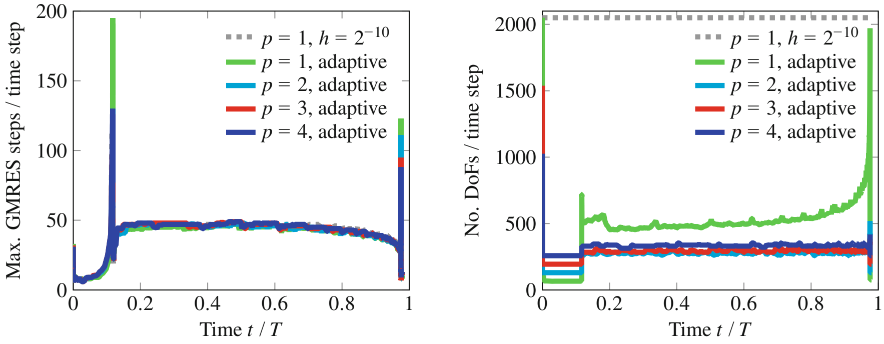


Fig. 4 Maximum number of GMRES steps per time step for different polynomial degrees with adaptive mesh refinement (left) and number of DoFs (right)

of magnitude. In contrast, during the migration of the phase transition a large time step size can be used. Furthermore, a full resolution of the whole spatial domain, respecting the width of the phase transition, is obviously not necessary, since the solution is approximately constant in regions away from the phase transition.

To show the performance of our proposed preconditioner we plotted the maximum number of GMRES steps needed to solve a time step in Fig. 3. For this we solved the model equation (1) for a series of successively uniform refined meshes, (2) for finite element methods with increasing polynomial degree holding the number of unknowns approximately constant. Once the phase transition is fully resolved, the number of GMRES steps is practically independent of further mesh refinement. The variation of the polynomial degree of the finite element method also has no significant influence on the number of iteration steps.

Allowing adaptive mesh refinement as described in the adaptive algorithm in Sect. 2, we see in Fig. 4 that the preconditioner shows the same performance as in the uniform refined case. In particular, compared to the uniformly refined case, the savings in degrees of freedom are enormous, especially for higher order finite element methods. As marking parameter we used $\theta = 0.2$ and for all polynomial degrees we used the tolerances $\mathrm{RelTol}_x = 10^{-2}$, $\mathrm{AbsTol}_x = 10^{-10}$, except in the case $p = 1$, where we used $\mathrm{RelTol}_x = 5 \times 10^{-2}$ instead.

4 Conclusion

Summing up, we introduced the model equations for phase separation in electrode particles of lithium ion batteries during lithium insertion. For this initial boundary value problem we presented a space-time adaptive algorithm for a finite element solver. In particular, we developed an easy to implement and matrix-free applicable preconditioner, which respects the nonlinear character of the PDEs. Numerical experiments showed the high demand for adaptive methods as an indispensable tool for the fast and accurate solution of this complex application problem. Furthermore with the proposed matrix-free preconditioner the number of GMRES steps showed practically no dependence on the spatial resolution even for locally refined meshes.

The presented results in one space dimension give rise to exploit the capabilities of the developed adaptive matrix-free finite element solver for future simulations of more realistic cases, such as phase separation in arbitrary shaped electrode particles including also a thermodynamically consistent mechanics theory.

The proof of a theoretical result for the proposed matrix-free preconditioner will also be part of a future work.

Acknowledgments The authors thank M. Kamlah and T. Zhang for discussions about the model equations. G.F. Castelli acknowledges financial support by the German Research Foundation (DFG) through RTG 2218 SiMET—Simulation of mechano-electro-thermal processes in lithium-ion batteries, project number 281041241.

References

1. Ainsworth, M., Oden, J.T.: A Posteriori Error Estimation in Finite Element Analysis. John Wiley, New York (2000)
2. Bangerth, W., Hartmann, R., Kanschat, G.: deal.II—a general-purpose object-oriented finite element library. ACM Trans. Math. Software **33**(4), 24/1–24/27 (2007)
3. Bosch, J., Stoll, M.: Preconditioning for vector-valued Cahn–Hilliard equations. SIAM J. Sci. Comput. **37**(5), S216–S243 (2015)
4. Brenner, S.C., Diegel, A.E., Sung, L.: A robust solver for a mixed finite element method for the Cahn–Hilliard equation. J. Sci. Comput. **77**(2), 1234–1249 (2018)
5. Castelli, G.F., Dörfler, W.: The numerical study of a microscale model for lithium-ion batteries. Comput. Math. Appl. **77**(6), 1527–1540 (2019)
6. Huttin, M., Kamlah, M.: Phase-field modeling of stress generation in electrode particles of lithium ion batteries. Appl. Phys. Lett. **101**(13), 133902–1–133902–4 (2012)
7. Kronbichler, M., Kormann, K.: A generic interface for parallel cell-based finite element operator application. Comput. Fluids **63**, 135–147 (2012)

8. Shampine, L.F., Reichelt, M.W.: The MATLAB ODE suite. SIAM J. Sci. Comput. **18**(1), 1–22 (1997)
9. Shampine, L.F., Reichelt, M.W., Kierzenka, J.A.: Solving index-1 DAEs in MATLAB and Simulink. SIAM Rev. **41**(3), 538–552 (1999)
10. The MathWorks Inc.: MATLAB. http://www.mathworks.com
11. Walk, A., Huttin, M., Kamlah, M.: Comparison of a phase-field model for intercalation induced stresses in electrode particles of lithium ion batteries for small and finite deformation theory. Eur. J. Mech. A Solids **48**, 74–82 (2014)
12. Xu, B., Zhao, Y., Stein, P.: Phase field modeling of electrochemically induced fracture in Li-ion battery with large deformation and phase segregation. GAMM-Mitt. **39**(1), 92–109 (2016)
13. Zhang, T., Kamlah, M.: Sodium ion batteries particles: Phase-field modeling with coupling of Cahn–Hilliard equation and finite deformation elasticity. J. Electrochem. Soc. **165**(10), A1997–A2007 (2018)

Numerical Study of the Fracture Diffusion-Dispersion Coefficient for Passive Transport in Fractured Porous Media

Florent Chave

Abstract We propose a new definition of the normal fracture diffusion-dispersion coefficient for a reduced model of passive transport in fractured porous media.

MSC (2010) 65N08, 65N12, 65N30

1 Introduction

In this paper, we focus on the reduced model introduced in [1, 3] describing the Passive Transport of a solute in a Fractured Porous Media, which will be now referred to as (PTFPM). By reduced model, we assume that the fracture is treated as a surface of codimension one. The reduced model (PTFPM) consists of two advection–diffusion–reaction equations, one in the porous media and one in the fracture, with advective velocity fields taken as the solution of a decoupled problem, and where the coupling is done by subtle transmission conditions describing the exchanges between the different regions. A notable feature of the reduced model (PTFPM) is that the transmission conditions between the porous media and the fracture mimic at the discrete level the property that the advection terms do not contribute to the energy balance of the system, allowing us to handle both conducting and blocking fractures by letting the concentration of the solute jumps across the fracture; see also [4] in the context of advection of a passive scalar in a fractured porous media. However, the description of the fracture diffusion-dispersion in both the normal and tangential directions considered in the reduced model (PTFPM) is meaningless from the physical viewpoint. Indeed, in (PTFPM) those coefficients are assumed to be independent from the surrounding unknowns: this is irrelevant since they play an important role in the description of (1) the exchanges between the porous media and the fracture, and (2) the behavior of the

F. Chave (✉)
Inria, Univ. Lille, CNRS, UMR 8524 – Laboratoire Paul Painlevé, Lille, France
e-mail: florent.chave@inria.fr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_24

solute at the neighborhood of the fracture. The aim of this paper is to propose a more physical definition of the fracture diffusion-dispersion coefficient, and to present some test cases based on the previous works [2, 3]. The rest of this paper is organized as follows: in Sect. 2 we present the main equations and in Sect. 3 we perform numerical experiments.

2 The Differential Model

In this section, we present the reduced model for the passive transport in a fractured porous media. We first introduce notation, then define the velocity fields and diffusion-dispersion tensors, and finally introduce the main equations.

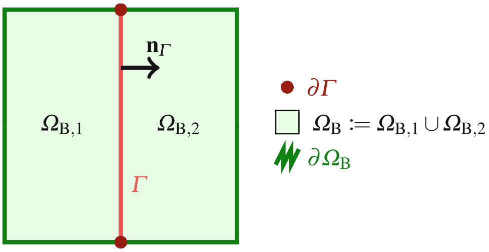
2.1 Notation

We consider a porous medium saturated by an incompressible fluid that occupies a space region $\Omega \subset \mathbb{R}^2$ traversed by a fracture Γ. We assume that Ω is an open, bounded, connected, polygonal set with Lipschitz boundary $\partial\Omega$, and denote by $\mathbf{n}_{\partial\Omega}$ the unit normal vector on $\partial\Omega$ pointing out of Ω. The fracture Γ is represented by an open line segment of nonzero length which cuts Ω into two disjoint connected polygonal subdomains $\Omega_{\mathrm{B},1}$ and $\Omega_{\mathrm{B},2}$ with Lipschitz boundary. The sets $\Omega_{\mathrm{B}} := \Omega \setminus \overline{\Gamma} = \Omega_{\mathrm{B},1} \cup \Omega_{\mathrm{B},2}$ and $\partial\Omega_{\mathrm{B}} := \cup_{i+1}^{2}(\partial\Omega_{\mathrm{B},i} \setminus \overline{\Gamma})$ correspond to the bulk region and the external boundary of the bulk region, respectively. The boundary of the fracture Γ is denoted by $\partial\Gamma$, and the corresponding outward unit tangential vector is $\boldsymbol{\tau}_{\partial\Gamma}$. Finally, $\mathbf{n}_\Gamma$ denotes the unit normal vector to Γ pointing out of $\Omega_{\mathrm{B},1}$ This notation is illustrated in Fig. 1.

For any scalar- or vector-valued function φ sufficiently regular to admit a (possibly two-valued) trace on Γ, we define the jump and average operators such that

$$[\![\varphi]\!]_\Gamma := (\varphi_{|\Omega_{\mathrm{B},1}} - \varphi_{|\Omega_{\mathrm{B},2}})_{|\Gamma}, \qquad \{\!\{\varphi\}\!\}_\Gamma := \frac{1}{2}(\varphi_{|\Omega_{\mathrm{B},1}} + \varphi_{|\Omega_{\mathrm{B},2}})_{|\Gamma}.$$

Fig. 1 Illustration of the notation introduced in Sect. 2.1


$\mathbf{n}_\Gamma$
$\Omega_{\mathrm{B},1}$
$\Omega_{\mathrm{B},2}$
Γ
● $\partial\Gamma$
□ $\Omega_{\mathrm{B}} := \Omega_{\mathrm{B},1} \cup \Omega_{\mathrm{B},2}$
$\partial\Omega_{\mathrm{B}}$

2.2 Advective Velocity Fields

We assume that the advective Darcy velocities follow from the decoupled reduced model [5], which describes the flow in a fractured porous medium. This model reads as follows: Find the bulk Darcy velocity $\mathbf{u} : \Omega_\mathrm{B} \to \mathbb{R}^2$, the bulk pressure $p : \Omega_\mathrm{B} \to \mathbb{R}$ and the fracture pressure $p_\Gamma : \Gamma \to \mathbb{R}$ such that

$$\begin{aligned}
\mathbf{u} + \mathbf{K}\nabla p &= 0 && \text{in } \Omega_\mathrm{B}, && \text{(1a)}\\
\nabla \cdot \mathbf{u} &= f && \text{in } \Omega_\mathrm{B}, && \text{(1b)}\\
\mathbf{u} \cdot \mathbf{n}_{\partial\Omega} &= 0 && \text{on } \partial\Omega_\mathrm{B}, && \text{(1c)}\\
\nabla_\tau \cdot (-K_\Gamma \nabla_\tau p_\Gamma) &= \ell_\Gamma f_\Gamma + [\![\mathbf{u}]\!]_\Gamma \cdot \mathbf{n}_\Gamma && \text{in } \Gamma, && \text{(1d)}\\
-K_\Gamma \nabla_\tau p_\Gamma \cdot \boldsymbol{\tau}_{\partial\Gamma} &= 0 && \text{on } \partial\Gamma, && \text{(1e)}\\
\int_\Gamma p_\Gamma &= 0, && && \text{(1f)}
\end{aligned}$$

where $f \in L^2(\Omega_\mathrm{B})$ and $f_\Gamma \in L^2(\Gamma)$ verify $\int_{\Omega_\mathrm{B}} f + \int_\Gamma \ell_\Gamma f_\Gamma = 0$ and denote source or sink terms, $\mathbf{K} : \Omega_\mathrm{B} \to \mathbb{R}^{2\times 2}$ is the bulk permeability tensor, and we have set $K_\Gamma := \kappa_\Gamma^\tau \ell_\Gamma$, with $\kappa_\Gamma^\tau : \Gamma \to \mathbb{R}$ denoting the tangential permeability inside the fracture and $\ell_\Gamma : \Gamma \to \mathbb{R}$ the fracture thickness. In (1d) and (1e), ∇_τ and $\nabla_\tau \cdot$ denote the tangential gradient and divergence operators along Γ, respectively. The following transmission conditions across the fracture close the problem:

$$\{\!\{\mathbf{u}\}\!\}_\Gamma \cdot \mathbf{n}_\Gamma = \frac{\kappa_\Gamma^n}{\ell_\Gamma} [\![p]\!]_\Gamma \text{ on } \Gamma, \qquad [\![\mathbf{u}]\!]_\Gamma \cdot \mathbf{n}_\Gamma = \frac{\kappa_\Gamma^n}{\ell_\Gamma} \xi^{-1} (\{\!\{p\}\!\}_\Gamma - p_\Gamma) \text{ on } \Gamma, \tag{2}$$

where $\xi \in \left(0, \frac{1}{2}\right]$ is a user-dependent model parameter and $\kappa_\Gamma^n : \Gamma \to \mathbb{R}$ represents the normal permeability inside the fracture. From now, we refer to the advective velocity fields as the bulk Darcy velocity $\mathbf{u}$ and the tangential fracture Darcy velocity $\mathbf{u}_\Gamma := -K_\Gamma \nabla_\tau p_\Gamma$.

2.3 Diffusion-Dispersion Tensors

Following [6], we assume that the bulk diffusion-dispersion tensor $\mathbf{D} : \Omega_\mathrm{B} \to \mathbb{R}^{2\times 2}$ and the fracture diffusion-dispersion coefficient $D_\Gamma : \Gamma \to \mathbb{R}$ are such that

$$\mathbf{D} := \phi \left(d_\mathrm{m} \mathbf{I}_2 + |\mathbf{u}| (d_\mathrm{l} \mathbf{E}(\mathbf{u}) + d_\mathrm{t} (\mathbf{I}_2 - \mathbf{E}(\mathbf{u}))) \right), \tag{3a}$$

$$D_\Gamma := \phi_\Gamma \left(\ell_\Gamma d_\mathrm{m}^\Gamma + |\mathbf{u}_\Gamma| d_\mathrm{l}^\Gamma \right), \tag{3b}$$

where $\mathbf{u}$ and $\mathbf{u}_\Gamma$ are defined in Sect. 2.2, $|\cdot|$ is the euclidian norm, and the scalar functions $\phi, d_\mathrm{m}, d_\mathrm{l}, d_\mathrm{t} : \Omega \to \mathbb{R}$ and $\phi_\Gamma, d_\mathrm{m}^\Gamma, d_\mathrm{l}^\Gamma : \Gamma \to \mathbb{R}$ are, respectively, the bulk porosity, molecular diffusion, longitudinal and transverse dispersion coefficients, and the fracture porosity, molecular diffusion and longitudinal dispersion coefficients. In (3a), $\mathbf{I}_2 \in \mathbb{R}^{2\times 2}$ is the identity matrix and $\mathbf{E}(\mathbf{u}) := |\mathbf{u}|^{-2}(\mathbf{u}\otimes\mathbf{u}) \in \mathbb{R}^{2\times 2}$ denotes the orthogonal projection matrix in the direction of $\mathbf{u}$. In the reduced model (PTFPM), the fracture diffusion-dispersion coefficient D_Γ depends on a fracture transverse dispersion coefficient. Here, the fracture transverse dispersion coefficient is rather integrated into the transmission conditions; see Remark 1.

2.4 The Reduced Model

For a fixed $T > 0$, we denote by $\Omega_\mathrm{B}^T := (0,T)\times\Omega_\mathrm{B}$ and $\Gamma^T := (0,T)\times\Gamma$ the temporal-spatial domains of interest, and by $\partial\Omega_\mathrm{B}^T := (0,T)\times\partial\Omega_\mathrm{B}$ and $\partial\Gamma^T := (0,T)\times\Gamma$ their respective boundaries. The reduced model for the passive transport of a solute in a fractured porous medium hinges into seeking the concentration of the solute in the bulk $c : \Omega^T \to \mathbb{R}$ and in the fracture $c_\Gamma : \Gamma^T \to \mathbb{R}$ such that

$$\phi\partial_t c + \nabla\cdot(\mathbf{u}c - \mathbf{D}\nabla c) + f^- c = f^+\widehat{c} \qquad \text{in } \Omega_\mathrm{B}^T, \tag{4a}$$

$$-\mathbf{D}\nabla c\cdot\mathbf{n}_{\partial\Omega} = 0 \qquad \text{on } \partial\Omega_\mathrm{B}^T, \tag{4b}$$

$$\ell_\Gamma\phi_\Gamma\partial_t c_\Gamma + \nabla_\tau\cdot(\mathbf{u}_\Gamma c_\Gamma - D_\Gamma\nabla_\tau c_\Gamma) + \ell_\Gamma f_\Gamma^- c_\Gamma = \ell_\Gamma f_\Gamma^+\widehat{c_\Gamma} + [\![\mathbf{u}c - \mathbf{D}\nabla c]\!]_\Gamma\cdot\mathbf{n}_\Gamma \qquad \text{in } \Gamma^T, \tag{4c}$$

$$-D_\Gamma\nabla_\tau c_\Gamma\cdot\boldsymbol{\tau}_{\partial\Gamma} = 0 \qquad \text{on } \partial\Gamma^T, \tag{4d}$$

where $\mathbf{u}$ and $\mathbf{u}_\Gamma$ are defined in Sect. 2.2, $\mathbf{D}$ and D_Γ are defined in Sect. 2.3, the terms $f^\pm := 1/2(|f| \pm f)$ and $f_\Gamma^\pm := 1/2(|f_\Gamma| \pm f_\Gamma)$ denote the positive or negative part of f and f_Γ, respectively, and the scalar functions $\widehat{c} : \Omega_\mathrm{B}^T \to \mathbb{R}$ and $\widehat{c_\Gamma} : \Gamma^T \to \mathbb{R}$ stand for the concentration of solute as it is injected in the bulk and in the fracture, respectively. The following transmission conditions, along with initial conditions $c(t=0) = c_0$ in Ω_B and $c_\Gamma(t=0) = c_{\Gamma,0}$ in Γ, close the problem:

$$\begin{aligned}
\{\!\{\mathbf{u}c - \mathbf{D}\nabla c\}\!\}_\Gamma\cdot\mathbf{n}_\Gamma &= \frac{\mathscr{D}_\Gamma^n}{\ell_\Gamma}[\![c]\!]_\Gamma + \{\!\{c\}\!\}_\Gamma\{\!\{\mathbf{u}\}\!\}_\Gamma\cdot\mathbf{n}_\Gamma + \frac{1}{8}[\![c]\!]_\Gamma[\![\mathbf{u}]\!]_\Gamma\cdot\mathbf{n}_\Gamma \text{ on } \Gamma,\\
[\![\mathbf{u}c - \mathbf{D}\nabla c]\!]_\Gamma\cdot\mathbf{n}_\Gamma &= \frac{\mathscr{D}_\Gamma^n}{\ell_\Gamma}\xi^{-1}(\{\!\{c\}\!\}_\Gamma - c_\Gamma) + \frac{1}{2}(\{\!\{c\}\!\}_\Gamma + c_\Gamma)[\![\mathbf{u}]\!]_\Gamma\cdot\mathbf{n}_\Gamma \text{ on } \Gamma,
\end{aligned} \tag{5}$$

where ξ is the user-dependent model parameter introduced in Sect. 2.2. The term $\mathscr{D}_{\Gamma}^{n} : \Gamma \to \mathbb{R}$ represents the normal diffusion-dispersion coefficient of the fracture. In the reduced model (PTFPM), $\mathscr{D}_{\Gamma}^{n}$ does not depend on the surrounding unknowns. For a more accurate description of the exchange between the bulk and the fracture, we propose the following definition:

$$\mathscr{D}_{\Gamma}^{n} := \phi_{\Gamma}(d_{\mathrm{m}}^{\Gamma} + d_{\mathrm{t}}^{\Gamma} |\{\{\mathbf{u}\}\}_{\Gamma} \cdot \mathbf{n}_{\Gamma}|), \tag{6}$$

that depends on (1) the porosity of the fracture ϕ_{Γ}, (2) the fracture molecular diffusion coefficient d_{m}^{Γ}, and (3) on the fracture transverse dispersion $d_{\mathrm{t}}^{\Gamma} : \Gamma \to \mathbb{R}$ weighted by the normal component of the average of the bulk Darcy velocity $\mathbf{u}$. From now, we refer to the reduced model (4)–(6) as (PTFPM$\star$).

Remark 1 The fracture transverse dispersion d_{t}^{Γ} describes the property of the solute to diffuse in the orthogonal directions of the fracture advective velocity field $\mathbf{u}_{\Gamma}$. In the framework of reduced models, it is assumed that the normal component of the fracture Darcy velocity is a linear combination of the normal component of the surrounding bulk Darcy velocity; see [5]. Therefore, it seems natural to integrate the fracture transverse dispersion coefficient into the transmission conditions (5).

3 Numerical Experiments

In this section we numerically compare the two reduced models (PTFPM) and (PTFPM$\star$). For the sake of brevity, we refer to the previous works [2, 3] for the space discretization aspects and to [3, Section 5] for an in-depth description of the test case configurations considered in this section. To discretize in time, we use a backward Euler scheme and consider a uniform partition $(t^n)_{0\leq n\leq N}$ of the time interval $(0, T)$ with $t^0 = 0$, $t^N = T$ and $t^n - t^{n-1} = \Delta t$ for all $1 \leq n \leq N$.

3.1 Injection and Production Wells

In petroleum engineering, the source terms f and f_{Γ} are used to model injection and production wells in the bulk and in the fracture, respectively; see [7]. Through this section, the injection well sits in $\mathbf{x}_{\mathrm{i}} \in \Omega_{\mathrm{B}}$, the production one in $\mathbf{x}_{\mathrm{p}} \in \Omega_{\mathrm{B}}$, and both are modeled by the source term f defined such that

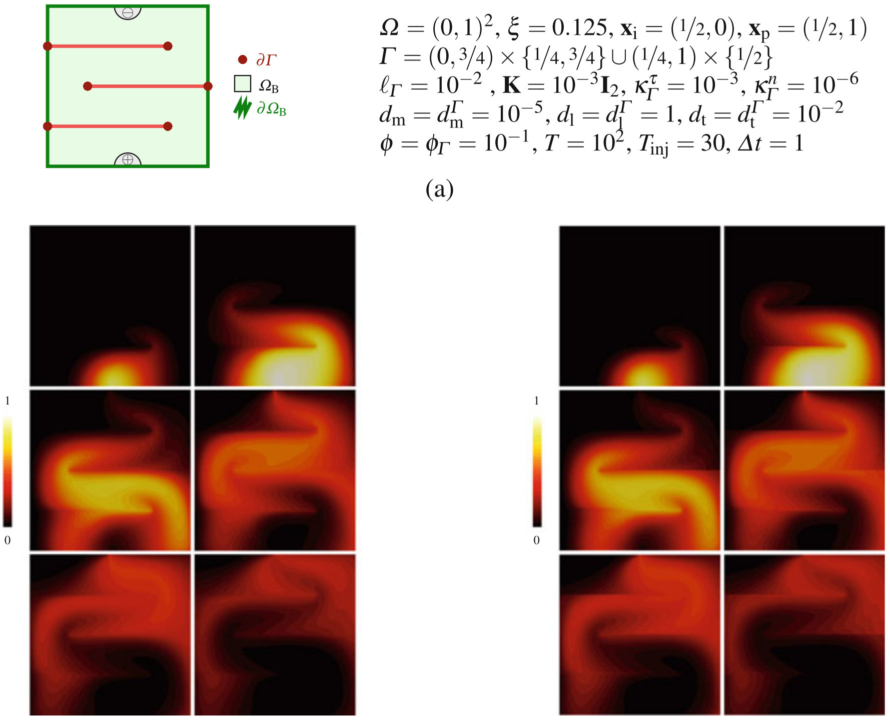
$$f(\mathbf{x}) = \frac{1}{2}\left(\tanh\left(200\left(0.025 - |\mathbf{x} - \mathbf{x}_{\mathrm{i}}|\right)\right) - \tanh\left(200\left(0.025 - |\mathbf{x} - \mathbf{x}_{\mathrm{p}}|\right)\right)\right).$$

For a fixed $T_{\mathrm{inj}} > 0$, the concentration of solute as it is injected is defined as $\widehat{c}(t, \mathbf{x}) = 1$ if $t < T_{\mathrm{inj}}$ and $\widehat{c}(t, \mathbf{x}) = 0$ otherwise. In the fracture, we set

$f_\Gamma \equiv \widehat{c_\Gamma}(t, \mathbf{x}) \equiv 0$. We assume that the initial concentration of solute is zero in Ω_{B} and Γ.

3.2 Impermeable Fractures

We first consider a test case modelling the passive displacement of a solute in a fractured porous medium where fractures act as barriers. The domain configuration and user parameters are detailed in Fig. 2a. With this configuration the solute is expected to go from the injection well toward the production well by avoiding the fractures; see [3, Section 5.2]. In Fig. 2b and c, we display the bulk concentrations of both reduced models (PTFPM) and (PTFPM⋆) obtained at different time t. In both cases, the solute follows the corridors designed by the fractures acting as barriers and goes from the injection to the production well for the two configurations.



$\Omega = (0,1)^2$, $\xi = 0.125$, $\mathbf{x}_{\mathrm{i}} = (1/2, 0)$, $\mathbf{x}_{\mathrm{p}} = (1/2, 1)$
$\Gamma = (0, 3/4) \times \{1/4, 3/4\} \cup (1/4, 1) \times \{1/2\}$
$\ell_\Gamma = 10^{-2}$, $\mathbf{K} = 10^{-3}\mathbf{I}_2$, $\kappa_\Gamma^\tau = 10^{-3}$, $\kappa_\Gamma^n = 10^{-6}$
$d_{\mathrm{m}} = d_{\mathrm{m}}^\Gamma = 10^{-5}$, $d_{\mathrm{l}} = d_{\mathrm{l}}^\Gamma = 1$, $d_{\mathrm{t}} = d_{\mathrm{t}}^\Gamma = 10^{-2}$
$\phi = \phi_\Gamma = 10^{-1}$, $T = 10^2$, $T_{\mathrm{inj}} = 30$, $\Delta t = 1$

(a)

(b)

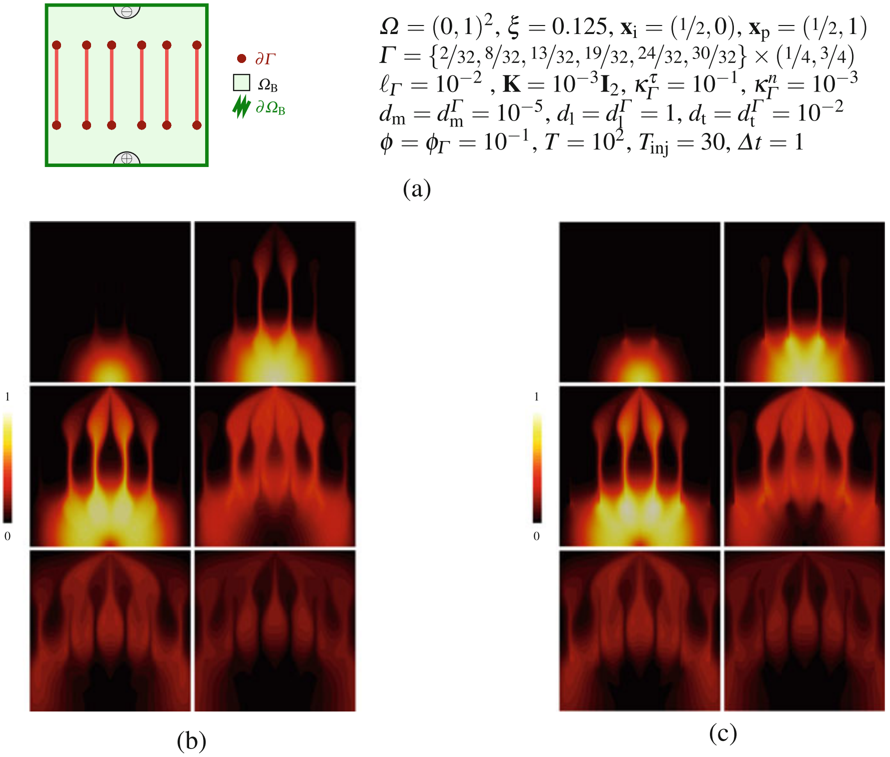
(c)

Fig. 2 Domain configuration (left) and parameters (right) (top, **a**), and snapshots of the bulk concentrations c (bottom) for the test case of Sect. 3.2 (impermeable fractures). Displayed times (from left to right, top to bottom): $t = 5,\ 20,\ 40,\ 60,\ 80,\ 100$. (**b**) Reduced models (PTFPM), (**c**) Reduced models (PTFPM⋆)

However, discontinuities of the bulk concentration c across the fractures are more pronounced in the reduced model (PTFPM⋆). This arises from the fact that the fracture transverse coefficient d_t^{Γ} depends on the surrounding bulk Darcy velocity, which, in this case, avoids fractures.

3.3 Permeable Fractures

We now consider fractures acting as conduits. Both the domain configuration and user parameters are displayed in Fig. 3a. With this choice, it is expected that the solute is attracted by the fractures; see [3, Section 5.3]. In Fig. 3b and c, we display bulk concentrations c of both reduced models (PTFPM) and (PTFPM⋆), at different time t. In both cases, we can distinctly see that the solute channeled by the fractures flows towards the production well faster than the solute in the surrounding bulk



$\Omega = (0,1)^2$, $\xi = 0.125$, $\mathbf{x}_i = (1/2, 0)$, $\mathbf{x}_p = (1/2, 1)$
$\Gamma = \{2/32, 8/32, 13/32, 19/32, 24/32, 30/32\} \times (1/4, 3/4)$
$\ell_\Gamma = 10^{-2}$, $\mathbf{K} = 10^{-3}\mathbf{I}_2$, $\kappa_\Gamma^\tau = 10^{-1}$, $\kappa_\Gamma^n = 10^{-3}$
$d_m = d_m^\Gamma = 10^{-5}$, $d_l = d_l^\Gamma = 1$, $d_t = d_t^\Gamma = 10^{-2}$
$\phi = \phi_\Gamma = 10^{-1}$, $T = 10^2$, $T_{inj} = 30$, $\Delta t = 1$

(a)

(b)

(c)

Fig. 3 Domain configuration (left) and parameters (right) (top, **a**), and snapshots of the bulk concentrations c (bottom) for the test of Sect. 3.3 (permeable fractures). Displayed times (from left to right, top to bottom): $t = 5, 15, 30, 50, 80, 100$. (**b**) Reduced models (PTFPM), (**c**) Reduced models (PTFPM⋆)

medium. We remark that the discontinuities of the concentration c are also in this case more pronounced at the neighbourhood of the fracture tips located near the injection well for the reduced model (PTFPM$\star$).

In practice, the molecular diffusion coefficients are set to zero. This delicate case is prone to instabilities since the diffusion-dispersion tensors can be degenerate in some parts of the domain where the Darcy velocities vanish. Moreover, the fracture normal diffusion-dispersion coefficient depends, in this case, only on the Darcy velocity $\mathbf{u}$. In Fig. 4, we display the concentrations obtained by the two reduced models (PTFPM) and (PTFPM$\star$) at different time t upon setting $d_{\mathrm{m}} = d_{\mathrm{m}}^{\Gamma} = 0$. Clearly, one can see instabilities at the neighborhood of the fractures for the reduced model (PTFPM); see Fig. 4a. On the other hand, the reduced model (PTFPM$\star$) seems to handle without difficulty this particular case; see Fig. 4b. We also note that the discontinuities are more pronounced in the reduced model (PTFPM$\star$), and that the concentrations of the two reduced models (PTFPM) and (PTFPM$\star$) behave differently at the vicinity of the fractures.

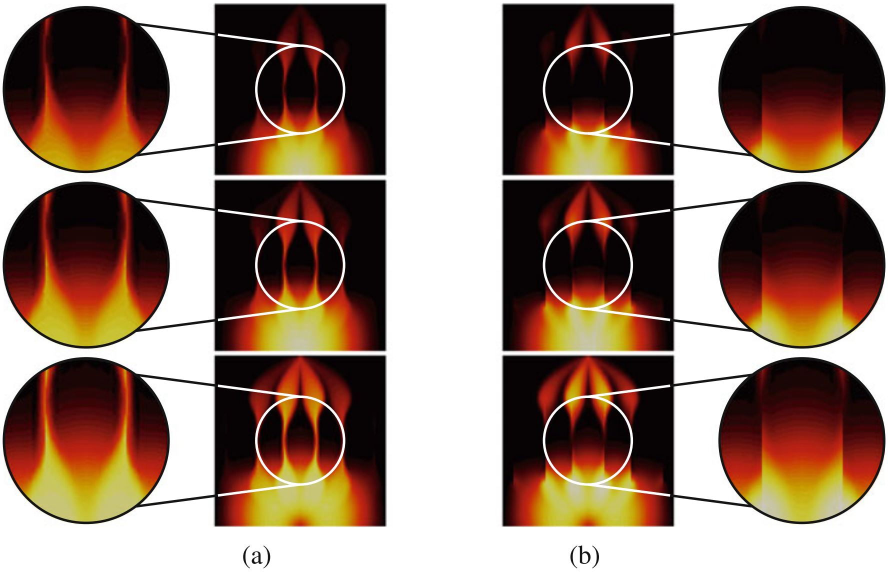


Fig. 4 Snapshots of the bulk concentration c and zoom on the vicinity of the fracture for the test case of Sect. 3.3 (permeable fracture, vanishing molecular diffusion). Displayed times: $t = 15,\ 20,\ 30$. (**a**) Reduced models (PTFPM), (**b**) Reduced models (PTFPM$\star$)

References

1. Chave, F.: Hybrid High-Order methods for interface problems. Ph.D. thesis, University of Montpellier, France and Polytechnic University of Milan, Italy (2018). https://tel.archives-ouvertes.fr/tel-01926061v2/document
2. Chave, F., Di Pietro, D.A., Formaggia, L.: A Hybrid High-Order method for Darcy flows in fractured porous media. SIAM J. Sci. Comput. **40**(2), A1063–A1094 (2018). https://doi.org/10.1137/17M1119500
3. Chave, F., Di Pietro, D.A., Formaggia, L.: A Hybrid High-Order method for passive transport in fractured porous media. Int. J. Geomath. **10**(12) (2019). https://doi.org/10.1007/s13137-019-0114-x
4. Fumagalli, A., Keilegavlen, E.: Dual virtual element methods for discrete fracture matrix models. Oil Gas Sci. Technol. – Rev. IFP Energies nouvelles **74**(41) (2019). https://doi.org/10.2516/ogst/2019008
5. Martin, V., Jaffré, J., Roberts, J.E.: Modeling fractures and barriers as interfaces for flow in porous media. SIAM J. Matrix Analysis and Applications **26**(5), 1667–1691 (2005). https://doi.org/10.1137/S1064827503429363
6. Peaceman, D.W.: Improved treatment of dispersion in numerical calculation of multidimensional miscible displacement. Soc. Petrol. Eng. J. **6**, 213–216 (1966). https://doi.org/10.2118/1362-PA
7. Todd, M.R., O'Dell, P.M., Hirasaki, G.J.: Methods for increased accuracy in numerical reservoir simulators. Soc. Petrol. Eng. J. **12**, 515–530 (1972). https://doi.org/10.2118/3516-PA

Several Agent-Based and Cellular Automata Mathematical Frameworks for Modeling Pancreatic Cancer

Jiao Chen and Fred J. Vermolen

Abstract Mathematical modeling sheds light on cancer research. In addition to reducing animal-based experiments, mathematical modeling is able to provide predictions and prevalidate hypotheses quantitatively. In this work, two different agent-based frameworks regarding cancer modeling are summarised. In contrast, cell-based models focus on the behavior of every single cell and presents the interaction of cells on a small scale, whereas, cellular automata models are used to simulate the interaction of cells with their microenvironment on a large tissue scale.

1 Introduction

In agent-based modeling, a collection of autonomous decision-making entities (called agents) is utilized to model a system. Based on a set of rules, each agent makes the decision individually and executes various behaviors for the whole system [8]. Therefore, agent-based modeling represents a dynamic and interactive system, which has been applied in various fields like biomedical research [5], chemistry [10], market analysis [1], etc.

J. Chen (✉)
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands

Department of Biomedical Engineering and Physics, Amsterdam UMC, Amsterdam,
The Netherlands
e-mail: j.chen-6@tudelft.nl

F. J. Vermolen
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands

Division of Mathematics and Statistics, Faculty of Sciences, Hasselt University, Diepenbeek,
Belgium
e-mail: fred.vermolen@uhasselt.be

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_25

Agent-based modeling is capable of simulating a broad spectrum of length-scales, which has been classified by Van Liedekerke et al. [11] into the following three types:

- **Lattice-based model**, where the model is developed based on regular lattice sites in a spatial computational domain. In biomedical modeling, cell bioprocesses are represented by transitions of each lattice state such that the model shows the evolution of a system by a discrete time-stepping mechanism or a continuous-time framework [9]. According to [11], the lattice-based model can be further classified into cellular automata models, lattice gas cellular and cellular potts models.
- **Off-lattice model**, which means the model is lattice-free and each agent is allowed to move in any direction rather than restricting agents to lattice sites. Some examples are center-based models, deformable cell models, and vertex model, etc. [11].
- **Hybrid discrete-continuum model**. To solve large multicellular systems, discrete agent-based models need large computational time since individual cells are concerned. The continuum model is able to solve PDEs for tissue dynamics or other complicated issues. Therefore, a hybrid discrete-continuum model is proposed to simulate multiscale models [11].

2 Agent-Based Models

Agent-based (or cell-based) models deal with biological cells as discrete entities in a computational domain. One of the advantages is the straightforward integration of cell-level processes like cell proliferation, cell death, cell mutation, etc. and the intracellular interactions. We develop a cell-based model with an application to pancreatic cancer therapy at early stages [5]. In this work, we consider three cell phenotypes, i.e. epithelial cells, cancer cells, T-lymphocytes, which are visualized as blue, red and green colored circles in Fig. 1, respectively. Figure 1 shows consecutive snapshots of the migration of T-lymphocytes in pancreatic cancer at an early stage. Since pancreatic cancer cells accumulate in rounded (three dimensional) clusters, we model the computational domain as a circular structure [5]. To visualize cell mutation, epithelial cells change color from blue to filled red. Moreover, other cell bioprocesses such as cell division and cell death are incorporated. Typically, in a competitive environment, cancer cells have a growth and proliferation (division) advantage over other healthy cells, therefore, the number of cancer cells in Fig. 1 accounts for the majority at time = 150 h.

In this model, the migration of epithelial and cancer cells is mechanotaxis updated by

$$\mathbf{r}_i^n = \mathbf{r}_i^{n-1} + \Delta t \alpha_i \hat{M}_i(\mathbf{r}^n) + \eta \Delta \mathbf{W}(t), \tag{1}$$

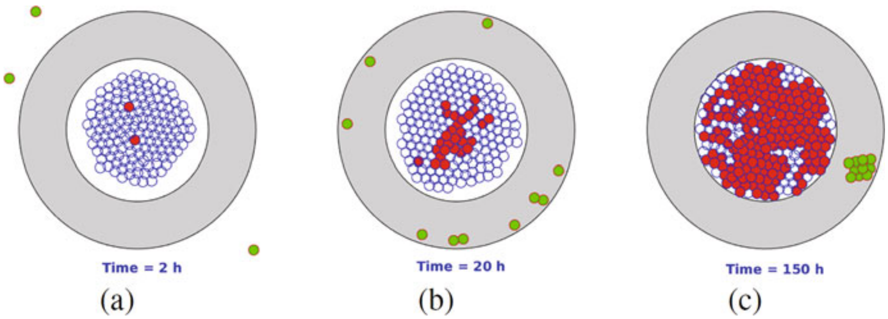


Fig. 1 Consecutive snapshots of cancer progression and T-lymphocytes migration when time = 2 h (**a**), time = 20 h (**b**) and time = 150 h (**c**), respectively. The blue, red and green color denote epithelial cells, cancer cells and T-lymphocytes

where $\mathbf{r}_i$ and α_i represents the position of cell i and its velocity parameter. The $\hat{M}_i(\mathbf{r})$ is the total mechanical signal comprising of traction force caused by strain energy density and a repulsive force. In addition, η denotes a constant and $\Delta\mathbf{W}(t)$ takes care of random walk (diffusion), which is a Wiener process. In contrast, the locomotion of T-lymphocytes is chemo-mechanotaxis, where T-lymphocytes are attracted by a type of chemokine secreted by cancer cells. The displacement of T-lymphocytes is described as

$$\mathbf{r}_j^n = \mathbf{r}_j^{n-1} + \beta\nabla c(t, \mathbf{r}_j^{n-1})\Delta t + \eta\Delta\mathbf{W} - M^{\mathrm{mc}}(\mathbf{r}_j^{n-1})\mathbf{z}_j^{n-1}\Delta t. \tag{2}$$

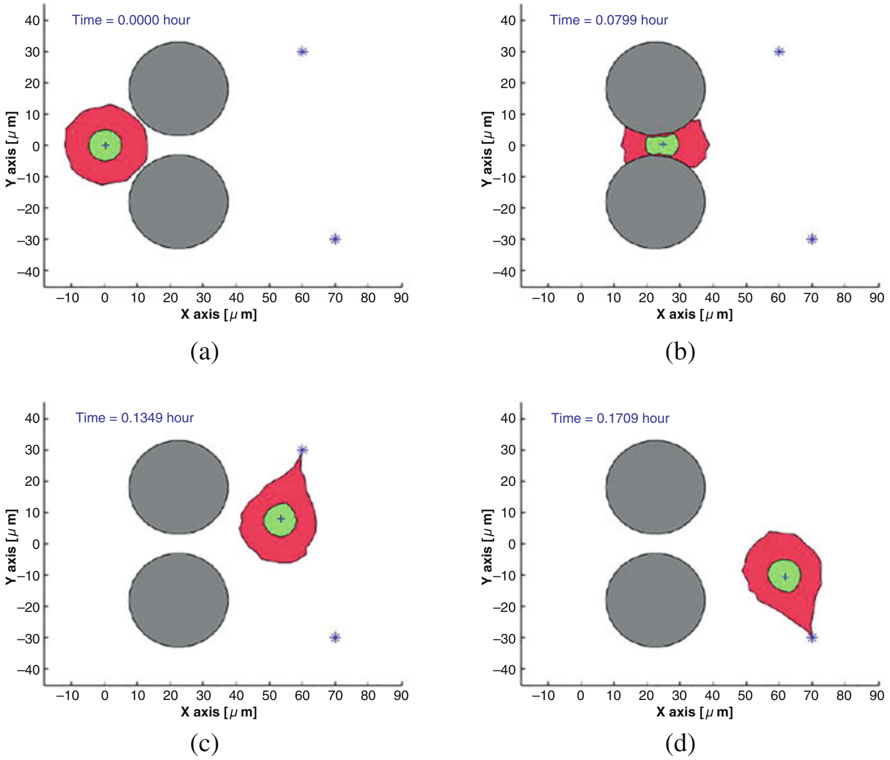
Here $c(t, \mathbf{r}_j^{n-1})$ denotes the concentration of chemokine secreted by cancer cells at time step $n-1$ and β is a constant. Whenever any two cells contact with each other, the repulsive force $M^{\mathrm{mc}}(\mathbf{r}_j)$ repels two cells with direction $\mathbf{z}_j$.

Next we consider a deformable cell model. The deformable cell model simulates the evolution of cell shape during the interaction with the microenvironment, see an example in [3]. In Fig. 2, some snapshots at consecutive times are plotted to show the deformation of a migrating cell and its nucleus denoted in red and green color, respectively. Furthermore, circles in grey color are regarded as two stiff obstacles and the cell penetrates the cavity by the attraction of two source points (blue asterisk). The migration of the cell and its nucleus is determined by chemotaxis, which can be expressed as

$$\mathbf{x}_i(t^{p+1}) = \mathbf{x}_i(t^p) + \Delta t \cdot (\beta\nabla c_i(t^{p+1}) + \alpha(\mathbf{x}_i^n(t^p) + \hat{\mathbf{x}}_i - \mathbf{x}_i(t^{p+1}))) + \eta\Delta\mathbf{W}, \tag{3}$$

and

$$\mathbf{x}_i^n(t^{p+1}) = \mathbf{x}_i^n(t^p) + \Delta t \cdot (-\alpha(\mathbf{x}_i^n(t^p) + \hat{\mathbf{x}}_i - \mathbf{x}_i(t^{p+1})) + \alpha^n(\mathbf{x}_c(t^p) + \hat{\mathbf{x}}_i^n - \mathbf{x}_i^n(t^{p+1}))) + \eta\Delta\mathbf{W}. \tag{4}$$



(a) (b) (c) (d)

Fig. 2 Consecutive snapshots of the deformation of one migrating cell and its nucleus when time $= 0$ h, time $= 0.0799$ h, time $= 0.1349$ h and time $= 0.1709$ h, in red and green color, respectively. Two stiff obstacles are visualized in grey circles and source points are denoted by blue asterisks

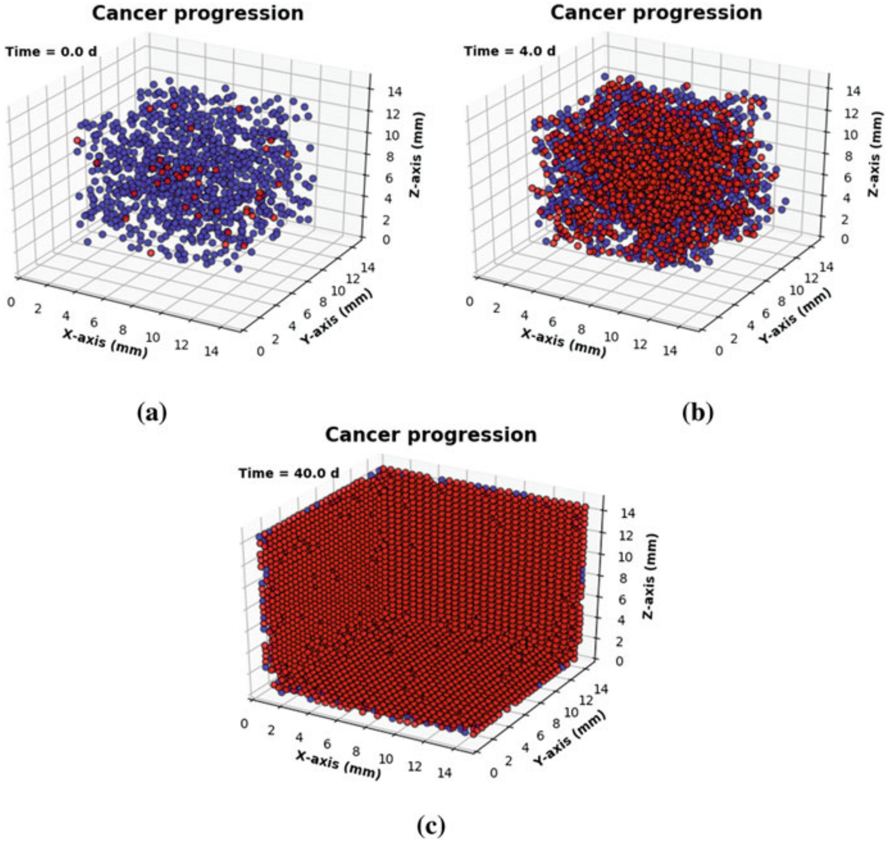
Note that $\mathbf{x}_i$ and $\mathbf{x}_i^n$ denote the location of a node i on the cell membrane and nucleus surface, respectively. The second term in Eqs. (3) and (4) represents the interaction between the nucleus surface and cell membrane. Analogously, we model random walk by using a Wiener process $\Delta\mathbf{W}$, where η is a constant.

Cells are subject to large deformation during migration to adapt to the environment. This cell-based model can be applied to the deformation of an immune cell with the attraction of a pathogen source. In addition, it also can be used to describe the deformation of a cancer cell during the migration to the oxygen source as part of the metastasis process.

3 The Cellular Automata Model

The cellular automata model is a lattice-based method, which has been used in various fields. Specifically, a computational domain is divided into lattice sites, where each lattice site can be occupied by one cell or multiple cells. Each lattice site can be in a discrete state and is able to 'jump' from one state into another state. Moreover, one single cell is able to share a few lattice sites in some cases. We develop a three-dimensional model to simulate the cancer progression and recession under virotherapy [2], in which one lattice point is occupied by multiple cells. As a result, Fig. 3 shows cancer progression at early stages in a 15×15×15 mm^3 domain. To mimic cell mutation, epithelial cells (in blue color) are allowed to turn into cancer cells (in red color). As mentioned earlier, cancer cells have more growth and division rates than normal cells in a competitive environment with limited space and nutrition. The number of cancer cells increases significantly and thereby cancer progresses to a large volumetric fraction in the simulations.

In the model, any lattice site has three discrete states, i.e. unoccupied (or dead cell) state, epithelial cell state, cancer cell state. Under certain conditions, a lattice



(a)

(b)

(c)

Fig. 3 Consecutive snapshots of cancer progression when time = 0 days, time = 4 days and time = 40 days, respectively, in cellular automata model. The blue and red color represent epithelial cells and cancer cells. The computational domain is 15×15×15 mm^3

point, i, can change state and the transition probability P within a time interval (t_0, t_0 + Δt) is defined as

$$P = \int_{t_0}^{t_0+\Delta t} f(\lambda_i, t)\mathrm{d}t \simeq 1 - \exp(-\lambda_i \Delta t). \tag{5}$$

where $f(\lambda_i, t)$ is an exponential distribution and λ_i denotes the probability rate at grid node i per unit of time of state transition. Note that the probability rate for the change of state depends on the two states between which the grid node undergoes the change. Regarding our model, one of the merits is the flexibility of the input parameters. With proper input variables, our numerical results can reproduce experimental results very well, see Fig. 4 [2], where curves show cancer growth during 50 days. Taking the animal-based experimental results from [6], cancer grows under gemcitabine intervention compared with a control experiment showing in the blue line and black line in Fig. 4. In comparison, modeled results indicated by the red lines are able to predict the cancer progression well according to experimental curves.

Subsequently, this cellular automata model is extended to oncolytic virotherapy in pancreatic cancer at early stages [2]. We assume that a three-dimensional domain is fully colonized by cancer cells and at a certain time a dose of viruses is given

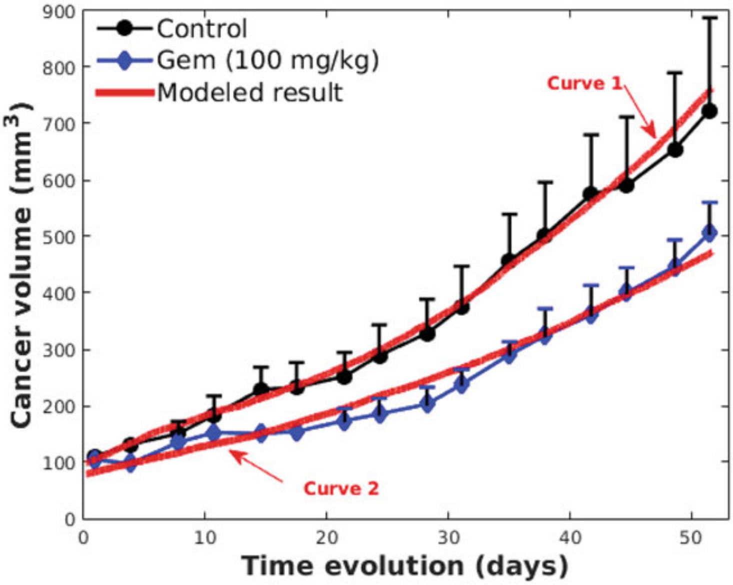


Fig. 4 Cancer growth with the respect of time in days [2]. The red curves show the numerical results from the cellular automata model, whereas the black and blue lines represent the cancer growth without gemcitabine and with gemcitabine, respectively. The experimental results are taken from the work [6]

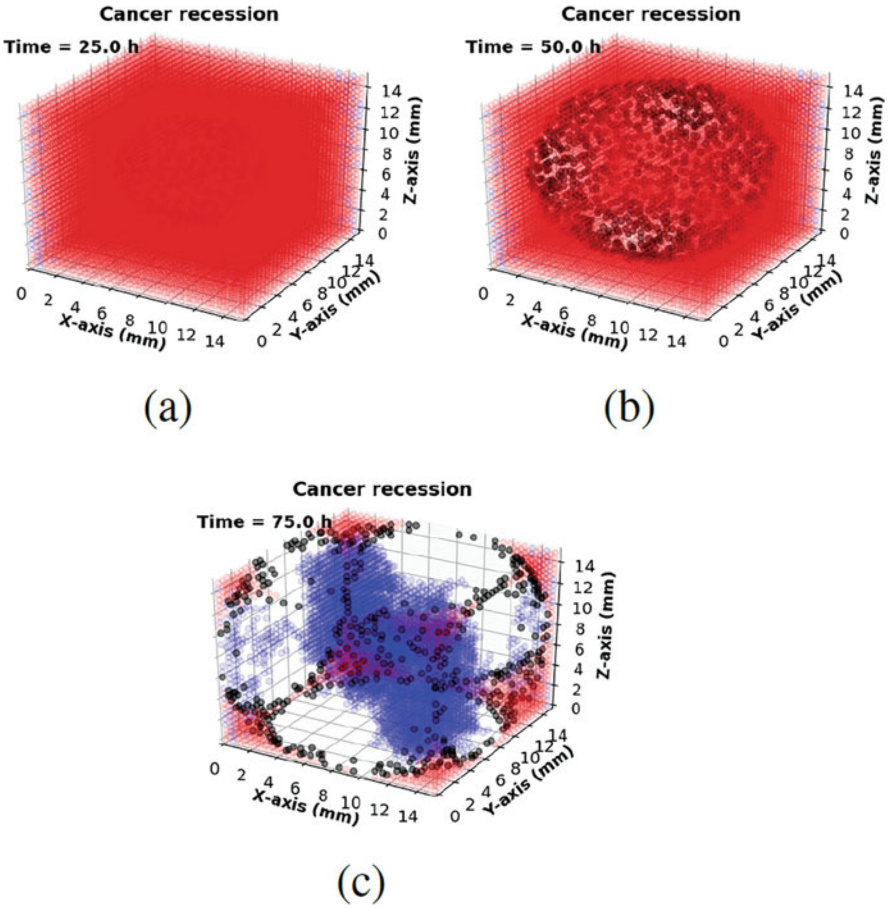


Fig. 5 Consecutive snapshots of cancer recession when time = 25 h, time = 50 h and time = 75 h, respectively, in cellular automata model with an application to virotherapy in pancreatic cancer. In the computational domain 15×15×15 mm^3, the epithelial cells, cancer cells, infected cells are denoted in blue, red and black color, respectively. In addition, the lattice sites in white color represent the dead cells or unoccupied states

intratumorally by injection (see Fig. 5). Figure 5 shows cancer recession under virotherapy, where cancer cells, epithelial cells, infected cells are visualized in red, blue and black color, respectively. Once cancer cells die due to viral replication, the lattice points will transform from the cancer state to the unoccupied state, which is indicated in white color. Since the viruses are injected in the center of the domain, viruses diffuse and infect cancer cells from the central lattice points with the evolution of time (see Fig. 5b). The model of viral diffusion is defined as

$$\begin{cases} \frac{\partial c(\mathbf{r})}{\partial t} = D\Delta c(\mathbf{r}) + \gamma(t)\delta(\mathbf{x}- \mathbf{x}_p) + \beta c(\mathbf{r})(1-\frac{c(\mathbf{r})}{N_v}) \\ D\frac{\partial c(\mathbf{r})}{\partial n} + Tc(\mathbf{r}) = 0, \quad \text{on } \partial\Gamma \end{cases}, \tag{6}$$

where c($\mathbf{r}$) is the viral concentration at any lattice point and D denotes the viral diffusivity. The Dirac delta function $\delta(\mathbf{x})$ mimics the viral source with a time-related secretion rate $\gamma(t)$ at position $\mathbf{x}_p$. Note that $\beta c(\mathbf{r})(1 - \frac{c(\mathbf{r})}{N_v})$ is a reaction term to simulate the viral replication, which only takes place in the grid nodes that are in the cancer state. Here β denotes the proliferation rate of virus and N_v represents a burst size of viruses. On the boundary Γ, viruses are able to disperse to the neighbor tissue or organs with a mass transfer rate coefficient T. As more and more cancer cells are eliminated by viruses, there is a 'wound' region, characterized by cells in the 'unoccupied state' appearing in the tissue as a result. However, healthy cells migrate to this wound from neighbor tissue or organs and hence fill in this gap by proliferation. In other words, this model could also be used for simulating wound healing.

4 Uncertainty Quantification

Using the cell deformation model, see Eqs. (3)–(4) and Fig. 2, we quantify the influence of uncertainty in the input data on the time of metastasis, which is modeled by the time at which a cancer cell exists a blood vessel. In the modeling set-up, cancer cells transmigrate through the walls of a blood vessel and subsequently they are transported by the bloodstream to enter at a different part of the body where they can colonize by forming new tumors. The set-up deviates from Fig. 2, more details can be found in [4]. The uncertainty quantification is performed by Monte Carlo simulations, see [7], in which the input parameters, here the cell size and the size of the aperture of the blood vessel are sampled from statistical distributions. The results indicate a significant positive correlation (sample correlation coefficient r $= 0.79$) between the metastasis time and the cell size. Hence the larger the cancer cell, the more time it takes to metastasize since transmigration through a blood vessel is more difficult for larger cells. Furthermore, the Monte Carlo simulations hint at a weaker negative correlation (r $= -0.17$) between the metastasis time and the size of the aperture of the vessel. This confirms the intuition that a permeable vessel facilitates the transmigration of the cell, and hence enhances metastasis (Fig. 6).

Moreover, the Monte Carlo method is further used to predict the likelihood of successful cancer treatment in our other works [2, 5]. The corresponding results are hopeful to aid experiment design and prevalidation before clinical trials.

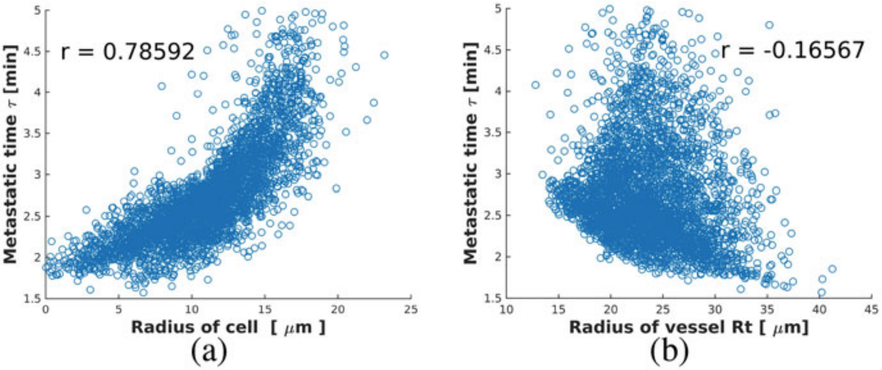


Fig. 6 Scatter plots of Monte Carlo simulations [4]. (**a**) Correlation between cell size and cell metastatic time with coefficient r = 0.78592; (**b**) correlation between vessel size and cell metastatic time with coefficient r = −0.16567

5 Discussion and Conclusions

Regarding cancer modeling, we develop different agent-based frameworks, namely the cell-based model and cellular automata model, which are compared in this paper. The cell-based model, where each individual cell is considered, is beneficial for modeling at small scales. The morphology of the cells can be fixed as in the model applied in pancreatic cancer at early stages [5]. Furthermore, one can zoom into the process of cell migration where one models morphological changes of each cell, such as in the simulation framework with an application to cancer metastasis [3]. Furthermore, the intercellular biomechanics and interactions between cells and their microenvironment are incorporated. However, with an increase in the number of cells, the cell-based model will be time-consuming, and therefore cellular automata model could be a computationally 'cheap' alternative. Besides the cellular automata model, a continuum model for the viral spread is taken into account by using the reaction-diffusion equation [2]. As we expected, the numerical results show consistency with the results from the experiments in the literature.

Computational modeling has played and will continue to play a pivotal role in cancer research and treatment. The computational framework will possess aspects from both complicated physics-based approaches as well as from 'simple' tractable phenomenological modeling approaches.

References

1. A Charania and D DePasquale. Economic modeling of future space markets. In *NewSpace 2006 Conference, Las Vegas, Nevada*, 2006.
2. Jiao Chen, Daphne Weihs, and Fred J Vermolen. A cellular automata model of oncolytic virotherapy in pancreatic cancer. *Bulletin of Mathematical Biology*, 82(8):1–25, 2020.

3. Jiao Chen, Daphne Weihs, Marcel Van Dijk, and Fred J Vermolen. A phenomenological model for cell and nucleus deformation during cancer metastasis. *Biomechanics and modeling in mechanobiology*, 17(5):1429–1450, 2018a.
4. Jiao Chen, Daphne Weihs, and Fred J Vermolen. Monte Carlo uncertainty quantification in modelling cell deformation during cancer metastasis. *Proceedings of the CMBBE2018*, 2018b.
5. Jiao Chen, Daphne Weihs, and Fred J Vermolen. Computational modeling of therapy on pancreatic cancer in its early stages. *Biomechanics and modeling in mechanobiology*, pages 1–18, 2019.
6. David E Durrant, Anindita Das, Samya Dyer, Seyedmehrad Tavallai, Paul Dent, and Rakesh C Kukreja. Targeted inhibition of phosphoinositide 3-kinase/mammalian target of rapamycin sensitizes pancreatic cancer cells to doxorubicin without exacerbating cardiac toxicity. *Molecular pharmacology*, 88(3):512–523, 2015.
7. John Hammersley. *Monte Carlo methods*. Springer Science & Business Media, 2013.
8. Charles M Macal and Michael J North. Tutorial on agent-based modeling and simulation. In *Proceedings of the Winter Simulation Conference, 2005.*, pages 14–pp. IEEE, 2005.
9. Michael J Plank and Matthew J Simpson. Models of collective cell behaviour with crowding effects: comparing lattice-based and lattice-free approaches. *Journal of the Royal Society Interface*, 9(76):2983–2996, 2012.
10. Alessandro Troisi, Vance Wong, and Mark A Ratner. An agent-based approach for modeling molecular self-organization. *Proceedings of the National Academy of Sciences*, 102(2):255–260, 2005.
11. Paul Van Liedekerke, MM Palm, N Jagiella, and Dirk Drasdo. Simulating tissue mechanics with agent-based models: concepts, perspectives and some novel results. *Computational particle mechanics*, 2(4):401–444, 2015.

Error Bounds for Some Approximate Posterior Measures in Bayesian Inference

Han Cheng Lie, T. J. Sullivan, and Aretha Teckentrup

Abstract In certain applications involving the solution of a Bayesian inverse problem, it may not be possible or desirable to evaluate the full posterior, e.g. due to the high computational cost of doing so. This problem motivates the use of approximate posteriors that arise from approximating the data misfit or forward model. We review some error bounds for random and deterministic approximate posteriors that arise when the approximate data misfits and approximate forward models are random.

1 Introduction

An inverse problem consists of recovering an unknown parameter u that belongs to a possibly infinite-dimensional space $\mathcal{U}$ from noisy data y of the form

$$y = G(u) + \eta \in \mathcal{Y}, \tag{1}$$

where $\mathcal{Y}$ is the 'data space', $G : \mathcal{U} \to \mathcal{Y}$ is a known 'forward operator', and η is a random variable. In many problems of interest, the parameter space $\mathcal{U}$ is a subset of

H. C. Lie (✉)
Institut für Mathematik, Universität Potsdam, Potsdam, Germany
e-mail: hanlie@uni-potsdam.de

T. J. Sullivan
Mathematics Institute and School of Engineering, The University of Warwick, Coventry, UK
e-mail: t.j.sullivan@warwick.ac.uk

T. J. Sullivan
Zuse-Institut Berlin, Berlin, Germany
e-mail: sullivan@zib.de

A. Teckentrup
School of Mathematics, University of Edinburgh, Edinburgh, UK
e-mail: a.teckentrup@ed.ac.uk

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_26

an infinite-dimensional Banach space, the data space $\mathcal{Y}$ is often taken to be $\mathbb{R}^d$ for some possibly large $d \in \mathbb{N}$, and η is assumed to be Gaussian.

One of the main difficulties with inverse problems is that they often do not satisfy Hadamard's definition of well-posedness. To circumvent this difficulty, one may use the Bayesian approach, in which one incorporates information about the unknown u from existing data and from new data in the 'prior' probability measure μ_0 on $\mathcal{U}$ and in the 'data misfit' $\Phi : \mathcal{Y} \times \mathcal{U} \to \mathbb{R}$ respectively. If $\eta \in \mathbb{R}^d$ in (1) is distributed according to the normal distribution $N(0, \Gamma)$ with positive definite $\Gamma \in \mathbb{R}^{d\times d}$, then

$$\Phi(y, u) := \frac{1}{2} \| \Gamma^{-1/2} (y - G(u)) \|^2. \tag{2}$$

By Bayes' formula, the posterior μ^y is a probability measure on $\mathcal{U}$ that is absolutely continuous with respect to the prior μ_0, and has Radon–Nikodym derivative

$$\frac{\mathrm{d}\mu^y}{\mathrm{d}\mu_0}(u) := \frac{\exp(-\Phi(y, u))}{Z(y)}, \quad Z(y) := \int_{\mathcal{U}} \exp(-\Phi(y, u'))\mathrm{d}\mu_0(u'). \tag{3}$$

The posterior μ^y describes the distribution of the unknown u, conditioned upon the data y. By imposing conditions jointly upon Φ and μ_0, one can show that the Bayesian solution μ^y to the inverse problem depends continuously on the data, and one can prove the well-posedness of the Bayesian inverse problem; see [1].

For simplicity, we shall assume that the data y is given and fixed, and omit the dependence of the posterior, data misfit, and normalisation constant Z on y.

One challenge with solving Bayesian inverse problems in practice is that it is often not possible or desirable to evaluate the data misfit $\Phi(u)$ exactly. It then becomes necessary to find approximations Φ_N of the true data misfit Φ that can be computed more efficiently, such that for sufficiently large values of N, inference using the approximate misfit Φ_N effectively approximates inference using the true misfit Φ. Thus, one needs to identify conditions on Φ_N such that two criteria are fulfilled: first, an approximate posterior measure μ_N defined by

$$\frac{\mathrm{d}\mu_N}{\mathrm{d}\mu_0}(u) := \frac{\exp(-\Phi_N(u))}{Z_N}, \quad Z_N := \int_{\mathcal{U}} \exp(-\Phi_N(u'))\mathrm{d}\mu_0(u') \tag{4}$$

exists and is well-defined; and second, the approximate posterior μ_N provides an increasingly good approximation of the true posterior μ as the approximation parameter N increases. In this paper, we review results from [2] that guarantee well-definedness of μ_N and establish error bounds for μ_N in terms of error bounds for Φ_N.

In recent years, randomised numerical methods have been developed in order to overcome limitations of their deterministic counterparts. The field of probabilistic numerical methods [3] injects randomness into existing deterministic solvers for differential equations in order to model the uncertainty due to unresolved subgrid-scale dynamics. Random approximations of the forward model have been applied for forward uncertainty propagation in a range of applications; see e.g. [4, 5].

Randomisation has been shown to yield gains in computational efficiency. Results from [6, Section 5.7] showed a reduction by a factor of almost 25 in the CPU time needed for generating an independent sample with the Metropolis-Hastings algorithm, while in [7], a multilevel Markov Chain Monte Carlo method uses randomisation in the form of control variates for variance reduction. Stochastic programming ideas were used for more efficient posterior sampling in [8]. The results we describe provide theoretical support for the use of randomisation in Bayesian inference, and extend the pioneering results from [9], which concerned Gaussian process approximations of data misfits and forward models.

To motivate the use of random approximate misfits, consider the following example: Let X be any $\mathbb{R}^d$-valued random variable such that $\mathbb{E}[X] = 0$ and $\mathbb{E}[XX^\top]$ is the $d \times d$ identity matrix, and let $\{X_i\}_{i\in\mathbb{N}}$ be i.i.d. copies of X. Given (2),

$$\begin{aligned}\Phi(u) =& \frac{1}{2}\left(\Gamma^{-1/2}(y - G(u)\right)\mathbb{E}\left[XX^\top\right]\left(\Gamma^{-1/2}(y - G(u)\right)\\ =& \frac{1}{2}\mathbb{E}\left[\left|X^\top\left(\Gamma^{-1/2}(y - G(u))\right)\right|\right] \approx \frac{1}{2N}\sum_{j=1}^{N}\left|X_j^\top\left(\Gamma^{-1/2}(y - G(u))\right)\right| =: \Phi_N(u).\end{aligned}$$

In [10], the misfit Φ_N above was used to obtain computational cost savings when solving inverse problems associated to PDE boundary value problems. The results we present below can be specialised to the case of X with bounded support [2, Proposition 4.1]. For example, we can use the *ℓ-sparse distribution* for some $0 \leq \ell < 1$; for $\ell = 0$, this is the Rademacher distribution. Similar ideas have been applied for full waveform inversion in seismic tomography [11], for example.

2 Error Bounds for Approximate Posteriors

In what follows, we shall assume that the parameter space $\mathcal{U}$ admits a Borel σ-algebra, and we shall denote by $\mathcal{M}_1(\mathcal{U})$ the set of Borel probability measures on $\mathcal{U}$. Recall that the Hellinger metric $d_{\mathrm{H}} : \mathcal{M}_1(\mathcal{U}) \times \mathcal{M}_1(\mathcal{U}) \to [0, 1]$ is defined by

$$d_{\mathrm{H}}(\mu, \nu)^2 := \frac{1}{2}\int_{\mathcal{U}}\left|\sqrt{\frac{\mathrm{d}\mu}{\mathrm{d}\pi}(u')} - \sqrt{\frac{\mathrm{d}\nu}{\mathrm{d}\pi}(u')}\right|^2 \mathrm{d}\pi(u'),$$

where $\pi \in \mathcal{M}_1(\mathcal{U})$ is any measure such that μ and ν are both absolutely continuous with respect to π. It is known that d_{H} does not depend on the choice of π.

2.1 Error Bounds for Random Approximate Posteriors

We first present error bounds on random approximate posteriors μ_N associated to random misfits Φ_N, where $N \in \mathbb{N}$. That is, given a probability space $(\Omega, \mathcal{F}, \mathbb{P})$, we shall view a random misfit as a measurable function $\Phi_N : \Omega \times \mathcal{U} \to \mathbb{R}$. Furthermore, we shall assume that the randomness associated to the approximate misfit Φ_N is independent of the randomness associated to the unknown parameter u. In what follows, ν_N denotes a probability measure on Ω with the property that the distribution of the random function Φ_N is given by $\nu_N \otimes \mu_0$.

Given (3) and (4), a natural question is to establish an appropriate bound on the Hellinger distance between the true posterior μ and the approximate posterior μ_N in terms of some norm of the error between the true misfit Φ and the approximate misfit Φ_N. We emphasise that the approximate posterior μ_N in (4) is random in the sense that it depends on ω, since the approximate misfit Φ_N depends on ω. Therefore, the Hellinger distance $d_{\mathrm{H}}(\mu, \mu_N)$ will depend on ω as well. To describe such a bound, we shall take the expectation of the Hellinger distance with respect to ν_N, and let

$$\left\|\mathbb{E}_{\nu_N}\left[f\left(\Phi_N\right)\right]\right\|_{L^q_{\mu_0}(\mathcal{U})} := \left(\int_{\mathcal{U}}\left|\int_{\Omega} f\left(\Phi_N(\omega, u)\right) \mathrm{d}\nu_N(\omega)\right|^q \mathrm{d}\mu_0(u)\right)^{1/q}$$

for any Borel-measurable function $f : \mathbb{R} \to \mathbb{R}$ and $q \in [1, \infty)$. We define the quantity $\|\mathbb{E}_{\nu_N}[f(\Phi_N)]\|_{L^\infty_{\mu_0}(\mathcal{U})}$ analogously. With these preparations, we present the following theorem, which was given in [2, Theorem 3.2].

Theorem 1 (Error Bound for Random Approximate Posterior) *Let (q_1, q_1') and (q_2, q_2') be pairs of Hölder conjugate exponents, and let D_1, D_2 be positive scalars that depend only on q_1 and q_2. Suppose the following conditions hold:*

$$\left\|\mathbb{E}_{\nu_N}\left[\left(\exp\left(-\tfrac{1}{2}\Phi\right) + \exp\left(-\tfrac{1}{2}\Phi_N\right)\right)^{2q_1}\right]^{1/q_1}\right\|_{L^{q_2}_{\mu_0}(\mathcal{U})} \leq D_1 \tag{5}$$

$$\left\|\mathbb{E}_{\nu_N}\left[\left(Z_N \max\{Z^{-3}, Z_N^{-3}\}\left(\exp\left(-\Phi\right) + \exp\left(-\Phi_N\right)\right)^2\right)^{q_1}\right]^{1/q_1}\right\|_{L^{q_2}_{\mu_0}(\mathcal{U})} \leq D_2 \tag{6}$$

Then

$$\mathbb{E}_{\nu_N}\left[d_{\mathrm{H}}\left(\mu, \mu_N\right)^2\right]^{1/2} \leq (D_1 + D_2)\left\|\mathbb{E}_{\nu_N}\left[\left|\Phi - \Phi_N\right|^{2q_1'}\right]^{1/2q_1'}\right\|_{L^{2q_2'}_{\mu_0}(\mathcal{U})}.$$

Theorem 1 provides a bound on the mean square Hellinger distance between the true posterior μ and the random approximate posterior μ_N, in terms of an appropriate norm of the error $\Phi - \Phi_N$. The bound (5) implies that the negative tails of both Φ and Φ_N must decay exponentially quickly with respect to the $\nu_N \otimes \mu_0$-measure, and is satisfied, for example, when both Φ and Φ_N are bounded from below. Since $Z_N \max\{Z^{-3}, Z_N^{-3}\} = \max\{Z_N Z^{-3}, Z_N^{-2}\}$, it follows that the constraint imposed on the misfit Φ_N by (6) is that $\exp(-\Phi_N)$ should be neither too concentrated nor too broad. Together, conditions (5) and (6) ensure that the random approximate posterior μ_N exists, is well-defined, and satisfies the desired bound on the mean square Hellinger distance with respect to the true posterior μ.

An alternative way to generate an approximate posterior measure given a random approximate misfit is to compute a marginal approximate posterior μ_N^{M}, defined by

$$\frac{\mathrm{d}\mu_N^{\mathrm{M}}}{\mathrm{d}\mu_0}(u) := \frac{\mathbb{E}_{\nu_N}\left[\exp(-\phi_N(u))\right]}{\mathbb{E}_{\nu_N}\left[Z_N\right]}. \tag{7}$$

Note that, since we have taken expectations with respect to ν_N, the marginal approximate posterior does not depend on ω, and is in this sense deterministic. The following theorem was given in [2, Theorem 3.1].

Theorem 2 (Error Bound for Marginal Approximate Posterior) *Let* (p_1, p_1'), (p_2, p_2'), *and* (p_3, p_3') *be Hölder conjugate exponent pairs, and suppose there exist finite, positive scalars* C_1, C_2, *and* C_3 *that depend only on* p_1, p_2, *and* p_3, *such that the following conditions hold:*

$$\min\left\{\left\|\mathbb{E}_{\nu_N}\left[\exp\left(-\Phi_N\right)\right]^{-1}\right\|_{L^{p_1}_{\mu_0}(\mathcal{U})}, \|\exp(\Phi)\|_{L^{p_1}_{\mu_0}(\mathcal{U})}\right\} \leq C_1 \tag{8}$$

$$\left\|\mathbb{E}_{\nu_N}\left[(\exp(-\Phi) + \exp(-\Phi_N))^{p_2}\right]^{1/p_2}\right\|_{L^{2p_1'p_3}_{\mu_0}(\mathcal{U})} \leq C_2 \tag{9}$$

$$C_3^{-1} \leq \mathbb{E}_{\nu_N}\left[Z_N\right] \leq C_3. \tag{10}$$

Then there exists $C > 0$ *that does not depend on* N *such that*

$$d_{\mathrm{H}}(\mu, \mu_N^M) \leq C \left\|\mathbb{E}_{\nu_N}\left[|\Phi - \Phi_N|^{p_2'}\right]^{1/p_2'}\right\|_{L^{2p_1'p_3'}_{\mu_0}(\mathcal{U})}.$$

The bounds in (10) ensure that the denominator in (7) is strictly positive and finite. Thus, these bounds play a fundamental role in ensuring that the marginal approximate posterior exists and is well-defined. The bound in (9) reiterates the bound (5), modulo the $\frac{1}{2}$ factor, and thus serves a similar purpose as (5). The bound in (8) serves a similar purpose as (6). However, the minimum operator implies that it is not necessary for both Φ and Φ_N to be well-behaved.

The following result is a corollary of Theorems 1, 2, and [2, Lemma 3.5]. The main idea is to specify sufficient conditions for the hypotheses of both Theorem 1 and Theorem 2 to hold.

Corollary 1 (Joint Conditions for Error Bounds on Both Approximate Posteriors) *Suppose the following conditions are satisfied:*

(i) There exists $C_0 \in \mathbb{R}$ that does not depend on N such that $\Phi \geq -C_0$ on $\mathcal{U}$ and, for all $N \in \mathbb{N}$, $\nu_N(\Phi_N \geq -C_0) = 1$,

(ii) For any $0 < C_3 < \infty$ such that $C_3^{-1} < Z < C_3$, there exists $N^*(C_3) \in \mathbb{N}$ such that $N \geq N^*$ implies

$$\left\| \mathbb{E}_{\nu_N}[|\Phi - \Phi_N|] \right\|_{L^1_{\mu_0}(\mathcal{U})} \leq \frac{1}{2} \exp(-C_0) \min\left\{ Z - C_3^{-1}, C_3 - Z \right\},$$

and

(iii) there exists some $2 < \rho^* < +\infty$ such that $\|\mathbb{E}_{\nu_N}[\exp(\rho^*\Phi_N)]\|_{L^1_{\mu_0}(\mathcal{U})}$ is finite.

Then for each $N \geq N^*(C_3)$,

$$d_{\mathrm{H}}\left(\mu, \mu_N^M\right) \leq C \left\| \mathbb{E}_{\nu_N}\left[|\Phi - \Phi_N|\right] \right\|_{L^{2\rho^*/(\rho^*-1)}_{\mu_0}(\mathcal{U})} \tag{11}$$

and

$$\mathbb{E}_{\nu_N}\left[d_{\mathrm{H}}(\mu, \mu_N)^2 \right]^{1/2} \leq D \left\| \mathbb{E}_{\nu_N}\left[|\Phi - \Phi_N|^{2\rho^*/(\rho^*-2)} \right]^{(\rho^*-2)/(2\rho^*)} \right\|_{L^1_{\mu_0}(\mathcal{U})}, \tag{12}$$

where $C, D > 0$ depend on $\|\mathbb{E}_{\nu_N}[\exp(\rho^*\Phi_N)]\|^{1/\rho^*}_{L^1_{\mu_0}(\mathcal{U})}$. If in addition to conditions (i)–(iii) it holds that

$$\sup_{N \geq N^*(C_3)} \left\| \mathbb{E}_{\nu_N}\left[\exp(\rho^*\Phi_N)\right] \right\|_{L^1_{\mu_0}(\mathcal{U})} < \infty,$$

then the constants C and D in (11) and (12) do not depend on N.

Condition (i) amounts to a common uniform lower bound on all the misfits, both the true misfit and the collection of random approximate misfits, and thus plays a role in ensuring that (5) and (9) are satisfied. Condition (ii) makes precise the assumption that Φ_N approximates Φ in the $L^1_{\nu_N \otimes \mu_0}$ topology, which is a necessary condition for ensuring that the right-hand sides of the conclusions of Theorems 1 and 2 are finite. Condition (iii) describes an exponential integrability condition on the random approximate misfits and ensures that (6) and (8) are satisfied. Thus the additional condition amounts to a uniform exponential integrability condition over all sufficiently large values of N.

Remark 1 Neither Theorem 1 nor Theorem 2 require boundedness from below of either Φ or the Φ_N. However, the negative tails of both Φ and Φ_N must decay exponentially quickly at a sufficiently high rate, as specified by (9) and (5) respectively.

2.2 Error Bounds for Random Forward Models

Next, we consider approximate posterior measures that arise as a result of approximating the forward model G in (1). For simplicity, we shall consider only the case when the data misfit Φ and forward model G are related via the quadratic potential (2). In particular, this means that if $G_N : \mathcal{U} \to \mathcal{Y}$ is an approximation of the true forward model G, then the resulting approximate data misfit is given by

$$\Phi_N(u) := \frac{1}{2} \|\Gamma^{-1} \left(y - G_N(u)\right) \|^2.$$

The following theorem is a nonasymptotic reformulation of [2, Theorem 3.9 (b)].

Theorem 3 (Error Bounds for Approximate Posteriors) *Suppose there exists* $2 < \rho^* < \infty$ *such that* $\sup_N \mathbb{E}_{\nu_N}[\exp(\rho^* \Phi_N)] \in L^1_{\mu_0}(\mathcal{U})$ *is finite. If there exists an* $N^* \in \mathbb{N}$ *such that, for all* $N \geq N^*$,

$$\left\| \mathbb{E}_{\nu_N} \left[\|G - G_N\|^{4\rho^*/(\rho^*-2)} \right]^{(\rho^*-2)/(2\rho^*)} \right\|_{L^{2\rho^*/(\rho^*-1)}_{\mu_0}(\mathcal{U})} \leq 1,$$

then

$$d_{\mathrm{H}}\left(\mu, \mu_N^M\right) \leq C \left\| \mathbb{E}_{\nu_N} \left[\|G_N - G\|^2 \right] \right\|^{1/2}_{L^{2\rho^*/(\rho^*-1)}_{\mu_0}(\mathcal{U})}$$

and

$$\mathbb{E}_{\nu_N} \left[d_{\mathrm{H}}(\mu, \mu_N))^2 \right]^{1/2} \leq D \left\| \mathbb{E}_{\nu_N} \left[\|G_N - G\|^{4\rho^*/(\rho^*-2)} \right]^{(\rho^*-2)/(2\rho^*)} \right\|^{1/2}_{L^2_{\mu_0}(\mathcal{U})}$$

for $C, D > 0$ *that do not depend on* N.

The theorem can be rewritten so that, instead of imposing a uniform exponential integrability condition on the approximate quadratic potentials Φ_N, one instead imposes an exponential integrability condition on the true data misfit Φ; see [2, Theorem 3.9 (a)]. An additional hypothesis in this case is that the expectations of the approximate data misfit functions are ν_N-almost surely bounded, in the sense that $\nu_N(\Phi_N \mid \mathbb{E}_{\mu_0}[\Phi_N] \leq C_4) = 1$ for some $C_4 \in \mathbb{R}$ that does not depend on N.

3 Conclusions and Directions for Future Work

This paper has reviewed the main error bounds of [2] concerning deterministic and random approximate posteriors that arise when performing Bayesian inference with random approximate data misfits or random forward models. The error bounds on the approximate posterior measures are given with respect to the Hellinger metric on the space of Borel probability measures $\mathcal{M}_1(\mathcal{U})$. Given a fixed prior measure μ_0, these error bounds describe—with specific exponents of integrability and problem-dependent constants—the local or global Lipschitz continuity of the map that takes a data misfit as input and produces the corresponding posterior measure as output. Aside from the regularity assumptions made on the random approximations, the error bounds shown above make no structural assumptions on the approximations used. For example, we do not assume that the random approximations involve Gaussian random variables, or random variables with bounded support.

Recent work has highlighted the importance of considering other metrics on $\mathcal{M}_1(\mathcal{U})$, and also of proving well-posedness of the solution of a Bayesian inverse problem by establishing continuous (instead of Lipschitz continuous) dependence on either the data, prior, or data misfit. The well-posedness of Bayesian inverse problems in the sense of continuous dependence with respect to the data of the posterior for given prior and data misfit was established in [12]. Local Lipschitz continuity with respect to *deterministic* perturbations in the prior or data misfit was shown in [13]. In both [12, 13], continuity is with respect to the topologies induced by the total variation metric, by Wasserstein p-metrics, or by the Kullback-Leibler divergence.

A key assumption made in [13] when establishing local Lipschitz continuity for a fixed prior μ_0 with respect to perturbations in the data misfit is that the deterministic perturbed data misfit is μ_0-almost surely bounded from below. As highlighted in Remark 1, the analysis of [2] does not require that either the true data misfit or the random approximate log-likelihood are μ_0-almost surely bounded from below. For future work, we aim to establish similar continuity results with respect to different metrics, as demonstrated in [12, 13], but at the same level of generality of [2].

Acknowledgments The research of HCL has been partially funded by Deutsche Forschungsgemeinschaft (DFG)—SFB1294/1—318763901 and by Universität Potsdam. The work of TJS has been partially supported by the Freie Universität Berlin within the Excellence Initiative of the German Research Foundation (DFG). The authors thank an anonymous referee for their feedback.

References

1. Dashti, M., Stuart, A. M.: The Bayesian Approach to Inverse Problems. In: Ghanem, R., Higdon, D., Owhadi, H. (eds) Handbook of Uncertainty Quantification. Springer, Cham. https://doi.org/10.1007/978-3-319-12385-1_7
2. Lie, H. C., Sullivan, T. J., Teckentrup, A. L.: Random Forward Models and Log-Likelihoods in Bayesian Inverse Problems. SIAM/ASA J. Uncertain. Quantif. (2018). https://doi.org/10.1137/18M1166523

3. Cockayne, J., Oates, C., Sullivan, T. J., Girolami, M.: Bayesian probabilistic numerical methods. SIAM Rev. (2019). https://doi.org/10.1137/17M1139357
4. Marzouk, Y., Xiu, D.: A stochastic collocation approach to Bayesian inference in inverse problems. Commun. Comput. Phys. (2009). https://doi.org/10.4208/cicp.2009.v6.p826
5. Birolleau, A., Poëtte, G., Lucor, D.: Adaptive Bayesian inference for discontinuous inverse problems, application to hyperbolic conservation laws. Commun. Comput. Phys. (2014). https://doi.org/10.4208/cicp.240113.071113a
6. Christen, J. A., Fox, C.: Markov chain Monte Carlo using an approximation. J. Comput. Graph. Statist. (2005). https://doi.org/10.1198/106186005X76983
7. Dodwell, T. J., Ketelsen, C., Scheichl, R., Teckentrup, A. L.: Multilevel Markov Chain Monte Carlo. SIAM Rev. (2019). https://doi.org/10.1137/19M126966X
8. Wang, K., Bui-Thanh, T., Ghattas, O.: A randomised maximum a posteriori method for posterior sampling of high dimensional nonlinear Bayesian inverse problems. SIAM J. Sci. Comput. (2018). https://doi.org/10.1137/16M1060625
9. Stuart, A. M., Teckentrup, A. L.: Posterior consistency for Gaussian process approximations of Bayesian posterior distributions. Math. Comput. (2010). https://doi.org/10.1090/mcom/3244
10. Le, E. B., Myers, A., Bui-Thanh, T., Nguyen, Q. P.: A data-scalable randomized misfit approach for solving large-scale PDE-constrained inverse problems. Inverse Probl. (2017). https://doi.org/10.1088/1361-6420/aa6cbd
11. Aravkind, A., Friedlander, M. P., Herrmann, F. J., van Leeuwen, T.: Robust inversion, dimensionality reduction, and randomized sampling. Math. Program., Ser. B (2012). https://doi.org/10.1007/s10107-012-0571-6
12. Latz, J.: On the well-posedness of Bayesian inverse problems. SIAM/ASA J. Uncertain. Quantif. (2020). https://doi.org/10.1137/19M1247176
13. Sprungk, B.: On the local Lipschitz robustness of Bayesian inverse problems. Inverse Probl. (2020). https://doi.org/10.1088/1361-6420/ab6f43

High-Order Two and Three Level Schemes for Solving Fractional Powers of Elliptic Operators

Raimondas Čiegis and Petr Vabishchevich

Abstract In this paper we develop and investigate numerical algorithms for solving the fractional powers of discrete elliptic operators $\mathcal{A}_h^\alpha U = F$, $0 < \alpha < 1$, for $F \in V_h$ with V_h a finite element or finite difference approximation space. Our goal is to construct efficient time stepping schemes for the implementation of the method based on the solution of a pseudo-parabolic problem. The second and fourth order approximations are constructed by using two- and three-level schemes. In order to increase the accuracy of approximations the geometric graded time grid is constructed which compensates the singular behavior of the solution for t close to 0. This apriori adaptive grid is compared with aposteriori adaptive grids. Results of numerical experiments are presented, they agree well with the theoretical results.

1 Introduction

There are different definitions of fractional power of elliptic operators [1]. We consider the definition based on the spectral decomposition of an elliptic operator. Let us define $H = H_0^1(\Omega)$, where $\Omega \subset \mathbb{R}^d$. On $H \times H$ we consider the weak formulation of the elliptic problem: find $u \in H$ such that

$$A(u, v) := \int_\Omega \big(k(x)\nabla u \cdot \nabla v + q(x)uv\big)dx = \int_\Omega f(x)v(x)dx, \quad \forall v \in H. \tag{1}$$

We define the elliptic operator $\mathcal{A}$, where $\mathcal{A}$ is an isomorphism $H_0^1(\Omega) \to H^{-1}(\Omega)$ given by $u \to a(u, \cdot)$.

R. Čiegis (✉)
Vilnius Gediminas Technical University, Vilnius, Lithuania
e-mail: rc@vgtu.lt

P. Vabishchevich
Nuclear Safety Institute, Russian Academy of Sciences, Moscow, Russia

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_27

Operator $\mathcal{A}$ is symmetric and positive definite on $L^2(\Omega)$. Let us denote the eigenpairs of this operator ψ_j, λ_j. Due to the properties of operator $\mathcal{A}$ its eigenvectors ψ_j provide an orthonormal basis for $L^2(\Omega)$.

Then for functions $u \in L^2(\Omega)$ such that $\sum_{j=1}^{\infty} \lambda_j^{2\alpha} |(u, \psi_j)|^2 < \infty$ the spectral fractional powers $\mathcal{A}^\alpha$ for $0 < \alpha < 1$ are defined by eigenvector expansions:

$$\mathcal{A}^\alpha u := \sum_{j=1}^{\infty} \lambda_j^\alpha (u, \psi_j) \psi_j, \tag{2}$$

where (u, v) denotes the standard scalar product $(u, v) = \int_\Omega u(x) v(x) dx$.

The Dirichlet problem for the fractional elliptic operator is defined as follows: given a function f and $\alpha \in (0, 1)$, we seek $u \in H_0^1$ such that

$$\mathcal{A}^\alpha u = f. \tag{3}$$

For functions f such that $\sum_{j=1}^{\infty} \lambda_j^{-2\alpha} |(f, \psi_j)|^2 < \infty$ negative fractional powers $\mathcal{A}^{-\alpha}$ for $0 < \alpha < 1$ can be defined by eigenvector expansions:

$$\mathcal{A}^{-\alpha} f := \sum_{j=1}^{\infty} \lambda_j^{-\alpha} (f, \psi_j) \psi_j. \tag{4}$$

For such problems the state of the art numerical methods are based on the following quite general approach. The given non-local differential problem is transformed to some local differential problem of elliptic or parabolic type, but this new problem is formulated in the extended $d + 1$ dimension space $\mathbb{R}^{d+1}$. There are a few interesting implementations of this general idea, see [1, 4, 6–8].

Our main goal is to construct and study numerical algorithms for the transformation of the non-local problem (2) to a pseudo-parabolic problem [5, 9]. The unique solution $u = \mathcal{A}^{-\alpha} f$ of the fractional power problem can be represented as a mapping

$$v(t) = \big(\delta \mathcal{I} + t\mathcal{B}\big)^{-\alpha} f, \tag{5}$$

where $\mathcal{B} = \mathcal{A} - \delta \mathcal{I}$. Then $u = v(1)$.

The next step is to find a nonstationary PDE for which $v(t)$ is the exact solution. This approach can lead to different PDEs, one such equation was proposed in the original paper [9]. It is shown that $v(t)$ satisfies the pseudo-parabolic problem

$$\begin{aligned} &\big(\delta \mathcal{I} + t\mathcal{B}\big) \frac{\partial v}{\partial t} + \alpha \mathcal{B} v = 0, \quad 0 < t \le 1, \\ &v(0) = \delta^{-\alpha} f. \end{aligned} \tag{6}$$

Then different time stepping approximations can be used to solve the obtained nonstationary PDE problem (6).

The rest of this paper is organized as follows. In Sect. 2, the symmetrical Euler method is applied to solve the obtained pseudo-parabolic problem. It is shown that this difference scheme is unconditionally stable. It is interesting to note that this scheme is equivalent to the time-stepping algorithm based on the first order diagonal Padé approximation for function $(1 + x)^{-\alpha}$ (see [2, 5]). Results of numerical experiments are provided to show that for nonregular solutions and uniform time grids the symmetrical Euler scheme regains the second order convergence rate only for sufficiently small time step sizes when the high modes are resolved correctly.

In order to increase the accuracy of approximations the geometric graded time grid is constructed in Sect. 3. It compensates the singular behavior of the solution for t close to 0. This apriori adaptive grid is compared with the aposteriori adaptive grid, which is constructed by using the Runge rule. In Sect. 4 a family of three-level finite difference schemes is constructed to solve the given pseudo-parabolic problem. A general nonuniform time mesh is used and the stability of the discrete problem is proved. It is noted that for a uniform time mesh a special value of the weight parameter exists which leads to the fourth order scheme. In Sect. 5 a high-order two-level finite difference scheme is developed and investigated. It is based on the method of modified equations. Results of numerical experiments are presented. Some final conclusions and remarks are done in Sect. 6.

2 Symmetrical Euler Method

We approximate the solution u of (2) by using the finite element approximation U on $V_h \subset V$, where h is the discretization parameter. Then we get the discrete operators $\mathcal{A}_h$, $\mathcal{B}_h = \mathcal{A}_h - \delta \mathcal{I}_h > 0$. Let $V^n \in V_h$ be the approximation of $v(t_n)$ on V_h. The pseudo-parabolic problem (6) is approximated by the symmetrical Euler method

$$\begin{aligned}&\big(\delta \mathcal{I}_h + t_{n-\frac{1}{2}} \mathcal{B}_h\big) \frac{V^n - V^{n-1}}{\tau_n} + \alpha \mathcal{B}_h V^{n-\frac{1}{2}} = 0, \quad n = 1, \ldots, N, \qquad (7)\\ &V^0 = \delta^{-\alpha} F_h,\end{aligned}$$

where $t_{n-\frac{1}{2}} = 0.5(t_n + t_{n-1})$ and $V^{n-\frac{1}{2}} = 0.5(V^n + V^{n-1})$.

The stability of this scheme is investigated in [2]. It is proved that (7) is unconditionally stable.

For smooth solutions this scheme approximates the differential problem with the second order. Still, it is well-known that the solution $u = \mathcal{A}^{-\alpha} f$ of problem (2)

Table 1 The error E_N of the discrete solution (7) and the experimental convergence order O_N for varying $\alpha = 0.1, 0.5$. The uniform space grid is used with $J = 100$

	$N = 10$	$N = 20$	$N = 40$	$N = 80$
$\alpha = 0.1$, E_N	0.11041	0.0818953	0.0577244	0.0380742
O_N		0.431	0.505	0.600
$\alpha = 0.5$, E_N	0.025209	0.0154431	0.00905345	0.0050120
O_N		0.707	0.770	0.853
	$N = 5000$	$N = 10000$	$N = 20000$	$N = 40000$
$\alpha = 0.1$, E_N	2.2866e−04	6.0332e−05	1.5332e−05	3.8499e−06
O_N		1.922	1.976	1.994
$\alpha = 0.5$, E_N	1.6317e−05	4.2534e−06	1.0769e−06	2.70125e−07
O_N		1.940	1.982	1.995

exhibit less regularity. It is proved in [5] that the error of the discrete solution due to time stepping algorithm (7) can be estimated as

$$\|\mathcal{A}_h{}^{-\alpha} F - V^N\| \le C\tau_N^{\alpha+\gamma} \|\mathcal{A}_h{}^{\gamma} F\|, \quad \alpha + \gamma \le 2. \tag{8}$$

The given estimate is valid uniformly for a broad set of time step sizes τ_N. Still, the asymptotic convergence order should be obtained for time step sizes resolving the high modes of the solution, i.e. when $\tau_N \lambda_{max} \le C \approx 1$.

In order to illustrate these estimates we present results of numerical experiments for the one dimensional problem $\Omega = (0, 1)$:

$$\mathcal{A}_h U = -\frac{U_{j+1} - 2U_j + U_{j-1}}{h^2}, \quad j = 1, \dots, J-1, \quad U_0 = U_J = 0 \tag{9}$$

with the function $f = 1$, $x \in \Omega$ (the case (d) in [5]). We report the error in the maximum norm $E_N = \|\mathcal{A}_h{}^{-\alpha} F - V^N\|_\infty$ and the experimental convergence order going from $N = m$ to $N = 2m$ grid points $O_{2m} = \log(E_m/E_{2m})/\log(2)$.

Table 1 gives results for $J = 100$ and varying $\alpha = 0.1, 0.5$.

3 Non-uniform and Adaptive Time Meshes

The error of the symmetrical Euler scheme (7) depends on the accuracy with which we approximate the transfer operator of the scheme, i.e. on the smallness of the factor $\tau_n \widetilde{\lambda}_{\max}/\big(\delta + t_{n-1}\widetilde{\lambda}_{\max}\big)$, where $\widetilde{\lambda}_j$ are eigenvalues of $\mathcal{B}_h$. We will construct a non-uniform mesh by using the regularization property of $t_{n-1}\widetilde{\lambda}$. Then the step sizes of the refined time mesh are defined from the equation

$$\tau_n \widetilde{\lambda}_{\max}/\big(\delta + t_{n-1}\widetilde{\lambda}_{\max}\big) = q \le 1. \tag{10}$$

Table 2 The error E_N of the discrete solution (7) and the experimental convergence order O_N for varying $\alpha = 0.1, 0.5$. The graded geometric time mesh and uniform space grid with $J = 100$ are used

	$N = 20$	$N = 40$	$N = 80$	$N = 160$
$\alpha = 0.1$, E_N	4.5694e−03	1.1159e−03	2.9076e−04	7.2757e−05
O_N		2.033	1.940	1.999
$\alpha = 0.5$, E_N	8.9964e−04	2.2572e−04	5.6479e−05	1.4123e−05
O_N		1.995	1.999	1.9997

Simple computations show that $\tau_n = \tau_1(1+q)^{n-1}, n \geq 2$, i.e. we construct a geometric graded mesh [2]. The number of discrete points is defined by $N = \log(\lambda_{\max})/\log(1+q) + 1$.

Next we give an alternative possibility to introduce a geometric graded mesh. The uniform time mesh for $s_n = n\tilde{\tau},\ n = 0, \dots, N$ is mapped to the non-uniform mesh for t_n by using a nonlinear function $t_n = \chi(s_n)$, where

$$\chi(s) = \frac{e^{\gamma s} - 1}{e^{\gamma} - 1}, \quad 0 \leq s \leq 1 \tag{11}$$

with some parameter $\gamma > 0$. It follows from (11) that sizes of adjacent time steps for τ_n and τ_{n-1} satisfy the relation

$$\frac{\chi(s_n) - \chi(s_{n-1})}{\chi(s_{n-1}) - \chi(s_{n-2})} = e^{\gamma \tilde{\tau}}.$$

By taking $\gamma = \log(\lambda_{\max})$ we again get the geometric graded mesh.

The results of computational experiments are presented in Table 2: one dimensional test problem (9) is solved by using the symmetrical Euler scheme (7) for $J = 100$ and varying $\alpha = 0.1, 0.5$. The second order convergence rate is clearly seen from experiments.

3.1 Adaptive Mesh

In this paragraph we apply a simple time step-size control method. For a given approximation V^{n-1} at t_{n-1} we apply the symmetrical Euler scheme (7) and compute the discrete solution $\widetilde{V}^n$ at $t_n = t_{n-1} + \tau_n$. Then we repeat the process with two times smaller step-size $\frac{1}{2}\tau_n$, apply the scheme (7) twice and compute one more approximation $\widehat{V}^n$. If the estimate $\|\widetilde{V}^n - \hat{V}^n\| < tol$ is valid, where tol denotes the required tolerance for the local error, then the current computational step is accepted $V^n = \widehat{V}^n$. Additionally we check if $\|\widetilde{V}^n - \widehat{V}^n\| < tol/2.5$, then the step-size of the next step is increased $\tau_{n+1} = 1.25\tau_n$, otherwise $\tau_{n+1} = \tau_n$. If the local error

Table 3 The error E_N of the discrete solution (7) for adaptive grids and $\alpha = 0.1$. The uniform space grid is used with $J = 100$

	$N = 35\,(126)$	$N = 57\,(189)$	$N = 70\,(235)$	$N = 113\,(360)$	$N = 139\,(441)$
E_N	2.579e−04	9.507−05	6.127−05	2.366−05	1.536−05

is larger than tol, then computations from mesh point t_{n-1} are repeated with the smaller time step-size $\tau_n = \frac{1}{2}\tau_n$.

Some results of computational experiments for $\alpha = 0.1$ are presented in Table 3. The number of successful full time steps is denoted by N. We also present the total number of time steps, required to solve the given problem (approx. three times larger than N). The structure of the adaptive mesh is very similar to a piecewise constant geometric graded mesh proposed in [5]. It follows from the presented results that geometric graded mesh is a simple but very efficient tool to increase the accuracy of time integration algorithms.

4 Three Level Scheme

In order to resolve the singularity of the solution we use the same mapping (11) as in previous section $t = \chi(s)$. Then instead of solving the Cauchy problem for the pseudo-parabolic equation (6) we get the Cauchy problem

$$\begin{aligned} &\big(\delta \mathcal{I}_h + \chi(s)\mathcal{B}_h\big)\frac{dv}{ds} + \alpha \frac{d\chi}{ds}\mathcal{B}_h v = 0, \quad 0 < s \le 1, \\ &v(0) = \delta^{-\alpha}\varphi. \end{aligned} \tag{12}$$

For solving problem (12) we use the symmetrical three level finite difference scheme ($\tau = 1/N$):

$$\begin{aligned} &\Big(\frac{d\chi}{ds}(s_n)\Big)^{-1}\big(\delta \mathcal{I}_h + \chi(s_n)\mathcal{B}_h\big)\frac{w_{n+1} - w_{n-1}}{2\tau} + \alpha \mathcal{B}_h \\ &\qquad \times(\sigma w_{n+1} + (1-2\sigma)w_n + \sigma w_{n-1}) = 0, \quad n = 1, 2, \ldots, N-1, \\ &w_0 = \delta^{-\alpha}\varphi, \quad w_1 = \overline{w}_1. \end{aligned} \tag{13}$$

We note that the initial condition $\overline{w}_1$ should be computed by applying some two level numerical algorithm and the accuracy of this approximation should be the same as of the main scheme (13).

For sufficiently smooth solutions of (12), the scheme (13) approximates the differential problem with the second order. It is interesting to note, that for a uniform time mesh, when $\chi(s) = s$ and taking $\sigma_0 = (2+\alpha)/(6\alpha)$ we get the discrete scheme of the fourth approximation order with respect to time t. By using the energy method and applying the analysis presented in [3] the following theorem is proved.

Table 4 The error E_N of the discrete solution (13) and the experimental convergence order O_N for varying $\alpha = 0.1, 0.5$

	$N = 20$	$N = 40$	$N = 80$	$N = 160$
$\alpha = 0.1$, E_N	9.2238e−04	2.2887e−04	5.7105e−05	1.4269e−05
O_N		2.011	2.003	2.001
$\alpha = 0.5$, E_N	2.2578e−03	5.5786e−04	1.3904e−04	3.4733e−05
O_N		2.016	2.004	2.001

Theorem 1 *For $\sigma > 0.25$ the three-level scheme (13) is unconditionally stable with respect to the initial data.*

The results of computational experiments are presented in Table 4. The second initial condition is computed applying the spectral method. The second order convergence rate is clearly seen from experiments.

5 High-Order Schemes

In this section, starting from the symmetrical Euler scheme (7) we construct a high-order scheme. By using the Taylor expansion of the scheme residual with respect to $t_{n-\frac{1}{2}}$ and applying the modified equations technique we construct a high-order two-level finite difference scheme

$$\begin{gathered}\Big(\mathcal{D}_h(t_{n-\frac{1}{2}}) - \frac{\tau_n^2}{12}(1-\alpha^2)\mathcal{B}_h\mathcal{D}_h{}^{-1}(t_{n-\frac{1}{2}})\mathcal{B}_h\Big)\frac{v^n - v^{n-1}}{\tau_n} \\ +\alpha\mathcal{B}_h\frac{v^n + v^{n-1}}{2} = 0, \quad n = 1, \ldots, N, \quad v^0 = \delta^{-\alpha}\varphi,\end{gathered} \tag{14}$$

where $\mathcal{D}_h(t) = \delta\mathcal{I}_h + t\mathcal{B}_h$. This scheme approximates the differential equation with the fourth order.

Theorem 2 *The high-order finite difference scheme (14) is unconditionally stable with respect to the initial data.*

The proof of this theorem follows from the spectral analysis of the self-adjoint transfer operator (for full details, see [2]). The results of computational experiments are presented in Table 5. The fourth order convergence rate is clearly seen from experiments.

Table 5 The error E_N of the discrete solution (13) and the experimental convergence order O_N for varying $\alpha = 0.1, 0.5$

	$N = 10$	$N = 20$	$N = 40$	$N = 80$
$\alpha = 0.1$, E_N	1.0811e−03	7.4307e−05	4.7576e−06	2.9916e−07
O_N		3.863	3.965	3.991
$\alpha = 0.5$, E_N	1.7533e−04	1.1891e−05	7.5907e−07	4.7696e−08
O_N		3.882	3.969	3.992

6 Conclusions

Two main directions are identified to construct efficient and high order discrete algorithms for solving the pseudo-parabolic version of the fractional power elliptic problems. The first one is based on the special geometric graded meshes. For three-level discrete scheme such a non-uniform mesh is introduced by using a special mapping of time coordinate. In the second approach the method of modified equations used to construct high-order finite difference scheme. Results of computational experiments have shown that a combination of high-order two level scheme with geometric graded mesh is the most efficient algorithm for solving the given pseudo-parabolic problem.

References

1. Bonito, A, Borthagaray, J.P., Nochetto, R.H., Otarola, E., Salgado, A. J.: Numerical methods for fractional diffusion. Computing and Visualization in Science. **19** (5–6), 19–46, (2018)
2. Čiegis,R., Vabishchevich, P.N.: Two-level schemes of Cauchy problem method for solving fractional powers of elliptic operators. Computers and Mathematics with Applications. (2019) https://doi.org/10.1016/j.camwa.2019.08.012
3. Čiegis,R., Vabishchevich, P.N.: High order numerical schemes for solving fractional powers of elliptic operators. Journal of Computational and Applied Mathematics. (2019) (accepted)
4. Cusimano, N., Del Teso, F., Gerardo-Giorda, L.: Numerical approximations for fractional elliptic equations via the method of semigroups, arXiv preprint arXiv:1812.01518.
5. Duan, B., Lazarov, R., Pasciak, J.: Numerical approximation of fractional powers of elliptic operators, arXiv preprint arXiv:1803.10055.
6. Hofreither, C.: A unified view of some numerical methods for fractional diffusion, RICAM-Report 2019-12.
7. Meidner, D., Pfefferer, J., Schürholz, K., Vexler, B.: hp-finite elements for fractional diffusion, SIAM Journal on Numerical Analysis. **56**(4) 2345–2374 (2018)
8. Nochetto, R. H., Otárola, E., Salgado, A. J.: A PDE approach to fractional diffusion in general domains: a priori error analysis. Foundations of Computational Mathematics. **15**(3), 733–791, (2015)
9. Vabishchevich, P. N.: Numerically solving an equation for fractional powers of elliptic operators. Journal of Computational Physics. **282**(1), 289–302, (2015)

Numerical Investigation of the Boussinesq Equations Through a Subgrid Artificial Viscosity Method

Medine Demir and Songül Kaya

Abstract This study presents a subgrid artificial viscosity method for approximating solutions to the Boussinesq equations. The stability is obtained by adding a term via an artificial viscosity and then removing it only on the coarse mesh scale. The method includes both vorticity in the viscous term and a grad-div stabilization. We analyze the method from both analytical and computational point of view and show that it is unconditionally stable and optimally convergent. Several numerical experiments are provided that support the derived theoretical results and demonstrate the efficiency and accuracy of the method.

1 Introduction

Natural convection is induced by the buoyancy force arising from the density differences due to temperature gradients along with the gravitational impacts. Because of the density differences, a full analysis of such flow problems becomes quite complex. Therefore, fluid flow and heat transfer are generally governed by the partial differential equation system of mass, momentum and energy conservation along with Boussinesq approximation which states that the density differences can be neglected, except in the buoyancy term, [1]. The governing equations for natural convection under Boussinesq approximation can be written as

$$\begin{aligned} u_t + (u\cdot\nabla)u - \nu\Delta u + \nabla p &= Ri\langle 0, T\rangle + f && \text{in } \Omega,\\ \nabla\cdot u &= 0 && \text{in } \Omega,\\ T_t + (u\cdot\nabla)T - \kappa\Delta T &= \gamma && \text{in } \Omega,\\ u(0,x) = u_0 \text{ and } T(0,x) &= T_0 && \text{in } \Omega,\\ u = 0 \text{ and } T &= 0 && \text{on } \partial\Omega \end{aligned} \tag{1}$$

M. Demir (✉) · S. Kaya
Department of Mathematics, Middle East Technical University, Ankara, Turkey
e-mail: dmedine@metu.edu.tr; smerdan@metu.edu.tr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_28

where u is the velocity, T is the temperature, $\nu = O(Re^{-1})$ is the kinematic viscosity, $\kappa = 1/PrRe$ is the thermal diffusivity parameter, $Pr = \nu/\kappa$ is Prandtl number and $Ri = PrRaRe^2$ is Richardson number.

Since Galerkin finite element discretization of Boussinesq system is itself unstable in the case of high Reynolds number, introducing a turbulence model becomes necessary. In this study, we propose, analyze and test an accurate regularization of subgrid artificial viscosity method for the Boussinesq system. We consider the extension of an earlier study of [2] for the Navier-Stokes equation based on the pioneering work of [3]. The underlying idea of this method is based on the variational multiscale method of [3] and stabilization via an artificial viscosity. In this method, the stability is achieved by adding an artificial viscosity and then removing it only on the coarse mesh scale. The stability process is applied to the viscous term by using the vector identity $\Delta u = -\nabla \times (\nabla \times u) + \nabla(\nabla \cdot u)$ and thus results in a two level method including both vorticity in the viscous term and grad-div stabilization. One can find many studies using similar methods to the discussed method [4]. However, our method is more efficient for some reasons. Using a mixed method for both velocity and vorticity significantly reduces extra storage in $3d$ compared to velocity and its gradient. Furthermore, the method improves the conditioning of the system, that is, instead of the full velocity gradient with nine variables it leads to coarse grid storage of vorticity with just three variables. Moreover, one can obtain more accurate numerical solutions in the presence of high Reynolds number without choosing a computationally inefficient time-step. Hence, it is important to extend this methodology to flows governed by the Boussinesq system. We aimed to obtain a much better quality solution with less computational effort.

2 Subgrid Artificial Viscosity Scheme

In this section, we present a fully discrete numerical algorithm of the proposed method. For this purpose, we choose the natural function spaces $X := H_0^1(\Omega)^d$, $W := H_0^1(\Omega)$ and $Q := L_0^2(\Omega)$ for the continuous velocity, temperature and pressure spaces, respectively. Let $X_h \subset X$, $W_h \subset W$, $Q_h \subset Q$ be conforming finite element spaces where the velocity, temperature and pressure spaces fulfill the discrete inf-sup condition. We use the usual $L^2(\Omega)$ norm and the inner product denoted by $\|\cdot\|$ and $(\cdot,\cdot)$, respectively. Define the skew-symmetric forms of the convective terms by

$$b^*(u, v, w) = \frac{1}{2}(u \cdot \nabla v, w) - \frac{1}{2}(u \cdot \nabla w, v), \tag{2}$$

$$c^*(u, T, \chi) = \frac{1}{2}(u \cdot \nabla T, \chi) - \frac{1}{2}(u \cdot \nabla \chi, T), \tag{3}$$

We also define $L_H \subset L^2(\Omega)^d$ to be a large scale space defined on a regular coarse mesh π^H which is a conforming triangulation of Ω. For the numerical analysis, we need to define the L^2 projection $P_{L_H} : (L^2(\Omega))^{d\times d} \longrightarrow L_H$ by

$$(P_{L_H}\phi - \phi, l_H) = 0 \qquad \forall l_H \in L_H. \tag{4}$$

We divide the time interval $[0, T]$ into N equal sub-interval with the time-step $\Delta t = T/N$ and $t_{n+1} = (n+1)\Delta t$ with $n = 0, 1, 2, \ldots, N$. Then, the subgrid artificial viscosity method based on backward Euler time stepping scheme reads as follows.

Algorithm Let D_H be the new coarse mesh variable and the initial conditions u^0, T^0, the forcing function f and the heat source γ be given. Define u_h^0 and T_h^0 as the nodal interpolants of u^0 and T^0, respectively. Then, given u_h^n, T_h^n, p_h^n, find $(u_h^{n+1}, T_h^{n+1}, p_h^{n+1}) \in (X_h, W_h, Q_h)$ satisfying $\forall (v_h, S_h, q_h, l_H) \in (X_h, W_h, Q_h, L_H)$

$$\begin{aligned}
(\frac{u_h^{n+1} - u_h^n}{\Delta t}, v_h) + \nu(\nabla u_h^{n+1}, \nabla v_h) + b^*(u_h^n, u_h^{n+1}, v_h) - (p_h^{n+1}, \nabla \cdot v_h)& \\
+\alpha_1(\nabla \times u_h^{n+1}, \nabla \times v_h) - \alpha_1(D_H^{n+1}, \nabla \times v_h) + \alpha_2(\nabla \cdot u_h^{n+1}, \nabla \cdot v_h)& \\
= Ri(\langle 0, T_h^n\rangle, v_h) + (f^{n+1}, v_h),& \qquad (5)\\
(\nabla \cdot u_h^{n+1}, q_h) = 0,& \qquad (6)\\
(D_H^{n+1} - \nabla \times u_h^n, l_H) = 0,& \qquad (7)\\
(T_h^{n+1}, S_h) + \kappa(\nabla T_h^{n+1}, \nabla S_h) + c^*(u_h^n, T_h^{n+1}, S_h) = (\gamma^{n+1}, S_h).& \qquad (8)
\end{aligned}$$

where $\alpha_1 = \alpha_1(x, h)$ is a known, positive, bounded function and constant elementwise and α_2 is called the grad-div stabilization parameter. In our analysis, we propose α_1 and α_2 as $O(h^2)$ and $O(1)$ constants, respectively.

3 Numerical Analysis

In this section, we present the numerical analysis of the Boussinesq equations based on the finite element formulation (5)–(8). We first prove the stability of the method by using standard energy arguments.

Lemma 1 *The solution of (5)–(8) is unconditionally stable in the following sense: for any* $\Delta t > 0$

$$\|T_h^N\|^2 + \kappa\Delta t \sum_{n=0}^{N-1} \|\nabla T_h\|^2 \leq \|T_h^0\|^2 + \Delta t\kappa^{-1} \sum_{n=0}^{n-1} \|\gamma^{n+1}\|_{-1}^2$$

$$\|u_N^h\|^2 + \Delta t \sum_{n=0}^{N-1} \left(\nu\|\nabla u_h^{n+1}\|^2 + \alpha_2\|\nabla \cdot u_h^{n+1}\|^2\right) + \alpha_1\Delta t\|\nabla \times u_h^N\|^2$$
$$\leq \|u_h^0\|^2 + \alpha_1\Delta t\|\nabla \times u_h^0\|^2 + 2\nu^{-1}\Delta t \sum_{n=0}^{N-1} \|f^{n+1}\|_{-1}^2$$
$$+\tilde{C}T\left(\|T_h^0\|^2 + \Delta t\kappa^{-1}\sum_{n=0}^{n-1}\|\gamma^{n+1}\|_{-1}^2\right)$$

where $\tilde{C} = C\nu^{-1}Ri^2$*.*

Proof Choosing $S_h = T_h^{n+1}$ in (8) and $v_h = u_h^{n+1}$ in (5), using the triangle, Cauchy-Schwarz and Young's inequalities gives the stated result. □

We now give the error analysis of the method. We assume that the exact solution satisfies the following regularity assumptions for the optimal asymptotic error estimation:

$$\begin{aligned} u, T &\in L^\infty(0,T;H^1(\Omega)) \cap H^1(0,T;H^{k+1}(\Omega)) \cap H^3(0,T;L^2(\Omega)) \cap H^2(0,T;H^1(\Omega)) \\ p &\in L^2(0,T;H^{s+1}(\Omega)) \cap H^2(0,T;L^2(\Omega)) \qquad (9) \\ f, \gamma &\in L^2(0,T;L^2(\Omega)) \end{aligned}$$

Theorem 1 *Let* (u, p, T) *be the solution of the Boussinesq system. In addition to the regularity assumptions (9), let* $(X_h, W_h, Q_h) = (P_2, P_2, P_1)$ *be the Taylor-Hood finite element spaces satisfying theoretical approximation estimations. Then, the following asymptotic error estimation is satisfied for the errors* $e_u^n = u^n - u_h^n$ *and* $e_T^n = T^n - T_h^n$*:*

$$\|e_u^N\|^2 + \|e_T^N\|^2 + \Delta t\sum_{n=0}^{N-1}\left(\nu\|\nabla e_u^{n+1}\|^2 + \kappa\|\nabla e_T^{n+1}\|^2\right) + \alpha_1\Delta t\sum_{n=0}^{N-1}\|\nabla \times e_u^{n+1}\|^2$$
$$+\alpha_2\Delta t\sum_{n=0}^{N-1}\|\nabla \cdot e_u^{n+1}\|^2 \leq C((\Delta t)^2 + h^4 + \|e_u^0\|^2 + \|e_T^0\|^2).$$

where C is a generic constant.

Proof The process of this proof is similar to the proof of [2]. One can adapt it to the proof of the error estimate of [5]. □

Remark 1 In order to obtain optimal order of accuracy, the initial approximations of u_h^0 and T_h^0 need to be suitably interpolated in X_h in such a way that $\|e_u^0\|$ and $\|e_T^0\|$ are optimal, that is $\|e_u^0\| \leq Ch^2\|u\|_2$ and $\|e_T^0\| \leq Ch^2\|T\|_2$. We consider $u_h^0 = I_u(u^0)$ and $T_h^0 = I_T(T^0)$ for some interpolations I_u in X_h and I_T in W_h, respectively. Existence of such operators can be found in [12]. Thus, Theorem 1 implies that, the error in velocity is $O(h^2)$ and the error in temperature is $O(\Delta t)$, which are optimal convergence rates for the scheme.

4 Numerical Experiments

In this section, we provide three numerical experiments to test the theoretical findings and to show the efficiency of the proposed method. Firstly, we verify the order of numerical convergence rates which are predicted in Theorem 1. Secondly, we provide the so-called Marsigli's flow example to prove that the method captures correct flow patterns by using a coarse mesh discretization. Lastly, we present the well-known Buoyancy driven cavity example and compare the Nusselt numbers obtained by the proposed method to previously obtained ones in literature. All computations are carried out with the finite element software package FreeFem++ [6]. In all simulations, we use (P_2, P_2, P_1) Taylor-Hood finite spaces for velocity, temperature and pressure on uniform triangular grids and P_1 for the large scale space L_H.

4.1 *Numerical Convergence Study*

In this subsection, we test the optimal convergence rates of the scheme (5)–(8) with a known analytic solution

$$\begin{aligned} u &= \begin{pmatrix} (1+0.1t)cos(\pi x) \\ (1+0.1t)sin(\pi y) \end{pmatrix}, \\ p &= (1+0.2t)sin(\pi(x+y)), \\ T &= \sin(\pi x)y\exp(t) \end{aligned}$$

on the unit square domain $\Omega := [0, 1]^2$. We take the parameters $Re = Ri = \kappa = 1$, stabilization parameters $\alpha_1 = h^2, \alpha_2 = 0.01$ and coarse mesh size $H = \sqrt{h}$. The right hand side functions f, γ are determined by the given true solution.

To test the spatial errors, we fix the time-step $\Delta t = T/8$ with end time $T = 10^{-4}$ and calculate the errors in $L^2(0, T; H_0^1)$ for varying h. To see the temporal errors,

Table 1 Spatial errors and rates of convergence for SAV method

h	$\|\mathbf{u}-\mathbf{u}_h\|_{2,1}$	Rate	$\|T-T_h\|_{2,1}$	Rate
1/4	7.128e−4	–	5.045e−4	–
1/8	1.771e−4	2.00	1.256e−4	2.00
1/16	4.353e−5	2.024	3.069e−5	2.033
1/32	1.119e−5	1.959	7.458e−6	2.040
1/64	3.002e−6	1.898	1.844e−6	2.146

Table 2 Temporal errors and rates of convergence for SAV method

Δt	$\|\mathbf{u}-\mathbf{u}_h\|_{2,1}$	Rate	$\|T-T_h\|_{2,1}$	Rate
1	1.096e−2	–	1.302e−1	–
1/2	1.373e−2	0.513	6.005e−2	1.121
1/4	9.542e−3	0.525	2.843e−2	1.078
1/8	6.542e−3	0.547	1.380e−2	1.042
1/16	4.348e−3	0.573	6.802e−3	1.021
1/32	2.786e−3	0.839	3.376e−3	1.010
1/64	1.702e−3	0.918	1.682e−3	1.004

we fix the mesh size $h = 1/128$ with an end time $t = 1$ and calculate the errors in $L^2(0, T; H_0^1)$ for varying Δt. Errors and rates are presented in Tables 1 and 2. As expected, we observe first order convergence in time and second order convergence in space which are optimal rate of convergence for velocity and temperature for the Taylor-Hood finite element spaces.

4.2 Marsigli's Flow Experiment

In this subsection, we test Marsigli's flow. In 1679, Marsigli figured out that the reason of ocean currents is due to the density differences. He observed that the fluid of lower density moves on the top of the fluid with higher density. In this experiment, we simulate this physical situation on a much coarser mesh than is needed by a direct numerical simulation which is known to fail even for finer meshes, see [7]. The aim is to capture correct flow patterns with less computational effort. The problem domain is a rectangular box $\Omega := (0, 1) \times (0, 8)$. No slip velocity boundary conditions are applied and the temperature gradients are taken to be zero at all boundaries. The initial temperature is given precisely as

$$T_0 = \begin{cases} 1.5 \;\; x \leq 4.0 \\ 1.0 \; x > 4.05 \end{cases}$$

and the initial velocity is zero. The flow parameters are taken as $Pr = 1$, $Re = 1000$, $Ri = 4$. We choose a large time-step size $\Delta t = 0.02$ and plot the temperature contours and velocity streamlines at $t = 2, 4, 6, 8$.

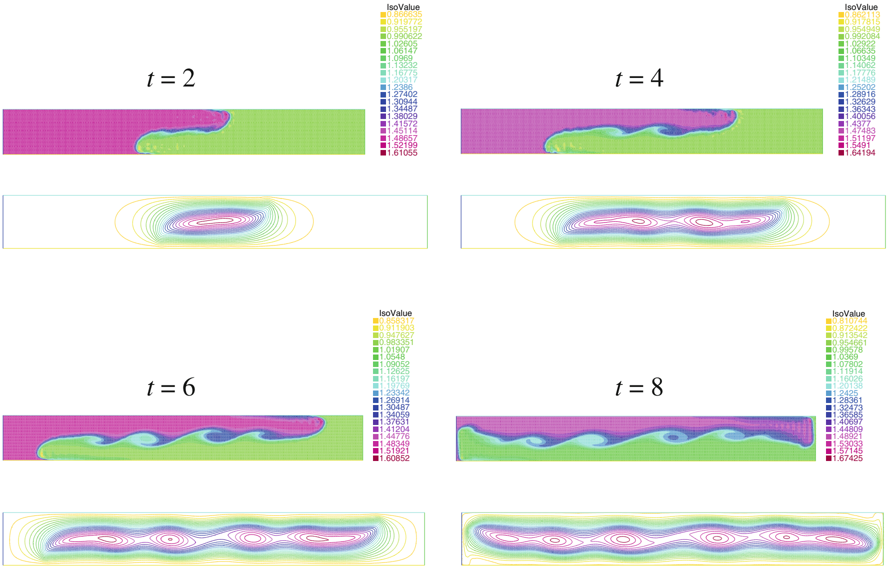


Fig. 1 Temperature contours and velocity streamlines at $t = 2, 4, 6, 8$

The resulting patterns are given in Fig. 1. As expected, the fluid at different temperatures mix at the interface and as time evolves the warmer fluid tends to spread out on the colder one. One can easily deduce from the Fig. 1 that the flow patterns of our solution match perfectly with the patterns of [8] in which a fourth order finite difference scheme is used for the Boussinesq equations. This comparison proves the promise of the method in this sense.

4.3 Thermal Distribution in Buoyancy Driven Cavity

In engineering, it is very important to know the thermal distribution along the hot and cold walls. The parameter of interest is called Nusselt number (Nu) which measures this distribution. The local and average Nusselt numbers are defined in [5]. Also, the problem domain is a unit square and boundary conditions are described as in [5]. The flow parameters are chosen as $Pr = \kappa = \nu = 1$ in this test.

In Fig. 2, we plot the temperature contours and velocity streamlines for $Ra = 10^4, 10^5$ with the time-step $\Delta t = 0.01$ and for $Ra = 10^6$ with the time-step $\Delta t = 0.001$. In addition, we provide the average Nusselt numbers of the proposed algorithm for $Ra = 10^4, 10^5, 10^6$. As seen in Table 3, we obtain acceptable results using a much coarser mesh than the literature known to obtain such results on a very fine mesh, see [11]. This shows the computational power and advantage of the proposed method against others.

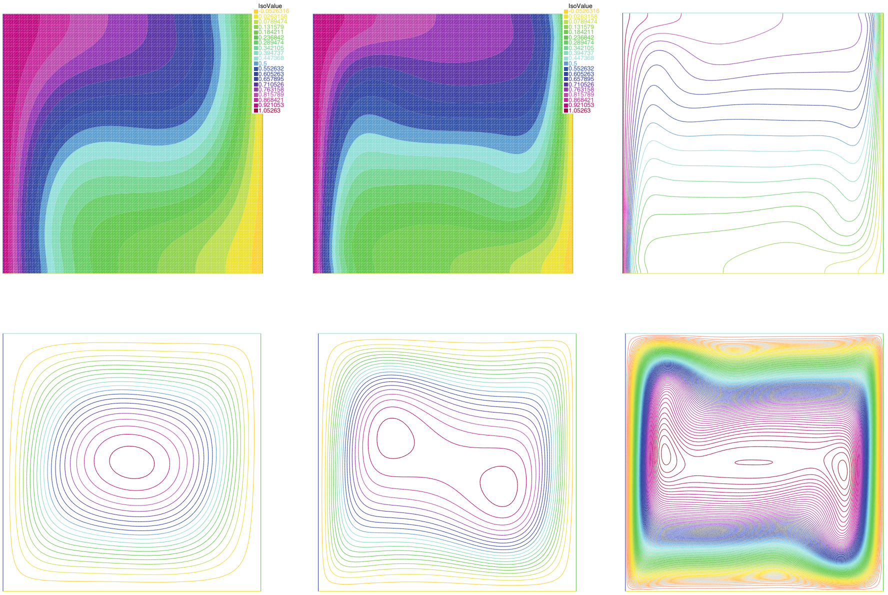


Fig. 2 Temperature contours (up) and streamlines (down) for $Ra = 10^4, 10^5, 10^6$ (from left to right)

Table 3 Comparison of average Nusselt number on hot wall for varying Rayleigh Numbers

Ra	Proposed method	Ref. [9]	Ref. [10]	Ref. [11]
10^4	2.257 (32 × 32)	2.254 (301 × 301)	2.201 (142 × 142)	2.258 (101 × 101)
10^5	4.600 (64 × 64)	4.598 (301 × 301)	4.532 (142 × 142)	4.511 (101 × 101)
10^6	8.984 (100 × 100)	8.976 (301 × 301)	8.90 (142 × 142)	8.97 (101 × 101)

5 Conclusion

In this paper, we proposed and analyzed a subgrid artificial viscosity method for solving the Boussinesq system based on the backward Euler time discretization scheme. We proved that the approximate solutions of the proposed algorithm are uniformly bounded at all time without any restriction on timestep. We also showed that the method is optimally convergent with suitable choices of artificial viscosity and the grad-div stabilization parameter. Finally, the efficiency and accuracy of the method is demonstrated on several numerical tests which revealed that the method gives superior results with a less computational effort over the previously obtained ones in literature.

References

1. Bejan, A. & Krauss, A. D.: Heat Transfer Handbook. John Wiley & Sons (2003)
2. Galvin, K. J.: New subgrid artificial viscosity Galerkin methods for the Navier-Stokes equations. Comput. Methods Appl. Mech. Engrg. **200**, 242–250 (2011)
3. Layton, W.: A connection between subgrid scale eddy viscosity and mixed methods. Appl. Math. Comput. **133**, 147–157 (2002)
4. Borggaard, J., Illiescu, T., Lee, H. & Roop, J. P.: A two level Smagorinsky model. Multiscale Model. Simul. Mech. Engrg. **7**, 599–621 (2008)
5. Çıbık, A. & Kaya, S.: A projection-based stabilized finite element method for steady-state natural convection problem. Int. J. Math. Anal. **381**, 469–484 (2011)
6. Hecht, F.: New development in FreeFem++.J. Numer. Math. **20**, 251–265 (2012)
7. Belenli, M. A., Kaya, S. & Rebholz, L.G.: An explicitly decoupled variational multiscale method for incompressible, non-isothermal flows. Comput. Meth.Appl. Math. **15**, 1–20 (2015)
8. Liu, J. G., Wang, C. & Johnston, H.: A fourth order scheme for incompressible Boussinesq equations. J. Sci. Comput. **18**, 253–285 (2003)
9. Wan, D. C., Patnaik, B. S. V. & Wei, G. W.: A new benchmark quality solution for the buoyancy driven- cavity by discrete singular convolution. Numerical Heat Transfer, Part B **40**, 199–228 (2001)
10. Younes, A., Makrad, A., Zidane, A., Shao, Q. & Bouhala, L.: A combination of Crouzeix-Raviart, Discontinuous Galerkin and MPFA methods for buoyancy-driven flows. Int. J. Numer. Meth. Heat Fluid Flow. **24(3)**, 735–759 (2014)
11. Kosec, G. & Šarler, B.: Solution of a low Prandtl number natural convection benchmark by a local meshless method. Int. J. Numer. Meth. Heat Fluid Flow.**23(1)**, 189–204 (2013)
12. Girault, V. & Raviart, P. A.: Finite element method for Navier-Stokes equations. Springer, Berlin (1986)

FFT-Based Solution Schemes for the Unit Cell Problem in Periodic Homogenization of Magneto-Elastic Coupling

Felix Dietrich

Abstract Starting from the linear equations for magneto-elastic coupling, the unit cell problem and the homogenized problem are derived as limits of a two-scale convergence process in a periodic homogenization setting. Exploiting the periodicity of the cell problem and the properties of its Fourier series representation allows for a reformulation as a Lippmann–Schwinger type equation. Iterative algorithms to solve these equations are presented and validated by an analytically solvable test problem.

1 Introduction

Coupling effects from piezomagnetic or (biased) magnetostrictive materials make them suitable for the development and production of actuators or sensors [1, 2]. Aiming at the creation of materials that follow a certain behavior, several such phases are combined to potentially amplify these effects [3, 4]. These composite materials may exhibit a complex micro-structural geometry that demands homogenization techniques [5] such as an asymptotic expansion series [6] or two-scale limits [7] to be treated efficiently. Due to the nature of the resulting elliptic partial differential equation spectral methods have been proven to solve the arising equations in a fast and cost efficient manner [8, 9]. In recent years these methods were developed further to include voids [10], to handle nonlinearities [11] or to be applied on general periodic anisotropic translation invariant spaces [12].

The following work has three goals in mind. First, it wishes to show the homogenization procedure and its therein derived quantities and problems for the coupled system. Next, it is explained how spectral schemes for this problem class are derived and how the coupling can be treated in numerical schemes. In the end,

F. Dietrich (✉)
Technische Universität Kaiserslautern, Kaiserslautern, Germany
e-mail: fdietric@rhrk.uni-kl.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_29

the correctness of the algorithms and their implementation shall be validated by an appropriately chosen benchmark.

2 Mathematical Formulation

Let $\Omega \subseteq \mathbb{R}^d$ be a bounded domain with Lipschitz boundary $\partial\Omega = \Lambda_{\mathrm{D}}^{\mathrm{mech}} \sqcup \Lambda_{\mathrm{N}}^{\mathrm{mech}} = \Lambda_{\mathrm{D}}^{\mathrm{mag}} \sqcup \Lambda_{\mathrm{N}}^{\mathrm{mag}}$. The strong form of the coupled linear magneto-elastic system with periodic coefficients reads as

$$-\operatorname{div}\left(\mathsf{C}\left(\frac{\mathbf{x}}{\kappa}\right)\boldsymbol{\varepsilon}\left(\mathbf{u}_\kappa\right)(\mathbf{x}) - \mathsf{e}\left(\frac{\mathbf{x}}{\kappa}\right)\mathbf{H}\left(\Phi_\kappa\right)(\mathbf{x})\right) = \mathbf{f}^{\mathrm{mech}}(\mathbf{x}) \quad \text{in } \Omega\,, \tag{1}$$

$$-\operatorname{div}\left(\mathsf{e}^{\mathrm{T}}\left(\frac{\mathbf{x}}{\kappa}\right)\boldsymbol{\varepsilon}\left(\mathbf{u}_\kappa\right)(\mathbf{x}) + \upmu\left(\frac{\mathbf{x}}{\kappa}\right)\mathbf{H}\left(\Phi_\kappa\right)(\mathbf{x})\right) = \mathrm{f}^{\mathrm{mag}}(\mathbf{x}) \quad \text{in } \Omega\,, \tag{2}$$

with Dirichlet boundary conditions $\mathbf{u}_\kappa = \mathbf{0}$ on $\Lambda_{\mathrm{D}}^{\mathrm{mech}}$ and $\Phi_\kappa = 0$ on $\Lambda_{\mathrm{D}}^{\mathrm{mag}}$, as well as Neumann boundary conditions

$$\left(\mathsf{C}\left(\frac{\mathbf{x}}{\kappa}\right)\boldsymbol{\varepsilon}\left(\mathbf{u}_\kappa\right)(\mathbf{x}) - \mathsf{e}\left(\frac{\mathbf{x}}{\kappa}\right)\mathbf{H}\left(\Phi_\kappa\right)(\mathbf{x})\right)\cdot\mathbf{n} = \tilde{\boldsymbol{\sigma}}(\mathbf{x}) \quad \text{on } \Lambda_{\mathrm{N}}^{\mathrm{mech}}\,, \tag{3}$$

$$\left(\mathsf{e}^{\mathrm{T}}\left(\frac{\mathbf{x}}{\kappa}\right)\boldsymbol{\varepsilon}\left(\mathbf{u}_\kappa\right)(\mathbf{x}) + \upmu\left(\frac{\mathbf{x}}{\kappa}\right)\mathbf{H}\left(\Phi_\kappa\right)(\mathbf{x})\right)\cdot\mathbf{n} = -\tilde{\mathrm{B}}(\mathbf{x}) \quad \text{on } \Lambda_{\mathrm{N}}^{\mathrm{mag}}\,. \tag{4}$$

The material tensors, namely the *stiffness* C, the *permeability* $\upmu$, and the *coupling tensor* e, are defined on the d-dimensional torus $\mathbb{T}^d \cong [0,1)^d$. The period length of the material's geometry is denoted by $\kappa \in \mathbb{R}_{>0}$ and visualized in Fig. 1. The *strain operator* $\boldsymbol{\varepsilon}$ takes the symmetric gradient of a *displacement field* $\mathbf{u}_\kappa$, whereas the *magnetic field operator* $\mathbf{H}$ denotes the negative gradient of a *magnetic scalar potential* Φ_κ. The divergence operator div in (1) is applied column-wise to the resulting stress field. For the purpose of simplification however, C, e, and $\boldsymbol{\varepsilon}$ will refer to the Mandel-notation of these quantities.

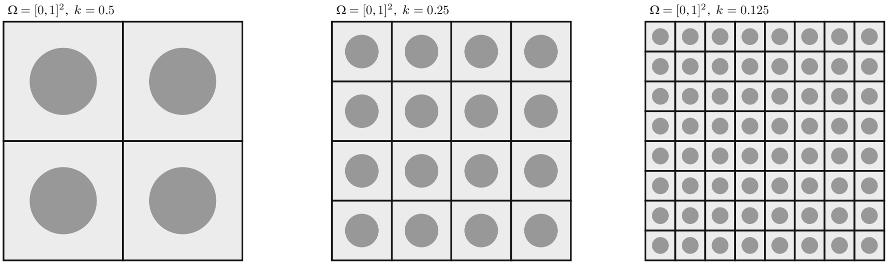

$\Omega = [0,1]^2,\ k = 0.5$
$\Omega = [0,1]^2,\ k = 0.25$
$\Omega = [0,1]^2,\ k = 0.125$

Fig. 1 Depiction of the domain $\Omega = [0,1]^2$ with periodically repeating inclusions for $\kappa = 0.5$, $\kappa = 0.25$, and $\kappa = 0.125$ (from left to right)

In the case of composite materials, one can assume the material tensors to be L^∞-functions and furthermore for C and $\upmu$ to be symmetric positive definite matrices fulfilling an ellipticity condition. If additionally all right-hand sides are at least L^2-functions, the Lax–Milgram theorem ensures the existence of unique solutions $\mathbf{u}_\kappa \in \mathcal{H}^1\left(\Omega, \mathbb{R}^d\right)$ and $\Phi_\kappa \in \mathcal{H}^1\left(\Omega, \mathbb{R}\right)$.

3 Periodic Homogenization

The process of periodic homogenization is equivalent to posing the question whether the formal limits of the solutions $\mathbf{u}_\kappa$ and Φ_κ in (1) and (2) exist for $\kappa \to 0$ and to which problem they correspond. The answer to this question is given by the *two-scale convergence method* which states that there exist functions

$$(\mathbf{u}_0, \mathbf{u}_1) \in \mathcal{H}^1_{\text{mech}}\left(\Omega, \mathbb{R}^d\right) \times L^2\left(\Omega, \mathcal{H}^1\left(\mathbb{T}^d, \mathbb{R}^d\right) \setminus \mathbb{R}\right) , \tag{5}$$

$$(\Phi_0, \Phi_1) \in \mathcal{H}^1_{\text{mag}}\left(\Omega, \mathbb{R}\right) \times L^2\left(\Omega, \mathcal{H}^1\left(\mathbb{T}^d, \mathbb{R}\right) \setminus \mathbb{R}\right) , \tag{6}$$

such that $\boldsymbol{\varepsilon}\left(\mathbf{u}_\kappa\right) \twoheadrightarrow \boldsymbol{\varepsilon}_{\mathbf{x}}\left(\mathbf{u}_0\right) + \boldsymbol{\varepsilon}_{\mathbf{y}}\left(\mathbf{u}_1\right)$ and $\mathbf{H}\left(\Phi_\kappa\right) \twoheadrightarrow \mathbf{H}_{\mathbf{x}}\left(\Phi_0\right) + \mathbf{H}_{\mathbf{y}}\left(\Phi_1\right)$. Here, the subscripts for $\mathcal{H}^1$ are each indicative of the incorporated Dirichlet boundary conditions in the sense of the trace mapping theorem. These two-scale limits lead to a variational formulation that allows to separate between the sought-after *homogenized problem* whose material tensors are not spatially dependent anymore and the underlying *cell problem* which contains all geometrical information of a period.

3.1 Cell Problem

The cell problem consists of solving the system of equations on the torus with multiple right-hand sides. More precisely, instead of prescribing macroscopic strains and magnetic fields directly, one only prescribes unit vectors and reconstructs the solution of the cell problem afterwards.

Sticking with the Mandel notation for mechanical quantities, one solves for unit vectors $\mathbf{z}_i^{\text{mech}} \in \mathbb{R}^{d_V}$, $i = 1, \ldots, d_V := d\,(d+1)/2$, the system

$$-\operatorname{div}_{\mathbf{y}}\left(\mathsf{C}\boldsymbol{\varepsilon}_{\mathbf{y}}\left(\boldsymbol{\omega}_i^{\text{mech}}\right) - \mathsf{e}\mathbf{H}_{\mathbf{y}}\left(\varrho_i^{\text{mech}}\right)\right) = \operatorname{div}_{\mathbf{y}}\left(\mathsf{C}\mathbf{z}_i^{\text{mech}}\right) \quad \text{in } \mathbb{T}^d , \tag{7}$$

$$-\operatorname{div}_{\mathbf{y}}\left(\mathsf{e}^{\mathrm{T}}\boldsymbol{\varepsilon}_{\mathbf{y}}\left(\boldsymbol{\omega}_i^{\text{mech}}\right) + \upmu\mathbf{H}_{\mathbf{y}}\left(\varrho_i^{\text{mech}}\right)\right) = \operatorname{div}_{\mathbf{y}}\left(\mathsf{e}^{\mathrm{T}}\mathbf{z}_i^{\text{mech}}\right) \quad \text{in } \mathbb{T}^d , \tag{8}$$

which corresponds to macroscopic strain fields, and analogously for unit vectors $\mathbf{z}_j^{\text{mag}} \in \mathbb{R}^d, \ j = 1, \ldots, d$ the system

$$-\operatorname{div}_{\mathbf{y}}\left(\mathsf{C}\boldsymbol{\varepsilon}_{\mathbf{y}}\left(\boldsymbol{\omega}_j^{\text{mag}}\right) - \mathsf{e}\mathbf{H}_{\mathbf{y}}\left(\varrho_j^{\text{mag}}\right)\right) = \operatorname{div}_{\mathbf{y}}\left(-\mathsf{e}\mathbf{z}_j^{\text{mag}}\right) \quad \text{in } \mathbb{T}^d , \tag{9}$$

$$-\operatorname{div}_{\mathbf{y}}\left(\mathsf{e}^{\mathrm{T}}\boldsymbol{\varepsilon}_{\mathbf{y}}\left(\boldsymbol{\omega}_j^{\text{mag}}\right) + \mu\mathbf{H}_{\mathbf{y}}\left(\varrho_j^{\text{mag}}\right)\right) = \operatorname{div}_{\mathbf{y}}\left(\mu^{\mathrm{T}}\mathbf{z}_j^{\text{mag}}\right) \quad \text{in } \mathbb{T}^d , \tag{10}$$

corresponding to macroscopic magnetic fields. Here, all derivatives are taken only with respect to the microscopic variable $\mathbf{y}$.

The solutions of (7)–(10) are called *correctors* and are used to reconstruct the unit cell solutions

$$\mathbf{u}_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d_V} \left(\boldsymbol{\varepsilon}_{\mathbf{x}}\left(\mathbf{u}_0\right)(\mathbf{x})\right)_i \boldsymbol{\omega}_i^{\text{mech}}(\mathbf{y}) + \sum_{j=1}^{d}\left(-\frac{\partial \Phi_0}{\partial x_j}(\mathbf{x})\right)\boldsymbol{\omega}_j^{\text{mag}}(\mathbf{y}) , \tag{11}$$

$$\Phi_1(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{d_V} \left(\boldsymbol{\varepsilon}_{\mathbf{x}}\left(\mathbf{u}_0\right)(\mathbf{x})\right)_i \varrho_i^{\text{mech}}(\mathbf{y}) + \sum_{j=1}^{d}\left(-\frac{\partial \Phi_0}{\partial x_j}(\mathbf{x})\right)\varrho_j^{\text{mag}}(\mathbf{y}) , \tag{12}$$

where $\left(\boldsymbol{\varepsilon}_{\mathbf{x}}\left(\mathbf{u}_0\right)(\mathbf{x})\right)_i$ refers to the i-th component. It should be noted that the correctors only depend on $\mathbf{y}$ whereas the coefficients only depend on the variable $\mathbf{x}$.

3.2 Homogenized Problem

The reason for computing the correctors instead of $\mathbf{u}_1$ and Φ_1 directly is their use in the computation of the *effective material tensors*. These constant tensors are no longer spatially dependent while simulating a homogeneous material that behaves the same as the composite.

The entries of the effective tensors for stiffness, permeability and the coupling tensor can be computed as

$$\mathsf{C}_{mn}^{\text{eff}} = \int_{\mathbb{T}^d} \mathsf{C}_{mn} + \sum_{k=1}^{d_V} \mathsf{C}_{mk}\left(\boldsymbol{\varepsilon}_{\mathbf{y}}\left(\boldsymbol{\omega}_n^{\text{mech}}\right)\right)_k + \sum_{l=1}^{d} \mathsf{e}_{ml}\frac{\partial \varrho_n^{\text{mech}}}{\partial y_l}\,\mathrm{d}\mathbf{y} , \tag{13}$$

$$\mu_{mn}^{\text{eff}} = \int_{\mathbb{T}^d} \mu_{mn} + \sum_{k=1}^{d_V} \mathsf{e}_{km}\left(\boldsymbol{\varepsilon}_{\mathbf{y}}\left(\boldsymbol{\omega}_n^{\text{mag}}\right)\right)_k - \sum_{l=1}^{d} \mu_{ml}\frac{\partial \varrho_n^{\text{mag}}}{\partial y_l}\,\mathrm{d}\mathbf{y} , \tag{14}$$

$$
\begin{aligned}
\mathsf{e}_{mn}^{\mathrm{eff}} &= \int\limits_{\mathbb{T}^d} \mathsf{e}_{mn} - \sum_{k=1}^{d_V} \mathsf{C}_{mk} \left(\boldsymbol{\varepsilon}_{\mathbf{y}} \left(\boldsymbol{\omega}_n^{\mathrm{mag}}\right)\right)_k - \sum_{l=1}^{d} \mathsf{e}_{ml} \frac{\partial \varrho_n^{\mathrm{mag}}}{\partial y_l} \, \mathrm{d}\mathbf{y} \\
&= \int\limits_{\mathbb{T}^d} \mathsf{e}_{mn} + \sum_{k=1}^{d_V} \mathsf{e}_{kn} \left(\boldsymbol{\varepsilon}_{\mathbf{y}} \left(\boldsymbol{\omega}_m^{\mathrm{mech}}\right)\right)_k - \sum_{l=1}^{d} \upmu_{nl} \frac{\partial \varrho_m^{\mathrm{mech}}}{\partial y_l} \, \mathrm{d}\mathbf{y} \,,
\end{aligned} \tag{15}
$$

with the correctors stemming from the previous cell problems (7)–(10).

Using (13)–(15), the homogenized problem reads as

$$
-\operatorname{div}_{\mathbf{x}} \left(\mathsf{C}^{\mathrm{eff}} \boldsymbol{\varepsilon}_{\mathbf{x}} \left(\mathbf{u}_0\right) - \mathsf{e}^{\mathrm{eff}} \mathbf{H}_{\mathbf{x}} \left(\Phi_0\right)\right) = \mathbf{f}^{\mathrm{mech}} \quad \text{in } \Omega \,, \tag{16}
$$

$$
-\operatorname{div}_{\mathbf{x}} \left(\left(\mathsf{e}^{\mathrm{eff}}\right)^{\mathrm{T}} \boldsymbol{\varepsilon}_{\mathbf{x}} \left(\mathbf{u}_0\right) + \upmu^{\mathrm{eff}} \mathbf{H}_{\mathbf{x}} \left(\Phi_0\right)\right) = \mathrm{f}^{\mathrm{mag}} \quad \text{in } \Omega \,, \tag{17}
$$

where $\mathbf{u}_0$ and Φ_0 need to fulfill the boundary conditions of the original problem given in Sect. 2. Note that contrary to the cell problem, the homogenized problem only contains derivatives with respect to $\mathbf{x}$.

4 Spectral Schemes on the Unit Cell

While Finite Elements may be used for the homogenized problem, the periodic nature of the cell problem offers the possibility to work with purely algebraic expressions. Expressing all quantities as a *Fourier Series* and exploiting its properties with respect to differentiation results in integral equations of *Lippmann–Schwinger type* whose fundamental solution operators are explicitly given in the frequency domain.

4.1 Periodic Lippmann–Schwinger Equations

In the following, the strain field is split into its spatial average $\bar{\boldsymbol{\varepsilon}} := \int_{\mathbb{T}^d} \boldsymbol{\varepsilon} \, \mathrm{d}\mathbf{y}$ and its remaining fluctuating part $\tilde{\boldsymbol{\varepsilon}} := \boldsymbol{\varepsilon} - \bar{\boldsymbol{\varepsilon}}$. The same is done for the magnetic field. Furthermore, the dependence on $\mathbf{u}$ and Φ is omitted for better readability.

With this notion, Eqs. (7)–(10) fit the more general problems

$$
-\operatorname{div} \left(\mathsf{C}^0 \tilde{\boldsymbol{\varepsilon}} + \boldsymbol{\tau}_{\boldsymbol{\varepsilon}}\right) = \mathbf{g}_{\boldsymbol{\varepsilon}} \,, \tag{18}
$$

$$
-\operatorname{div} \left(\upmu^0 \tilde{\mathbf{H}} + \boldsymbol{\tau}_{\mathbf{H}}\right) = \mathbf{g}_{\mathbf{H}} \,, \tag{19}
$$

if one chooses $\mathbf{g}_{\boldsymbol{\varepsilon}}$ and $\mathbf{g}_{\mathbf{H}}$ to be the corresponding right-hand sides given before and sets

$$\boldsymbol{\tau}_{\boldsymbol{\varepsilon}} := \left(\mathsf{C} - \mathsf{C}^0\right)\tilde{\boldsymbol{\varepsilon}} - \mathsf{e}\tilde{\mathbf{H}} \, , \tag{20}$$

$$\boldsymbol{\tau}_{\mathbf{H}} := \left(\mu - \mu^0\right)\tilde{\mathbf{H}} + \mathsf{e}^{\mathrm{T}}\tilde{\boldsymbol{\varepsilon}} \, , \tag{21}$$

with C^0 and μ^0 being arbitrarily chosen constant *reference tensors*.

For each frequency vector $\mathbf{k} \in \mathbb{Z}^d$ one defines

$$\hat{\mathsf{Z}}^0_{\boldsymbol{\varepsilon}}(\mathbf{k}) := \left(\mathbf{k}^{\mathrm{T}}\mathsf{C}^0\mathbf{k}\right)^{-1} \quad \text{and} \quad \hat{Z}^0_{\mathbf{H}}(\mathbf{k}) := \left(\mathbf{k}^{\mathrm{T}}\mu^0\mathbf{k}\right)^{-1} \, , \tag{22}$$

where the hat is indicative of the Fourier domain. From this, the entries of the *solution operators* can be calculated explicitly as

$$\left(\hat{\Gamma}^0_{\boldsymbol{\varepsilon}}(\mathbf{k})\right)_{mnop} = -\frac{1}{2}\left(k_n k_p \big(\hat{\mathsf{Z}}^0_{\boldsymbol{\varepsilon}}(\mathbf{k})\big)_{mo} + k_m k_p \big(\hat{\mathsf{Z}}^0_{\boldsymbol{\varepsilon}}(\mathbf{k})\big)_{no}\right) \, , \tag{23}$$

$$\left(\hat{\Theta}^0_{\boldsymbol{\varepsilon}}(\mathbf{k})\right)_{mno} = -\frac{\mathrm{i}}{4\pi}\left(k_n \big(\hat{\mathsf{Z}}^0_{\boldsymbol{\varepsilon}}(\mathbf{k})\big)_{mo} + k_m \big(\hat{\mathsf{Z}}^0_{\boldsymbol{\varepsilon}}(\mathbf{k})\big)_{no}\right) \, , \tag{24}$$

$$\left(\hat{\Gamma}^0_{\mathbf{H}}(\mathbf{k})\right)_{mn} = -k_m k_n \hat{Z}^0_{\mathbf{H}}(\mathbf{k}) \, , \tag{25}$$

$$\left(\hat{\Theta}^0_{\mathbf{H}}(\mathbf{k})\right)_{m} = \frac{\mathrm{i}}{2\pi} k_m \hat{Z}^0_{\mathbf{H}}(\mathbf{k}) \, . \tag{26}$$

The solution of (18) and (19) can thus be written as

$$\tilde{\boldsymbol{\varepsilon}} = \Gamma^0_{\boldsymbol{\varepsilon}} * \boldsymbol{\tau}_{\boldsymbol{\varepsilon}} + \Theta^0_{\boldsymbol{\varepsilon}} * \mathbf{g}_{\boldsymbol{\varepsilon}} \, , \tag{27}$$

$$\tilde{\mathbf{H}} = \Gamma^0_{\mathbf{H}} * \boldsymbol{\tau}_{\mathbf{H}} + \Theta^0_{\mathbf{H}} * \mathbf{g}_{\mathbf{H}} \, , \tag{28}$$

where the asterisk denotes the convolution operation.

4.2 Algorithms

The advantage of resorting to spectral methods for the cell problem lies in them being applicable to regular grids arising directly from imaging techniques. Most of the algorithmic operations work on each pixel or voxel independently and can therefore be parallelized. The only exception to that would be the *Discrete Fourier Transform*, which is a well-known, highly optimized computational routine at this point. The discretization is based on collocation methods and follows straightforward from the truncation of the Fourier Series. While sticking to the same notation as before, all quantities will be understood as their discretized equivalents from now on.

Solving the discrete system can be achieved in one of two ways. The *staggered method* makes use of the fact that (27) and (28) share the same structure and can thus be solved by the same algorithm individually. Iterating between both equations where the approximated solution of one equation is then transferred unto the right-hand side of the other one is an intuitive approach in which already established solvers can be reused. The initially proposed *Basic Scheme* solves a single equation through a Neumann iteration but its convergence depends strongly on the choice of the reference tensors. It is also possible to rearrange the equation to a linear system that, despite not being symmetric, was shown to be solvable by conjugate gradient methods.

The *monolithic method* combines the strain $\tilde{\boldsymbol{\varepsilon}}$ and the magnetic field $\tilde{\mathbf{H}}$ into a single solution vector and sets up a common linear system at once. Let $\mathsf{F}_{\boldsymbol{\varepsilon}}$ and $\mathsf{F}_{\mathbf{H}}$ denote the Fourier matrices of appropriate size. Analogously, one defines the identity matrices $\mathsf{Id}_{\boldsymbol{\varepsilon}}$ and $\mathsf{Id}_{\mathbf{H}}$. The linear system with system matrix

$$\mathsf{A} = \begin{pmatrix} \mathsf{Id}_{\boldsymbol{\varepsilon}} & 0 \\ 0 & \mathsf{Id}_{\mathbf{H}} \end{pmatrix} - \begin{pmatrix} \mathsf{F}_{\boldsymbol{\varepsilon}}^{-1}\hat{\Gamma}_{\boldsymbol{\varepsilon}}^{0}\mathsf{F}_{\boldsymbol{\varepsilon}} & 0 \\ 0 & \mathsf{F}_{\mathbf{H}}^{-1}\hat{\Gamma}_{\mathbf{H}}^{0}\mathsf{F}_{\mathbf{H}} \end{pmatrix} \begin{pmatrix} \mathsf{C}-\mathsf{C}^{0} & \mathsf{e} \\ -\mathsf{e}^{\mathrm{T}} & \mu-\mu^{0} \end{pmatrix} \tag{29}$$

and right-hand side vector

$$\mathbf{b} = \begin{pmatrix} \mathsf{F}_{\boldsymbol{\varepsilon}}^{-1}\hat{\Gamma}_{\boldsymbol{\varepsilon}}^{0}\mathsf{F}_{\boldsymbol{\varepsilon}} & 0 \\ 0 & \mathsf{F}_{\mathbf{H}}^{-1}\hat{\Gamma}_{\mathbf{H}}^{0}\mathsf{F}_{\mathbf{H}} \end{pmatrix} \begin{pmatrix} \mathsf{C}\bar{\boldsymbol{\varepsilon}} - \mathsf{e}\bar{\mathbf{H}} \\ \mu\bar{\mathbf{H}} + \mathsf{e}^{\mathrm{T}}\bar{\boldsymbol{\varepsilon}} \end{pmatrix} + \begin{pmatrix} \mathsf{F}_{\boldsymbol{\varepsilon}}^{-1}\hat{\Theta}_{\boldsymbol{\varepsilon}}^{0}\mathsf{F}_{\boldsymbol{\varepsilon}} & 0 \\ 0 & \mathsf{F}_{\mathbf{H}}^{-1}\hat{\Theta}_{\mathbf{H}}^{0}\mathsf{F}_{\mathbf{H}} \end{pmatrix} \begin{pmatrix} \mathbf{g}_{\boldsymbol{\varepsilon}} \\ \mathbf{g}_{\mathbf{H}} \end{pmatrix} \tag{30}$$

can again be solved with classical iterative solvers such as conjugate gradient schemes.

5 Numerics

First, an analytically solvable test case is presented which proves useful in validating the implemented schemes. Consider a unit cell in 2D with two different phases denoted by superscripts α and β. Assume the stiffness tensors to be isotropic with Lamé parameters λ and μ, diagonal permeability tensors with entries p_{11} and p_{22}, and coupling tensors whose third row is all zeros. The spatial averages $\bar{\boldsymbol{\varepsilon}} = (1, 1, 0)^{\mathrm{T}}$ and $\bar{\mathbf{H}} = (1, 1)$ are prescribed. If the material tensors were to fulfill the additional constraints

$$\begin{aligned} &\lambda^{\alpha} = \lambda^{\beta}\,,\; e_{12}^{\alpha} = e_{12}^{\beta}\,,\; e_{21}^{\alpha} = e_{21}^{\beta}\,, \\ &p_{11}^{\beta} - p_{11}^{\alpha} = p_{22}^{\beta} - p_{22}^{\alpha} = e_{11}^{\alpha} - e_{11}^{\beta} = e_{22}^{\alpha} - e_{22}^{\beta} = 2\left(\mu^{\alpha} - \mu^{\beta}\right)\,, \end{aligned} \tag{31}$$

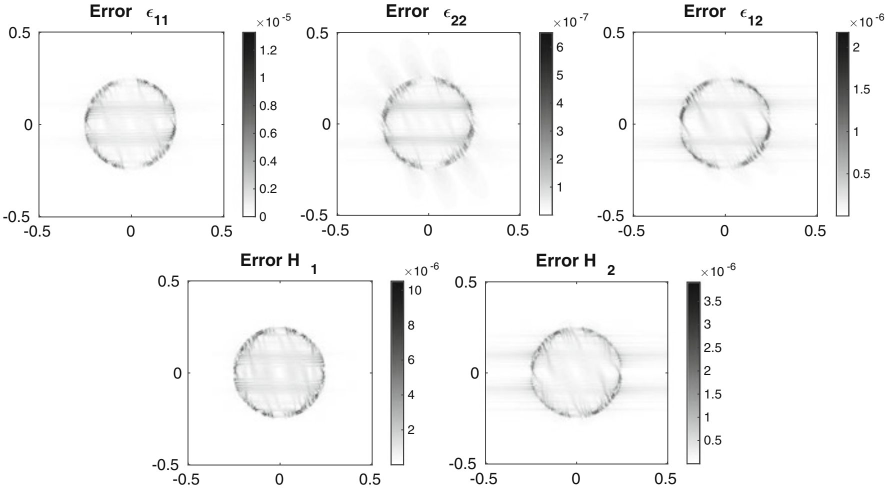


Fig. 2 Absolute error between numerical and analytical solution for the test case given in Sect. 5

and one was to prescribe the outer forces as

$$\mathbf{g}_{\boldsymbol{\varepsilon}} = -2\pi \begin{pmatrix} \left(2\mu^{\alpha} + \lambda^{\alpha} - e_{11}^{\alpha}\right) \cos\left(2\pi y_1\right) \\ \left(2\mu^{\alpha} + \lambda^{\alpha} - e_{22}^{\alpha}\right) \cos\left(2\pi y_2\right) \end{pmatrix} , \tag{32}$$

$$\mathbf{g}_{\mathbf{H}} = -2\pi \left(\left(e_{11}^{\alpha} + p_{11}^{\alpha}\right) \cos(2\pi y_1) + \left(e_{22}^{\alpha} + p_{22}^{\alpha}\right) \cos(2\pi y_2) \right) , \tag{33}$$

the solution of the coupled problem would be given by smooth solution fields

$$\boldsymbol{\varepsilon} = \begin{pmatrix} \sin\left(2\pi y_1\right) \\ \sin\left(2\pi y_2\right) \\ 0 \end{pmatrix} \quad \text{and} \quad \mathbf{H} = \begin{pmatrix} \sin\left(2\pi y_1\right) \\ \sin\left(2\pi y_2\right) \end{pmatrix} . \tag{34}$$

It is interesting to note that the solution of this example does actually not depend on the underlying geometry as long as the constraints (31) are met. For the case where the α-phase was centered as a circular inclusion in the β-phase, the error between the analytic and numerical solution on a 256×256 grid is shown in Fig. 2. A residual tolerance of 10^{-6} was used.

6 Conclusion

The cell and homogenized problem were obtained for a system of coupled elliptic partial differential equations in form of magneto-elastic coupling. Explicit formulas for solution operators were derived in the spectral domain. Numerical schemes following this method were outlined and validated through an analytically solvable test case.

Acknowledgments The author would like to thank his supervisor Bernd Simeon and former colleagues Mané Harutyunyan and Dennis Merkert for their work and support.

References

1. Brown Jr., W. F.: Magnetoelastic Interactions. Springer, Berlin, Heidelberg (1966) https://doi.org/10.1007/978-3-642-87396-6
2. Harutyunyan, M., Simeon, B.: On a saddle point problem arising from magneto-elastic coupling. Appl. Math. Lett. (2018) https://doi.org/10.1016/j.aml.2018.03.029
3. Lanotte, L., Ausanio, G., Hison, C., Iannotti, V., Luponio, C.: The potentiality of composite elastic magnets as novel materials for sensors and actuators. Sensors and Actuators A (2003) https://doi.org/10.1016/S0924-4247(03)00133-X
4. Lapine, M., Shadrivov, I. V., Powell, D. A., Kivshar, Y. S.: Magnetoelastic metamaterials. Nature Materials (2012) https://doi.org/10.1038/nmat3168
5. Cioranescu, D., Donato, P.: An Introduction to Homogenization. Oxford Lecture Series in Mathematics and Its Applications (1999)
6. Boutin, C.:Microstructural effects in elastic composites. Int. J. Solids Structures (1996) https://doi.org/10.1016/0020-7683(95)00089-5
7. Allaire, G.: Homogenization and Two-Scale Convergence. SIAM J. Math. Anal. (1992) https://doi.org/10.1137/0523084
8. Moulince, H., Suquet, P.: A numerical method for computing the overall response of nonlinear composites with complex microstructure. Comput. Methods Appl. Mech. Engrg. (1998) https://doi.org/10.1016/S0045-7825(97)00218-1
9. Vondřejc, J., Zeman, J., Marek, I.: An FFT-based Galerkin method for homogenization of periodic media. Comput. Math. Appl. (2014) https://doi.org/10.1016/j.camwa.2014.05.014
10. Michel, J. C., Moulince, H., Suquet, P.: A computational scheme for linear and non-linear composites with arbitrary phase contrast. Int. J. Numer. Meth. Engng. (2001) https://doi.org/10.1002/nme.275
11. Kabel, M., Böhlke, T., Schneider, M.: Efficient fixed point and Newton–Krylov solvers for FFT-based homogenization of elasticity at large deformations. Comput. Mech. (2014) https://doi.org/10.1007/s00466-014-1071-8
12. Bergmann, R., Merkert, D.: FFT-based homogenization on periodic anisotropic translation invariant spaces. Appl. Comput. Harmon. Anal. (2020) https://doi.org/10.1016/j.acha.2018.05.003

Novel Flux Approximation Schemes for Systems of Coupled Advection-Diffusion-Reaction Equations

J. van Dijk, R.A.M. van Gestel, C.E.M. Schoutrop, and J.H.M. ten Thije Boonkkamp

Abstract The physical modeling of transport in multi-component mixtures results in systems of coupled equations for the mass fractions. This contribution discusses the mathematical structure of such transport systems and presents a novel approximation scheme for the associated mass fluxes. The scheme respects the coupled nature of the equations and allows for a linearized source term. An illustrative example is presented.

1 Introduction

Conservation laws of advection-diffusion-reaction type are omnipresent in physics. Examples are those describing transfer of momentum and energy in a flowing substance and the mass balance equations for a multi-component medium. For such medium the equations that describe the components are coupled via the source terms, which express the result of chemical reactions, but also through the expressions for the transport coefficients, which usually depend on *all* mixture variables.

In the last decade the strong coupling between the mixture variables has triggered the development of new flux approximation schemes that respect that coupling [1, 2] and result in computations that guarantee conservation of mass regardless of the number of grid points that is used for the computation. Later extensions of the scheme take into account the source terms in the derivation of the flux approximation scheme [3]. This Complete Flux Scheme can be shown to be of second-order accuracy, regardless of the Péclet matrix that expresses the relative importance of advection compared to diffusive transport of the individual components [4].

J. van Dijk (✉) · R.A.M. van Gestel · C.E.M. Schoutrop · J.H.M. ten Thije Boonkkamp
Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: j.v.dijk@tue.nl; r.a.m.v.gestel@tue.nl; c.e.m.schoutrop@tue.nl; j.h.m.tenthijeboonkkamp@tue.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_30

In recent years there has been a renewed interest in the further development of flux approximation schemes for multi-component mixtures. This can be explained by the development of new plasma sources that feature a bewildering chemical complexity. As an example, research on *Solar Fuels* has resulted in the development of microwave plasma reactors in which water and carbon-dioxide are transformed into methane and other value-added chemicals in a process that is perhaps best described as ‘inverse combustion’. In such devices one typically needs to consider dozens to hundreds of components that are involved in many thousands of reactions [5].

The second-order differential equations that describe the mass fractions of the components in such a mixture are of advection-diffusion-reaction type and form a set of coupled quasi-linear equations. In recent years the authors have made significant progress in the development of flux approximation schemes for such systems and their application in finite-volume simulations. This work presents the physical model for multi-component diffusion that underlies those efforts, which is based on the work of Giovangigli et al. [6] (Sect. 2) and demonstrates a novel flux approximation scheme that allows a linearization of the source terms (Sect. 3). A numerical example is presented in Sect. 4.

2 Mathematical Model of Multi-Component Diffusion

For a one-dimensional Cartesian coordinate system the mass balance for a component i in a multi-component mixture takes the form

$$\partial_t \rho_i + \partial_x(\rho_i v_i) = \omega_i, \tag{1}$$

where ρ_i is the mass density of component i, v_i its velocity and ω_i its volumetric mass production rate. Summation of these equations over all components results in the continuity equation,

$$\partial_t \rho + \partial_x(\rho v) = 0, \tag{2}$$

where the total mass density ρ is given by $\rho = \sum_i \rho_i$. Note that $\sum_i \omega_i = 0$ since no net mass is produced in chemical reactions. Finally, the mass-averaged or barycentric velocity field v is defined by the relation

$$\rho v = \sum_i \rho_i v_i. \tag{3}$$

Instead of the mass densities ρ_i one usually adopts the components’ mass fractions $y_i = \rho_i/\rho$ as solution variables. In terms of these variables the mass balance equations (1) take the form

$$\partial_t(\rho y_i) + \partial_x(\rho y_i v_i) = \omega_i. \tag{4}$$

The further analysis of the subject is greatly facilitated by collecting the properties of the components in vectors. As an example, the mass fractions can be combined into the vector $\boldsymbol{y} = (y_1 \ \ldots \ y_N)^{\mathrm{T}}$, the mass flux densities $J_i = \rho y_i v_i$ into $\boldsymbol{J} = (J_1 \ \ldots \ J_N)^{\mathrm{T}}$, etc. Here T indicates transposition. The mass balance equations then read

$$\partial_t(\rho \boldsymbol{y}) + \partial_x \boldsymbol{J} = \boldsymbol{\omega}. \tag{5}$$

In order to solve equation (5) for the mass fractions y_i one needs, in addition to the expressions for the chemical source terms ω_i, knowledge of the component velocities v_i. In principle these can be obtained from a set of momentum balance equations. Each such equation is of the Navier-Stokes type, but carries additional terms due to the forces that the individual components exert on each other.

Solving such momentum balance for each individual component is excessively time-consuming for mixtures that consist of many components. Fortunately, in many cases of practical interest the momentum balance equations can be simplified to a set of coupled algebraic equations for the local *diffusion velocities*. The diffusion velocity u_i of component i is defined as the velocity of that component relative to the mass-averaged velocity,

$$u_i = v_i - v. \tag{6}$$

From the definition of the diffusion velocity it follows that diffusion does not result in net mass transport: multiplying equation (6) with ρ_i and summing over all components one finds

$$\sum_i \rho_i u_i = \sum_i \rho_i v_i - \Big(\sum_i \rho_i\Big) v = \rho v - \rho v = 0. \tag{7}$$

The result of the Stefan-Maxwell approach is that the Navier-Stokes equation needs to be solved only for the mass-averaged velocity. The component velocities are then obtained by adding the diffusion velocities to the mass-averaged velocity.

The diffusion velocities u_i are governed by a set of algebraic equations which are known as the *Stefan-Maxwell equations*. For a component i the result is

$$\sum_{j \neq i} f_{ij}(u_i - u_j) = -d_i, \tag{8}$$

where $f_{ij} = f_{ji}$ is a *friction coefficient*, which describes the momentum exchange between components i and j. Furthermore d_i is the *diffusive driving force* that acts on component i. The latter is related to spatial inhomogeneities and will be discussed later. For a further discussion of the Stefan-Maxwell equations and the calculation of the friction coefficients we refer to references [6] and [7]. In the vector notation the Stefan-Maxwell equations take the form

$$\boldsymbol{F}\boldsymbol{u} = -\boldsymbol{d}, \tag{9}$$

where the matrix $\boldsymbol{F}$ has elements

$$F_{ij} = \begin{cases} -f_{ij} & i \neq j, \\ \sum_{k\neq i} f_{ik} = \sum_{k\neq j} f_{kj} & i = j. \end{cases} \tag{10}$$

From equation (8) it is immediately clear that the Stefan-Maxwell equations are linearly dependent: it only gives relations between the *differences* of the diffusion velocities; the sum of the left-hand sides over all components vanishes. In order to solve the system (9) for the diffusion velocities one must therefore combine the Stefan-Maxwell equations with an additional constraint. That is given by equation (7), which, in matrix-vector form can be written as

$$\boldsymbol{y}^{\mathrm{T}}\boldsymbol{u} = 0. \tag{11}$$

In order to arrive at a non-singular formulation of the Stefan-Maxwell equations one could replace one of the equations with the constraint (11). However, we adopt another formulation, proposed by Giovangigli [6], to regularize the Stefan-Maxwell equations in a symmetric way. The idea is to left-multiply equation (11) with the column vector $\alpha\boldsymbol{y}$, where $\alpha > 0$ is a constant. By adding the result to the Stefan-Maxwell equations one obtains

$$\widetilde{\boldsymbol{F}}\boldsymbol{u} = -\boldsymbol{d}, \quad \text{with} \quad \widetilde{\boldsymbol{F}} = \boldsymbol{F} + \alpha\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}. \tag{12}$$

Giovangigli has demonstrated that the modified friction matrix $\widetilde{\boldsymbol{F}}$ is non-singular for $\alpha > 0$. Numerical considerations suggest that the value of α is chosen such that the diagonal elements of $\boldsymbol{F}$ and those of $\alpha\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}$ are balanced. The resulting system can be solved to obtain an expression for the diffusion velocities,

$$\boldsymbol{u} = -\left(\boldsymbol{F} + \alpha\boldsymbol{y}\boldsymbol{y}^{\mathrm{T}}\right)^{-1}\boldsymbol{d}. \tag{13}$$

It has been shown [6, section 4.1] that the driving forces related to ordinary diffusion can be written in the form

$$\boldsymbol{d} = \widetilde{\boldsymbol{M}}\partial_x\boldsymbol{y}, \tag{14}$$

where $\widetilde{\boldsymbol{M}}$ is a non-singular matrix that depends solely on the mass fractions. If we furthermore add the advective velocity to the diffusion velocities of the species and multiply each term with the mass density ρy_i of that component we obtain an expression for the total mass flux densities. The result is

$$\boldsymbol{J} = \rho v \boldsymbol{y} - \rho \,\mathrm{diag}(\boldsymbol{y}) \left(\boldsymbol{F} + \alpha \boldsymbol{y} \boldsymbol{y}^{\mathrm{T}}\right)^{-1} \widetilde{\boldsymbol{M}} \partial_x \boldsymbol{y}. \tag{15}$$

The reader will notice the resemblance of this expression with the single-variable flux expressions of advection-diffusion type: it is a quasi-linear system with one term that is proportional to the solution variables $\boldsymbol{y}$ and one that is proportional to the gradient of this variable. In order to emphasize this structure it useful to write this expression in the generic form

$$\boldsymbol{J} = \boldsymbol{U}(\boldsymbol{y})\boldsymbol{y} - \mathcal{E}(\boldsymbol{y})\partial_x \boldsymbol{y}. \tag{16a}$$

For the present case we find that

$$\boldsymbol{U} = \rho v \boldsymbol{I}, \quad \mathcal{E} = \rho \,\mathrm{diag}(\boldsymbol{y}) \left(\boldsymbol{F} + \alpha \boldsymbol{y} \boldsymbol{y}^{\mathrm{T}}\right)^{-1} \widetilde{\boldsymbol{M}}. \tag{16b}$$

The mass flux diffusion matrix $\mathcal{E}$ is positive definite (but not symmetric) and diagonalizable. When other types of diffusion are considered the matrices $\boldsymbol{U}$ and $\mathcal{E}$ take different forms, but the general structure of equation (16a) will stay the same. For the example of *ambipolar* diffusion in plasmas, see [1], for an application to transport in magnetized plasmas, see [2].

3 Special Case: Homogeneous Flux Scheme for a Coupled Diffusion-Reaction System

As a special case, we outline the derivation of the homogeneous flux scheme for a system of diffusion-reaction equations containing a linear source term. The derivation of the complete flux scheme for a generic system of advection-diffusion-reaction equations is covered in [3].

Consider the following system of one-dimensional, stationary conservation laws for the vector of unknowns $\boldsymbol{y}$, i.e.,

$$\boldsymbol{J}' = \boldsymbol{C}\boldsymbol{y} + \boldsymbol{s}(\boldsymbol{y}), \quad \boldsymbol{J} = -\mathcal{E}\boldsymbol{y}', \tag{17}$$

where the prime ($'$) denotes differentiation w.r.t. x. $\boldsymbol{J}$ is the (diffusive) mass flux vector and $\mathcal{E}$ the mass flux diffusion matrix, which we assume to be positive definite. Note that the source in the right hand side contains a linear term $\boldsymbol{C}\boldsymbol{y}$, typically resulting from a linearisation about an equilibrium solution, and a nonlinear part $\boldsymbol{s}(\boldsymbol{y})$. For discretisation of (17) we employ the (cell-centred) finite volume method. We introduce an equidistant spatial grid $\{x_j\}$ with grid spacing Δx and cover the domain with a disjunct set of control volumes $V_j = [x_{j-1/2}, x_{j+1/2}]$ with $x_{j\pm 1/2} = \frac{1}{2}(x_j + x_{j\pm 1})$. Integrating the first equation in (17) over V_j and applying the midpoint rule for the right hand side, we obtain the discrete conservation law for the numerical approximation $\boldsymbol{y}_j \approx \boldsymbol{y}(x_j)$, i.e.,

$$\boldsymbol{J}_{j+1/2} - \boldsymbol{J}_{j-1/2} = \Delta x\big(\boldsymbol{C}\boldsymbol{y}_j + \boldsymbol{s}(\boldsymbol{y}_j)\big), \tag{18}$$

where $\boldsymbol{J}_{j+1/2}$ denotes the numerical flux at the cell interface $x = x_{j+1/2}$ approximating $\boldsymbol{J}(x_{j+1/2})$.

In the following we assume that $\mathcal{E}$ and $\boldsymbol{C}$ are piecewise constant, i.e., $\mathcal{E} = \mathcal{E}(\boldsymbol{y}_{j+1/2})$ and $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{y}_{j+1/2})$ for $x_j \le x \le x_{j+1}$. Let $\boldsymbol{A} = \mathcal{E}^{-1}\boldsymbol{C}$. The key idea is to compute the numerical flux $\boldsymbol{J}_{j+1/2}$ from the local BVP

$$-\boldsymbol{J}' + \boldsymbol{C}\boldsymbol{y} = \mathcal{E}\big(\boldsymbol{y}'' + \boldsymbol{A}\boldsymbol{y}\big) = \boldsymbol{0}, \quad x_j < x < x_{j+1}, \tag{19a}$$

$$\boldsymbol{y}(x_j) = \boldsymbol{y}_j, \quad \boldsymbol{y}(x_{j+1}) = \boldsymbol{y}_{j+1}, \tag{19b}$$

ignoring the nonlinear source term $\boldsymbol{s}(\boldsymbol{y})$, however, including the linear term $\boldsymbol{C}\boldsymbol{y}$ to account for exponential and oscillatory solution components. From this BVP we determine a representation of the solution $\boldsymbol{y}(x)$ on the interval $[x_j, x_{j+1}]$ and subsequently compute the numerical flux according to $\boldsymbol{J}_{j+1/2} = -\mathcal{E}\boldsymbol{y}'(x_{j+1/2})$.

In the derivation that follows, we need to evaluate several matrix functions. Therefore we compute the eigenvalues λ_k and corresponding eigenvectors $\boldsymbol{v}_k$ of the matrix $\boldsymbol{A}$, which satisfy the eigenvalue problem

$$\big(\boldsymbol{C} - \lambda\mathcal{E}\big)\boldsymbol{v} = \boldsymbol{0}. \tag{20}$$

We assume that $\boldsymbol{A}$ has a complete set of eigenvectors, thus $\boldsymbol{A} = \boldsymbol{V}\boldsymbol{\Lambda}\boldsymbol{V}^{-1}$, where $\boldsymbol{\Lambda}$ and $\boldsymbol{V}$ are defined by

$$\boldsymbol{\Lambda} = \mathrm{diag}(\lambda_1, \lambda_2, \ldots, \lambda_m\big), \quad \boldsymbol{V} = \big(\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_m\big). \tag{21}$$

Note that by assumption $\boldsymbol{V}$ is regular. Multiplying the ODE (19a) with the inverse $\boldsymbol{V}^{-1}\mathcal{E}^{-1}$, we obtain the decoupled system $\boldsymbol{\psi}'' + \boldsymbol{\Lambda}\boldsymbol{\psi} = \boldsymbol{0}$ for the variable $\boldsymbol{\psi} = \boldsymbol{V}^{-1}\boldsymbol{y}$, or written componentwise

$$\psi_k'' + \lambda_k\psi_k = 0, \quad (k = 1, 2, \ldots, m). \tag{22}$$

The solutions of (22) can be written as

$$\psi_k(x) = \alpha_k \mathrm{e}^{\nu_k\omega_k x} + \beta_k \mathrm{e}^{-\nu_k\omega_k x}, \tag{23a}$$

$$\omega_k = \sqrt{|\lambda_k|}, \quad \nu_k = \begin{cases} 1 & \text{if} \quad \lambda_k < 0, \\ \mathrm{i} & \text{if} \quad \lambda_k > 0, \end{cases} \qquad (k = 1, 2, \ldots, m). \tag{23b}$$

Next, assembling the solution components in the vector $\boldsymbol{\psi} = (\psi_k)$, we find

$$\boldsymbol{\psi}(x) = \mathrm{e}^{x\boldsymbol{\Omega}}\boldsymbol{\alpha} + \mathrm{e}^{-x\boldsymbol{\Omega}}\boldsymbol{\beta}, \quad \boldsymbol{\Omega} = \mathrm{diag}\big(\nu_k\omega_k\big) \tag{24}$$

with $\boldsymbol{\alpha} = (\alpha_k)$ and $\boldsymbol{\beta} = (\beta_k)$. Transforming back to $\boldsymbol{y} = \boldsymbol{V}\boldsymbol{\psi}$ and applying the boundary condition (19b) we can determine the solution of (19). Introducing the matrix $\boldsymbol{B} = \boldsymbol{V}\boldsymbol{\Omega}\boldsymbol{V}^{-1}$, we obtain the representation

$$\boldsymbol{y}(x) = \big(\sinh(\Delta x\boldsymbol{B})\big)^{-1}\big(-\sinh((x-x_{j+1})\boldsymbol{B})\boldsymbol{y}_j + \sinh((x-x_j)\boldsymbol{B})\boldsymbol{y}_{j+1}\big), \tag{25a}$$

where the matrix function $\sinh(z\boldsymbol{B})$ is defined by

$$\sinh(z\boldsymbol{B}) = \boldsymbol{V}\sinh(z\boldsymbol{\Omega})\boldsymbol{V}^{-1}, \quad \sinh(z\boldsymbol{\Omega}) = \mathrm{diag}\big(\sinh(\nu_k\omega_k z)\big). \tag{25b}$$

Alternatively, $\sinh(z\boldsymbol{B})$ can be evaluated as $\sinh(z\boldsymbol{B}) = \frac{1}{2}(\mathrm{e}^{z\boldsymbol{B}} - \mathrm{e}^{-z\boldsymbol{B}})$. Finally, by straightforward differentiation we find for the numerical flux

$$\begin{aligned}\boldsymbol{J}_{j+1/2} &= -\frac{1}{\Delta x}\mathcal{E}\big(\tfrac{1}{2}\Delta x\boldsymbol{B}\big)\Big(\sinh\big(\tfrac{1}{2}\Delta x\boldsymbol{B}\big)\Big)^{-1}\big(\boldsymbol{y}_{j+1} - \boldsymbol{y}_j\big)\\ &= -\frac{1}{\Delta x}\mathcal{E}\big(\mathrm{Sinhc}(\tfrac{1}{2}\Delta x\boldsymbol{B})\big)^{-1}\big(\boldsymbol{y}_{j+1} - \boldsymbol{y}_j\big),\end{aligned} \tag{26}$$

referred to as the homogeneous flux approximation scheme, where the Sinhc function is defined as $\mathrm{Sinhc}(z) = \sinh(z)/z$.

4 Numerical Example

As an example we simulate the dissociation of NO in a nitrogen-oxygen mixture at low temperature, typically $T = 300\mathrm{K}$. We assume there is no flow and a uniform pressure of $p = 2 \times 10^4\mathrm{Pa}$. Thus, consider a mixture of N_2, O_2 and NO. Since the temperature is low, we consider one single reaction $2\mathrm{NO} \rightarrow \mathrm{N}_2 + \mathrm{O}_2$. Moreover, we assume the mixture is confined between two walls, located at $x = 0$ and $x = 1\mathrm{m}$, such that in the center of the physical domain (vessel) the multi-species diffusion process can be considered one-dimensional. The governing system of equations for the mass fractions $\boldsymbol{y} = \big(y_{\mathrm{N}_2}\; y_{\mathrm{O}_2}\; y_{\mathrm{NO}}\big)^{\mathrm{T}}$ then reads $-\big(\mathcal{E}(\boldsymbol{y})\boldsymbol{y}'\big)' = \boldsymbol{C}(\boldsymbol{y})\boldsymbol{y}$, where $\mathcal{E}(\boldsymbol{y})$ is defined as in (16b) and where $\boldsymbol{C} = \boldsymbol{C}(\boldsymbol{y})$ is defined by

$$\boldsymbol{C}(\boldsymbol{y}) = k(T)\rho^2\frac{y_{\mathrm{NO}}}{m_{\mathrm{NO}}^2}\,\mathrm{diag}(\boldsymbol{m})\begin{pmatrix}0 & 0 & 1\\ 0 & 0 & 1\\ 0 & 0 & -2\end{pmatrix}, \tag{27a}$$

$$k(T) = A(T/K)^q\mathrm{e}^{-E_{\mathrm{a}}/(RT)}, \tag{27b}$$

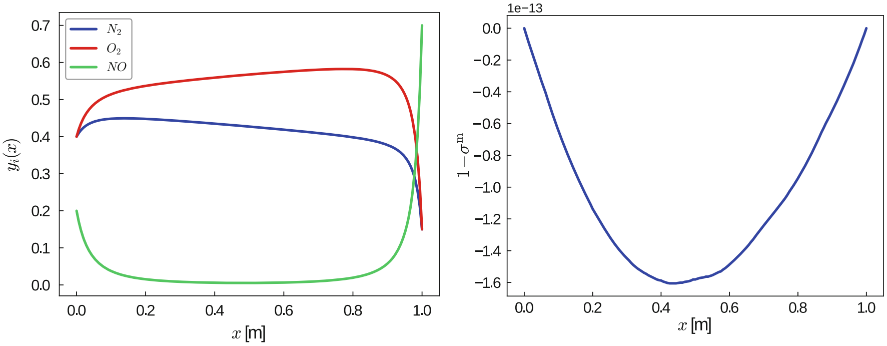


Fig. 1 Dissociation of NO: mass fractions (left) and mass deficit (right)

where $k = k(T)$ is the reaction rate, $\boldsymbol{m} = \left(m_{\mathrm{N_2}}\, m_{\mathrm{O_2}}\, m_{\mathrm{NO}}\right)^{\mathrm{T}}$ the vector of the component masses and R the gas constant. Parameter values are $A = 1.83 \times 10^{-17}\mathrm{m}^3/\mathrm{s}$, $q = 0.3$ and $E_{\mathrm{a}} = 52.5\mathrm{kJ/mol}$.

For space discretisation we employ the finite volume method with the numerical flux vector given by (26). Our numerical results are shown in Fig. 1 computed with grid size $\Delta x = (1/160)\mathrm{m}$. In the left figure we present the mass fractions. Observe that the solution contains boundary layers at both ends of the interval, and that the mass fraction of NO is rapidly decreasing towards the centre of the interval. In the right figure, we present mass deficit $1 - \sigma^{\mathrm{m}}$, where σ^{m} is the sum of all mass fraction, which should obviously sum to 1. Clearly, mass is conserved almost up to the machine precision.

Finally, in convergence tests we observed second order convergence of the scheme. Moreover, the scheme turned out to be slightly more accurate than the standard central difference scheme.

References

1. K.S.C. Peerenboom, J. van Dijk, W.J. Goedheer and J.J.A.M. van der Mullen, *On the ambipolar constraint in multi-component diffusion problems*, J. Comp. Phys. **230**(10), 3651–3655 (2011)
2. K.S.C. Peerenboom, J. van Dijk, W.J. Goedheer, G. Degrez and J.J.A.M. van der Mullen, *A finite volume model for multi-component diffusion in magnetically confined plasmas*, J. Phys. D: Appl. Phys. **44**(19) 194006 (2011)
3. J.H.M. ten Thije Boonkkamp, J. van Dijk, L. Liu, K.S.C. Peerenboom, *Extension of the complete flux scheme to systems of conservation laws*, J. Sci. Comput. **53**, 552–568 (2012)
4. L. Liu, J. van Dijk, J.H.M. ten Thije Boonkkamp, D.B. Mihailova and J.J.A.M. van der Mullen, *The complete flux scheme—Error analysis and application to plasma simulation*, Journal of Computational and Applied Mathematics, **250**, 229–243 (2013)
5. P.M.J. Koelman, S. Heijkers, S. Tadayon Mousavi, W.A.A.D. Graef, D.B. Mihailova, T. Kozak, A. Bogaerts and J. van Dijk, *A Comprehensive Chemical Model for the Splitting of CO2 in Non-Equilibrium Plasmas*, Plasma Processes and Polymers, **15**(4–5) 1600155 (2017)

6. Vincent Giovangigli, *Mass conservation and singular multicomponent diffusion algorithms*, IMPACT of Computing in Science and Engineering, **2**(1), 73–97 (1990)
7. J.D. Ramshaw and C.H. Chang, Plasma Chem. Plasma Proc., **12**(3) 299 (1992)

PDE-Constrained Optimization: Optimal control with L_1-Regularization, State and Control Box Constraints

Ivo Dravins and Maya Neytcheva

Abstract We present a method for solving optimal control problems constrained by a partial differential equation, where we simultaneously impose sparsity-promoting L_1-regularization on the control as well as box constraints on both the control and the state. We focus on numerical implementation aspects and on preconditioners used when solving the arising linear systems.

1 Introduction

We consider a distributed optimal control problem where the constraining PDE is Poisson's equation although the presented methods are also applicable to related PDEs such as the convection-diffusion equation. The main novelty of this paper is the combination of several additional constraints, namely, L_1 regularization on the control in order to promote sparsity and box constraints on both the control and the state. In earlier related studies these extra constraints have been considered separately but, to the authors knowledge, they have not been combined except briefly mentioned in [1]. The L_1-regularization and the box constraints on the control are implemented following [2] while the state-constraints are implemented by Moreau-Yosida forcing terms [3–6]. For completeness and to aid reproducibility, a demonstration code is made available at https://github.com/dravinsi/demoPDEOPT.

I. Dravins (✉) · M. Neytcheva
Department of Information Technology, Uppsala University, Uppsala, Sweden
e-mail: ivo.dravins@it.uu.se; maya.neytcheva@it.uu.se

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_31

2 Problem Formulation and Optimality System

We consider the task to minimize the cost functional

$$\begin{aligned}\mathcal{J}(y,u) = {} & \tfrac{1}{2}\|y-y_d\|^2_{L^2(\Omega)} + \tfrac{\alpha}{2}\|u\|^2_{L^2(\Omega)} + \beta\|u\|_{L^1(\Omega)} \\ & + \tfrac{1}{2\varepsilon}\|\max\{0,y-y_b\}\|^2_{L^2(\Omega)} + \tfrac{1}{2\varepsilon}\|\min\{0,y-y_a\}\|^2_{L^2(\Omega)},\end{aligned}$$

such that: $-\Delta y = f + u$ in Ω, $y = g$ on $\partial\Omega$ and $u_a \le u \le u_b$ a.e. in Ω, where $u_a < 0 < u_b$. Following [2] and [3], the first order necessary conditions in strong form read:

$$\begin{aligned}
&(\mathbf{1s})\ y + \tfrac{1}{\varepsilon}\max\{0,y-y_b\} + \tfrac{1}{\varepsilon}\min\{0,y-y_a\} - \Delta p = y_d \text{ in } \Omega\ ;\ \ p = 0 \text{ on } \partial\Omega \\
&(\mathbf{2s})\ -\Delta y - u = f \text{ in } \Omega;\ \ y = g \text{ on } \partial\Omega \\
&(\mathbf{3s})\ \alpha u - p + \lambda = 0 \text{ in } \Omega \\
&(\mathbf{4s})\ u - \max\{0, u + c_1(\lambda-\beta)\} - \min\{0, u + c_1(\lambda+\beta)\} + \max\{0, c_2(u-u_b) + \\
&\qquad c_1(\lambda-\beta)\} + \min\{0, c_2(u-u_a) + c_1(\lambda+\beta)\} = 0\ \text{ in } \Omega\ ;\ \forall c_{1,2} > 0.
\end{aligned}$$

The presence of conditional terms in the above system result in a non-linear problem. These terms are to be understood point-wise (cf., e.g., [7]), thus these are checked in each individual point. In a finite element setting this means we check the conditions on every mesh node. The sets of points where the various conditions are not fulfilled, are referred to as the "active sets". The constant c_2 in (**4s**) is not present in [2], it is added to facilitate and stabilize the convergence of the non-linear problem for a broader range of parameter values. To see why this can be added, it is useful to write out the possible outcomes of (**4s**) and their impact on the control:

$$(\mathbf{4s}) \Leftrightarrow \begin{cases}
(\mathbf{4s1})\ u = 0\ ;\ \text{Sparsity enforcing: } u = 0, \\
(\mathbf{4s2})\ \ -c_1(\lambda-\beta) = 0\ ;\ \text{Non-zero positive: } 0 < u < u_b, \\
(\mathbf{4s3})\ \ -c_1(\lambda+\beta) = 0\ ;\ \text{Non-zero negative: } u_a < u < 0, \\
(\mathbf{4s4})\ c_2(u-u_b) = 0\ ;\ \text{Positive box-constraint: } u = u_b, \\
(\mathbf{4s5})\ c_2(u-u_a) = 0\ ;\ \text{Negative box-constraint: } u = u_a.
\end{cases}$$

The latter shows that c_2 can be added and chosen as any positive number, indeed we could also multiply the u-term in the sparsity-enforcing case by a third constant. As we are interested in controlling the relations between the terms, it suffices with two constants. The values of the constants in (**4s**) regulate the rules for classifying a point as *active* or *inactive*, so while they do not appear explicitly in the linear system, they are of importance for the non-linear behavior of the problem. We next use (**3s**) to reduce the system by $u = \frac{1}{\alpha}(p-\lambda)$ and we are left with y, p and λ. To reduce the system further, we can use (**4s**) to eliminate λ, though this reduction depends on the current non-linear iterate. After solving the reduced linear system, λ needs to be recovered by back-substitution in order to generate the next linear system, necessitating that we keep track of the active sets (**4s1**) through (**4s5**). As the unknowns that appear in (**1s**) are unaffected by the substitutions, we need only to consider the resulting cases for (**2s**) $-\Delta y - \frac{1}{\alpha}(p - \lambda) = f$, namely,

$$\begin{cases} (\mathbf{2s1}) \; -\Delta y = f \;;\; \text{Sparsity enforcing: } u = \frac{1}{\alpha}(p-\lambda) = 0, \\ (\mathbf{2s2}) \; -\Delta y - \frac{1}{\alpha}p = f - \frac{\beta}{\alpha} \;;\; \text{Non-zero positive: } 0 < u < u_b \;;\; \lambda = \beta, \\ (\mathbf{2s3}) \; -\Delta y - \frac{1}{\alpha}p = f + \frac{\beta}{\alpha} \;;\; \text{Non-zero negative: } u_a < u < 0 \;;\; \lambda = -\beta, \\ (\mathbf{2s4}) \; -\Delta y = f + u_b \;;\; \text{Positive box-constraint: } u = \frac{1}{\alpha}(p-\lambda) = u_b, \\ (\mathbf{2s5}) \; -\Delta y = f + u_a \;;\; \text{Negative box-constraint: } u = \frac{1}{\alpha}(p-\lambda) = u_a. \end{cases}$$

In discrete (FEM) notation with K denoting the stiffness matrix and M - the mass matrix, (**2s**) becomes

$$(\mathbf{2s}) \quad K\boldsymbol{y} - \frac{1}{\alpha}I_pM\boldsymbol{p} = M(\boldsymbol{f} + \boldsymbol{b}_u),$$

where I_p is a diagonal matrix with ones in the rows corresponding to nodes where conditions (**2s2**) and (**2s3**) are active, and zeros elsewhere. As we zero out rows of the mass matrix, to retain symmetry, it is useful to work with the lumped mass matrix. The vector $\boldsymbol{b}_u$ accounts for the conditional terms in the right-hand-side of the cases (**2s1**) through (**2s5**). For brevity we denote $M_p^{(i)} = I_pM$. Similarly, we write (**1s**) in discrete form as

$$(\mathbf{1s}) \quad (M + \frac{1}{\varepsilon}I_yM)\boldsymbol{y} + K\boldsymbol{p} = M(\boldsymbol{y}_d + \boldsymbol{b}_y),$$

where I_y is a diagonal matrix with ones in the rows corresponding to nodes where any of the two min/max conditions are active and zeros elsewhere. The conditional terms in the right-hand-side are in the vector $\boldsymbol{b}_y$. We denote $M_y^{(a)} = I_yM$ and combine (**1s**) and (**2s**) to obtain the linear system

$$\begin{bmatrix} M + \frac{1}{\varepsilon}M_y^{(a)} & K \\ K & -\frac{1}{\alpha}M_p^{(i)} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \boldsymbol{p} \end{bmatrix} = \begin{bmatrix} M(\boldsymbol{y}_d + \boldsymbol{b}_y) \\ M(\boldsymbol{f} + \boldsymbol{b}_u) \end{bmatrix}. \tag{1}$$

This is the linear system we solve in each non-linear step. We then use **p** together with the active sets (**2s1**) through (**2s5**) to recover λ and generate the system for the next Newton step. In order to reduce the impact of ε and α on the conditioning of the system matrix, we rescale the system as follows. We first introduce $\widehat{\boldsymbol{p}} = -\sqrt{\alpha}\boldsymbol{p}$ and multiply the bottom block-row by $\sqrt{\alpha}$:

$$\begin{bmatrix} M + \frac{1}{\varepsilon}M_y^{(a)} & -\sqrt{\alpha}K \\ \sqrt{\alpha}K & M_p^{(i)} \end{bmatrix} \begin{bmatrix} \boldsymbol{y} \\ \widehat{\boldsymbol{p}} \end{bmatrix} = \begin{bmatrix} M(\boldsymbol{y}_d + \boldsymbol{b}_y) \\ \sqrt{\alpha}M(\boldsymbol{f} + \boldsymbol{b}_u) \end{bmatrix}.$$

Note that $M + \frac{1}{\varepsilon}M_y^{(a)} = (I + \frac{1}{\varepsilon}I_y)M$. Since $\mathbb{I}_y = (I + \frac{1}{\varepsilon}I_y)$ is a positive definite diagonal matrix, we can use both its square-root and inverse. We introduce $\widehat{\boldsymbol{y}} = \mathbb{I}_y^{\frac{1}{2}}\boldsymbol{y}$ and multiply the top block-row by $\mathbb{I}_y^{-\frac{1}{2}}$ to obtain (assuming lumped mass matrix)

$$\begin{bmatrix} M & -\mathbb{I}_y^{-\frac{1}{2}}\sqrt{\alpha}K \\ \sqrt{\alpha}K\mathbb{I}_y^{-\frac{1}{2}} & M_p^{(i)} \end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{y}} \\ \widehat{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix}\mathbb{I}_y^{-\frac{1}{2}}M(\boldsymbol{y}_d + \boldsymbol{b}_y) \\ \sqrt{\alpha}M(\boldsymbol{f} + \boldsymbol{b}_u)\end{bmatrix}.$$

Denoting $\widehat{K} = \sqrt{\alpha}K\mathbb{I}_y^{-\frac{1}{2}}$ we obtain the form of the system, used in the sequel,

$$A\begin{bmatrix}\widehat{\boldsymbol{y}} \\ \widehat{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix} M & -\widehat{K}^T \\ \widehat{K} & M_p^{(i)} \end{bmatrix}\begin{bmatrix}\widehat{\boldsymbol{y}} \\ \widehat{\boldsymbol{p}}\end{bmatrix} = \begin{bmatrix}\mathbb{I}_y^{-\frac{1}{2}}M(\boldsymbol{y}_d + \boldsymbol{b}_y) \\ \sqrt{\alpha}M(\boldsymbol{f} + \boldsymbol{b}_u)\end{bmatrix}. \tag{2}$$

3 Preconditioning

For solving systems with the matrix A in (2) we suggest the preconditioner

$$P = \begin{bmatrix} M & -\widehat{K}^T \\ \widehat{K} + M_p^{(i)} & \widehat{K} + M_p^{(i)} \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \widehat{K} + M_p^{(i)} \end{bmatrix}\begin{bmatrix} M & -\widehat{K}^T \\ I & I \end{bmatrix} = P_1P_2. \tag{3}$$

Solutions with P consist of solving systems with the factors P_1 and P_2. For P_1 we need to solve for the lower diagonal block

$$(\sqrt{\alpha}K\mathbb{I}_y^{-\frac{1}{2}} + M_p^{(i)})\boldsymbol{x} = \boldsymbol{b} \Leftrightarrow (\sqrt{\alpha}K + M_p^{(i)}\mathbb{I}_y^{\frac{1}{2}})\mathbb{I}_y^{-\frac{1}{2}}\boldsymbol{x} = \boldsymbol{b}.$$

As M is diagonal, the matrix $\sqrt{\alpha}K + M_p^{(i)}\mathbb{I}_y^{\frac{1}{2}}$ is symmetric positive definite (spd) and can be efficiently solved for, using the conjugate gradient (CG) method, preconditioned by an algebraic multigrid (AMG). To solve for $P_2(\boldsymbol{x}_1, \boldsymbol{x}_2)^T = (\boldsymbol{b}_1, \boldsymbol{b}_2)^T$ we use the bottom block row to reduce $\boldsymbol{x}_2 = \boldsymbol{b}_2 - \boldsymbol{x}_1$. Inserting $\boldsymbol{x}_2$ into the top block row we obtain

$$M\boldsymbol{x_1} - \widehat{K}^T(\boldsymbol{b}_2 - \boldsymbol{x}_1) = \boldsymbol{b}_1 \Leftrightarrow (\widehat{K}^T + M)\boldsymbol{x}_1 = \boldsymbol{b}_1 + \widehat{K}^T\boldsymbol{b}_2,$$

which we reformulate into an spd system

$$(\sqrt{\alpha}K + \mathbb{I}_y^{\frac{1}{2}}M)\boldsymbol{x}_1 = \mathbb{I}_y^{\frac{1}{2}}(\boldsymbol{b}_1 + \widehat{K}^T\boldsymbol{b}_2)$$

and also solve by an AMG-preconditioned CG. The total cost of solving with P is then two solutions of spd systems of half the dimension of the original system,

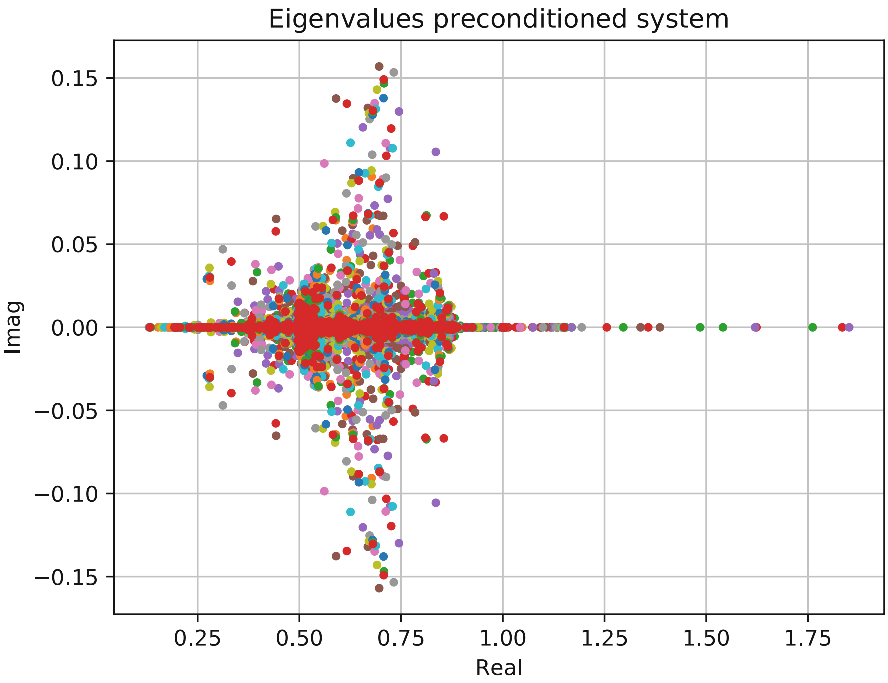


Fig. 1 Eigenvalues of the 14 linear systems we solve in Problem 1, $h = 2^{-5}$, $\alpha = 10^{-6}$

one matrix-vector multiplication with the matrix K and applying a few diagonal scalings.

An eigenvalue analysis of the preconditioned system $P^{-1}A$ is beyond the scope of this paper. To illustrate the behavior, in Fig. 1 we provide a plot of the eigenvalues of the preconditioned systems for all linear systems arising during the nonlinear iteration when solving Problem 1 (Sect. 5). For all tested parameter sets the eigenvalues of $P^{-1}A$ lie broadly in the same range.

4 Solvers, Tolerances and Termination Criteria

The conditional terms in the optimality system lead to a non-linear problem. Specifically we need to find x such that $f(x) = A(x)x - b(x) = 0$, where $A(x)$ is a piece-wise linear function. We thus start with some initial guess $x^{(0)}$ and generate the Jacobian and the right-hand-side. The next step is obtained by solving $A(x^{(0)})x^{(1)} = b(x^{(0)})$, i.e., we are using a semi-smooth Newton method. An important consideration when solving these problems is the choice of termination criteria, both for the linear and the non-linear solver. For the non-linear solver, a natural termination criterion is that we stop when the active sets stop changing, referred to as *set-convergence* and used in the tests. We observe however, that a stopping criterion for the nonlinear problem, based only on set-convergence may have to be combined with other measures, to avoid cycling of a few points between the active and inactive sets. There is also a certain interplay between the stopping tolerances for the linear and non-linear solvers. In general, the required tolerance for the linear solver to achieve set convergence is problem dependent. The semi-smooth Newton method is sensitive to the initial guess. A good initial guess can be found by utilizing mesh hierarchy, solving the problem on the coarser grids and interpolating the solution to a finer grid, to become the new initial guess. Alternatively, one can solve the problem with a larger α, gradually decreasing it, using the previous solution as initial guess.

As an outer solver for systems with the matrix A in (2) we use the Generalized Conjugate Residual (GCR) method, preconditioned from the right using P in (3). For the stopping criteria we use the relative preconditioned residual $\|P^{-1}(Ax - b)\|/\|P^{-1}b\| <$ tol. In the first step in each non-linear iteration we use a zero-initial guess for the GCR, in subsequent steps we use the solution from the previous step. Finding a stopping criterion that is both efficient and sufficient for a range of different parameters, problems and discretizations remains a challenge. For earlier work we refer, e.g., to [8, 9] and the references therein.

5 Numerical Illustrations

We illustrate the performance of the involved nonlinear and linear solvers with some numerical experiments. The tests are performed in `Julia` [10]. Common for all tests are: $\varepsilon = \alpha^{1/4}$, $c_1 = c_2 = 1/\alpha$, tolerance for the inner (CG) solver 10^{-6} and tolerance for the outer linear solver 10^{-6}, based on the norm of the preconditioned relative residual. We use a mesh hierarchy to generate initial guesses. On the coarsest grid, we start with an initial guess of all ones, for each subsequent grid the initial guess is obtained by interpolating the solution from the previous grid. For these tests we consider the non-linear iteration to be converged when all active sets stop changing, i.e., set convergence is achieved. Boundary state values are equal to the desired state. For all test problems we choose $\Omega = [0, 1]^2$, discretized using triangular finite elements and linear basis functions. The mesh is generated with `FEniCS` [11]. We consider the following three problems:

Problem 1 $y_d(x_1, x_2) = \sin(2\pi x_1)\sin(2\pi x_2)e^{2x_1}/6$, $-30 \le \boldsymbol{u} \le 30$, $y_a = -0.2$, $y_b = 0.3$, $\beta = 10^{-4}$.

Problem 2 $y_d(x_1, x_2) = -\exp(|x - \frac{1}{2}| + |y - \frac{1}{2}|)$, $-\infty < \boldsymbol{u} \le 30$, $y_a = -\infty$, $y_b = -1.3$, $\beta = 10^{-4}$.

Problem 3 $y_d(x_1, x_2) = |\sin(2\pi x_1)\sin(2\pi x_2)|$, $-100 \le \boldsymbol{u} \le 75$, $y_a = -\infty$, $y_b = 0.75$, $\beta = 0.5 \cdot 10^{-4}$.

Table 1 Numerical tests: (M) denotes that we reached the maximum number of iterations

		Problem 1		Problem 2		Problem 3	
		Nonlin. iter.	Av. lin. iter.	Nonlin. iter.	Av. lin. iter.	Nonlin. iter.	Av. lin. iter.
Level ↓	$h = 2^{-n}$	$\alpha = 10^{-6}$					
1	5	14	16	7	10	6	10
2	6	9	18	3	10	4	11
3	7	4	19	3	11	3	12
4	8	4	19	3	11	3	12
5	9	4	20	3	12	3	13
6	10	4	21	3	13	3	13
Level ↓	$h = 2^{-n}$	$\alpha = 10^{-7}$					
1	5	41(M)	38	8	14	41(M)	16
2	6	41(M)	30	6	18	22	16
3	7	41(M)	29	6	18	8	18
4	8	14	31	4	18	4	19
5	9	8	31	5	18	4	19
6	10	5	32	4	18	3	20
		(a)		(b)		(c)	

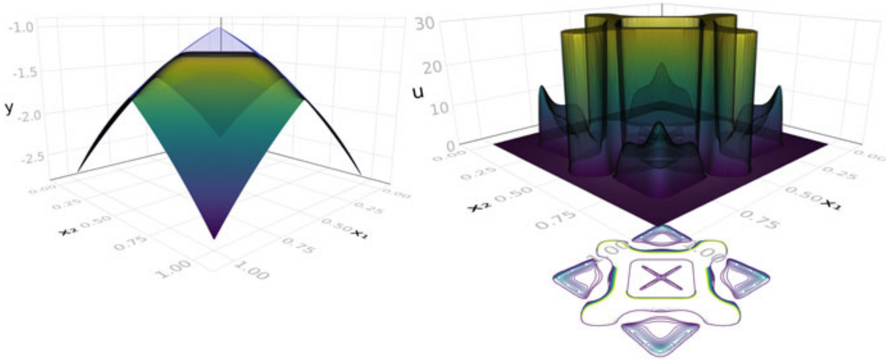


Fig. 2 Problem 2: $\alpha = 10^{-7}$: Left: achieved state, *blue*—desired state. Right: control with contour plot below

Iteration counts are listed in Table 1, plots of the solutions to Problems 2 and 3 are displayed in Figs. 2 and 3.

We note that for Problem 1 and 3 with $\alpha = 10^{-7}$ the non-linear iterations do not converge on the coarser grids. This seems to be due to the discretization not being able to resolve the problem well enough. We also note that the average linear iterations increase as α decreases. Examples not included here show that the iterations depend on the value of ε - both the linear and the nonlinear solvers encounter difficulties as $\varepsilon \to 0$. Alternative methods for similar problems in

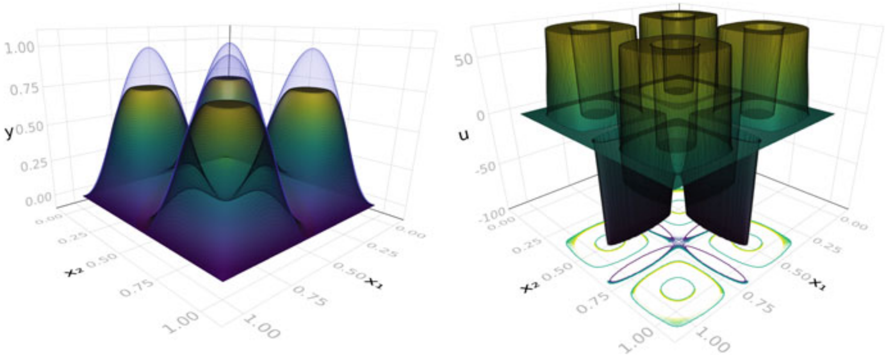


Fig. 3 Problem 3: $\alpha = 10^{-7}$: Left: achieved state, *blue*—desired state. Right: control with contour plot below

a different setting are considered in [12]. To summarize, the interplay between α, ε, c_1, c_2 and h is problem-dependent and needs further study.

6 Concluding Remarks

We have presented a method for solving a distributed control optimal control problem with box-constraints on the control and the state combined with sparsity enforcing L_1 regularization. The method allows for efficient solutions to many problems but the approach is problem dependent and difficulties remain in choosing efficient general stopping criteria.

Acknowledgments This work was supported by VR Grant 2017-03749 *Mathematics and numerics in PDE-constrained optimization problems with state and control constraints*, 2018–2022.

References

1. Dravins I., Neytcheva M.: PDE-Constrained Optimization: Matrix Structures and Preconditioners. Lirkov I., Margenov S. (eds) Large-Scale Scientific Computing. LSSC 2019. Lecture Notes in Comput. Sci. 11958, 315–323 (2020)
2. Stadler G.: Elliptic optimal control problems with L1-control cost and applications for the placement of control devices. Comput. Optim. Appl. 44, 159–181 (2009)
3. Herzog R., Sachs E. W.: Preconditioned conjugate gradient method for optimal control problems with control and state constraints. SIAM J. Matrix. Anal. Appl. 31, 2291–2317 (2010)
4. Pearson J.W., Stoll M., Wathen A.J.: Preconditioners for state–constrained optimal control problems with Moreau–Yosida penalty function. Numer. Lin. Alg. Appl. 21, 81–97 (2014)

5. Hintermüller M., Hinze M.: Moreau-Yosida regularization in state constrained elliptic control problems: error estimates and parameter adjustment. SIAM J. Numer. Anal. 47, 1666–1683 (2009)
6. Axelsson O., Neytcheva M., Ström A.: An efficient preconditioning method for state box-constrained optimal control problems. J. Numer. Math. 26(4), 185–207 (2018)
7. Ito K., Kunisch K.: Semi-smooth Newton methods for state-constrained optimal control problems. Systems & Control Letters 5, 221–228 (2003)
8. Axelsson O., Kaporin I.: On a class of nonlinear equation solvers based on the residual norm reduction over a sequence of affine subspaces. SIAM J. Sci. Comput. 16(1), 228–249 (1995)
9. Axelsson O., On mesh independence and Newton-type methods. Appl. Math. 39, 249–265 (1993)
10. Bezanson J., Edelman A., Karpinski S., Shah V.: Julia: A Fresh Approach to Numerical Computing. SIAM Review 50, 65–98 (2017)
11. Alnaes M. S., Blechta J., Hake J., Johansson A., Kehlet B., Logg A., Richardson C., Ring J., Rognes M. E., Wells G. N.: The FEniCS Project Version 1.5. Arch. Num. Soft. (2015)
12. Pearson J.W., Porcelli M., Stoll M.: Interior Point Methods and Preconditioning for PDE-Constrained Optimization Problems Involving Sparsity Terms. arXiv: 1806.05896 (2019)

A Time-Simultaneous Multigrid Method for Parabolic Evolution Equations

J. Dünnebacke, S. Turek, P. Zajac, and A. Sokolov

Abstract We present a time-simultaneous multigrid scheme for parabolic equations that is motivated by blocking multiple time steps together. The resulting method is closely related to multigrid waveform relaxation and is robust with respect to the spatial and temporal grid size and the number of simultaneously computed time steps. We give an intuitive understanding of the convergence behavior and briefly discuss how the theory for multigrid waveform relaxation can be applied in some special cases. Finally, some numerical results for linear and also nonlinear test cases are shown.

1 Motivation

Modern high performance computing systems feature a growing number of processors and massively parallel co-processors, e.g. GPUs, while the performance of each processor does barely increase or even stagnates. To efficiently use such supercomputers the algorithms have to be massively parallel. The usual time stepping approach to solve time dependent partial differential equations (PDEs) is inherently sequential and does only allow spatial parallelization. If we want to simulate problems with a relatively low number of spatial degrees of freedoms (DOFs), we can only use a certain degree of parallelism, while the number of time steps may be very high due to a long time frame or short time steps. These simulations can not be sped up even if there is more parallel compute power available.

J. Dünnebacke (✉) · S. Turek · P. Zajac · A. Sokolov
Institute of Applied Mathematics (LS III), TU Dortmund University, Dortmund, Germany
e-mail: jonas.duennebacke@math.tu-dortmund.de; stefan.turek@math.tu-dortmund.de; peter.zajac@math.tu-dortmund.de; andriy.sokolov@math.tu-dortmund.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_32

The parallel scalability is limited because the communication between the processes will outweigh the actual computation time, if too many processes are used. It is important to note that usually the main cost of the communication stems from latency and not from limited bandwidth. If the number of communication operations is reduced by communicating more data at once, the scaling behavior can be improved, so that more processors can be used efficiently in such simulations (see Fig. 3). To achieve this we have to abandon the sequential time stepping.

There already exists a lot of work on time parallel integration. Many methods are based on integrating ODEs parallel in time. The most prominent examples of this group are Parareal [6] and its variants. Another group of time parallel methods is based on solving a global discrete system with multigrid methods. The first parabolic multigrid was developed by Hackbusch [3]. Other representers of such schemes are the one developed by Horton and Vandewalle [4] as well as the recent variant by Gander and Neumüller [2]. The method our approach resembles the most is multigrid waveform relaxation which was first published by Lubich and Ostermann [7]. For a more complete overview on parallel in time methods, we refer to [1].

2 Time-Simultaneous Multigrid

In the following, we propose a multigrid scheme that computes many time steps simultaneously but relies solely on spatial parallelization. Here, we start with a second order parabolic evolution equation

$$\partial_t u(x,t) + \mathcal{L}(t)u(x,t) = f(x,t) \quad (x,t) \in \Omega \times (0,T) \tag{1}$$

with suitable initial and boundary conditions. $\mathcal{L}(t)$ is a linear elliptic operator for every $t \in (0,T)$. As discretization schemes we consider linear one- or multistep methods in time and finite element (FE) or finite difference (FD) methods in space so that the discrete linear systems of equations (LSE) can be written as

$$\sum_{m=0}^{M} A_{k,m}\mathbf{u}_{k-m} = \mathbf{f}_k\,, \quad k = 1,\dots,K\,, \tag{2}$$

with matrices $A_{k,m} \in \mathbb{R}^{N\times N}$. $N \in \mathbb{N}$ is the number of spatial degrees of freedom, $K \in \mathbb{N}$ is the number of time steps and $M \in \mathbb{N}$ is the number of steps in the multistep scheme, e.g. $M = 1$ for Crank-Nicolson and Euler schemes or $M = R$ for linear R-step methods. Then we can gather the time stepping equations (2) in one global system of the form

$$\underbrace{\begin{bmatrix} A_{1,0} & & & & & \\ \vdots & \ddots & & & & \\ A_{M,M-1} \cdots & A_{M,0} & & & & \\ A_{M+1,M} \cdots & A_{M+1,1} & A_{M+1,0} & & & \\ & \ddots & & \ddots & \ddots & \\ & & A_{K,M} & \cdots & A_{K,1} & A_{K,0} \end{bmatrix}}_{=:\bar{A}\in\mathbb{R}^{NK\times NK}} \underbrace{\begin{bmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_M \\ \mathbf{u}_{M+1} \\ \vdots \\ \mathbf{u}_K \end{bmatrix}}_{=:\bar{\mathbf{u}}\in\mathbb{R}^{NK}} = \underbrace{\begin{bmatrix} \mathbf{f}_1 - \sum_{m=1}^{M} A_{1,m}\mathbf{u}_{1-m} \\ \vdots \\ \mathbf{f}_M - A_{M,M}\mathbf{u}_0 \\ \mathbf{f}_{M+1} \\ \vdots \\ \mathbf{f}_K \end{bmatrix}}_{=:\bar{\mathbf{f}}\in\mathbb{R}^{NK}} . \tag{3}$$

The main idea is to reorder the unknowns from a *space-major* ordering

$$\bar{\mathbf{u}} = [u_{1,1}, \dots, u_{1,N}, u_{2,1}, \dots, u_{2,N}, \dots, u_{K,1}, \dots, u_{K,N}]$$

to a *time-major* ordering

$$\mathbf{u} = [u_{1,1}, \dots, u_{K,1}, u_{1,2}, \dots, u_{K,2}, \dots, u_{1,N}, \dots, u_{K,N}] ,$$

where $u_{k,i} = (\mathbf{u}_k)_i$ denotes the i-th degree of freedom at the k-th time step. Reordering the right hand side vector $\bar{\mathbf{f}}$ and the global matrix $\bar{A}$ accordingly leads to the *time-blocked* system matrix A and the vector $\mathbf{f}$. The Matrix A has the same outer block-structure as the matrices $A_{k,l}$, but each block is a lower triangular $K \times K$ matrix with $M + 1$ diagonals.

Now, when we adapt the spatial multigrid method for those systems, we treat each block of the matrix as one entry and use the same transfers and smoothers we would use in a sequential time stepping approach. In our work, we take a Jacobi smoother given by the iteration

$$\mathbf{u}^{m+1} = \mathbf{u}^m + \omega D^{-1}(\mathbf{f} - A\mathbf{u}^m) , \tag{4}$$

where D is the block-diagonal part of the reordered matrix A and $\omega \in \mathbb{R}$ is the damping parameter. As we are formally treating the matrix A as a matrix of blocks, we have to use the complete block-diagonal of A to construct the matrix D instead of only using the main diagonal of A. This leads to a block-Jacobi smoother with block dimension K. Different smoothers that can be written in the form of Eq. (4) with different block-matrices D are applicable in the same manner. The transfer operators are constructed by the same reasoning leading to *semi-coarsening in space* which means that the transfers in space are applied to each time step independently and the temporal grid stays the same across all levels. With these transfer and smoothing operators the usual multigrid algorithm can be used to solve the LSE incorporating multiple time steps simultaneously.

2.1 Intuitive Explanation for Small and Large Time Steps

We want to give a short intuitive understanding of two special cases that can help to tweak the algorithm in practice. To do this we consider the one dimensional heat equation. In the most simplistic case of central differences as space discretization and an implicit Euler time discretization the discrete scheme is given by

$$\frac{1}{\tau}(u_{k,i} - u_{k-1,i}) - \frac{1}{h^2}(u_{k,i+1} - 2u_{k,i} + u_{k,i-1}) = f_{k,i} \tag{5}$$

with the (fixed) spatial grid size h and the (fixed) time step size τ.

Therefore, the matrix entries belonging to the time derivative are of size $\mathcal{O}(\tau^{-1})$ whereas the values belonging to discrete Laplace operator are of size $\mathcal{O}(h^{-2})$. To describe the ratio between them we introduce the anisotropy factor $\lambda = \frac{\tau}{h^2}$ that is widely used in the convergence analysis of space-time multigrid methods [2, 4, 11].

As this parameter depends on the temporal and spatial grids, it changes on different levels of the multigrid scheme. Furthermore, it changes locally on each level, if local refinements or space and time dependent diffusion coefficients are used. Consequently the multigrid method should yield fast convergence for all possible λ.

In the extreme case $\lambda \to \infty$ the matrix entries belonging to the spatial discretizations prevail. If we ignore the significantly smaller values with a factor of τ^{-1}, each block of the global matrix becomes diagonal, so that the global system consists of K independent $N \times N$ systems. Thus, using the time-blocked multigrid is equivalent to solving each time step with a multigrid scheme on its own. This consideration holds true for all BDF-like time discretizations. Other time discretizations show a similar behavior (see Sect. 3). In the opposite case of $\lambda \searrow 0$ the values of the time derivative dominate and therefore the global system becomes block-diagonal, if the mass matrix is diagonal. A diagonal mass matrix arises naturally in FD discretizations or can be created by using finite elements with mass lumping. With those block-diagonal matrices the undamped block-Jacobi smoother ($\omega = 1.0$) becomes exact and the multigrid solver converges in one step.

An undamped Jacobi smoother is not a suitable smoother generally and we do not want to choose the damping parameter ω based on λ manually. Instead, we suggest to use different smoothers, like the Krylov subspace methods BiCGSTAB [9] or GMRES [8] with the block-diagonal matrix D as a preconditioner. These smoothers yield convergence rates similar to the Jacobi smoothing with comparable effort for large λ, while they can recover the convergence in one step in the case of $\lambda \searrow 0$ and a diagonal mass matrix (see Sect. 3).

2.2 Characteristics of the Proposed Method

The time-simultaneous multigrid scheme can be interpreted as a variation of *multigrid waveform relaxation* (WRMG) (c.f. [5, 7]). Multigrid waveform relaxation methods are based on discretizing the PDE in space and applying a multigrid splitting to the stiffness matrix of the semi-discrete ODE system. When using finite elements such a splitting has to be applied to the mass matrix as well to be able to solve the ODEs that arise in every step of the algorithm independently. These methods are equivalent to the time-simultaneous algorithm if a multigrid splitting with a smoother of the form (4) is used for the mass and stiffness matrices and if the same linear multistep method is used to solve every ODE in the multigrid waveform relaxation scheme. Therefore, we do not provide a more detailed convergence analysis but refer to the literature on WRMG [5, 11].

Remark 1 As was shown by Janssen and Vandewalle [5] the time discrete WRMG method for finite elements with a time constant operator $\mathcal{L}$ converges and yields the same asymptotic convergence rates as the traditional multigrid algorithm in the time stepping case, if the coarse grid system matrix A_0 and the preconditioning matrices D_l on each level l are regular. Due to the equivalence of both methods this result holds true for the time-simultaneous algorithm.

The spectral radius of the iteration matrix is bounded, but that does not imply that the defect reduction in each iteration is bounded as well, since the iteration matrix is not symmetric. For more complex smoothers like BiCGSTAB and GMRES with a time-blocked preconditioner the result mentioned in Remark 1—that is based on the spectral radius of the iteration matrix—cannot be applied, because the resulting multigrid iteration is not linear.

The number of necessary floating point operations (FLOPs) in each iteration of the time-simultaneous method with K blocked time steps is still linear in the number of unknowns NK. Compared to the time stepping case where a $N \times N$ system is solved by a multigrid method in K time steps, the cost of the grid transfer per iteration and time step is the same. The cost of the defect calculation per iteration and time step is slightly higher in the time-simultaneous case, because the global matrix has a higher bandwidth. The application of the block-diagonal preconditioner D in the smoothing operation (4) also has linear complexity, as each block is a lower triangular matrix with M bands and can be solved by forward substitution.

While the number of required FLOPs of the time-simultaneous method is slightly higher, the number of required communications per multigrid iteration and time step is reduced by a factor of K^{-1}, because one multigrid solve yields the solution to K time steps. Consequently, the latency induced time of the communications can be lowered and better parallel scaling is possible. In order to actually achieve this a telescopic multigrid scheme, where on coarser levels fewer processes are used, needs to be applied. When only a single process is used on the coarse grid, the coarse solve can also be done by time stepping, since no communication is necessary.

The lower triangular solves are inherently sequential, therefore, parallelization in time direction is not trivial. Nevertheless, it is still possible using parallel triangular solvers (c.f. [10]).

To solve non-linear evolution equations we use a time-simultaneous fixed-point or Newton iteration. Using a time stepping scheme we would discretize the equation in time and apply the linearization in each time step, but now we want to solve multiple time steps simultaneously. Therefore, we have to linearize the PDE itself or the global non-linear discrete system.

3 Numerical Results

In the following, we provide some exemplary results. As a linear test problem we choose the heat equation

$$\begin{aligned} \partial_t u - \Delta u &= 1 + 0.1\sin(t) \quad && (x,t) \in (0,1)^2 \times (0,T) \\ u(0,t) = u(1,t) &= 0 && t \in (0,T) \\ u(x,0) &= 0 && x \in (0,1) \end{aligned}$$

with linear finite elements with mass-lumping as space and a Crank-Nicolson scheme as time discretization. The time-blocked multigrid algorithm uses the F-cycle with one block-Jacobi preconditioned BiCGSTAB pre- and post-smoothing step. For each test 1000 time steps were computed using a different number of blocked time steps. Additionally, we solve the same problem by time stepping and the stationary problem with the same multigrid configuration to create reference results. This was done using spatial grids with grid sizes $h = \frac{1}{32}$ and $h = \frac{1}{128}$. The results are shown in Figs. 1 and 2.

The number of iterations for very small and large time steps behaves as expected. For $\lambda \gg 1$ the number of iterations needed to reduce the global defect by a factor of 10^{-8} is independent of the block size and corresponds to the number of iterations that are needed in the stationary test. In the case of $\lambda \ll 1$, the multigrid algorithm converges in one step and in between the number of iteration is at most slightly higher than in the case of large time steps. The only major difference between different block sizes is that the transition area between small and large time steps shifts to smaller time steps if the block size increases.

Comparing the results of different spatial grids shows that the grid size only affects the convergence speed due to its influence on λ. Other linear multistep methods, higher order finite elements and different test cases show the same qualitative behavior.

To demonstrate the possible benefits of this approach we show the results of a strong scaling test with the same configuration (see Fig. 3) and grid sizes of $h =$

Fig. 1 Number of iterations in the heat eq. test case with different time step sizes and block dimensions, $h = 1/32$

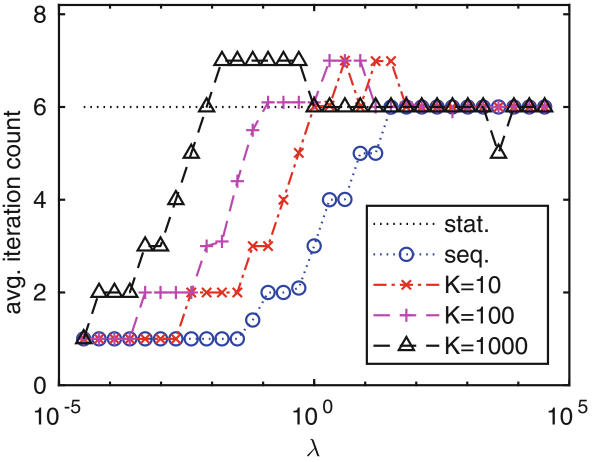
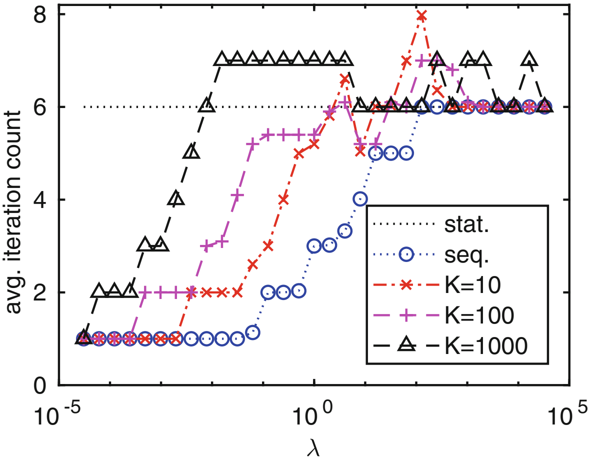


Fig. 2 Number of iterations in the heat eq. test case with different time step sizes and block dimensions, $h = 1/128$



$1/256$ and $\tau = 0.001$. The method was implemented using the C++ based software package *FEAT3*[1] and the tests were executed on the LiDO3 cluster.[2]

With sequential time stepping the best run time is achievable with 32 CPUs and using more processors yields no benefit. Due to the computational overhead, the time-simultaneous approach needs approximately twice the time for low core counts but provides better scaling. Even with a small block size of 20 time steps, more processors can be efficiently used and the run time can be reduced, but with greater block sizes the time-simultaneous scheme scales even better.

[1] http://www.featflow.de/en/software/feat3.html.

[2] https://www.lido.tu-dortmund.de/.

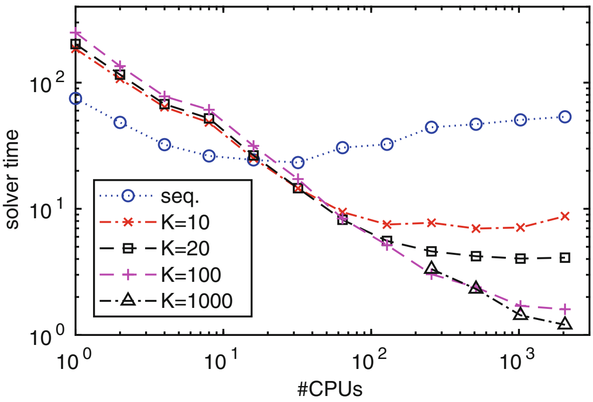


Fig. 3 Strong scaling test: Solver time for a increasing number of processors, $h = 1/256$ (65536 spatial elements), $\tau = 0.001$, $T = 1$ (1000 time steps)

To investigate whether this method can be used for non-linear problems we study the behavior of a time-simultaneous linearization with the one-dimensional viscous Burgers' equation

$$\begin{aligned}
&\partial_t u - \varepsilon \partial_{xx} u + u \partial_x u = 0 && (x,t) \in (0,1) \times (0,T) \\
&u(0,t) = 1\,,\ u(1,t) = 0 && t \in (0,T) \\
&u(x,0) = \max(1 - 5x, 0) && x \in (0,1)
\end{aligned}$$

with the viscosity $0 < \varepsilon \in \mathbb{R}$. Here, we use a FD-discretization with upwind stabilization as discretization in space and Crank-Nicolson in time.

The number of necessary fixed point iteration it to achieve a global defect reduction by 10^{-6} depends on the simulated time frame in the case of small diffusion coefficients ε. For example, in the case of $T = 1$, $\varepsilon = 10^{-3}$ the non-linear solver does not converge in 50 iterations, whereas the averaged number of iterations per time step it_{ref} is only 13.95 in the time stepping approach with time step size $\tau = 0.05$ and decreases with smaller time step sizes. Therefore, a time-simultaneous fixed-point iteration is not suitable for the Burgers' equation with a small viscosity (Table 1).

The Newton scheme provides quadratic convergence if the initial guess is close to the solution. Thus, we compute the solution for the same problem with $2h$, τ and 2ε and use it as the initial guess for the simulation with the grid sizes τ, h and the viscosity ε. Using those starting values the number of iterations shows only a slight increase if a longer time frame is calculated simultaneously and in those tests at most 5 iterations are necessary to achieve the desired defect reduction (see Table 2).

Table 1 Burgers' equation: number of fixed-point iterations, $h = \frac{1}{2048}$

		$\varepsilon = 1$		$\varepsilon = 10^{-2}$		$\varepsilon = 10^{-3}$	
T	τ	it	it_{ref}	it	it_{ref}	it	it_{ref}
0.1	0.050	4	4.00	8	7.50	10	8.50
0.1	0.005	4	3.00	7	4.00	8	4.00
0.1	0.001	4	2.23	7	3.00	8	3.00
1.0	0.050	5	3.55	25	7.95	–	13.95
1.0	0.005	5	3.00	25	4.00	–	6.47
1.0	0.001	5	2.02	25	3.00	–	3.82

Table 2 Burgers' equation: number of Newton iterations, $h = \frac{1}{2048}$

		$\varepsilon = 1$		$\varepsilon = 10^{-2}$		$\varepsilon = 10^{-3}$	
T	τ	it	it_{ref}	it	it_{ref}	it	it_{ref}
0.1	0.050	2	2.50	3	3.00	3	3.00
0.1	0.005	2	2.00	3	2.00	3	2.40
0.1	0.001	2	2.00	3	2.00	3	2.00
1.0	0.050	2	2.90	3	3.80	4	–
1.0	0.005	2	2.00	4	2.88	5	3.25
1.0	0.001	2	2.00	4	2.00	5	2.82

4 Conclusion

We have presented an algebraic approach leading to a time-simultaneous multigrid method that is closely related to multigrid waveform relaxation. The proposed method shows convergence rates that are stable with respect to the number of simultaneous time steps, the grid size and the time step size. The computational cost is slightly higher than in the time stepping case and no parallelization in time direction was done, but the time-simultaneous multigrid method enhances the scalability of the spatial parallelization. The application of this scheme to non-linear equations is also possible by using a time-simultaneous Newton scheme with suitable initial guesses whose choice remains challenging and has to be further examined.

Acknowledgments Calculations have been carried out on the LiDO3 cluster at TU Dortmund. The support by the LiDO3 team at the ITMC at TU Dortmund is gratefully acknowledged.

References

1. Gander, M.J.: 50 Years of Time Parallel Time Integration. Contrib. Math. Comput. Sci. (2015) https://doi.org/10.1007/978-3-319-23321-5_3
2. Gander, M.J., Neumüller, M.: Analysis of a new space-time parallel multigrid algortihm for parabolic problems. SIAM J. Sci. Comput. **38**(4), A2173–A2208 (2016)
3. Hackbusch, W.: Parabolic Multi-grid methods. In R. Glowinski and J.-L. Lions (eds.) Computing Methods in Applied Science and Engineering VI, pp. 189–197, North-Holland (1984)

4. Horton, G., Vandewalle, S.: A space-time multigrid method for parabolic partial differential equations. SIAM J. Sci. Comput. **16**(4), 848–864 (1995)
5. Janssen, J., Vandewalle, S.: Multigrid waveform relaxation on spatial finite element meshes: The discrete-time case. SIAM J. Sci. Comput. **17**(1), 133–155 (1996)
6. Lions, J-L., Maday, Y., Turinici, G.: Résolution d'EDP par un schéma en temps «pararéel» (2001) https://doi.org/10.1016/S0764-4442(00)01793-6
7. Lubich, C., Ostermann, A.: Multi-grid dynamic iteration for parabolic equations. BIT Numer. Math. **27**(2), 216–234 (1987)
8. Saad, Y., Schultz, M.L.: GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**(3), 856–869 (1986)
9. Van der Vorst, H.A.: Bi-CGSTAB: A fast and smoothly converging cariant of Bi-CG for the solution of nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **13**(2), 631–644 (1992)
10. Vandewalle, S., Van de Velde, E.: Space-time concurrent multigrid waveform relaxation. Ann. Numer. Math. **1**(1–4), 347–363 (1994)
11. Vandewalle, S., Horton, G.: Fourier mode analysis of the multigrid waveform relaxation and time-parallel multigrid methods. Computing (1995) https://doi.org/10.1007/BF02238230

Computing Function of Large Matrices by a Preconditioned Rational Krylov Method

Daniele Bertaccini and Fabio Durastante

Abstract Rational Krylov methods are a powerful alternative for computing the product of a function of a large matrix times a given vector. However, the creation of the underlying rational subspaces requires solving sequences of large linear systems, a delicate task that can require intensive computational resources and should be monitored to avoid the creation of subspace different to those required whenever, e.g., the underlying matrices are ill-conditioned. We propose the use of robust preconditioned iterative techniques to speedup the underlying process. We also discuss briefly how the inexact solution of these linear systems can affect the computed subspace. A preliminary test approximating a fractional power of the Laplacian matrix is included.

1 Rationale

Many applications in science and engineering require the evaluation of expressions of the form $f(A)$ or $f(A)v$, where $A \in \mathbb{C}^{n\times n}$, $v \in \mathbb{C}^n$. Among them we recall the numerical integration of (fractional) partial differential equations, of stiff differential equations, solution of linear systems, exponential integrators, which require the computation of the matrix exponential $\exp(A)$, simulating chiral fermions in lattice quantum chromodynamics (QCD), computation of relevant quantities in complex networks and many of others from very different fields of applications.

D. Bertaccini
Università di Roma "Tor Vergata", Dipartimento di Matematica, viale della Ricerca Scientifica 1, Roma, Italy

Istituto per le Applicazioni del Calcolo (IAC) "M. Picone", National Research Council (CNR), Roma, Italy
e-mail: bertaccini@mat.uniroma2.it

F. Durastante (✉)
Istituto per le Applicazioni del Calcolo (IAC) "M. Picone", National Research Council (CNR), Napoli, Italy
e-mail: f.durastante@na.iac.cnr.it

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_33

We consider here a rational Krylov method approximating the action of the product of a function of a large matrix A_n by a given vector using new poles recently introduced in [1] used to compute the underlying rational Krylov subspaces. To build up the rational Krylov subspace requires solving large linear systems thus requiring often heavy computational resources. To defray this, we propose the use of Krylov methods to solve the underlying linear systems. We note that in the literature the related linear systems are mostly solved by direct methods even when the underlying matrices are large and sparse or structured, or without using the "shifted" structure of the matrices in the underlying linear systems. Moreover, in order to have convergence in a reasonable number of iterations, a strategy based on a combination of nonpreconditioned and sequences of preconditioners is proposed and compared with other popular solvers.

Note that the entire process of evaluating $f(A_n)\mathbf{v}$ can be completely *matrix-free*, without forming explicitly the matrices. In particular, for the creation of the rational Krylov subspace by using our preconditioned iterative solver we need just to be able to form matrix-vector products with the underlying matrix. Details, analysis and practical consequences will be pursued in a future work.

Preliminary numerical experiments, with notes on the convergence and the role of the inner tolerances of the Krylov inner accelerators are also given.

2 Rational Krylov Approximation for $f(A_n)\mathbf{v}$

Let V_k be an orthogonal matrix whose columns $\mathbf{v}_1, \ldots, \mathbf{v}_k$ span an arbitrary subspace $\mathcal{W}_k$ of dimension k. An approximation of $f(A_n)\mathbf{v}$ is

$$\begin{aligned} f(A_n)\mathbf{v} =& \frac{1}{2\pi i}\int_\Gamma f(z)(zI - A_n)^{-1}\mathbf{v}\,dz \\ \approx& \frac{1}{2\pi i}\int_\Gamma f(z)V_k(zI - V_k^T A_n V_k)^{-1}V_k^T\mathbf{v}\,dz = V_k f(V_k^T A_n V_k)V_k^T\mathbf{v}, \end{aligned}$$

where the function f is analytic on and inside a closed contour Γ that encloses the spectrum of A_n.

Different methods for the approximation of matrix functions can be provided for different choices of the projection spaces $\mathcal{W}_k$. Given a set of scalars $\{\sigma_1, \ldots, \sigma_{k-1}\} \subset \overline{\mathbb{C}}$ (the extended complex plane), that are not eigenvalues of A_n, let

$$q_{k-1}(z) = \prod\nolimits_{j=1}^{k-1}(\sigma_j - z).$$

We select $\mathcal{W}_k$ as the rational Krylov subspace of order k associated with A_n, $\mathbf{v}$ and q_{k-1} defined by

$$\mathcal{Q}_k(A_n, \mathbf{v}) = [q_{k-1}(A_n)]^{-1}\mathcal{K}_k(A_n, \mathbf{v}), \text{ with } \mathcal{K}_k(A_n, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, A_n\mathbf{v}, \ldots, A_n^{k-1}\mathbf{v}\}.$$

By defining the matrices

$$C_j = \left(\mu_j\sigma_j A_n - I\right)(\sigma_j I - A_n)^{-1}, \text{ for } \{\mu_1, \ldots, \mu_{k-1}\} \subset \overline{\mathbb{C}} \text{ and } \mu_j \neq \sigma_j^{-2}$$

the rational Krylov space $\mathcal{Q}_k(A_n, \mathbf{v})$ can also be written as follows (see [11])

$$\mathcal{Q}_k(A_n, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, C_1\mathbf{v}, \ldots, C_{k-1}\cdots C_2C_1\mathbf{v}\}.$$

This general formulation allows to recast most of the standard Krylov methods in terms of a rational Krylov method with a specific choice of σ_j and μ_j. In particular, the standard (polynomial) Krylov method in which $\mathcal{W}_k = \mathcal{K}_k(A_n, \mathbf{v})$ can be recovered by defining $\mu_j = 1$ and $\sigma_j = \infty$ for each j. Other rational Krylov approaches such as *extended Krylov* and the well-known *shift-and-invert* techniques and some bibliography are briefly recalled in [1, 2]. Rational approximations with numerator different from 1 can be used, e.g., trying to optimizing a rational approximation like with *RKFIT*; see [6].

Here we focus on the fast computation of a rational Krylov subspace by using a preconditioned iterative solver. For brevity, we consider the approach in which $\mu_j = 0$ for each j, and where $-\sigma_j = \xi_j > 0$ are all positive and real, computed from poles positive, real and simple; see [1, Proposition 1] for details. The rational Krylov method is then built by choosing $\mathcal{W}_k = \mathcal{Q}_k(A_n, \mathbf{v})$, where

$$\mathcal{Q}_k(A_n, \mathbf{v}) = \mathrm{Span}\{\mathbf{v}, (\xi_1 I + A_n)^{-1}\mathbf{v}, \ldots, (\xi_{k-1} I + A_n)^{-1}\cdots(\xi_1 I + A_n)^{-1}\mathbf{v}\}. \tag{1}$$

The rational Krylov subspace (1) is computed starting from $\mathbf{v}_1 = \mathbf{v}/\beta$, where $\beta = \|\mathbf{v}\|_2$. Then $\mathbf{v}_{j+1}$ is determined by orthogonalizing the vector $\mathbf{w}_j = (\xi_j I + A_n)^{-1}\mathbf{v}_j$ against $\mathbf{v}_1, \ldots, \mathbf{v}_j$ and normalizing it. Since it is highly unrecommended to invert a matrix (usually the inverse of a sparse matrix is full) $\mathbf{w}_j$ is computed by solving instead the linear system

$$(A_n + \xi_j I)\mathbf{w}_j = \mathbf{v}_j. \tag{2}$$

Note that the matrix in (2) is a shifted version of A_n and can be of very large dimension. Therefore, the handling of this step is crucial. In this way, a sequence of vectors $\{\mathbf{v}_j\}_{j=1}^{k}$ is generated such that

$$\mathbf{v}_j = (\xi_j I + A_n) \sum_{i=1}^{j+1} h_{i,j} \mathbf{v}_i, \qquad \text{for } j \leq k-1,$$

$$\mathbf{v}_k = (\xi_k I + A_n) \sum_{i=1}^{k} h_{i,k} \mathbf{v}_i + (\xi_k I + A_n) h_{k+1,k} \mathbf{v}_{k+1}$$

and the following Arnoldi-like decomposition is obtained

$$V_k^T A_n V_k = (I - H_k D_k) H_k^{-1} - h_{k+1,k} V_k^T A_n \mathbf{v}_{k+1} \mathbf{e}_k^T H_k^{-1},$$

where $D_k = \operatorname{diag}(\{\xi_j^{-1}\}_{j=1}^{k})$ and H_k is the Hessemberg matrix $H_k = [h_{i,j}]$. Since

$$h_{k+1,k} V_k^T A_n \mathbf{v}_{k+1} \mathbf{e}_k^T H_k^{-1} = h_{k+1,k} V_k^T V_{k+1} H_{k+1} \mathbf{e}_{k+1} \mathbf{e}_k^T H_k^{-1} = O_k,$$

the following expression for the projected matrix is true:

$$V_k^T A_n V_k = (I - H_k D_k) H_k^{-1}.$$

Therefore, $f(A_n)\mathbf{v}$ can be approximated as

$$f(A_n)\mathbf{v} \approx \beta V_k f(V_k^T A_n V_k) \mathbf{e}_1, \text{ where } \mathbf{e}_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^k.$$

We observed that often rational Krylov methods exhibit a fast convergence in term of iteration numbers compared to polynomial Krylov methods whenever A_n represents an unbounded self-adjoint operator; see, e.g., [1] and references on rational Krylov methods therein. This comes at the cost of the solution of the linear systems (2) that is the most computationally demanding part of these methods. The computational cost of an iterative solver for (2), whenever A_n is large, is highly problem-dependent, and determines its competitiveness with respect to polynomial methods.

Here we present some preliminary tests using an *iterative* solver with and without some *preconditioners*; see, e.g., [3] for iterative solvers and preconditioners, to solve the sequence of linear systems (2) necessary to generate the underlying rational Krylov space. In particular, we also apply some ideas developed successfully in the past years in [4, 5, 7–10].

3 A Sequence of Preconditioners for the Rational Krylov Subspace

In order to produce the rational Krylov subspace (1) we need to solve, for each new vector of the basis of $\mathcal{W}_k(A_n, \mathbf{v})$, the systems (2). To simplify the treatise, suppose A_n symmetric and positive definite. The discussion will be extended to positive definite normal matrices A_n, not necessarily symmetric, in a future research. By [1, Proposition 1], the zeros of polynomials related to Gauss-Jacobi quadrature formulas σ_j, are all negative and therefore the matrices

$$M_n^{(j)} = (A_n - \sigma_j\, I) = A_n + \xi_j\, I, \quad \xi_j = -\sigma_j \quad j = 1, 2, \ldots, \tag{3}$$

are again positive definite matrices. When A_n is a large sparse (or structured) matrix, an iterative solver for the linear systems (2) can provide a computationally effective alternative to direct solvers usually used, e.g., in [1, 11, 15] and many others. In particular, for symmetric and positive definite linear systems, the natural choice is the Conjugate Gradient method (CG). In order to do this efficiently, we should take care of the value of ξ_j: in general (in this setting ξ_j are all positive; see Sect. 2 and [1]), if $|\xi_j|$ is larger or equal to a certain threshold $\bar{\xi}$, dependent on A_n, then we can try to approximate $\mathbf{w}_j$ by $\mathbf{v}_j/\xi_j$. When the convergence of CG is slow, we experienced that a preconditioning strategy can be beneficial; see the numerical experiments in Sect. 5. However, computation of a new preconditioner for each of the values of j such that $|\xi_j| > \bar{\xi}$ can be expensive but using the same preconditioner for all vectors can give again a slow convergence; see also Fig. 1. A possibility is to use the information we have in order to generate all the needed preconditioners with a computational complexity linear in the number of the unknowns, whenever A_n is sparse.

Suppose that there exists an incomplete factorization for $A_n$in the form

$$P_0 = L_n\, D_n\, U_n^H = L_n\, D_n\, L_n^H, \tag{4}$$

with L_n and U_n sparse unit lower triangular and D_n a diagonal matrix with positive entries. This can be extended easily by using another starting matrix instead of A_n by computing a decomposition (4) for $A_n - \sigma_k I$ for a given value of k; see [7].

In order to determine a sequence of preconditioners for (2), let us consider the matrices related to $M_n^{(j)}$ in (3)

$$P_n^{(j)} = L_n\, \left(D_n + \xi_j\, E_n\right)\, L_n^H, \tag{5}$$

where E_n is a matrix chosen such that $D_n + \xi_j\, E_n$ is nonsingular and

$$||P_n^{(j)} - M_n^{(j)}|| \le c\, \tau, \tag{6}$$

where τ is the drop tolerance used to compute (4). By using arguments similar to those in [5, 7, 10], if $Z_n = L_n^{-H}$, taking

$$E_n = L_n^{-1}\, L_n^{-H} = Z_n^H\, Z_n, \tag{7}$$

we get the desired result and (6) is still valid. However, since the inverse of L_n can be full even if L_n is sparse, we cannot use $Z_n = L_n^{-H}$ in general but, formally $Z_n = g(L_n^{-H})$, where g extracts a sparse matrix from the (full) L_n^{-1}. In [5] and other papers, to solve linear systems coming from a time-dependent partial differential equation, the authors chose to extract a narrow band or even diagonal matrix from L_n^{-1}, but it is not the only possible (or best) choice to be made. We can also use a preconditioner in inverse form, i.e. approximating the inverse of the matrix instead of the matrix itself, P_n^{-1}, that is generated directly as

$$K_n^{(j)} = \left(P_n^{(j)}\right)^{-1} = Z_n\, \left(D_n + \xi_j\, E_n\right)^{-1}\, Z_n^H, \tag{8}$$

because the sparse approximate inverse preconditioner generates the sparse Z_n and D_n^{-1} as an incomplete factorization for A_n^{-1}. However, here we use a sequence of preconditioners of type (5) produced by an inversion and sparsification process for incomplete Cholesky factorizations analyzed in [10] because is faster for our prototype Matlab implementation. Building the proposed strategy by using preconditioners in inverse form and a suitable parallel implementation will be the argument of a future research. For some approximate inverse preconditioners, see [3, 14] and for approximate inverse preconditioners suitable for a matrix-free implementation see [9, Chapter 3, Section 5]. Using an approximate inverse preconditioner, gives an algorithm that is more suitable for parallel implementation since no triangular linear systems are solved and only matrix-vector products are performed. Moreover, the usually computationally expensive setup phase is done mostly once (or few times) and updated many times and this can give an overall reasonable computational cost.

4 What Subspace Are We Computing in Practice?

Projection techniques are sensitive to propagation of rounding errors. For space reasons and because it is out of the scope of this brief note, we do not provide a detailed error analysis of the process of the creation of the basis of our rational Krylov subspaces but some comments are mandatory in order to understand what we are doing in practice. Note that in [1] and in many other papers, auxiliary linear systems are solved by Matlab's standard backslash and no discussions on which kind of subspace $\mathcal{Q}_k(A_n, \mathbf{v})$ is computed in practice are made.

Here, we propose to solve the underlying linear systems using an iterative solver, therefore adding also a (small) analytical error, due to stopping whenever the relative residual is less than a prescribed tolerance.

First, as observed in [15], we should note that residuals of the linear systems (2) must be small. This is a necessary (but not sufficient because the norm of the residual can be smaller than the norm of the error; see, e.g., [9, Section 2.1]) requirement for the correct representation of the underlying Krylov subspace.

What is less trivial is the effect of finite precision, in particular whenever the underlying matrices are severely ill-conditioned. This is the case of the function of matrices whose arguments come from the discretization of differential operators. Here we only observe that generating $Q_k(A_n, \mathbf{v})$ by the preconditioned iterative solution of (2) can benefit from a more favorable accumulation of rounding errors, in particular whenever preconditioning reduces the condition number.

5 Numerical Test

Let us consider one sample of various tests we performed in Matlab. The general settings are the same as those in [1]. We test our proposal by computing the function $f(A) = A_n^{-\alpha}\mathbf{v}$, $\alpha \in (0, 1)$, where the argument is the matrix A_n, the second order centered differences discretization of the bidimensional Laplacian discretized on the unit square $[0, 1]^2$ with an equispaced mesh and $\mathbf{v}$ is a given random vector. Similar results are observed also for the Laplacian with variable coefficients and in multiple dimensions. Note that, by considering the *Matrix Transfer Technique* (MTT) [12, 13], this function can be used to compute the numerical solution of the fractional partial differential equation

$$\begin{cases} (-\Delta)^{\alpha} u = s(\mathbf{x}, u), & \mathbf{x} \in \Omega, \\ u(\mathbf{x}) = 0, & \mathbf{x} \in \partial\Omega. \end{cases} \tag{9}$$

We perform several preliminary tests by comparing the performances of the underlying rational Krylov approximation for $f(A_n)\mathbf{v}$ by solving the linear systems (2) by (in brackets the legends for the Fig. 1 with the results): the Matlab's "backslash" ("Direct"); the conjugate gradient ("CG"); the conjugate gradient with a fixed preconditioner by choosing the zero fill-in incomplete Cholesky that gives the lowest number of overall iterations for all the matrices in the sequences (2) ("PCG fix"); the conjugate gradient recomputing the zero fill-in incomplete Cholesky for each linear system (2) ("PCG"); the conjugate gradient updating the incomplete Cholesky preconditioner as described in Sect. 3 with droptol 0.1 with the starting factorized preconditioner P_0 computed not for A_n but for $A_n - \sigma_{k+1} I$ ("PCG upd"); σ_k defined as in (3). The value k is selected such as, if $|\sigma_k| \geq |\sigma_*|$, σ_* a threshold value such that, for $j \leq k$, (2) can be solved with the conjugate gradient without preconditioning because of the major influence of the identity in the matrix of (2); see, e.g., [5, Section 4]. Of course, for the Laplacian with constant coefficients, a

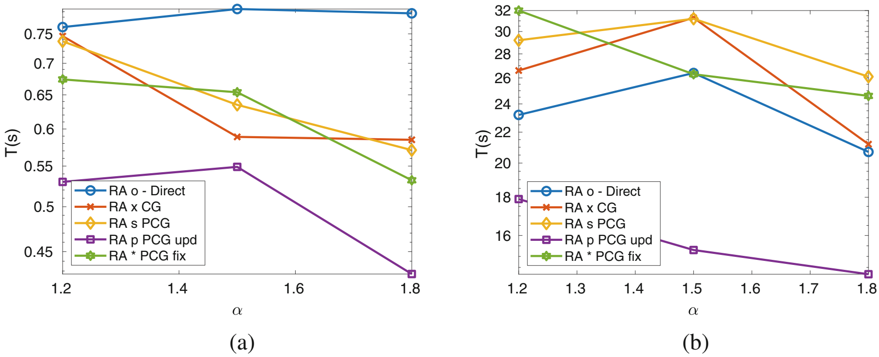


Fig. 1 Performance test plot on the computation of $f(z) = z^{\alpha}$, $\alpha = 1.2,\ 1.5,\ 1.8$, $z = A$, A generated by the 2nd order centered difference discretization of the Laplacian. (**a**) The test matrices are 16384 × 16384. (**b**) The test matrices are 262144 × 262144

structured preconditioner could certainly perform better (e.g., a geometric multigrid) and could be also updated, but here we prefer to test a black-box preconditioner like the incomplete factorizations in order to stress the advantages of the general framework over a specialized technique that is too problem dependent for a very well know problem such as the one considered. All the iterative solvers were initialized by the null vector. As can be observed from Fig. 1, computing the rational Krylov subspace by the updated preconditioned iterative solver can be beneficial over all the other solvers.

6 Conclusions

We propose to speed up the application of rational Krylov methods for computing the product of some function of a large matrix times a given vector. Indeed, they often require solving sequences of large linear systems, a task that can require intensive computational resources. Preliminary tests on some fractional partial differential equations show that the approach is promising.

Details, analysis and large tests using a parallel implementation will be pursued in a future work.

Acknowledgments We wish to thank two anonymous referees for their comments which have improved the readability of the paper.

This project was partially supported by the Tor Vergata University project “MISSION: SUSTAINABILITY” “NUMnoSIDS”, CUP E86C18000530005, and by the INDAM-GNCS 2019 project “Tecniche innovative e parallele per sistemi lineari e non lineari di grandi dimensioni, funzioni ed equazioni matriciali ed applicazioni”. D. Bertaccini gratefully acknowledges the MIUR Excellence Department Project awarded to the Department of Mathematics, University of Rome Tor Vergata, CUP E83C18000100006.

References

1. Aceto, L., Bertaccini, D., Durastante, F., Novati, P.: Rational Krylov methods for functions of matrices with applications to fractional partial differential equations. J. Comput. Phys. **396**, 470–482 (2019)
2. Aceto, L., Novati, P.: Rational approximations to fractional powers of self-adjoint positive operators. Numer. Math. **143**(1), 1–16 (2019)
3. Axelsson, O.: Iterative solution methods. Cambridge University press (1996)
4. Bellavia, S., Bertaccini, D., Morini, B.: Nonsymmetric preconditioner updates in Newton-Krylov methods for nonlinear systems. SIAM J. Sci. Comput. **33**(5), 2595–2619 (2011)
5. Benzi, M., Bertaccini, D.: Approximate inverse preconditioning for shifted linear systems. BIT **43**(2), 231–244 (2003)
6. Berljafa, M., Güttel, S.: The RKFIT algorithm for nonlinear rational approximation. SIAM J. Sci. Comput. **39**(5), A2049–A2071 (2017). https://doi.org/10.1137/15M1025426
7. Bertaccini, D.: Efficient preconditioning for sequences of parametric complex symmetric linear systems. Electron. Trans. Numer. Anal. **18**, 49–64 (2004)
8. Bertaccini, D., Durastante, F.: Interpolating preconditioners for the solution of sequence of linear systems. Comput. Math. Appl. **72**(4), 1118–1130 (2016)
9. Bertaccini, D., Durastante, F.: Iterative Methods and Preconditioning for Large and Sparse Linear Systems with Applications. Chapman & Hall/CRC Monographs and Research Notes in Mathematics. CRC Press (2018)
10. Bertaccini, D., Filippone, S.: Sparse approximate inverse preconditioners on high performance GPU platforms. Computer e Mathematics with Applications **71**, 693–711 (2016)
11. Güttel, S.: Rational Krylov approximation of matrix functions: numerical methods and optimal pole selection. GAMM-Mitt. **36**(1), 8–31 (2013)
12. Ilic, M., Liu, F., Turner, I., Anh, V.: Numerical approximation of a fractional-in-space diffusion equation. I. Fract. Calc. Appl. Anal. **8**(3), 323–341 (2005)
13. Ilic, M., Liu, F., Turner, I., Anh, V.: Numerical approximation of a fractional-in-space diffusion equation. II. With nonhomogeneous boundary conditions. Fract. Calc. Appl. Anal. **9**(4), 333–349 (2006)
14. Kolotina, L., Yeremin, A.: Factorized sparse approximate inverse precon- ditioning i: Theory. SIAM J. Matrix Anal. Appl. **14**(1), 45–58 (1993)
15. Lehoucq, R.B., Meerbergen, K.: Using generalized Cayley transformations within an inexact rational Krylov sequence method. SIAM J. Matrix Anal. Appl. **20**, 131–148 (1998)

On Energy Preserving High-Order Discretizations for Nonlinear Acoustics

Herbert Egger and Vsevolod Shashkov

Abstract This paper addresses the numerical solution of the Westervelt equation, which arises as one of the model equations in nonlinear acoustics. The problem is rewritten in a canonical form that allows the systematic discretization by Galerkin approximation in space and time. Exact energy preserving methods of formally arbitrary order are obtained and their efficient realization as well as the relation to other frequently used methods is discussed.

1 Introduction

The modeling of nonlinear effects arising in the presence of high intensity acoustic fields is one of the central subjects of nonlinear acoustics [11]. One widely used model in this area is the Westervelt equation [13, 20] which in dimensionless form can be written as

$$\partial_{tt}\psi - \Delta\psi = \alpha\Delta(\partial_t\psi) + \beta\partial_t(\partial_t\psi)^2. \tag{1}$$

The two terms on the right hand side, scaled with constants $\alpha, \beta \geq 0$, account for viscous and nonlinear effects of the medium and constitute the deviations from the standard linear wave equation. Equation (1) is written here in terms of the velocity potential ψ which is related to the acoustic velocity and pressure variations by

$$v = -\nabla\psi \qquad \text{and} \qquad p = \partial_t\psi. \tag{2}$$

Similar to the linear wave equation, the Westervelt equation also encodes the principle of energy conservation. Using (2), the dimensionless acoustic energy

H. Egger (✉) · V. Shashkov
TU Darmstadt, Darmstadt, Germany
e-mail: egger@mathematik.tu-darmstadt.de; shashkov@mathematik.tu-darmstadt.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_34

contained in a bounded domain Ω can be expressed in terms of the velocity potential by

$$\mathcal{E}(\psi, \partial_t \psi) = \int_\Omega \tfrac{1}{2}|\nabla \psi|^2 + \left(\tfrac{1}{2} - \tfrac{2\beta}{3}\partial_t \psi\right) |\partial_t \psi|^2 dx \tag{3}$$

One can verify by elementary computations that solutions of (1), when complemented, e.g., by homogeneous boundary conditions $\partial_n \psi = 0$, satisfy

$$\frac{d}{dt}\mathcal{E}(\psi, \partial_t \psi) = -\alpha \int_\Omega |\nabla(\partial_t \psi)|^2 dx. \tag{4}$$

This *energy identity* states that in a closed system the acoustic energy is conserved exactly up to dissipation caused by viscous effects. For $\alpha \geq 0$, the Westervelt equation (1) thus models a passive system. This property is of fundamental importance not only for the analysis of the problem [13] but also for the accuracy and long-term stability of discretization schemes; see [15] and the references given there.

Various discretization schemes for the linear wave equation can be extended to nonlinear acoustics. Among the most widely used approaches are the finite-difference-time-domain method [10, 14, 17], finite-volume schemes [5, 19], and finite-element methods together with Newmark time-stepping [2, 12, 18]. To the best of our knowledge, none of the mentioned approaches is capable to exactly reproduce the energy identity (4) on the discrete level in the presence of nonlinearities.

In this paper, we propose a systematic strategy for the high-order approximation of nonlinear acoustics in space and time which exactly satisfies an integral version of the energy identity (4) on the discrete level. Our approach utilizes the fact that the Westervelt equation (1) can be written as a generalized gradient system

$$C(u)\partial_t u = -\mathcal{H}'(u) \tag{5}$$

with $u = (\psi, \partial_t \psi)$ denoting the state and $\mathcal{H}(u) = \mathcal{E}(\psi, \partial_t \psi)$ the energy of the system. The energy identity (4) is then a direct consequence of the particular structure of this system; see below. As illustrated in [4], the structure-preserving discretization of (5) can be obtained in a systematic manner by Galerkin approximation in space and time. For the space discretization, we utilize a finite-element approximation with mass-lumping. The time-integration resulting from our approach can be interpreted as a variant of particular Runge-Kutta methods and is strongly related to discrete gradient and average vector field collocation methods [7, 8, 16].

The remainder of the manuscript is organized as follows: In Sect. 2, we rewrite the Westervelt equation (1) into the non-standard canonical form (5). Our discretization strategy is then introduced in Sect. 3, and we show that the energy identity remains valid after discretization. In Sect. 4, we briefly discuss some details of the numerical realization and the connection to other discretization methods. In

Sect. 5, we illustrate the exact energy-conservation in the absence of viscous effects for one-dimensional example.

2 A Canonical Form of the Westervelt Equation

We introduce $p = \partial_t \psi$ as new variable and write $u = (\psi, p)$ and $\mathcal{H}(u) = \mathcal{E}(\psi, p)$. The derivative $\mathcal{H}'(u)$ of the energy in direction $v = (\eta, q)$ is then given by

$$\langle \mathcal{H}'(u), v\rangle = \langle \mathcal{E}'(\psi, p), (\eta, q)\rangle = \int_\Omega \nabla\psi \cdot \nabla\eta + (1 - 2\beta p)p \cdot q \, dx.$$

Using integration-by-parts for the first term under the integral and homogeneous boundary conditions $\partial_n \psi = 0$ on $\partial\Omega$, we can now formally represent the negative derivative of the energy functional as a two-component function

$$-\mathcal{H}'(u) = (\Delta\psi, -(1 - 2\beta p)p)\,. \tag{6}$$

In order to bring equation (1) into the canonical form (5), we should thus derive an equivalent first order system with right hand sides given by $-\mathcal{H}'(u)$. By elementary computations, one can verify the following statements.

Lemma 1 *The Westervelt equation* (1) *is equivalent to the system*

$$(1 - 2\beta p)\partial_t p - \alpha\Delta\partial_t \psi = \Delta\psi. \tag{7}$$

$$-(1 - 2\beta p)\partial_t \psi = -(1 - 2\beta p)p. \tag{8}$$

Proof Differentiating the last term in (1) yields

$$\beta\partial_t(\partial_t\psi)^2 = 2\beta(\partial_t\psi)\partial_{tt}\psi.$$

Using this identity and a slight rearrangement of terms, the Westervelt equation can thus be rewritten equivalently as

$$(1 - 2\beta\partial_t\psi)\partial_{tt}\psi - \alpha\Delta(\partial_t\psi) = \Delta\psi.$$

By replacing $\partial_t\psi$ and $\partial_{tt}\psi$ in the first term by p and $\partial_t p$, we already obtain (7). The second equation (8) is an immediate consequence of the identity $p = \partial_t\psi$. □

Remark 1 Abbreviating $u = (\psi, p)$ and $\mathcal{H}(u) = \mathcal{E}(\psi, p)$ as above, the system (7)–(8) can be seen to formally be in the canonical form (5) with

$$C(u) = \begin{pmatrix} -\alpha\Delta & (1 - 2\beta p) \\ -(1 - 2\beta p) & 0 \end{pmatrix}.$$

The somewhat unconventional form of the system (7)–(8) is dictated by the underlying energy, whose derivative has to appear in the right hand side of the equations.

Our discretization will be based on the following weak formulation of (7)–(8).

Lemma 2 *Let (ψ, p) denote a smooth solution of the system (7)–(8) on Ω with homogeneous boundary values $\partial_n \psi = 0$ on $\partial\Omega$ for $0 \le t \le T$. Then*

$$\langle (1-2\beta p(t))\partial_t p(t), \eta\rangle + \alpha\langle \nabla \partial_t \psi(t), \nabla \eta\rangle = -\langle \nabla \psi(t), \nabla \eta\rangle \tag{9}$$

$$-\langle (1-2\beta p(t))\partial_t \psi(t), q\rangle = -\langle (1-2\beta p(t)) p(t), q\rangle \tag{10}$$

for all test functions $\eta, q \in H^1(\Omega)$ and all $0 \le t \le T$. The bracket $\langle u, v\rangle = \int_\Omega uv\, dx$ is used here to denote the scalar product on $L^2(\Omega)$.

Proof The two identities follow by multiplying (7)–(8) with appropriate test functions, integrating over Ω, and integration-by-parts for the terms with the Laplacian. The boundary terms vanish due to the homogeneous boundary conditions. □

We now show that the energy identity (4) follows directly from this weak formulation.

Lemma 3 *Let (ψ, p) denote a solution of the weak formulation (9)–(10). Then*

$$\frac{d}{dt}\mathcal{E}(\psi(t), p(t)) = -\alpha \int_\Omega |\nabla(\partial_t \psi(t))|^2 dx.$$

Proof Formal differentiation of the energy yields

$$\begin{aligned}\frac{d}{dt}\mathcal{E}(\psi, p) &= \langle \mathcal{E}'(\psi, p), (\partial_t \psi, \partial_t p)\rangle \\ &= \langle \nabla \psi, \nabla \partial_t \psi\rangle + \langle (1-2\beta p)p, \partial_t p\rangle,\end{aligned}$$

where we used the representation of the energy derivative derived above. The two terms correspond to the right hand sides of the weak formulation (9)–(10) with test functions $\eta = \partial_t \psi$ and $q = \partial_t p$. Using the weak formulation, we thus obtain

$$\frac{d}{dt}\mathcal{E}(\psi, p) = -\langle (1-2\beta p)\partial_t p, \partial_t \psi\rangle - \alpha\langle \nabla \partial_t \psi, \nabla \partial_t \psi\rangle + \langle (1-2\beta p)\partial_t \psi, \partial_t p\rangle.$$

Now the first and last term on the right hand side cancel out and the assertion follows by noting that $\langle \nabla \partial_t \psi, \nabla \partial_t \psi\rangle = \int_\Omega |\nabla \partial_t \psi|^2 dx$ by definition of the bracket. □

Remark 2 The proof of the previous lemma reveals that the energy identity (4) is a direct consequence already of the particular structure of the weak formulation (9)–(10). Since this form is preserved automatically under projection, one can obtain a structure preserving discretization by Galerkin approximation; see [4] for details.

In the following section, we discuss a particular approximation based on finite elements.

3 Structure-Preserving Discretization

Let $\mathcal{T}_h = \{K\}$ denote a mesh, i.e., a geometrically conforming and uniformly shape-regular simplicial partition, of the domain Ω. We write h_K and $h = \max_K h_K$ for the local and global mesh size. We further denote by

$$V_h = \{v \in H^1(\Omega) : v|_K \in P_k(K) \quad \forall K \in T_h\}$$

the standard finite element space consisting of continuous piecewise polynomial functions of degree $\leq k$. Let $I_\tau = \{0 = t^0 < t^1 < \ldots < t^N = T\}$ denote a partition of the time interval $[0, T]$ into elements $[t^{n-1}, t^n]$ of size $\tau_n = t^n - t^{n-1}$ and, as before, write $\tau = \max_n \tau_n$ for the global time step size. We denote by

$$P_q(I_\tau; X) = \{v : v|_{[t^{n-1}, t^n]} \in P_q([t^{n-1}, t^n]; X)\}$$

the space of piecewise polynomial functions in time of degree $\leq q$ with values in X. As approximation for the Westervelt equation (1) we now consider the following inexact Galerkin-Petrov Galerkin approximation of the weak formulation (9)–(10).

Problem 1 Find $\psi_h, p_h \in P_q(I_\tau; V_h) \cap H^1([0; T]; V_h)$ such that $\psi_h(0) = \psi_{h,0}$, $p_h(0) = p_{h,0}$, for given initial values $\psi_{h,0}, p_{h,0} \in V_h$, and such that

$$\int_{t^m}^{t^n} \langle (1 - 2\beta p_h)\partial_t p_h, \widetilde{\eta}_h \rangle_h + \alpha \langle \nabla \partial_t \psi_h, \nabla \widetilde{\eta}_h \rangle \, dt = -\int_{t^m}^{t^n} \langle \nabla \psi_h, \nabla \widetilde{\eta}_h \rangle \, dt$$

$$-\int_{t^m}^{t^n} \langle (1 - 2\beta p_h)\partial_t \psi_h, \widetilde{q}_h \rangle_h \, dt = -\int_{t^m}^{t^n} \langle (1 - 2\beta p_h) p_h, \widetilde{q}_h \rangle_h \, dt.$$

for all $0 \leq t^m \leq t^n \leq T$ and all $\widetilde{\eta}_h, \widetilde{q}_h \in P_{q-1}(I_\tau; V_h)$. Here $\langle u, v \rangle_h$ is a symmetric positive definite approximation for $\langle u, v \rangle$ obtained by numerical integration.

Due to the inexact realization of the scalar product in some of the terms, we have to modify the discrete energy accordingly and define

$$\mathcal{E}_h(\psi_h, p_h) = \langle \tfrac{1}{2} \nabla \psi_h, \nabla \psi_h \rangle + \langle (\tfrac{1}{2} - \tfrac{2\beta}{3} p_h) p_h, p_h \rangle_h.$$

Note that $\mathcal{E}_h(\psi_h, p_h) = \mathcal{E}(\psi_h, p_h)$ when the scalar products are computed exactly, so this defines a natural modification of the energy on the discrete level. With similar arguments as used in Lemma 3, we now obtain the following discrete energy identity.

Lemma 4 *Let (ψ_h, p_h) denote a solution of Problem 1. Then one has*

$$\mathcal{E}_h(\psi_h(t^n), p_h(t^n)) = \mathcal{E}_h(\phi_h(t^m), p_h(t^m))) - \alpha \int_{t^m}^{t^n} \int_\Omega |\nabla \partial_t \psi_h(s)|^2 dx\, ds,$$

for all $0 \le t^m \le t^n \le T$*, which is the discrete equivalent of the integral form of* (4).

Proof Let $u^n = u(t^n)$ denote the value of a function a time t^n. Then by the fundamental theorem of calculus and the expression of the energy derivative, we obtain

$$\begin{aligned}\mathcal{E}_h(\psi_h^n, p_h^n) - \mathcal{E}_h(\psi_h^m, p_h^m) &= \int_{t^m}^{t^n} \frac{d}{dt}\mathcal{E}_h(\psi_h, p_h) dt \\ &= \int_{t^m}^{t^n} \langle \nabla\psi_h, \nabla\partial_t\psi_h\rangle + \langle (1-2\beta p_h)p_h, \partial_t p_h\rangle_h \, dt.\end{aligned}$$

The two terms in the second line correspond to the negative of the right hand side in Problem 1 with test functions $\widetilde{\eta}_h = \partial_t \psi_h$ and $\widetilde{q}_h = \partial_t p_h$, which directly leads to

$$\mathcal{E}_h(\psi_h^n, p_h^n) - \mathcal{E}_h(\psi_h^m, p_h^m) = -\alpha \int_{t^m}^{t^n} \langle \nabla\partial_t\psi_h, \nabla\partial_t\psi_h\rangle \, dt.$$

The assertion of the lemma now follows from the definition of the bracket $\langle\cdot,\cdot\rangle$. □

Remark 3 Let us note that, exactly in the same way as in the previous section, the discrete energy identity is a direct consequence of the particular structure of the weak formulation used in the definition of Problem 1, which adequately accounts for the underlying nonlinear discrete energy.

4 Remarks on the Implementation

Before we proceed to numerical tests, let us briefly comment on the implementation of the method resulting from Problem 1. For ease of presentation, we consider piecewise linear approximations in space and time, i.e., $k = q = 1$. We choose the standard nodal basis for the finite elements in space and utilize the vertex rule for numerical integration in $\langle u, v\rangle_h$, which gives rise to diagonal matrices associated with these integrals. The system to be solved on every time step then takes the form

$$D(1-2\beta p^{n+1/2})\frac{p^{n+1}-p^n}{\tau} + \alpha K(1)\frac{\psi^{n+1}-\psi^n}{\tau} = -K(1)\psi^{n+1/2}$$

$$-D(1-2\beta p^{n+1/2})\frac{\psi^{n+1}-\psi^n}{\tau} = -D(1-2\beta p^{n+1/2})p^{n+1/2} - \tfrac{\beta}{6}D(p^{n+1}-p^n)(p^{n+1}-p^n)$$

with $u^{n+1/2} = \frac{1}{2}(u^n + u^{n+1})$ denoting the value at the midpoint of the time interval. Furthermore, the matrices $D(a)$, $K(b)$ represent the integrals $\langle au, v\rangle_h$ and $\langle b\nabla u, \nabla v\rangle$.

Remark 4 Apart from the last term in the second equation, the time-step iteration amounts to the Gauß-Runge-Kutta method with $s = 1$ stages and could also be interpreted as an inexact realization of the Lobatto-IIIA method with $s = 2$ stages. Similar statements can be made for and order $q \geq 1$ in Problem 1. Using an inexact computation of the time integrals arising on the left-hand side in Problem 1 leads to the *average vector field collocation methods* discussed in [9]. The inexact realization $\langle\cdot,\cdot\rangle_h$ of the scalar product in space allows to utilize mass-lumping strategies which facilitates the handling of the nonlinear terms in the numerical realization, since they only appear in the diagonal matrices $D(\cdot)$. Using the considerations of [2, 6], mass lumping can be achieved in principle for any order of approximation $k \geq 1$ in space.

5 Numerical Tests

For illustration of our results, we now report about numerical tests for a simple example. We consider the Westervelt equation (1) on the domain $\Omega = (0, 16)$ with homogeneous boundary conditions $\partial_x\psi = 0$ at $\partial\Omega$. The model parameters are set to $\alpha = 0$ and $\beta = 0.3$, i.e., we consider a problem without dissipation. By Lemma 3, the acoustic energy of the system is then preserved for all times. As initial conditions for our computational tests, we choose $\psi_0(x) = 0$ and $p_0(x) = e^{-0.2x^2}$. Some snapshots of the numerical solution obtained with the method of Problem 1 with polynomial orders $k = q = 2$ are depicted in Fig. 1. In comparison to the solution

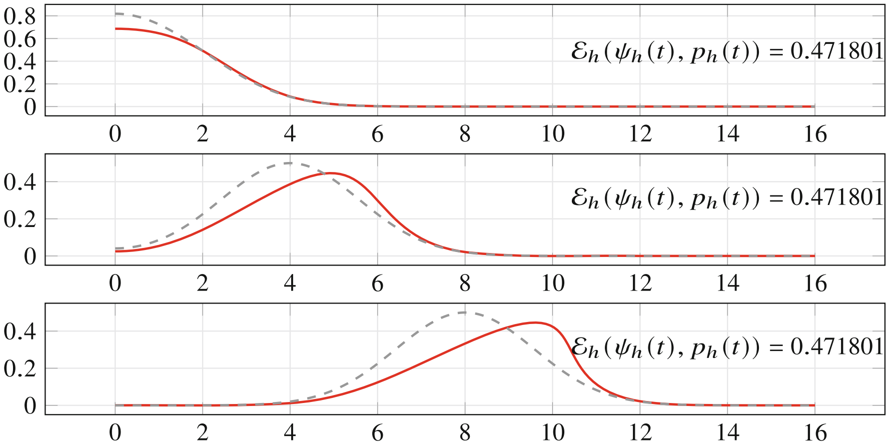


Fig. 1 Solution $p_h(t)$ of the Westervelt equation with $\alpha = 0$, $\beta = 0.3$ (red) and the linear wave equation with $\alpha = \beta = 0$ (black dashed) at time steps $t = 1$, $t = 4$, and $t = 8$

Table 1 Convergence rates for discrete error in the pressure at gridpoints for the nonlinear wave equation $\beta = 0.3$ (left) and the linear wave equation $\beta = 0$ (right) for comparison

$h = \tau$	err $\times 10^{-3}$	eoc	$h = \tau$	err$\times 10^{-5}$	eoc
0.25	1.7758	–	0.25	2.4964	–
0.125	0.1841	3.27	0.125	0.1565	3.99
0.0625	0.0131	3.81	0.0625	0.0098	4.00
0.03125	0.0008	4.03	0.03125	0.0006	4.03

of the linear wave equation, which corresponds to (1) with $\alpha = \beta = 0$, the presence of the nonlinear terms ($\beta = 0.3$) leads to a steepening of the wave front. In the absence of viscous damping, this leads to the formation of a shock in the long run. For the linear wave equation ($\beta = 0$), our method coincides with the Lobatto-IIIA method and the energy is preserved exactly for both schemes. While the proposed method still yields exact energy preservation also in the nonlinear case ($\beta > 0$), the Lobatto-IIIA method fails to do so. Similar statements also hold for the Gauß-Runge-Kutta and the Newmark scheme.

From the usual error analysis of Galerkin methods [1], we expect that the error

$$\text{err} = \max_{0 \le t_n \le T} \|p(t^n) - p_h^n\|_h$$

of the method resulting from Problem 1 with approximation orders $q = k$ converges with order $p = k+1$ in space and time. In Table 1, we report about the corresponding convergence rates observed in our numerical tests. For our numerical tests, we use polynomial orders $k = q = 2$ in space and time, and thus would expect third order convergence. As can be seen in Table 1, we here even observe fourth order convergence on grid-points. This kind of super-convergence on uniform grids can be observed also for finite-difference approximations of linear wave equations [3].

Acknowledgments The authors are grateful for support by the German Research Foundation (DFG) via grants TRR 146 C3, TRR 154 C4, Eg-331/1-1, and through the "Center for Computational Engineering" at TU Darmstadt.

References

1. G. Akrivis, C. Makridakis, and R. N. Nochetto. Galerkin and Runge-Kutta methods: unified formulation, a posteriori error estimates and nodal superconvergence. *Numer. Math.*, 118:429–456, 2011.
2. G. Cohen. *Higher-Order Numerical Methods for Transient Wave Equations*. Springer, 2002.
3. G. Cohen and P. Joly. Construction analysis of fourth-order finite difference schemes for the acoustic wave equation in nonhomogeneous media. *SIAM J. Numer. Anal.*, 33:1266–1302, 1996.
4. H. Egger. Energy stable Galerkin approximation of Hamiltonian and gradient systems. 2018. arXive:1812.04253.

5. K. Fagnan, R. J. LeVeque, T. J. Matula, and B. MacConaghy. High-resolution finite volume methods for extracorporeal shock wave therapy. In *Hyperbolic Problems: Theory, Numerics, Applications*, pages 503–510. Springer, New York, 2008.
6. S. Geevers, W. A. Mulder, and J. J. W. van der Vegt. New higher-order mass-lumped tetrahedral elements for wave propagation modelling. *SIAM J. Sci. Comput.*, 40:A2830–A2857, 2018.
7. O. Gonzales. Time integration and discrete Hamiltonian systems. *J. Nonl. Sci.*, 6:449–467, 1996.
8. E. Hairer and C. Lubich. Energy-diminishing integration of gradient systems. *IMA J. Numer. Anal.*, 34:452–461, 2014.
9. E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations; 2nd ed.* Springer, 2006.
10. I. M. Hallaj and R. O. Cleveland. FDTD simulation of finite-amplitude pressure and temperature fields for biomedical ultrasound. *J. Acoust. Soc. Am.*, 105:L7, 1999.
11. M. F. Hamilton and D. T. Blackstock. *Nonlinear Acoustics*. Academic Press, 1998.
12. J. Hoffelner, H. Landes, M. Kaltenbacher, and R. Lerch. Finite element simulation of nonlinear wave propagation in thermoviscous fluids including dissipation. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control*, 48:779–786, 2001.
13. B. Kaltenbacher and I. Lasiecka. Global existence and exponential decay rates for the Westervelt equation. *Discr. Cont. Dyn. Sys. Ser. S*, 2:503–523, 2009.
14. A. Karamalis, W. Wein, and N. Navab. Fast ultrasound image simulation using the Westervelt equation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2010*, pages 243–250. Springer, New York, 2010.
15. B. Leimkuhler and S. Reich. *Simulating Hamiltonian Dynamics*. Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 2004.
16. R. I. McLachlan, G. R. W. Quispel, and N. Robidoux. Geometric integration using discrete gradients. *R. Soc. Lond. Philos. Trans. Ser. A: Math. Phys. Eng. Sci.*, 357:1021–1045, 1999.
17. K. Okita, K. Ono, S. Takagi, and Y. Matsumoto. Development of high intensity focused ultrasound simulator for large-scale computing. *Int. J. Numer. Meth. Fluids*, 65:43–66, 2011.
18. T. Tsuchiya and Y. Kagawa. A simulation study on nonlinear sound propagation by finite element approach. *J. Acoust. Soc. Jpn.*, 13:223–230, 1992.
19. R. Velasco-Segura and P. L. Rendòn. A finite volume approach for the simulation of nonlinear dissipative acoustic wave propagation. *Wave Motion*, 58:180–195, 2015.
20. P. J. Westervelt. Parametric acoustic array. *J. Acoust. Soc. Am.*, 35:535–537, 1963.

Hierarchical DWR Error Estimates for the Navier-Stokes Equations: *h* and *p* Enrichment

B. Endtmayer, U. Langer, J.P. Thiele, and T. Wick

Abstract In this work, we further develop multigoal-oriented a posteriori error estimation for the nonlinear, stationary, incompressible Navier-Stokes equations. It is an extension of our previous work on two-side a posteriori error estimates for the DWR method. We now focus on h enrichment and p enrichment for the error estimator. These advancements are demonstrated with the help of a numerical example.

1 Introduction

Multigoal-oriented error estimation offers the opportunity to control several quantities of interest simultaneously. In recent years, we have developed a version [3, 4] which relies on the dual-weighted residual method [2], and also balances the discretization error with the nonlinear iteration error [12]. The localization is based on the weak formulation proposed in [13]. Our method uses hierarchical finite element spaces. Here, we investigate h-refinement along with p-refinement to generate enriched spaces. This we call h and p enrichment, respectively. It is an extension of our previous work on two-side a posteriori error estimates for the DWR method [4]. These ideas are applied to the stationary incompressible Navier-Stokes equations. It is well-known that the spaces for the velocities and the pressure must be balanced in order to satisfy an inf-sup condition [6]. These requirements must be reflected in the design of the adjoint problems in dual-weighted residual

B. Endtmayer (✉) · U. Langer
Johann Radon Institute for Computational and Applied Mathematics, Austrian Academy of Sciences, Linz, Austria
e-mail: bernhard.endtmayer@ricam.oeaw.ac.at; ulanger@numa.uni-linz.ac.at

J. P. Thiele · T. Wick
Institut für Angewandte Mathematik, Leibniz Universität Hannover, Hannover, Germany
e-mail: thiele@ifam.uni-hannover.de; thomas.wick@ifam.uni-hannover.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_35

error estimation and the p enrichment proposed in this paper. To demonstrate the performance of the error estimator, we adopt the 2D-1 fluid flow benchmark [14].

2 The Model Problem and Discretization

2.1 The Model Problem

We consider the stationary Navier-Stokes 2D-1 benchmark problem [14] as our model problem. This configuration was also considered in [4]. The domain $\Omega \subset \mathbb{R}^2$ is given by $(0, 2.2) \times (0, H) \setminus B$, and B is the ball with the center $(0.2, 0.2)$ and the radius 0.05 as given in [14] and visualized in Fig. 1. Find $\boldsymbol{u} = (u, p)$ such that

$$
\begin{aligned}
-\mathrm{div}(\nu(\nabla u + \nabla u^T)) + (u \cdot \nabla)u - \nabla p &= 0, && \text{in } \Omega,\\
-\mathrm{div}(u) &= 0, && \text{in } \Omega,\\
u &= u_{\text{in}} && \text{on } \Gamma_{\text{in}},\\
u &= 0 && \text{on } \Gamma_{\text{no-slip}},\\
\nu(\nabla u + \nabla u^T)\boldsymbol{n} + p\boldsymbol{n} &= 0 && \text{on } \Gamma_{\text{out}},
\end{aligned}
$$

where $\Gamma_{\text{in}} := \{x = 0\} \cap \partial\Omega$, $\Gamma_{\text{no-slip}} := \overline{\partial\Omega \setminus (\Gamma_{\text{in}} \cup \Gamma_{\text{out}})}$ and $\Gamma_{\text{out}} := (\{x = 2.2\} \cap \partial\Omega) \setminus \partial(\{x = 2.2\} \cap \partial\Omega)$. Furthermore, the viscosity $u = 10^{-3}$ and $u_{\text{in}}(x, y) = (0.3w(y), 0)$ with $w(y) = 4y(H - y)/H^2$ and $H = 0.41$. Let $[H^1(\Omega)]^2_{BC} := \{u \in [H^1(\Omega)]^2 : u_{|\Gamma_{\text{in}}} = u_{\text{in}} \wedge u_{|\Gamma_{\text{no-slip}}} = 0\}$ and $[H^1_0(\Omega)]^2 := \{v \in [H^1(\Omega)]^2 : v_{|\Gamma_{\text{in}}} = 0 \wedge v_{|\Gamma_{\text{no-slip}}} = 0\}$. The corresponding weak form reads as follows: Find $\boldsymbol{u} = (u, p) \in V_{BC} := [H^1(\Omega)]^2_{BC} \times L^2(\Omega)$ such that

$$
A(\boldsymbol{u})(\boldsymbol{v}) = 0 \quad \forall \boldsymbol{v} = (v_u, v_p) \in V_0 := [H^1_0(\Omega)]^2 \times L^2(\Omega), \tag{1}
$$

with

$$
\begin{aligned}
A(\boldsymbol{u})(\boldsymbol{v}) := (\nu(\nabla u + \nabla u^T), \nabla v_u)_{[L^2(\Omega)]^{2\times 2}} + ((u \cdot \nabla)u, v_u)_{[L^2(\Omega)]^2}\\
+ (p, \mathrm{div}(v_u))_{L^2(\Omega)} - (\mathrm{div}(u), v_p)_{L^2(\Omega)}.
\end{aligned}
$$

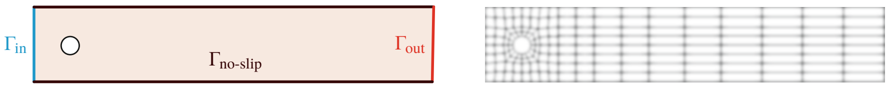


Fig. 1 The computational domain Ω (left) and the initial mesh (right)

2.2 Discretization

Let $\mathcal{T}_h$ be a decomposition of $\Omega \subset \mathbb{R}^2$ into quadrilateral elements. Furthermore, we assume that $\mathcal{T}_{\frac{h}{2}}$ is the uniform refinement of $\mathcal{T}_h$. We discretize our problem using piecewise bi-quadratic elements $[Q_c^2]^2$ for the velocity u, and piecewise bi-linear elements Q_c^1 for the pressure p. The resulting space using the mesh $\mathcal{T}_h$ will be denoted by V_h. For a more detailed explanation of the discretization, we refer to [4]. The resulting space using the mesh $\mathcal{T}_{\frac{h}{2}}$ will be denoted by $V_{\frac{h}{2}}$. We say $V_{\frac{h}{2}}$ is the (hierarchical) h-refined finite element space of V_h. Furthermore, we consider piecewise bi-quartic elements $[Q_c^4]^2$ for the velocity u, and piecewise bi-quadratic elements Q_c^2 for the pressure p. The resulting finite element space using the mesh $\mathcal{T}_h$ will be denoted by $V_h^{(2)}$. Here we have the property that $V_h \subset V_h^{(2)}$. We say $V_h^{(2)}$ is the (hierarchical) p-refined finite element space of V_h. The corresponding discretized problems read as: Find $\boldsymbol{u}_h \in V_h \cap V_{BC}$, $\boldsymbol{u}_{\frac{h}{2}} \in V_{\frac{h}{2}} \cap V_{BC}$ and $\boldsymbol{u}_h^{(2)} \in V_h^{(2)} \cap V_{BC}$ such that

$$\begin{aligned} A(\boldsymbol{u}_h)(\boldsymbol{v}_h) &= 0 \qquad \forall \boldsymbol{v}_h \in V_h \cap V_0, \\ A(\boldsymbol{u}_{\frac{h}{2}})(\boldsymbol{v}_{\frac{h}{2}}) &= 0 \qquad \forall \boldsymbol{v}_{\frac{h}{2}} \in V_{\frac{h}{2}} \cap V_0, \\ A(\boldsymbol{u}_h^{(2)})(\boldsymbol{v}_h^{(2)}) &= 0 \qquad \forall \boldsymbol{v}_h^{(2)} \in V_h^{(2)} \cap V_0. \end{aligned}$$

Remark 1 We would like to mention that the domain Ω is not of polygonal shape. Therefore, a decomposition into quadrilateral elements is not possible. However, we approximate the ball B by a polygonal domain, which is adapted after every refinement process by describing it as a spherical manifold in `deal.II` [1] using the command `Triangulation::set_manifold`.

3 Dual Weighted Residual Method and Error Representation

We are primarily interested in one or more particular quantities of interest. We employ the dual weighted residual (DWR) method [2] for estimating the error in these quantities. To connect the quantity of interest J with the model problem, we consider the adjoint problem.

3.1 The Adjoint Problem

The adjoint problem reads as follows: Find $\boldsymbol{z} \in V_0$ such that

$$A'(\boldsymbol{u})(\boldsymbol{v}, \boldsymbol{z}) = J'(\boldsymbol{u})(\boldsymbol{v}) \quad \forall \boldsymbol{v} \in V_0, \tag{2}$$

where A' and J' denote the Frechet derivative of A and J, respectively, and $\boldsymbol{u}$ is the solution of the model problem (1).

Theorem 1 *Let us assume that* $J \in C^3(V_{BC}, \mathbb{R})$. *If* $\boldsymbol{u}$ *solves the model problem (1) and* $\boldsymbol{z}$ *solves the adjoint problem (2), then, for arbitrary fixed* $\tilde{\boldsymbol{u}} \in V_{BC}$ *and* $\tilde{\boldsymbol{z}} \in V_0$, *the following error representation formula holds:*

$$J(\boldsymbol{u}) - J(\tilde{\boldsymbol{u}}) = \tfrac{1}{2}\rho(\tilde{\boldsymbol{u}})(\boldsymbol{z} - \tilde{\boldsymbol{z}}) + \tfrac{1}{2}\rho^*(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{z}})(\boldsymbol{u} - \tilde{\boldsymbol{u}}) + \rho(\tilde{\boldsymbol{u}})(\tilde{\boldsymbol{z}}) + \mathcal{R}^{(3)},$$

where $\rho(\tilde{\boldsymbol{u}})(\cdot) := -A(\tilde{\boldsymbol{u}})(\cdot)$, $\rho^*(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{z}})(\cdot) := J'(\tilde{\boldsymbol{u}})(\cdot) - A'(\tilde{\boldsymbol{u}})(\cdot, \tilde{\boldsymbol{z}})$, *and*

$$\mathcal{R}^{(3)} := \frac{1}{2}\int_0^1 [J'''(\tilde{\boldsymbol{u}} + s\boldsymbol{e})(\boldsymbol{e}, \boldsymbol{e}, \boldsymbol{e}) - A'''(\tilde{\boldsymbol{u}} + s\boldsymbol{e})(\boldsymbol{e}, \boldsymbol{e}, \boldsymbol{e}, \tilde{\boldsymbol{z}} + s\boldsymbol{e}^*) \\ -3A''(\tilde{\boldsymbol{u}} + s\boldsymbol{e})(\boldsymbol{e}, \boldsymbol{e}, \boldsymbol{e}^*)]s(s-1)\,ds, \qquad (3)$$

with $\boldsymbol{e} = \boldsymbol{u} - \tilde{\boldsymbol{u}}$ *and* $\boldsymbol{e}^* = \boldsymbol{z} - \tilde{\boldsymbol{z}}$.

Proof We refer the reader to [3] and [12]. □

Remark 2 In practice, the arbitrary elements $\tilde{\boldsymbol{u}} \in V_{BC}$ and $\tilde{\boldsymbol{z}} \in V_0$ will be replaced by approximations $\boldsymbol{u}_h$ and $\boldsymbol{z}_h$ to the corresponding finite element solutions.

Remark 3 The error representation formula in Theorem 1 is exact but not computable, because $\boldsymbol{u}$ and $\boldsymbol{z}$ are not known.

3.2 Error Estimation and Adaptive Algorithm

The different error estimator parts are discussed in [4]. In particular, it turns out that $\eta_h := \frac{1}{2}\rho(\tilde{\boldsymbol{u}})(\boldsymbol{z} - \tilde{\boldsymbol{z}}) + \frac{1}{2}\rho^*(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{z}})(\boldsymbol{u} - \tilde{\boldsymbol{u}})$ is related to the discretization error [3, 4, 12]. The idea is to replace the quantities $\boldsymbol{u} - \tilde{\boldsymbol{u}}$ and $\boldsymbol{z} - \tilde{\boldsymbol{z}}$ by some computable quantities. This can be done via higher order interpolation [2, 12] or hierarchically (via an additional solve on an enriched space) [2, 3, 10]. If $\boldsymbol{u}_{\boldsymbol{h}}^{+}, \boldsymbol{z}_{\boldsymbol{h}}^{+}$ are the solution, then we approximate $\boldsymbol{u} - \tilde{\boldsymbol{u}}$ and $\boldsymbol{z} - \tilde{\boldsymbol{z}}$ by $\boldsymbol{u}_{\boldsymbol{h}}^{+} - \tilde{\boldsymbol{u}}$ and $\boldsymbol{z}_{\boldsymbol{h}}^{+} - \tilde{\boldsymbol{z}}$, respectively. The new computable error estimator then reads as

$$\eta_h^+ := \frac{1}{2}\rho(\tilde{\boldsymbol{u}})(\boldsymbol{z}_{\boldsymbol{h}}^{+} - \tilde{\boldsymbol{z}}) + \frac{1}{2}\rho^*(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{z}})(\boldsymbol{u}_{\boldsymbol{h}}^{+} - \tilde{\boldsymbol{u}}).$$

Under some saturation assumption, it was shown in [4] that the resulting error estimator is efficient and reliable. We consider the two different error estimators

$$\eta_h^{(2)} := \tfrac{1}{2}\rho(\tilde{\boldsymbol{u}})(\boldsymbol{z}_h^{(2)} - \tilde{\boldsymbol{z}}) + \frac{1}{2}\rho^*(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{z}})(\boldsymbol{u}_h^{(2)} - \tilde{\boldsymbol{u}}),$$

$$\eta_{\frac{h}{2}} := \tfrac{1}{2}\rho(\tilde{\boldsymbol{u}})(\boldsymbol{z}_{\frac{h}{2}} - \tilde{\boldsymbol{z}}) + \frac{1}{2}\rho^*(\tilde{\boldsymbol{u}}, \tilde{\boldsymbol{z}})(\boldsymbol{u}_{\frac{h}{2}} - \tilde{\boldsymbol{u}}).$$

We call $\eta_h^{(2)}$ and $\eta_{\frac{h}{2}}$ the p enriched and h enriched error estimators, respectively. The error estimators are localized using the partition of unity technique proposed in [13]. The marking strategy and algorithms are the same as in [4].

Remark 4 The efficiency and reliability are not guaranteed under the corresponding saturation assumption in [4] for $\eta_{\frac{h}{2}}$, since the boundary is adapted in every refinement step.

Remark 5 We use the algorithm presented in [4]. The algorithm using p-enrichment coincides with Algorithm 3 in [4]. In the algorithm, where we use h enrichment, we replace $V_h^{(2)}$ by $V_{\frac{h}{2}}$.

4 Numerical Experiment

We compare the two error estimators introduced in Sect. 3.2. In the p-enriched case, we use uniform p-refinement for the hierarchical approximation. The results for p enrichment have already been computed in [4]. In the h-enriched case, we use uniform h-refinement. The configuration of the problem is given in Sect. 2.1.

4.1 Quantities of Interest

We use the quantities of interest defined in [4, 14]:

$$\begin{aligned} \Delta p(\boldsymbol{u}) &:= \qquad\qquad p(X_1) - p(X_2), \\ c_{\text{drag}}(\boldsymbol{u}) &:= C \int_{\partial B} \left[\nu(\nabla u + \nabla u^T)\boldsymbol{n} - p\boldsymbol{n}\right] \cdot \boldsymbol{e}_1 \ \mathrm{d}s_{(x,y)}, \\ c_{\text{lift}}(\boldsymbol{u}) &:= C \int_{\partial B} \left[\nu(\nabla u + \nabla u^T)\boldsymbol{n} - p\boldsymbol{n}\right] \cdot \boldsymbol{e}_2 \ \mathrm{d}s_{(x,y)}, \end{aligned}$$

where $C = 500$, $X_1 = (0.15, 0.2)$, $X_2 = (0.25, 0.2)$, $\boldsymbol{e}_1 := (1, 0)$, $\boldsymbol{e}_2 := (0, 1)$, and $\boldsymbol{n}$ denotes the outer normal vector. To do adaptivity for all of them at once, we combine them to one functional

$$J_{\mathfrak{E}}(\boldsymbol{v}_h) := \frac{|\Delta p(\boldsymbol{u}_h^+ - \boldsymbol{v}_h)|}{|\Delta p(\boldsymbol{u}_h)|} + \frac{|c_{\text{drag}}(\boldsymbol{u}_h^+ - \boldsymbol{v}_h)|}{|c_{\text{drag}}(\boldsymbol{u}_h)|} + \frac{|c_{\text{lift}}(\boldsymbol{u}_h^+ - \boldsymbol{v}_h)|}{|c_{\text{lift}}(\boldsymbol{u}_h)|}.$$

By $J_{\mathfrak{E}}^p$ or $J_{\mathfrak{E}}^h$, we denote the functionals where we replace $\boldsymbol{u}_h^+$ with $\boldsymbol{u}_h^{(2)}$ or $\boldsymbol{u}_{\frac{h}{2}}$, respectively. More information on how to treat multiple functionals at once can be found in [3–5, 7–9, 11, 15]. The implementation is done in the finite element library `deal.II` [1], and follows the code in [4]. In this section, we compare two different sequences of meshes. The sequences are generated by the error estimators $\eta_h^{(2)}$ and

$\eta_{\frac{h}{2}}$. First of all, let us define the effectivity indices by

$$I_{eff}^{p} := \frac{\eta_h^{(2)}}{|J_{\mathfrak{E}}^{p}(u) - J_{\mathfrak{E}}^{p}(u_h)|} \qquad \text{and} \qquad I_{eff}^{h} := \frac{\eta_{\frac{h}{2}}}{|J_{\mathfrak{E}}^{h}(u) - J_{\mathfrak{E}}^{h}(u_h)|}.$$

The p enriched discrete remainder part of the error estimator $\eta_{\mathcal{R}}^{(2)}$ is defined as the quantity (3), where we replace $\tilde{\boldsymbol{u}}, \tilde{z}, \boldsymbol{u}, z$ by $\boldsymbol{u}_h, z_h, \boldsymbol{u}_h^{(2)}, z_h^{(2)}$, respectively. The h-enriched discrete remainder part of the error estimator $\eta_{\mathcal{R},\frac{h}{2}}$ is defined as the quantity (3), where we replace $\tilde{\boldsymbol{u}}, \tilde{z}, \boldsymbol{u}, z$ by $\boldsymbol{u}_h, z_h, \boldsymbol{u}_{\frac{h}{2}}, z_{\frac{h}{2}}$, respectively. Finally, we define the gaps between the theoretical findings in [4] by

$$\eta_{\mathbb{E}}^{(2)} := \left| |J(\boldsymbol{u}_h^{(2)}) - J(\boldsymbol{u}_h)| - |\eta_h^{(2)} + \rho(\boldsymbol{u}_h, z_h) + \eta_{\mathcal{R}}^{(2)}| \right|,$$

and

$$\eta_{\mathbb{E},\frac{h}{2}} := \left| |J(\boldsymbol{u}_{\frac{h}{2}}) - J(\boldsymbol{u}_h)| - |\eta_{\frac{h}{2}} + \rho(\boldsymbol{u}_h, z_h) + \eta_{\mathcal{R},\frac{h}{2}}| \right|.$$

4.2 Discussion of the Results

In Fig. 2, the effectivity indices for the two different types of error estimators are shown on their respective grids. We see that h enrichment delivers effectivity indices which are very close to one, whereas, for p enrichment, we have effectivity indices in the range of $0.2 - 8.1$. This was also observed in [4]. In the case of p enrichment, the saturation assumption is violated multiple times, as we observe in Fig. 3. The saturation assumption is violated if the error $|J_{\mathfrak{E}}^{p}(\boldsymbol{u}_h^{(2)}) - J_{\mathfrak{E}}^{p}(\boldsymbol{u})|$ in the enriched solution is larger than $|J_{\mathfrak{E}}^{p}(\boldsymbol{u}_h) - J_{\mathfrak{E}}^{p}(\boldsymbol{u})|$. In the case of h enrichment, this always happens. If we compare the errors of the single functionals, which are monitored in Figs. 4, 5 and 6, we conclude that the meshes generated by the p enriched error estimator lead to smaller errors in the single functionals. If all the conditions in [4] are fulfilled, then $\eta_{\mathbb{E}}^{(2)}$ and $\eta_{\mathbb{E},\frac{h}{2}}$ are zero. However, in the computation of the error estimators, our overall round-off error is in the order of $\varepsilon(\text{double}) \times \text{DOFs}$, where $\varepsilon(\text{double}) = 2^{-52}$ is the machine precision for double floating point numbers.[1] In the case of p enrichment, we observe in Fig. 7 that $\eta_{\mathbb{E}}^{(2)}$ indeed is in the order or even better than the round off errors when summing up the different error contributions. In this case, all requirements are fulfilled. For h enrichment, we do not have the inclusion $V_h \subset V_{\frac{h}{2}}$ due to the geometrical approximation. Therefore, these conditions are violated. The effects are monitored in Fig. 7 as well. The quantity

[1] https://en.wikipedia.org/wiki/Machine_epsilon.

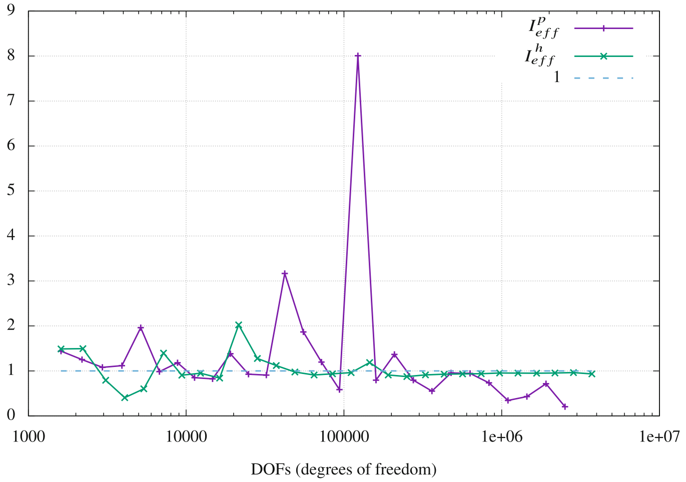


Fig. 2 The two effectivity indices on the corresponding meshes

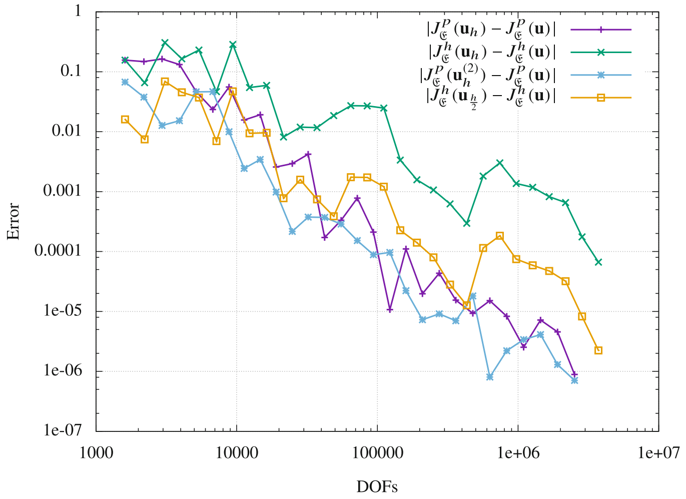


Fig. 3 Errors in $J^p_{\mathfrak{E}}$ and $J^h_{\mathfrak{E}}$ at the solution and the enriched solution

$\eta_{\mathbb{E},\frac{h}{2}}$ does not only contain numerical round off errors, but also errors coming from the geometrical approximation. However, this is a non-local quantity, and the localization is not straightforward.

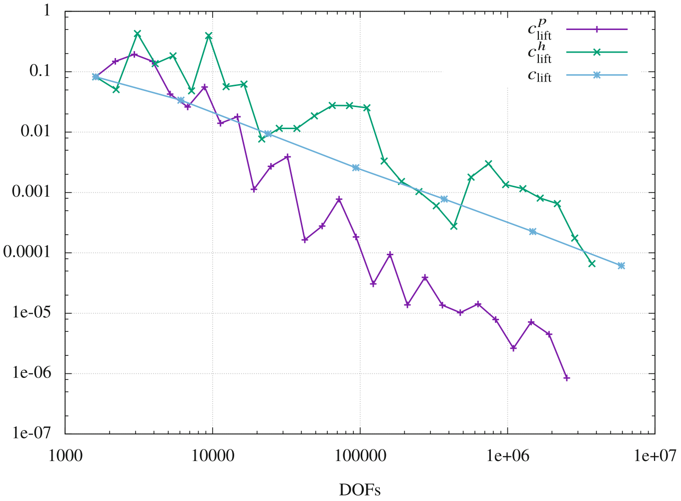


Fig. 4 The errors in c_{lift} for refinement with p enriched error estimation (c^p_{lift}), refinement with h enriched error estimation (c^h_{lift}), and uniform refinement (c_{lift})

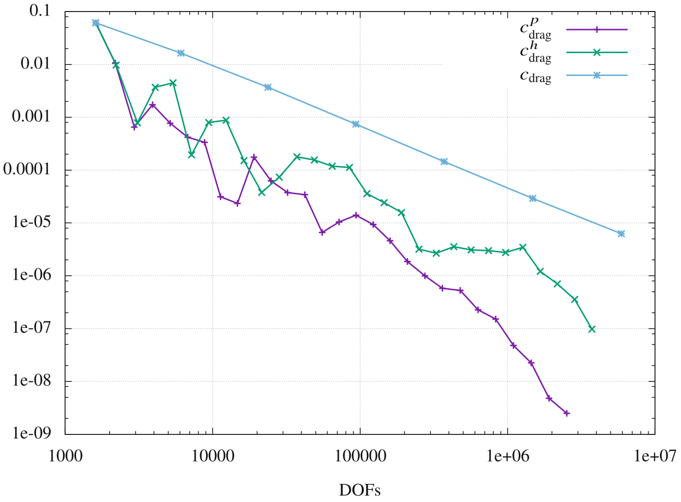


Fig. 5 The errors in c_{drag} for refinement with p enriched error estimation (c^p_{drag}), refinement with h enriched error estimation (c^h_{drag}), and uniform refinement (c_{drag})

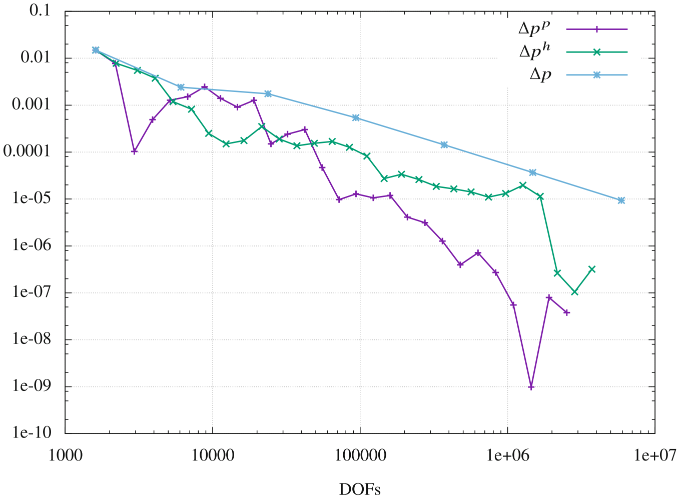


Fig. 6 The errors in Δp for refinement with p enriched error estimation (Δp^p), refinement with h enriched error estimation (Δp^h), and uniform refinement (Δp)

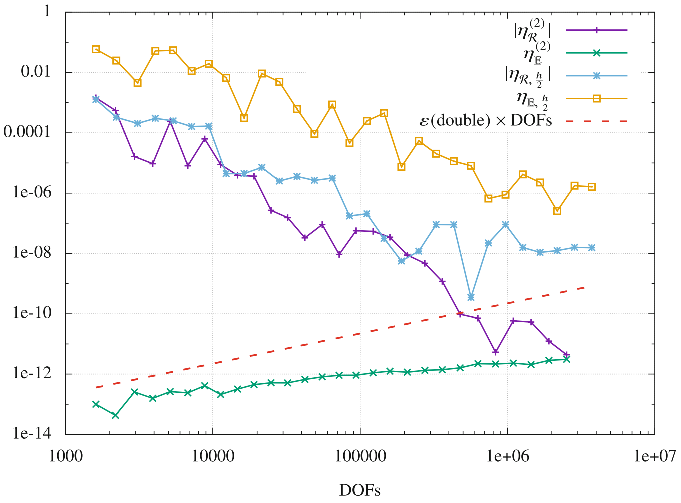


Fig. 7 The remainder parts $\eta_{\mathcal{R}}^{(2)}$, $\eta_{\mathcal{R},\frac{h}{2}}$ and gap parts $\eta_{\mathbb{E}}^{(2)}$, $\eta_{\mathbb{E},\frac{h}{2}}$ for p and h enrichment

Acknowledgments This work has been supported by the Austrian Science Fund (FWF) under the grant P 29181 'Goal-Oriented Error Control for Phase-Field Fracture Coupled to Multiphysics Problems'. Furthermore, we thank Daniel Jodlbauer for fruitful discussions.

References

1. G. Alzetta, D. Arndt, W. Bangerth, V. Boddu, B. Brands, D. Davydov, R. Gassmöller, T. Heister, L. Heltai, K. Kormann, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. The `deal.II` library, version 9.0. *J. Numer. Math.*, 26(4):173–183, 2018.
2. R. Becker and R. Rannacher. An optimal control approach to a posteriori error estimation in finite element methods. *Acta Numer.*, 10:1–102, 2001.
3. B. Endtmayer, U. Langer, and T. Wick. Multigoal-oriented error estimates for non-linear problems. *J. Numer. Math.*, 27(4):215–236, 2019.
4. B. Endtmayer, U. Langer, and T. Wick. Two-Side a Posteriori Error Estimates for the Dual-Weighted Residual Method. *SIAM J. Sci. Comput.*, 42(1):A371–A394, 2020.
5. B. Endtmayer and T. Wick. A Partition-of-Unity Dual-Weighted Residual Approach for Multi-Objective Goal Functional Error Estimation Applied to Elliptic Problems. *Comput. Methods Appl. Math.*, 17(4):575–599, 2017.
6. V. Girault and P.-A. Raviart. *Finite Element method for the Navier-Stokes equations: theory and algorithms*. Number 5 in Computer Series in Computational Mathematics. Springer-Verlag, 1986.
7. R. Hartmann. Multitarget error estimation and adaptivity in aerodynamic flow simulations. *SIAM J. Sci. Comput.*, 31(1):708–731, 2008.
8. R. Hartmann and P. Houston. Goal-oriented a posteriori error estimation for multiple target functionals. In *Hyperbolic problems: theory, numerics, applications*, pages 579–588. Springer, Berlin, 2003.
9. K. Kergrene, S. Prudhomme, L. Chamoin, and M. Laforest. A new goal-oriented formulation of the finite element method. *Comput. Methods Appl. Mech. Engrg.*, 327:256–276, 2017.
10. U. Köcher, M. P. Bruchhäuser, and M. Bause. Efficient and scalable data structures and algorithms for goal-oriented adaptivity of space–time FEM codes. *SoftwareX*, 10:100239, 2019.
11. D. Pardo. Multigoal-oriented adaptivity for hp-finite element methods. *Procedia Computer Science*, 1(1):1953–1961, 2010.
12. R. Rannacher and J. Vihharev. Adaptive finite element analysis of nonlinear problems: balancing of discretization and iteration errors. *J. Numer. Math.*, 21(1):23–61, 2013.
13. T. Richter and T. Wick. Variational localizations of the dual weighted residual estimator. *J. Comput. Appl. Math.*, 279:192–208, 2015.
14. M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. Benchmark computations of laminar flow around a cylinder. In *Flow simulation with high-performance computers II*, pages 547–566. Springer, 1996.
15. E. H. van Brummelen, S. Zhuk, and G. J. van Zwieten. Worst-case multi-objective error estimation and adaptivity. *Comput. Methods Appl. Mech. Engrg.*, 313:723–743, 2017.

Towards Confident Bayesian Parameter Estimation in Stochastic Chemical Kinetics

Stefan Engblom, Robin Eriksson, and Pedro Vilanova

Abstract We investigate the feasibility of Bayesian parameter inference for chemical reaction networks described in the low copy number regime. Here stochastic models are often favorable implying that the Bayesian approach becomes natural. Our discussion circles around a concrete oscillating system describing a circadian rhythm, and we ask if its parameters can be inferred from observational data. The main challenge is the lack of analytic likelihood and we circumvent this through the use of a synthetic likelihood based on summarizing statistics. We are particularly interested in the robustness and confidence of the inference procedure and therefore estimates *a priori* as well as *a posteriori* the information content available in the data. Our all-synthetic experiments are successful but also point out several challenges when it comes to real data sets.

1 Introduction

Systems Biology deals with the study of complex biological systems underpinning biological life, in particular chemical reaction networks (CRNs) and their qualitative properties. At low copy numbers, the dynamics of CRNs is accurately modeled as a continuous-time Markov chain. Such stochastic models are more difficult to analyze than their deterministic counterparts, but have gained widespread recognition since experimentalists have noticed that observed data variations can consistently be attributed to the inherent network noise. Prototypical cases for pronounced stochastic effects include gene transcription regulation processes [2, 14], robustness in

S. Engblom (✉) · R. Eriksson
Division of Scientific Computing, Department of Information Technology, Uppsala university, Uppsala, Sweden
e-mail: stefane@it.uu.se; robin.eriksson@it.uu.se

P. Vilanova
Department of Mathematical Sciences, New Jersey Institute of Technology, Newark, NJ, USA

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_36

biological oscillations [1, 10], stationary behavior [15], and cellular reprogramming [12].

Through the rapid development of imaging- and sequencing techniques, interfacing CRN models with data becomes an important and, as it turns out, a very challenging problem. For models driven by intrinsic stochasticity, *Bayesian* approaches are favorable since they are formulated in a consistent probabilistic framework. One of the major obstacles is the *likelihood function*, the conditional probability of the data given a proposed parameter. Computing the likelihood formally requires the solutions to high-dimensional forward evolution equations, e.g., the *Chemical Master Equation* (CME) [4–6]. Another challenge stems from the fact that the comparably large levels of noise forces the data volumes to go up to ensure an accurate identification. This also means that the problem of *identifiability* should be addressed, preferably already in a prior phase.

In this paper we attempt to highlight the feasibility of the Bayesian approach for CRNs. We select a fairly challenging network in the form of a *Circadian rhythm* [1] and set out to invert synthetic data from this model via Bayesian methods. To circumvent the issue with high-dimensional density equations, we follow the idea in [17] and develop suitable *summarizing statistics* for which the limiting distributions are multivariate normal, and the computational procedure then becomes completely simulation-driven. We stress the identifiability of the model by investigating its information theoretic properties, in a prior as well as in a posterior setting. We find that our proposed Bayesian set-up is promising in the setting of CRNs, but we also point out a few challenges.

2 Bayesian Inversion of the Circadian Rhythm

We investigate the qualities of a Bayesian approach to parameter inversion of CRNs by looking at a specific example in the form of an oscillating system modeling *Circadian clocks* as specified in Sect. 2.1. A prior investigation of the information content in synthetic measurements is presented in Sect. 2.2 and summarizing statistics are selected in Sect. 2.3. The obtained posterior distribution itself is finally investigated in Sect. 2.4.

2.1 The Circadian Rhythm by Vilar et al.

As a challenging example for Bayesian inversion we propose to use the circadian clock by Vilar *et al.* [1]. A circadian clock enjoys an oscillating dynamics and is used by various organisms to keep track of time. The model is defined by 9 species and 18 transitions,

$$\left.\begin{array}{rcl} D'_a & \xrightarrow{\theta_a D'_a} & D_a \\ D_a + A & \xrightarrow{\gamma_a D_a A} & D'_a \\ D'_r & \xrightarrow{\theta_r D'_r} & D_r \\ D_r + A & \xrightarrow{\gamma_r D_r A} & D'_r \end{array}\right\} \quad \begin{array}{l} \left.\begin{array}{rcl} \emptyset & \xrightarrow{\alpha'_a D'_a} & M_a \\ \emptyset & \xrightarrow{\alpha_a D_a} & M_a \\ M_a & \xrightarrow{\delta_{ma} M_a} & \emptyset \end{array}\right\} \\ \left.\begin{array}{rcl} \emptyset & \xrightarrow{\alpha'_r D'_r} & M_r \\ \emptyset & \xrightarrow{\alpha_r D_r} & M_r \\ M_r & \xrightarrow{\delta_{mr} M_r} & \emptyset \end{array}\right\} \end{array} \quad \left.\begin{array}{rcl} \emptyset & \xrightarrow{\beta_a M_a} & A \\ \emptyset & \xrightarrow{\theta_a D'_a} & A \\ \emptyset & \xrightarrow{\theta_r D'_r} & A \\ A & \xrightarrow{\delta_a A} & \emptyset \\ A + R & \xrightarrow{\gamma_c A R} & C \end{array}\right\} \quad \left.\begin{array}{rcl} \emptyset & \xrightarrow{\beta_r M_r} & R \\ R & \xrightarrow{\delta_r R} & \emptyset \\ C & \xrightarrow{\delta_a C} & R \end{array}\right\}$$

In terms of which the rate parameters are

α_a	50	α_r	0.01	β_a	50	γ_a	1	γ_c	2	δ_{ma}	10	δ_a	1	θ_a	50
α'_a	500	α'_r	50	β_r	5	γ_r	1			δ_{mr}	0.5	δ_r	0.20	θ_r	100

Interestingly, certain choices of the parameters are known to produce oscillating behavior using a stochastic formulation, whereas a deterministic formulation rapidly reaches a steady-state solution [16]. Hence the noise has a stabilizing effect on the rhythm making it an interesting test case.

In this work we assume perfect measurements consisting of samples of R and C gathered every 2 min for 2 h. This data, chiefly shown in Fig. 4, consists in all of roughly 5 full periods and we now set out to transform this data to posterior distributions of the model parameters.

2.2 Prior Information Content

The Hessian associated with the path-wise relative entropy, corresponding to a parameter vector θ, contains the information of the sensitive directions of the system, in the sense of the local index

$$S_k := \frac{\partial}{\partial \theta_k} \mathbb{E}_{P^\theta}[g(X_t)]\,,$$

for a suitable path-wise observable g of the stochastic reaction network process X_t with density P^θ. The following sensitivity bound holds:

$$|S_k| \leq \sqrt{\mathrm{Var}(g)}\sqrt{\mathcal{I}_{k,k}(P^\theta)}\,, \quad k = 1, \ldots, K,$$

where $\mathcal{I}(P^\theta)$ is the path-wise Fisher Information Matrix (pFIM) of the K-dimensional parameter vector θ (see [3]).

The FIM constitutes a classical criterion for local parameter identifiability, since it is a measure of how the process changes in response to infinitesimal changes in the parameters, in view of the expansion

$$\mathcal{R}(P^{\theta}|P^{\theta+\epsilon}) = \frac{1}{2}\epsilon^{T}\mathcal{I}(P^{\theta})\epsilon + O(|\epsilon|^{3}),$$

where $\mathcal{R}(P^{\theta}|P^{\theta+\epsilon})$ is the relative entropy (RE), which is the loss of information when ϵ-perturbing the parametric density P^{θ}. Thus, the FIM is the Hessian of the RE which describes the local curvature around the minimum value of the RE [3, 9, 13].

In the present context, the parameter vector θ is locally identifiable if and only if the pFIM has full rank [11], which is the case in the model considered in this work. In this respect, all the parameters of the model are, in principle, identifiable, but in practice the conditioning of the problem may well be prohibitive. The conditioning translates to small eigenvalues. In Fig. 1 we show the eigenvalues of the pFIM under a uniform prior distribution in $[0, 3\theta_0]$, where θ_0 are the parameters from Sect. 2.1.

The pFIM is also a key tool to parameter inference via the Cramer-Rao bound, since the inverse diagonal elements of the pFIM constitute a lower-bound for the variance of any unbiased estimator of elements of θ. From this and from the fact that the median eigenvalue corresponding to α_r under our prior is close to 10^{-6}, we argue that α_r cannot be retrieved at our sample volume of data. We therefore set a sharp delta-function prior at the true value for this parameter.

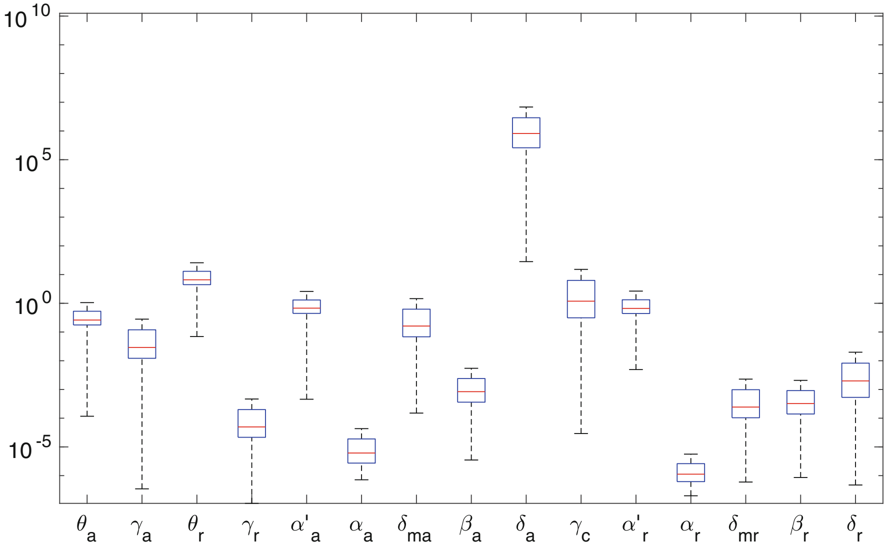


Fig. 1 Eigenvalues of the pFIM under a uniform prior in $[0, 3\theta_0]$

2.3 Summarizing Statistics

The Circadian rhythm has an oscillating dynamics and is challenging to summarize by suitable statistics. We propose 15 statistics consisting of first and second moments and statistics from the frequency domain. The list of the used statistics includes the maximum, the mean, the standard deviation, the 91% percentile, the index of dispersion (ID), the amplitude of fast Fourier transform (FFT) coefficients 1 and 6, and the correlation between R and C.

The summarizing statistics (SS) takes on some distribution given measurements of the stochastic process. The synthetic likelihood (SL) approach is convenient provided the distribution is multivariate normal [17]. We thus investigate the marginal distribution for each of the SS to support that the SL ansatz is reasonable. See Fig. 2 for selected examples.

Next we need to support that the SL includes enough information for parameter inversion. We investigated the convexity along each model parameter dimension as follows. We perturb one parameter at a time around the true value θ_0 and sample the SL on a grid, constructing a 95% credible interval (CI) for the SL and calculate the

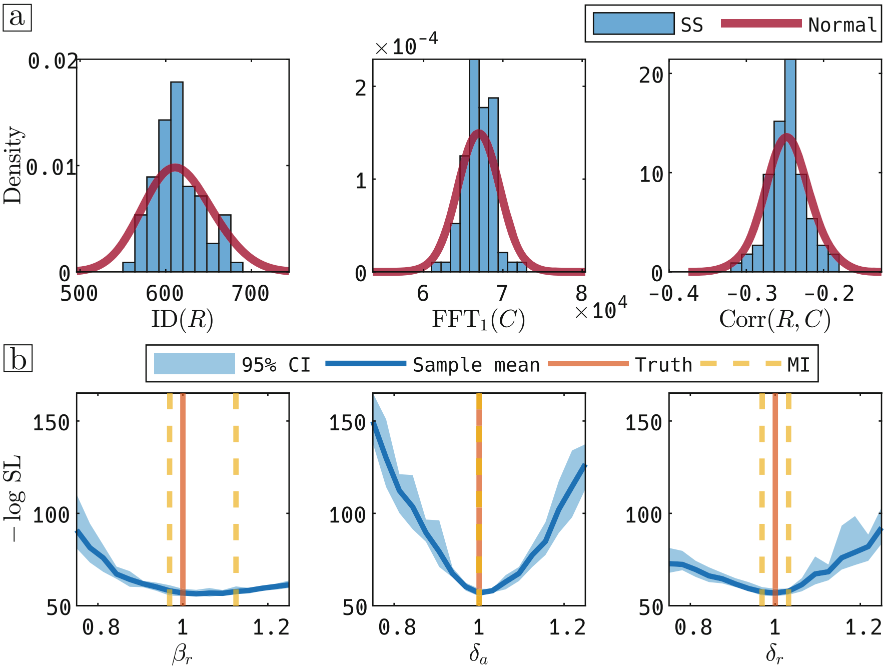


Fig. 2 (**a**) The distribution of 80 samples of three of the chosen summary statistics. From left to right: index of dispersion of R, amplitude of the dominating Fourier coefficient of C, and the correlation between R and C. The solid red line is a normal fit. (**b**) Check of single dimension convexity for three rates (scaled by the true values), together with the minimum interval (MI)

mean. We search for the minimum SL on the mean, and if this minimum is included in the CI of any other grid-point, that point is added to the minimum interval (MI). A narrow MI is of course indicative of a well defined parameter space. Sample outcomes of this procedures are illustrated in Fig. 2.

2.4 Posterior Distribution and Prediction

We employ an efficient Bayesian sampler, adaptive Metropolis (AM) [8], together with the SL induced by our SS. For more information on the combination of SL in AM ("SLAM"), see [7]. The prior knowledge we supply is a uniform prior $[0, 3\theta]$, motivated simply by considering too large values of the rate parameters to be unphysical. Note that the prior is unsymmetric around the true values.

In Fig. 3, we summarize the resulting posterior from $1.2 \cdot 10^5$ posterior samples after removal of burn-in. All 14 parameter values were retrieved reasonably well with our set-up, albeit with some outliers (notably γ_r and θ_r).

To evaluate the qualities of the posterior distribution, we can assess the *posterior residual*, i.e., a posterior predictive check of how well generated data from the posterior agrees with the observations. In Fig. 4, we evaluate 2000 realized samples

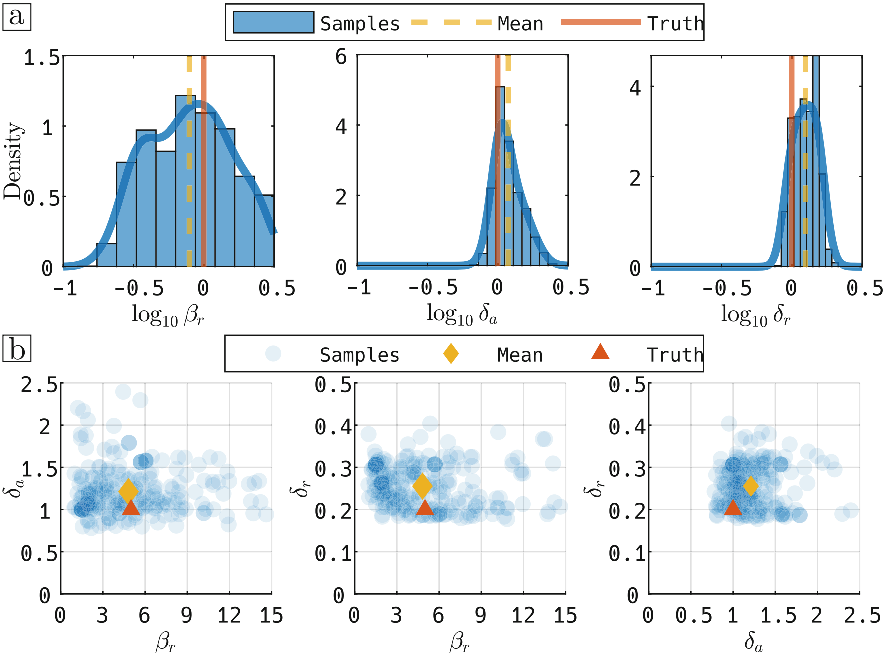


Fig. 3 (**a**) The marginal posterior scaled by the true value for three of the rates. (**b**) Sample pairwise marginal posterior density

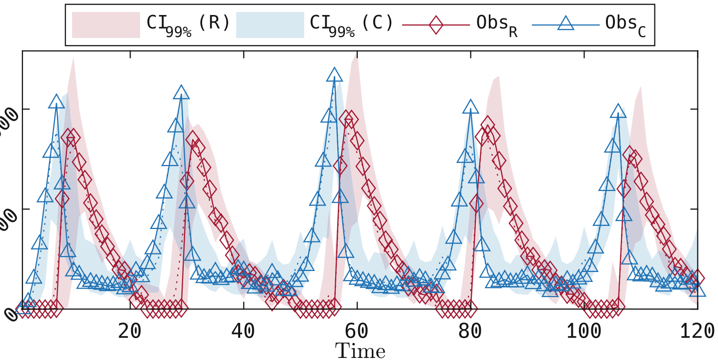


Fig. 4 2000 posterior samples demonstrating the posterior residual with a 99% CI. As an aid in visualization and to address the diverging phase of the model, the simulations are restarted from the true observations on every 5th data point. The displayed CI is then constructed from this sample CI by smoothing through an exponential moving average of window size 3

from the posterior for which we compute the 99% CI. All in all the posterior predictor covers the data rather convincingly.

3 Discussion

We have demonstrated that synthetic inversion of the Circadian rhythm is doable for qualities of data which appear reasonable to achieve under experimental conditions. During the course of experimenting it was noted that a bounded prior was necessary in order to obtained bounded posteriors. An extended set of summarizing statistics would likely make the problem more definite and allow also for non-informative prior distributions. Importantly, the initial analysis of the information content in measurements revealed that one parameter could not be retrieved from data and, in fact, that the associated reaction channel could be removed fully. We conclude that Bayesian methods are attractive in the context of inversion of CRNs, but that both the set-up and the end-result should be subjected to careful synthetic tests before considering real experimental data.

References

1. N. Barkai and S. Leibler. Circadian clocks limited by noise. *Nature*, 403:267–268, 2000. https://doi.org/10.1038/35002258.
2. W. J. Blake, M. Kærn, C. R. Cantor, and J. J. Collins. Noise in eukaryotic gene expression. *Nature*, 422(6932):633–637, 2003.

3. P. Dupuis, M. A. Katsoulakis, Y. Pantazis, and P. Plecháč. Path-space information bounds for uncertainty quantification and sensitivity analysis of stochastic dynamics. *SIAM/ASA Journal on Uncertainty Quantification*, 4(1):80–111, 2016.
4. S. Engblom. Galerkin spectral method applied to the chemical master equation. *Commun. Comput. Phys.*, 5(5):871–896, 2009.
5. S. Engblom. Spectral approximation of solutions to the chemical master equation. *J. Comput. Appl. Math.*, 229(1):208–221, 2009. https://doi.org/10.1016/j.cam.2008.10.029.
6. S. Engblom and V. Sunkara. Preconditioned Metropolis sampling as a strategy to improve efficiency in posterior exploration. *IFAC-PapersOnLine*, 49(26):89–94, 2016. https://doi.org/10.1016/j.ifacol.2016.12.108. Foundations of Systems Biology in Engineering, FOSBE 2016.
7. S. Engblom, R. Eriksson, and S. Widgren: Bayesian epidemiological modeling over high-resolution network data. *Epidemics*, 32, 2020. https://doi.org/10.1016/j.epidem.2020.100399.
8. H. Haario, E. Saksman, and J. Tamminen. An adaptive Metropolis algorithm. *Bernoulli*, 7(2): 223–242, 2001. https://doi.org/10.2307/3318737.
9. M. A. Katsoulakis and P. Vilanova. Data-driven, variational model reduction of high-dimensional reaction networks. *Journal of Computational Physics*, 401:108997, 2020. ISSN 0021–9991. https://doi.org/10.1016/j.jcp.2019.108997.
10. B. N. Kholodenko. Negative feedback and ultrasensitivity can bring about oscillations in the mitogen-activated protein kinase cascades. *European Journal of Biochemistry*, 267(6):1583–1588, 2000.
11. M. Komorowski, M. J. Costa, D. A. Rand, and M. P. H. Stumpf. Sensitivity, robustness, and identifiability in stochastic chemical kinetics models. *Proceedings of the National Academy of Sciences*, 108(21):8645–8650, 2011. ISSN 0027-8424. https://doi.org/10.1073/pnas.1015814108.
12. B. D. MacArthur, A. Ma'ayan, and I. R. Lemischka. Systems biology of stem cell fate and cellular reprogramming. *Nature Reviews Molecular Cell Biology*, 10(10):672–681, 2009.
13. Y. Pantazis, M. Katsoulakis, and D. Vlachos. Parametric sensitivity analysis for biochemical reaction networks based on pathwise information theory. *BMC Bioinformatics*, 14(1):311, 2013. ISSN 1471-2105. https://doi.org/10.1186/1471-2105-14-311.
14. J. Paulsson, O. G. Berg, and M. Ehrenberg. Stochastic focusing: Fluctuation-enhanced sensitivity of intracellular regulation. *Proc. Natl. Acad. Sci. USA*, 97(13):7148–7153, 2000. https://doi.org/10.1073/pnas.110057697.
15. Y. Togashi and K. Kaneko. Molecular discreteness in reaction-diffusion systems yields steady states not seen in the continuum limit. *Phys. Rev. E*, 70(2):020901–1, 2004. https://doi.org/10.1103/PhysRevE.70.020901.
16. J. M. G. Vilar, H. Y. Kueh, N. Barkai, and S. Leibler. Mechanism of noise-resistance in genetic oscillators. *Proc. Natl. Acad. Sci. USA*, 99:5988–5992, 2002. https://doi.org/10.1073/pnas.092133899.
17. S. N. Wood. Statistical inference for noisy nonlinear ecological dynamic systems. *Nature*, 466 (7310):1102–1104, 2010. https://doi.org/10.1038/nature09319.

Strategies for the Vectorized Block Conjugate Gradients Method

Nils-Arne Dreier and Christian Engwer

Abstract Block Krylov methods have recently gained a lot of attraction. Due to their increased arithmetic intensity they offer a promising way to improve performance on modern hardware. Recently Frommer et al. (Electron Trans Numer Anal 47:100–126, 2017). presented a block Krylov framework that combines the advantages of block Krylov methods and data parallel methods. We review this framework and apply it on the Block Conjugate Gradients method, to solve linear systems with multiple right hand sides. In this course we consider challenges that occur on modern hardware, like a limited memory bandwidth, the use of SIMD instructions and the communication overhead. We present a performance model to predict the efficiency of different Block CG variants and compare these with experimental numerical results.

1 Introduction

Developers of numerical software are facing multiple challenges on modern HPC-hardware. Firstly, multiple levels of concurrency must be exploited to achieve the full performance. Secondly, due to that parallelism, communication between nodes is needed, which must be cleverly organized to avoid an expensive overhead. And most importantly, modern CPUs have a low memory bandwidth, compared to the peak FLOP rate, such that for standard linear solvers the memory bandwidth is the bottleneck for the performance. Therefore only algorithms with a high arithmetic intensity will perform well.

Instruction level parallelism is now apparent on all modern CPU architectures. They provide dedicated vector (or SIMD) instructions, that allow to proceed multiple floating point operations with one instruction call, e.g. AVX-512 allows

N.-A. Dreier · C. Engwer (✉)
University of Münster, Münster, Germany
e-mail: n.dreier@uni-muenster.de; c.engwer@uni-muenster.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_37

processing 8 `double` at once. The efficient use of these instructions is a further challenge.

The Conjugate Gradients method (CG) is a standard tool for solving large, sparse, symmetric, positive definite, linear systems. The Block Conjugate Gradient (BCG) method was introduced in the 1980s to improve the convergence rate for systems with multiple right hand sides [12]. Recently these methods have been rediscovered to reduce the communication overhead in parallel computations [1, 6, 7].

In this paper we present a generalization of the BCG method, which makes it applicable to arbitrary many right-hand-sides. We consider a symmetric, positive definite matrix $A \in \mathbb{R}^{n\times n}$ and want to solve the matrix equation

$$AX = B, \qquad \text{with } B, X \in \mathbb{R}^{n\times k}. \tag{1}$$

This paper is structured as follows. In Sect. 2 we briefly review the theoretical background of block Krylov methods, using the notation of [5]. Then in Sect. 3 we apply this theory on the BCG method. The implementation of the method, a theoretical performance model and some numerical experiments are presented in Sect. 4.

2 Block Krylov Subspaces

Considering functions of matrices, Frommer et al. presented in [5] a generic framework for block Krylov methods. Further work on this framework can be found in the PhD thesis of Lund [11]. In the following we review the most important definitions, which we will need in Sect. 3. Frommer et al. used $\mathbb{C}$ as numeric field, for simplicity of the following numerics, we restrict our self to $\mathbb{R}$.

Definition 1 (Block Krylov Subspace) Let $\mathbb{S}$ be a *-subalgebra of $\mathbb{R}^{k\times k}$ and $R \in \mathbb{R}^{n\times k}$. The m-th block Krylov subspace with respect to A, R and $\mathbb{S}$ is defined by

$$\mathcal{K}^m_{\mathbb{S}}(A,R) = \left\{ \sum_{i=0}^{m-1} A^i R c_i \,\middle|\, c_0,\dots,c_{m-1} \in \mathbb{S} \right\} \subset \mathbb{R}^{n\times k}. \tag{2}$$

From that definition we find the following lemma directly.

Lemma 1 *If $\mathbb{S}_1$ and $\mathbb{S}_2$ are two *-subalgebras of $\mathbb{R}^{n\times n}$, with $\mathbb{S}_1 \subseteq \mathbb{S}_2$. Then*

$$\mathcal{K}^m_{\mathbb{S}_1}(A,R) \subseteq \mathcal{K}^m_{\mathbb{S}_2}(A,R) \tag{3}$$

holds.

In this paper we want to consider the following *-subalgebras and $\mathbb{S}$-products and the corresponding Krylov spaces.

Definition 2 (Relevant *-Subalgebras) Let $p \in \mathbb{N}$ be a divider of k. We define the following *-subalgebras and corresponding products:

$$\begin{aligned}
&\text{hybrid:} && \mathbb{S}^p_{\mathrm{Hy}} := \operatorname{diag}\left(\left(\mathbb{R}^{p\times p}\right)^q\right) && \Rightarrow && \langle\!\langle \cdot,\cdot \rangle\!\rangle_{\mathbb{S}^p_{\mathrm{Hy}}} = \operatorname{diag}\left(X_1^*Y_1,\ldots,X_q^*Y_q\right),\\
&\text{global:} && \mathbb{S}^p_{\mathrm{Gl}} := \mathbb{R}^{p\times p}\otimes I_q && \Rightarrow && \langle\!\langle \cdot,\cdot \rangle\!\rangle_{\mathbb{S}^p_{\mathrm{Gl}}} = \sum_{i=0}^{q} X_i^*Y_i \otimes I_q,
\end{aligned}$$

where I_q denotes the q dimensional identity matrix and $\operatorname{diag}\left(\left(\mathbb{R}^{p\times p}\right)^q\right)$ denotes the set of $k\times k$ matrices where only the $p\times p$ diagonal matrices have non-zero values. Furthermore we define the special cases

$$\begin{aligned}
&\text{classical:} && \mathbb{S}_{\mathrm{Cl}} := \mathbb{R}^{k\times k} = \mathbb{S}^k_{\mathrm{Hy}} = \mathbb{S}^k_{\mathrm{Gl}} && \text{and}\\
&\text{parallel:} && \mathbb{S}_{\mathrm{Pl}} := \operatorname{diag}\left(\mathbb{R}^k\right) = \mathbb{S}^1_{\mathrm{Hy}}.
\end{aligned}$$

The names result from the behavior of the resulting Krylov method; $\mathbb{S}_{\mathrm{Cl}}$ yields in the classical block Krylov method as presented by O'Leary [12], whereas $\mathbb{S}_{\mathrm{Pl}}$ results in a CG method, which is carried out for all right hand sides simultaneously, in a instruction level parallel fashion.

From that definition we could conclude the following embedding lemma.

Lemma 2 (Embeddings of *-Subalgebras) *For $p_1, p_2 \in \mathbb{N}$, where p_1 is a divisor of p_2 and p_2 is a divisor of k, we have the following embedding:*

$$\begin{array}{ccccccc}
\mathbb{S}_{Pl} & \subseteq & \mathbb{S}^{p_1}_{Hy} & \subseteq & \mathbb{S}^{p_2}_{Hy} & \subseteq & \mathbb{S}_{Cl}\\
\cup & & \rotatebox{90}{$\subseteq$} & & \rotatebox{90}{$\subseteq$} & & \parallel\\
\mathbb{S}^1_{Gl} & \subseteq & \mathbb{S}^{p_1}_{Gl} & \subseteq & \mathbb{S}^{p_2}_{Gl} & \subseteq & \mathbb{S}_{Cl}
\end{array} \tag{4}$$

3 Block Conjugate Gradient Method

Algorithm 1 shows the preconditioned BCG method. We recompute ρ^{i-1} in line 6 to improve the stability of the method. A more elaborate discussion of the stability of the BCG method can be found in the paper of Dubrulle [3]. This stabilization has only mild effect on the performance as the communication that is needed to compute the block product could be carried out together with the previous block product.

The algorithm is build-up from four kernels:

- `BDOT`: Computing the block product, $\gamma \leftarrow \langle\!\langle X, Y\rangle\!\rangle_{\mathbb{S}}$
- `BAXPY`: Generic vector update $X \leftarrow X + Y\gamma$
- `BOP`: Applying the operator (or preconditioner) on a block vector $Y \leftarrow AX$
- `BSOLVE`: Solve a block system in the *-subalgebra $\delta \leftarrow \gamma^{-1}\delta$

Algorithm 1 Preconditioned Block Conjugate Gradients method (stabilized)

```
$R^0 \leftarrow B - AX^0$
$P^1 \leftarrow M^{-1}R^0$
for $i = 1, \dots$ to convergence do
    $Q^i \leftarrow AP^i$
    $\alpha^i \leftarrow \langle\!\langle P^i, Q^i \rangle\!\rangle_{\mathbb{S}}$
    $\rho^{i-1} \leftarrow \langle\!\langle P^i, R^{i-1} \rangle\!\rangle_{\mathbb{S}}$                ▷ recompute
    $\lambda^i \leftarrow \left(\alpha^i\right)^{-1} \rho^{i-1}$
    $X^i \leftarrow X^{i-1} + P^i\lambda^i$
    $R^i \leftarrow R^{i-1} - Q^i\lambda^i$
    $Z^{i+1} \leftarrow M^{-1}R^i$
    $\rho^i \leftarrow \langle\!\langle Z^{i+1}, R^i \rangle\!\rangle_{\mathbb{S}}$
    $\beta^i \leftarrow {\rho^{i-1}}^{-1}\rho^i$
    $P^{i+1} \leftarrow Z^{i+1} - P^i\beta^i$
end for
```

O'Leary showed the following convergence result to estimate the error of the classical BCG method.

Theorem 1 (Convergence of Block Conjugate Gradients [12, Theorem 5]) *For the energy-error of the s-th column $\|e_s^i\|_A$ of the classical BCG method, the following estimation hold:*

$$\|e_s^i\|_A \leq c_1\mu^i$$

$$\textit{with } \mu = \frac{\sqrt{\kappa_k} - 1}{\sqrt{\kappa_k} + 1}, \kappa_k = \frac{\lambda_n}{\lambda_k} \textit{ and constant } c_1,$$

where $\lambda_1 \leq \dots \leq \lambda_N$ denotes the eigenvalues of the preconditioned matrix $M^{-\frac{1}{2}}AM^{-\frac{1}{2}}$. The constant c_1 depends on s and the initial error e^0 but not on i.

This theorem holds for the classical method. However as the hybrid method is only a data-parallel version of the classical block method the same convergence rate hold with $k = p$ for the $\mathbb{S}_{\text{Hy}}^p$ method. The following lemma gives us a convergence rate for the global methods.

Lemma 3 (Theoretical Convergence Rate of Global Methods) *The theoretical convergence rate of a global method using $\mathbb{S}_{Gl}^p$ is*

$$\hat{\mu} = \frac{\hat{\kappa}_p - 1}{\hat{\kappa}_p + 1}, \quad \textit{with} \quad \hat{\kappa}_p = \frac{\lambda_N}{\lambda_{\left\lceil \frac{p}{q} \right\rceil}}$$

Proof A global method is equivalent to solve the qn-dimensional system

$$\begin{pmatrix} A & & & \\ & A & & \\ & & \ddots & \\ & & & A \end{pmatrix} \begin{pmatrix} X_1 & \cdots & X_p \\ X_{p+1} & \cdots & X_{2p} \\ & \vdots & \\ X_{k-p+1} & \cdots & X_k \end{pmatrix} = \begin{pmatrix} B_1 & \cdots & B_p \\ B_{p+1} & \cdots & B_{2p} \\ & \vdots & \\ B_{k-p+1} & \cdots & B_k \end{pmatrix}$$

with the classical block Krylov method with p right hand sides. The matrix of this system has the same eigenvalues as A but with q times the multiplicity. Thus the p-smallest eigenvalue is $\lambda_{\lceil \frac{p}{q} \rceil}$. Therefore and by applying Theorem 1 we deduce the theoretical convergence rate. □

This result makes the global methods irrelevant for practical use. In particular for $q > 1$ the non-global hybrid method would perform better.

4 Implementation and Numerical Experiments

With the DUNE 2.6 release an abstraction for SIMD data types was introduced. The aim of these abstraction is the possibility to use the data types as a replacement for the numeric data type, like **`double`** or **`float`**, to create data parallel methods. For a more detailed review see Bastian et al. [2]. Usually these SIMD data types are provided by external libraries like Vc [9] or vectorclass [4], which usually provide data types with the length of the hardware SIMD (e.g. 4 or 8). For problems with more right hand sides we use the `LoopSIMD` type. This type is basically an array but implements the arithmetic operations by static sized loops.

Listing 1 shows the implementation for the `BAXPY` kernel for the case $p = k$.

Listing 1 Implementation of `BAXPY`

```
void baxpy(scalar_field_type alpha,
           const BlockProduct<scalar_field_type>& gamma,
           const X& x, X& y){
  for(size_t i=0;i<x.size();++i){
    field_type xi = x[i]
    field_type yi = y[i];
    for(size_t r=0;r<P;++r){
      yi += lane(r, xi)*gamma[r];
    }
    y[i] = yi;
  }
}
```

In a first test series we examine the runtime of the kernels BDOT, BAXPY and BOP. To check the efficiency of our implementation we, take the following performance model into account. This performance model is a simplified variant of the ECM model presented by Hofmann et al. [8]. We assume that the runtime of a kernel is bounded by the following three factors:

- $T_{\text{comp}} = \frac{\omega}{\text{peakflops}}$: The time the processor needs to perform the necessary number of floating point operations. Where ω is the number of floating point operations of the kernel.
- $T_{\text{mem}} = \frac{\beta}{\text{memory bandwidth}}$: The time to transfer the data from the main memory to the L1 cache. Where β is the amount of data that needs to be transferred in the kernel.
- $T_{\text{reg}} = \frac{\tau}{\text{register bandwidth}}$: The time to transfer data between L1 cache and registers. Where τ is the amount of data that needs to be transferred in the kernel.

Finally the expected runtime is given by

$$T = \max\left(T_{\text{comp}}, T_{\text{mem}}, T_{\text{reg}}\right). \tag{5}$$

Table 1 shows an overview of the performance relevant characteristics of the kernels. We observe that for the BOP kernel the runtime per right hand side decreases rapidly for small k, this is in accordance with our expectation. For larger k the runtime per right hand side increases slightly. We suppose that this effect is due to the fact that for larger k one row of a block vector occupies more space in the caches, hence fewer rows can be cached. This effects could be mitigated by using a different sparse matrix format, like the Sell-C-σ format [10].

Furthermore we see that the runtime of the BDOT and BAXPY kernels is constant up to an certain p ($p \lesssim 16$). This is in accordance with our expectation, as it is memory bound in that regime and β does not depend on p. At almost all p the runtimes for global and hybrid version coincide except for $p = 64$. The reason for that is, that a 64×64 takes 32 kbyte memory, which is exactly the L1 cache size. In the non-global setting two of these matrices are modified during the computation, which then exceeds the L1 cache. This explains as well why the runtime for the $p = 128$ case is so much higher than expected.

Figure 1 shows the measured data compared with the expected runtimes. All tests are performed on an Intel Skylake-SP Xeon Gold 6148 on one core. The

Table 1 Performance relevant characteristics for the BDOT and BAXPY kernels. Number of floating point operations (ω), amount of data loaded from main memory (β), number of data transfers between registers and L1-Cache (τ). z is the number of non-zeros in A

	ω	β	τ
BDOT	$2np^2q$	$2nk$	$2nqp^2 + 2nk$
BAXPY	$2np^2q$	$3nk$	$nqp^2 + 2nk$
BOP	$2kz$	$2z + 2kn$	$z(2 + 2k)$

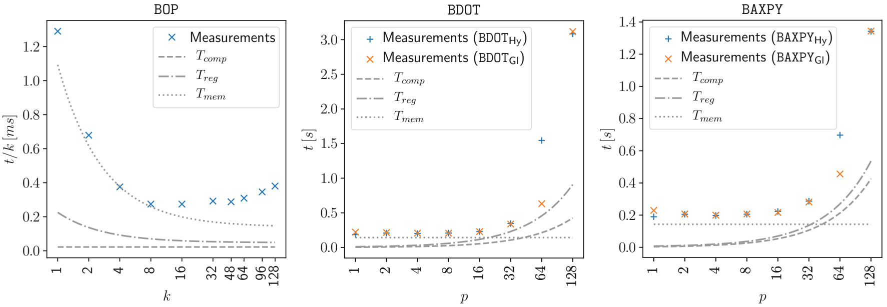


Fig. 1 Microbenchmarks for kernels BDOT, BAXPY and BOP using $k = 128$

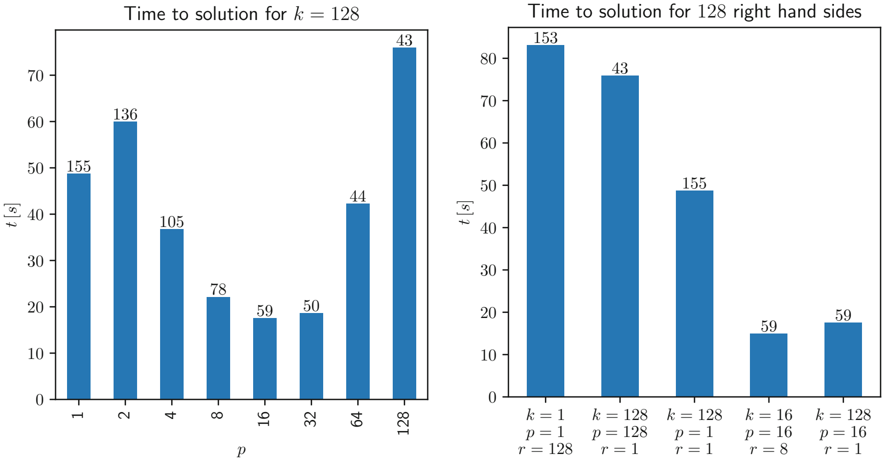


Fig. 2 Time to solution for different parameters. Numbers on top of the bars denote the number of iterations. Left: $k = 128$ Right: Different configurations: $r = 128/k$ is the number of repetitions to solve for all 128 right hand sides

theoretical peakflops are 76.8 GFLOP/s, the memory bandwidth is 13.345 Gbyte/s and the register bandwidth is 286.1 Gbyte/s.

In a second experiment we compare the runtime of the whole algorithm with each other. For that we discretized a 2D heterogeneous Poisson problem with a 5-point Finite Difference stencil. The right hand sides are initialized with random numbers. We iterate until the defect norm of each column has been decreased by a factor of 10^{-8}. An ILU preconditioner was used. Figure 2 shows the results. We see that the best block size is $p = 16$. In another test we compare the runtimes for different parameters, where the algorithm is executed r times until all 128 right hand sides are solved. In this case the $k = 16$, $p = 16$ case is the fastest but only slightly slower as the $k = 128$, $p = 16$. The reason for that is the worse cache behavior of the BOP kernel, like we have seen before. Note that on a distributed machine the latter case would need 8x less communication.

5 Conclusion and Outlook

In this paper we have presented strategies for the vectorized BCG method. For that we reviewed the block Krylov framework of Frommer et al. and apply it on the BCG method. This makes it possible to use the advantages of the BCG method as far as it is beneficial, while the number of right hand sides can be further increased. This helps to decrease the communication overhead and improve the arithmetic intensity of the kernels. We observed that the runtime of the individual kernels scale linearly with the number of right hand sides as long as they are memory bound ($p \lesssim 16$ on our machine). That means that it is always beneficial to use at least this block size p, depending on the problem it could also be beneficial to choose even larger p.

The found optimizations are also applicable to other block Krylov space methods like GMRes, MINRES or BiCG, and could be combined with pipelining techniques. These approaches are the objective of future work.

References

1. Al Daas, H., Grigori, L., Hénon, P., Ricoux, P.: Enlarged gmres for reducing communication (2017). Preprint hal.inria.fr/hal-01497943
2. Bastian, P., Blatt, M., Dedner, A., Dreier, N.A., Engwer, C., Fritze, R., Gräser, C., Kempf, D., Klöfkorn, R., Ohlberger, M., Sander, O.: The dune framework: Basic concepts and recent developments (2019). Preprint arXiv:1909.13672
3. Dubrulle, A.A.: Retooling the method of block conjugate gradients. Electron. Trans. Numer. Anal. **12**, 216–233 (2001). URL http://etna.mcs.kent.edu/vol.12.2001/pp216-233.dir/pp216-233.pdf
4. Fog, A.: C++ vector class library (2013). URL http://www.agner.org/optimize/vectorclass.pdf
5. Frommer, A., Szyld, D.B., Lund, K.: Block Krylov subspace methods for functions of matrices. Electron. Trans. Numer. Anal. **47**, 100–126 (2017). URL http://etna.math.kent.edu/vol.47.2017/pp100-126.dir/pp100-126.pdf
6. Grigori, L., Moufawad, S., Nataf, F.: Enlarged Krylov subspace conjugate gradient methods for reducing communication. SIAM Journal on Matrix Analysis and Applications **37**(2), 744–773 (2016). DOI https://doi.org/10.1137/140989492
7. Grigori, L., Tissot, O.: Reducing the communication and computational costs of enlarged Krylov subspaces conjugate gradient (2017). Preprint hal.inria.fr/hal-01451199/
8. Hofmann, J., Alappat, C.L., Hager, G., Fey, D., Wellein, G.: Bridging the architecture gap: Abstracting performance-relevant properties of modern server processors (2019). Preprint arXiv:1907.00048
9. Kretz, M., Lindenstruth, V.: Vc: A c++ library for explicit vectorization. Software: Practice and Experience **42**(11), 1409–1430 (2012). DOI https://doi.org/10.1002/spe.11490
10. Kreutzer, M., Hager, G., Wellein, G., Fehske, H., Bishop, A.R.: A unified sparse matrix data format for efficient general sparse matrix-vector multiplication on modern processors with wide SIMD units. SIAM Journal on Scientific Computing **36**(5), C401–C423 (2014). DOI https://doi.org/10.1137/130930352
11. Lund, K.: A new block Krylov subspace framework with applications to functions of matrices acting on multiple vectors. Ph.D. thesis, Temple University (2018)
12. O'Leary, D.P.: The block conjugate gradient algorithm and related methods. Linear algebra and its applications **29**, 293–322 (1980). DOI https://doi.org/10.1016/0024-3795(80)90247-5

The Unfitted HHO Method for the Stokes Problem on Curved Domains

Erik Burman, Guillaume Delay, and Alexandre Ern

Abstract We design a hybrid high-order (HHO) method to approximate the Stokes problem on curved domains using unfitted meshes. We prove inf-sup stability and a priori estimates with optimal convergence rates. Moreover, we provide numerical simulations that corroborate the theoretical convergence rates. A cell-agglomeration procedure is used to prevent the appearance of small cut cells.

1 Introduction

Generating meshes to solve problems posed on domains with a curved boundary can be a difficult task when high-order methods are used. The use of unfitted meshes that do not fit this boundary can circumvent this difficulty. In the framework of finite element methods, the main paradigm for unfitted methods [5] is the use of Nitsche's method [21] to enforce weakly the boundary conditions at the boundary. One difficulty with this method is the possible presence of small cut cells, i.e. cells that have only a small fraction of their volume inside the actual physical domain. These small cut cells can have an adverse effect on the conditioning of the system

E. Burman
Department of Mathematics, University College London, London, UK
e-mail: e.burman@ucl.ac.uk

G. Delay
Laboratoire Jacques-Louis Lions, Sorbonne Université, Paris, France

CERMICS, Ecole des Ponts, Marne-la-Vallée, France

INRIA, Paris, France
e-mail: guillaume.delay@sorbonne-universite.fr

A. Ern (✉)
CERMICS, Ecole des Ponts, Marne-la-Vallée, France

INRIA, Paris, France
e-mail: alexandre.ern@enpc.fr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_38

matrix and can even hamper convergence (see [11] for a recent study on the topic). The most common way to deal with the problem of small cut cells is to add a stabilizing term such as the ghost penalty [3]. In the present study we use a cell-agglomeration technique to prevent the appearance of small cut cells. Such a method has been considered in [17, 23] and more recently in [1, 4, 6, 7].

In order to easily handle the various shapes of the cells produced by the agglomeration process, we consider the hybrid high-order (HHO) method, which is a polyhedral method. HHO methods have been introduced recently in [12, 13]. As shown in [10], they are closely related to hybridizable discontinuous Galerkin methods and to nonconforming Virtual element methods. Moreover, the unfitted HHO method has already been studied in [4, 7] for elliptic interface problems. More precisely, it was adapted to the unfitted framework in [7], a mixed-order polynomial setting was considered with the cell unknowns being one degree higher than the face unknowns and a first algorithm for the cell-agglomeration procedure was provided. This study was continued in [4], where the use of a novel gradient reconstruction operator eliminated the requirement on the Nitsche's penalty parameter to be large enough. An improvement of the cell-agglomeration algorithm and numerical simulations were also provided.

The present study extends the unfitted HHO method presented in [4] to the Stokes problem. Unfitted schemes have already been used to approximate the Stokes problem in e.g. [8, 15, 16, 20]. A HHO scheme was already provided on fitted meshes for the Stokes problem in [14] and for the Navier–Stokes equations in [2]. In addition to the usual stabilization and gradient reconstruction operators, a divergence reconstruction operator is also defined. We here focus on the Stokes problem in curved domains. The extension to the Stokes problem for two immiscible fluids separated by a curved interface will be treated in a future work. We also mention that a HDG scheme for the Stokes problem in curved domains was devised in [22].

Let Ω be a smooth domain in $\mathbb{R}^d$, $d \in \{2, 3\}$ and $\Gamma = \partial\Omega$ its boundary. We consider the Stokes problem

$$-\Delta \boldsymbol{u} + \nabla p = \boldsymbol{f} \qquad \text{in } \Omega, \tag{1a}$$

$$\nabla{\cdot}\boldsymbol{u} = 0 \qquad \text{in } \Omega, \tag{1b}$$

$$\boldsymbol{u} = \boldsymbol{g} \qquad \text{on } \Gamma, \tag{1c}$$

where $\boldsymbol{u}$ and p are the velocity and pressure of the fluid. In the sequel, we denote by $\boldsymbol{n}_\Gamma$ the unit outward normal of Ω. For all $\boldsymbol{f} \in L^2(\Omega; \mathbb{R}^d)$ and $\boldsymbol{g} \in H^{1/2}(\Gamma; \mathbb{R}^d)$ with $\int_\Gamma \boldsymbol{g}{\cdot}\boldsymbol{n}_\Gamma = 0$, the problem (1) admits a unique solution in $H^1(\Omega; \mathbb{R}^d) \times L_0^2(\Omega)$, where $L_0^2(\Omega) := \{q \in L^2(\Omega) \mid \int_\Omega q = 0\}$.

In Sect. 2, we introduce the unfitted HHO method for the Stokes problem on curved domains. We also state the main stability result in the form of an inf-sup condition and the main error estimates. In Sect. 3, we present some numerical simulations.

2 The Unfitted HHO Method

We consider a larger domain $\widetilde{\Omega}$ such that $\Omega \subset \widetilde{\Omega}$. Let $\mathcal{T}_h$ be a discretization of $\widetilde{\Omega}$. We assume that $(\mathcal{T}_h)_{h>0}$ is a shape-regular polyhedral mesh sequence in the sense of [12]. In particular, all the cells $T \in \mathcal{T}_h$ are assumed to have planar faces and straight edges. We denote by $\rho > 0$ the parameter that quantifies the regularity of the mesh. We denote by h_T the diameter of the cell $T \in \mathcal{T}_h$ and $\boldsymbol{n}_T$ its unit outward normal. We set conventionally $h := \max_{T\in\mathcal{T}_h} h_T$. The meshes do not necessarily fit Ω.

For all $T \in \mathcal{T}_h$, we denote by $T^\circ := T \cap \Omega$, $(\partial T)^\circ := \partial T \cap \Omega$ and $T^\Gamma := T \cap \Gamma$. Let $\mathbb{P}^\ell(S)$ (resp. $\mathbb{P}^\ell(S;\mathbb{R}^d)$, $\mathbb{P}^\ell(S;\mathbb{R}^{d\times d})$) be the space composed of scalar (resp. vectorial, matricial) polynomials of degree at most $\ell \geq 0$ in S. We denote by $(\cdot,\cdot)_S$ the L^2-scalar product on S, $\|\cdot\|_S$ the associated norm, and $B(\boldsymbol{x}, r)$ the ball of center $\boldsymbol{x}$ and radius r. We assume that the meshes fulfill the following three assumptions.

Assumption 1 (Cut Cells) There is $\delta \in (0, 1)$ such that, for all $T \in \mathcal{T}_h$, there is $\tilde{\boldsymbol{x}}_T \in T^\circ$ such that $B(\tilde{\boldsymbol{x}}_T, \delta h_T) \subset T^\circ$.

Assumption 2 (Multiplicative Trace Inequality) There are $c_{\mathrm{mtr}} > 0$ and $\theta_{\mathrm{mtr}} \geq 1$, such that for all $T \in \mathcal{T}_h$, there is $\check{\boldsymbol{x}}_T \in \mathbb{R}^d$ so that for all $\boldsymbol{v} \in H^1(T^\dagger;\mathbb{R}^d)$ with $T^\dagger := B(\check{\boldsymbol{x}}_T; \theta_{\mathrm{mtr}} h_T)$, $\|\boldsymbol{v}\|_{(\partial T)^\circ} + \|\boldsymbol{v}\|_{T^\Gamma} \leq c_{\mathrm{mtr}}(h_T^{-1/2}\|\boldsymbol{v}\|_{T^\dagger} + \|\boldsymbol{v}\|_{T^\dagger}^{1/2}\|\nabla\boldsymbol{v}\|_{T^\dagger}^{1/2})$.

Assumption 3 (Resolving $T^\dagger$) There exists $N_0 \in \mathbb{N}$ (independent from h) such that for every $T \in \mathcal{T}_h$, $T^\dagger \subset \Delta_{N_0}(T)$, where $\Delta_0(T) := T$ and $\Delta_{j+1}(T) := \{T' \in \mathcal{T}_h \mid \overline{T'} \cap \overline{\Delta_j(T)} \neq \emptyset\}$ for all $j \geq 0$.

Assumption 1 means that there are no bad cut cells in the mesh. This assumption provides a discrete trace inequality [7], and it can be satisfied by the cell-agglomeration procedure described in [4] if the mesh is fine enough w.r.t. the curvature of Γ, see [7]. Assumption 2 is classical in the framework of unfitted finite element methods. It can be established if the mesh is fine enough w.r.t. the curvature of the boundary [7]. Assumption 3 is reasonable for meshes that are not too graded.

2.1 The Local Discrete Problem

Let $k \geq 0$ be the face polynomial degree in the unfitted HHO method. The velocity is represented by a vector-valued polynomial of degree at most $k+1$ in every cell and a vector-valued polynomial of degree at most k on every face. The pressure is represented by a polynomial of degree at most k in every cell. The local degrees of freedom are denoted $\hat{\boldsymbol{u}}_T = (\boldsymbol{u}_T, \boldsymbol{u}_{\partial T}) \in \mathbb{P}^{k+1}(T^\circ;\mathbb{R}^d) \times \mathbb{P}^k(\mathcal{F}_{(\partial T)^\circ};\mathbb{R}^d) =: \hat{\boldsymbol{U}}_T^k$ and $p_T \in \mathbb{P}^k(T^\circ)$, where $\mathbb{P}^k(\mathcal{F}_{(\partial T)^\circ};\mathbb{R}^d) := \prod_{F^\circ\in\mathcal{F}_{(\partial T)^\circ}} \mathbb{P}^k(F^\circ;\mathbb{R}^d)$ and $\mathcal{F}_{(\partial T)^\circ} := \{F^\circ := F \cap \Omega \mid F \in \mathcal{F}_h, F \subset \partial T\}$, with $\mathcal{F}_h$ the set of faces of $\mathcal{T}_h$.

We define the gradient reconstruction operator $\mathbb{G}_T^k : \hat{\boldsymbol{U}}_T^k \to \mathbb{P}^k(T^\circ; \mathbb{R}^{d\times d})$ such that for all $\hat{\boldsymbol{u}}_T \in \hat{\boldsymbol{U}}_T^k$ and all $\mathbb{q} \in \mathbb{P}^k(T^\circ; \mathbb{R}^{d\times d})$, we have

$$(\mathbb{G}_T^k(\hat{\boldsymbol{u}}_T), \mathbb{q})_{T^\circ} := (\nabla \boldsymbol{u}_T, \mathbb{q})_{T^\circ} + (\boldsymbol{u}_{\partial T} - \boldsymbol{u}_T, \mathbb{q}\boldsymbol{n}_T)_{(\partial T)^\circ} - (\boldsymbol{u}_T, \mathbb{q}\boldsymbol{n}_\Gamma)_{T^\Gamma}. \tag{2}$$

In a similar way, we define the divergence reconstruction operator $D_T^k : \hat{\boldsymbol{U}}_T^k \to \mathbb{P}^k(T^\circ)$ such that for all $\hat{\boldsymbol{u}}_T \in \hat{\boldsymbol{U}}_T^k$ and all $q \in \mathbb{P}^k(T^\circ)$, we have

$$(D_T^k(\hat{\boldsymbol{u}}_T), q)_{T^\circ} := (\nabla{\cdot}\boldsymbol{u}_T, q)_{T^\circ} + (\boldsymbol{u}_{\partial T} - \boldsymbol{u}_T, q\boldsymbol{n}_T)_{(\partial T)^\circ} - (\boldsymbol{u}_T, q\boldsymbol{n}_\Gamma)_{T^\Gamma}, \tag{3}$$

so that $D_T^k(\hat{\boldsymbol{u}}_T) = Tr(\mathbb{G}_T^k(\hat{\boldsymbol{u}}_T))$. Furthermore, we define the stabilization operator

$$s_T(\boldsymbol{u}_T, \boldsymbol{v}_T) := h_T^{-1}(\Pi_{(\partial T)^\circ}^k(\boldsymbol{u}_{\partial T} - \boldsymbol{u}_T), \boldsymbol{v}_{\partial T} - \boldsymbol{v}_T)_{(\partial T)^\circ} + h_T^{-1}(\boldsymbol{u}_T, \boldsymbol{v}_T)_{T^\Gamma}, \tag{4}$$

where $\Pi_{(\partial T)^\circ}^k$ denotes the L^2-orthogonal projection onto $\mathbb{P}^k(\mathcal{F}_{(\partial T)^\circ}; \mathbb{R}^d)$. We define the following bilinear and linear forms: For all $\hat{\boldsymbol{v}}_T, \hat{\boldsymbol{w}}_T \in \hat{\boldsymbol{U}}_T^k$ and all $q_T \in \mathbb{P}^k(T^\circ)$,

$$a_T(\hat{\boldsymbol{v}}_T, \hat{\boldsymbol{w}}_T) := (\mathbb{G}_T^k(\hat{\boldsymbol{v}}_T), \mathbb{G}_T^k(\hat{\boldsymbol{w}}_T))_{T^\circ} + s_T(\hat{\boldsymbol{v}}_T, \hat{\boldsymbol{w}}_T), \tag{5a}$$

$$b_T(\hat{\boldsymbol{v}}_T, q_T) := (D_T^k(\hat{\boldsymbol{v}}_T), q_T)_{T^\circ}, \tag{5b}$$

$$\ell_T^a(\hat{\boldsymbol{w}}_T) := (\boldsymbol{f}, \boldsymbol{w}_T)_{T^\circ} + (\boldsymbol{g}, h_T^{-1}\boldsymbol{w}_T - \mathbb{G}_T^k(\hat{\boldsymbol{w}}_T)\boldsymbol{n}_\Gamma)_{T^\Gamma}, \tag{5c}$$

$$\ell_T^b(q_T) := -(\boldsymbol{g}, q_T\boldsymbol{n}_\Gamma)_{T^\Gamma}. \tag{5d}$$

Note that s_T and $\mathbb{G}_T^k(\hat{\boldsymbol{u}}_T)$ are similar to the operators proposed in [4].

Remark 1 (Variants) The gradient reconstruction operators can also be defined in $\nabla\mathbb{P}^{k+1}(T^\circ; \mathbb{R}^d)$ instead of $\mathbb{P}^k(T^\circ; \mathbb{R}^{d\times d})$ (see for instance [7]). Moreover one can also use cell unknowns in $\mathbb{P}^k(T^\circ; \mathbb{R}^d)$ (instead of $\mathbb{P}^{k+1}(T^\circ; \mathbb{R}^d)$) away from the interface provided the stabilization operator from [12] is used.

2.2 The Global Discrete Problem

The global unknowns are $\hat{\boldsymbol{u}}_h \in \mathbb{P}^{k+1}(\mathcal{T}_h; \mathbb{R}^d) \times \mathbb{P}^k(\mathcal{F}_h; \mathbb{R}^d) =: \hat{\boldsymbol{U}}_h^k$ and $p_h \in \mathbb{P}^k(\mathcal{T}_h) =: P_h^k$. For all $T \in \mathcal{T}_h$, $\hat{\boldsymbol{u}}_T$ and p_T are the local components of $\hat{\boldsymbol{u}}_h$ and p_h attached to T (see Sect. 2.1). Let $P_{h*}^k := \{q_h \in P_h^k \mid \int_\Omega q_h = 0\}$. We define the global bilinear forms $a_h(\hat{\boldsymbol{v}}_h, \hat{\boldsymbol{w}}_h) := \sum_{T\in\mathcal{T}_h} a_T(\hat{\boldsymbol{v}}_T, \hat{\boldsymbol{w}}_T)$, $b_h(\hat{\boldsymbol{v}}_h, q_h) := \sum_{T\in\mathcal{T}_h} b_T(\hat{\boldsymbol{v}}_T, q_T)$, and the global linear forms $\ell_h^a(\hat{\boldsymbol{w}}_h) := \sum_{T\in\mathcal{T}_h} \ell_T^a(\hat{\boldsymbol{v}}_T)$, $\ell_h^b(q_h) := \sum_{T\in\mathcal{T}_h} \ell_T^b(q_T)$. The discrete global problem reads: find $(\hat{\boldsymbol{u}}_h, p_h) \in$

$Y_h^k := \hat{\boldsymbol{U}}_h^k \times P_{h*}^k$ such that

$$a_h(\hat{\boldsymbol{u}}_h, \hat{\boldsymbol{v}}_h) - b_h(\hat{\boldsymbol{v}}_h, p_h) = \ell_h^a(\hat{\boldsymbol{v}}_h), \tag{6a}$$

$$b_h(\hat{\boldsymbol{u}}_h, q_h) = \ell_h^b(q_h), \tag{6b}$$

for all $(\hat{\boldsymbol{v}}_h, q_h) \in Y_h^k$. This discrete global problem can be solved in an efficient way by means of a static condensation procedure as described e.g. in [9, 14]. Specifically, the global problem that actually has to be solved involves only the face degrees of freedom of the velocity and the mean pressure value in every cell. The other degrees of freedom can be computed in a post-processing step by means of local solves.

2.3 Stability and Error Estimates

For all $\hat{\boldsymbol{v}}_h \in \hat{\boldsymbol{U}}_h^k$, we denote by $\|\hat{\boldsymbol{v}}_h\|_*^2 := \sum_{T\in\mathcal{T}_h} \|\nabla \boldsymbol{v}_T\|_{T^\circ}^2 + h_T^{-1}\|\boldsymbol{v}_{\partial T} - \boldsymbol{v}_T\|_{(\partial T)^\circ}^2 + h_T^{-1}\|\boldsymbol{v}_T\|_{T^\Gamma}^2$ and for all $(\hat{\boldsymbol{v}}_h, q_h) \in Y_h^k$, we denote by $\|(\hat{\boldsymbol{v}}_h, q_h)\|_\#^2 := \|\hat{\boldsymbol{v}}_h\|_*^2 + \|q_h\|_\Omega^2$. We also denote by $A_h((\hat{\boldsymbol{v}}_h, q_h), (\hat{\boldsymbol{w}}_h, r_h)) := a_h(\hat{\boldsymbol{v}}_h, \hat{\boldsymbol{w}}_h) - b_h(\hat{\boldsymbol{w}}_h, q_h) + b_h(\hat{\boldsymbol{v}}_h, r_h)$. A numerical analysis leads to the following results. We only sketch the proof of the inf-sup condition; see [6, Lemma 10 & Theorem 12] for detailed proofs.

Theorem 4 (Inf-Sup Condition) *Under the Assumption 1, there exists $\beta > 0$, depending only on k, δ and ρ, such that for every $(\hat{\boldsymbol{v}}_h, q_h) \in Y_h^k$, we have*

$$\beta\|(\hat{\boldsymbol{v}}_h, q_h)\|_\# \le \sup_{(\hat{\boldsymbol{w}}_h, r_h)\in Y_h^k} \frac{A_h((\hat{\boldsymbol{v}}_h, q_h), (\hat{\boldsymbol{w}}_h, r_h))}{\|(\hat{\boldsymbol{w}}_h, r_h)\|_\#}.$$

Moreover, there exists a unique solution $(\hat{\boldsymbol{u}}_h, p_h) \in Y_h^k$ to (6).

Proof (Sketch of the Proof)

- We prove the coercivity and the continuity of a_h w.r.t. the norm $\|\cdot\|_*$ by proceeding as in [4].
- Let $(\hat{\boldsymbol{v}}_h, q_h) \in Y_h^k$. Using the test function $(\hat{\boldsymbol{w}}_h, r_h) := (\hat{\boldsymbol{v}}_h, q_h)$ proves $\|\hat{\boldsymbol{v}}_h\|_*^2 \le CS\|(\hat{\boldsymbol{v}}_h, q_h)\|_\#$, where $S := \sup_{(\hat{\boldsymbol{w}}_h, r_h)\in Y_h^k} \frac{A_h((\hat{\boldsymbol{v}}_h, q_h), (\hat{\boldsymbol{w}}_h, r_h))}{\|(\hat{\boldsymbol{w}}_h, r_h)\|_\#}$ and C is a generic constant that has the same dependencies as β.
- We use the surjectivity of the $\nabla\cdot$ operator to prove that $\|q_h\|_\Omega^2 \le C(S^2 + \|\hat{\boldsymbol{v}}_h\|_*^2 + \sum_{T\in\mathcal{T}_h} h_T^2\|\nabla q_T\|_{T^\circ}^2)$.
- Using the test function $(\hat{\boldsymbol{w}}_h, r_h) := (\hat{\boldsymbol{w}}_h, 0)$ where $\hat{\boldsymbol{w}}_T := (-h_T^2\nabla q_T, 0)$ proves $\sum_{T\in\mathcal{T}_h} h_T^2\|\nabla q_T\|_{T^\circ}^2 \le C(S^2 + \|\hat{\boldsymbol{v}}_h\|_*^2)$.
- This shows the inf-sup condition and thus the well-posedness of (6). □

Theorem 5 (Error Estimates) *Under the Assumptions 1, 2 and 3, there exists $C > 0$, depending on k, δ, c_{mtr}, N_0 and ρ, such that, if (u, p) belongs to $H^{k+2}(\Omega) \times H^{k+1}(\Omega)$, we have*

$$\Big(\sum_{T\in\mathcal{T}_h} \|\nabla u - \nabla u_T\|^2_{T^\circ} + \|p - p_T\|^2_{T^\circ}\Big)^{1/2} \le Ch^{k+1}(|u|_{H^{k+2}(\Omega)} + |p|_{H^{k+1}(\Omega)}). \quad (7)$$

Remark 2 Contrary to the classical Nitsche's method, we do not need here any parameter to be large enough. This is a consequence of the construction of $\mathbb{G}^k_T(\hat{\boldsymbol{u}}_T)$ in (2). For more details, the reader can refer to [4] and [18] for similar ideas using FEM.

3 Numerical Simulations

On the circular domain $\Omega := C((0.5, 0.5); 1/3)$, the disk of center $(0.5, 0.5)$ and radius $1/3$ (see Fig. 1), we consider the exact solution $u_1(x, y) := X^2(X^2 - 2X + 1)Y(4Y^2 - 6Y + 2)$, $u_2(x, y) := -Y^2(Y^2 - 2Y + 1)X(4X^2 - 6X + 2)$, and $p(x, y) := \sin(X + Y)$ where $X := x - 0.5$, $Y := y - 0.5$. The circular domain Ω is embedded into the unit square $\widetilde{\Omega} := (0, 1)^2$ which is meshed with a uniform Cartesian mesh. The mesh size h refers to the number of subdivisions of each side of $\widetilde{\Omega}$. In a pre-processing step, we use the cell-agglomeration technique described in [4]. Static condensation is used to decrease the total number of degrees of freedom. The global problem is solved by means of a sparse LU decomposition. The numerical developments follow the DiSk++ framework [9] and are available in the proton[1] library. The profiles of the Euclidean velocity norm and the pressure are shown in Figs. 2 and 3, respectively.

The cells lying completely outside the domain are not considered. The results of the numerical simulations are reported in Table 1. The rates of convergence are computed by comparing the result of one refinement step to the previous one. We recover the convergence rates stated in Theorem 5.

Note that, as in [4, 6], the interface is represented in every cut cell by $2^{n_{\mathrm{int}}}$ segments, and every cut cell is then triangulated for the purpose of numerical integration. In the computations in Table 1, we used $n_{\mathrm{int}} = 11$. We think that the slightly lower rate of convergence observed for the pressure for $k = 3$ and $h = 1/64$ is due to the error in the representation of the boundary since the rate improves when n_{int} is increased (at the price of a more expensive assembly of the system matrix). A similar observation was made in [6]. Instead of increasing n_{int}, one can also consider a higher-order representation of the boundary as for instance in [19].

[1] https://github.com/wareHHOuse.

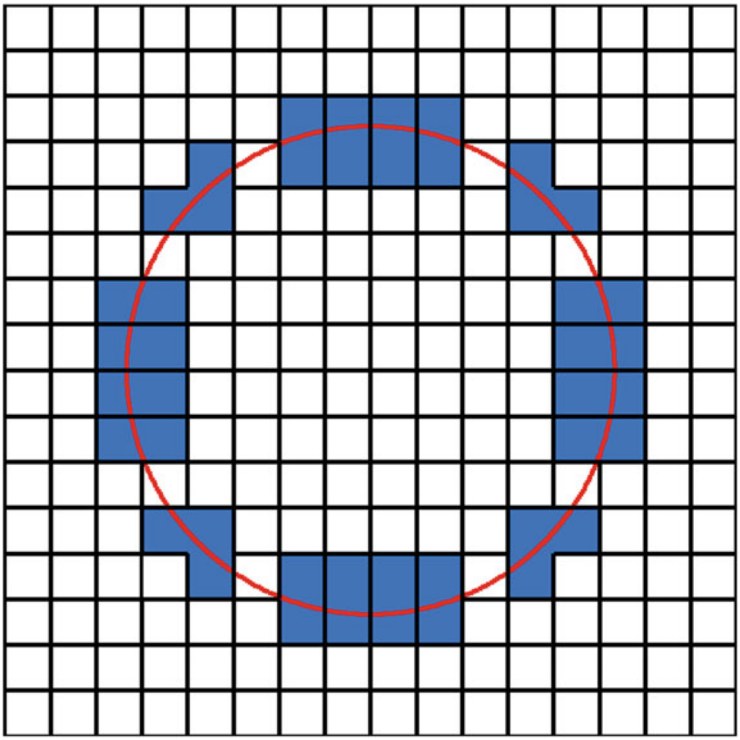


Fig. 1 The mesh and the computational domain for $h = 1/16$. The highlighted cells are the ones that are agglomerated. The boundary of the domain is in red

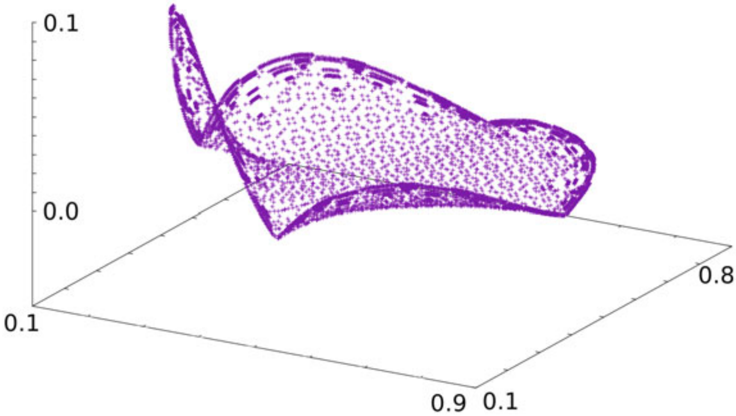


Fig. 2 Euclidean velocity norm ($h = 1/16$, $k = 1$)

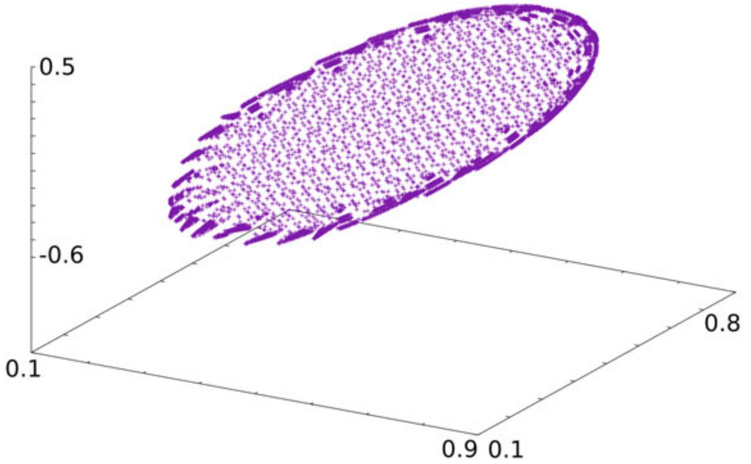


Fig. 3 Pressure field ($h = 1/16$, $k = 1$)

Table 1 Convergence of the errors for various polynomial orders

h^{-1}	$\boldsymbol{u}$ (H^1-seminorm)	Rate	p (L^2-norm)	Rate
$k = 0$				
8	9.54e−2	·	4.53e−2	·
16	3.85e−2	1.31	2.11e−2	1.11
32	1.71e−2	1.17	8.84e−3	1.25
64	8.60e−3	0.99	4.24e−3	1.06
$k = 1$				
8	4.80e−2	·	7.44e−3	·
16	9.36e−3	2.36	1.98e−3	1.91
32	1.68e−3	2.48	3.32e−4	2.57
64	4.15e−4	2.02	6.49e−5	2.35
$k = 2$				
8	7.41e−3	·	5.15e−4	·
16	7.69e−4	3.27	6.99e−5	2.88
32	6.63e−5	3.54	6.66e−6	3.39
64	8.89e−6	2.90	6.40e−7	3.38
$k = 3$				
8	7.60e−4	·	2.51e−5	·
16	3.44e−5	4.46	1.14e−6	4.46
32	1.44e−6	4.57	5.16e−8	4.47
64	9.89e−8	3.87	5.90e−9	3.13

References

1. Badia, S., Verdugo, F., Martín, A.F.: The aggregated unfitted finite element method for elliptic problems. Comput. Methods Appl. Mech. Engrg. **336**, 533–553 (2018)
2. Botti, L., Di Pietro, D. A. Droniou, J.: A hybrid high-order method for the incompressible Navier-Stokes equations based on Temam's device. J. Comput. Phys. **376**, 786–816 (2019)
3. Burman, E.: Ghost penalty. C. R. Math. Acad. Sci. Paris. **348(21–22)**, 1217–1220 (2010)
4. Burman, E., Cicuttin, M., Delay, G., Ern, A.: An unfitted Hybrid High-Order method with cell agglomeration for elliptic interface problems. submitted. https://hal.archives-ouvertes.fr/hal-02280426/
5. Burman, E., Claus, S., Hansbo, P., Larson, M. G., Massing, A.: CutFEM: discretizing geometry and partial differential equations. Internat. J. Numer. Methods Engrg. **104(7)**, 472–501 (2015)
6. Burman, E., Delay, G., Ern, A.: An unfitted hybrid high-order method for the Stokes interface problem, to appear in IMA J. Numer. Anal. https://hal.archives-ouvertes.fr/hal-02519896/
7. Burman, E., Ern, A.: An unfitted hybrid high-order method for elliptic interface problems. SIAM J. Numer. Anal. **56(3)**, 1525–1546 (2018)
8. Burman, E., Hansbo, P.: Fictitious domain methods using cut elements: III. A stabilized Nitsche method for Stokes' problem. ESAIM Math. Model. Anal. **48(3)**, 859–874 (2014)
9. Cicuttin, M., Di Pietro, D. A., Ern, A.: Implementation of discontinuous skeletal methods on arbitrary-dimensional, polytopal meshes using generic programming. J. Comput. Appl. Math. **344**, 852–874 (2018)
10. Cockburn, B., Di Pietro, D. A., Ern, A.: Bridging the Hybrid High-Order and Hybridizable Discontinuous Galerkin methods. ESAIM: Math. Model Numer. Anal. (M2AN) **50(3)**, 635–650 (2016)
11. de Prenter, F., Lehrenfeld, C., Massing, A.: A note on the stability parameter in Nitsche's method for unfitted boundary value problems. Comput. Math. Appl. **75(12)**, 4322–4336 (2018)
12. Di Pietro, D. A., Ern, A.: A Hybrid High-Order locking-free method for linear elasticity on general meshes. Comput. Meth. Appl. Mech. Engrg. **283**, 1–21 (2015)
13. Di Pietro, D. A., Ern, A., Lemaire, S.: An arbitrary-order and compact- stencil discretization of diffusion on general meshes based on local reconstruction operators. Comput. Meth. Appl. Math. **14(4)**, 461–472 (2014)
14. Di Pietro, D. A., Ern, A., Linke, A., Schieweck, F.: A discontinuous skeletal method for the viscosity-dependent Stokes problem. Comput. Methods Appl. Mech. Engrg. **306**, 175–195 (2016)
15. Fournié, M., Lozinski, A.: Stability and optimal convergence of unfitted extended finite element methods with Lagrange multipliers for the Stokes equations. In: Lect. Notes Comput. Sci. Eng., pp. 143–182, Springer, Cham (2017)
16. Guzmán, J., Olshanskii, M.: Inf-sup stability of geometrically unfitted Stokes finite elements. Math. Comp. **87(313)**, 2091–2112 (2018)
17. Johansson, A., Larson, M. G.: A high order discontinuous Galerkin Nitsche method for elliptic problems with fictitious boundary. Numer. Math. **123(4)**, 607–628 (2013)
18. Lehrenfeld, C.: Removing the stabilization parameter in fitted and unfitted symmetric Nitsche formulations. arXiv:1603.00617 (2016)
19. Lehrenfeld, C., Reusken, A.: Analysis of a high-order unfitted finite element method for elliptic interface problems. IMA J. Numer. Anal. **38**, 1351–1387 (2018)
20. Massing, A., Larson, M. G., Logg, A., Rognes, M. E.: A stabilized Nitsche fictitious domain method for the Stokes problem. J. Sci. Comput. **61(3)**, 604–628 (2014)
21. Nitsche, J.: Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. Abh. Math. Sem. Univ. Hamburg **36**, 9–15 (1971)
22. Solano, M., Vargas, F.: A high order HDG method for Stokes flow in curved domains. J. Sci. Comput.**79(3)**, 1505–1533 (2019)
23. Sollie, W. E. H., Bokhove, O., van der Vegt, J. J. W.: Space-time discontinuous Galerkin finite element method for two-fluid flows. J. Comput. Phys. **230(3)**, 789–817 (2011)

A Non-reflective Boundary Condition for LBM Based on the Assumption of Non-equilibrium Symmetry

R. Euser and C. Vuik

Abstract In this study a new type of non-reflective boundary condition (NRBC) for the Lattice Boltzmann Method (LBM) is proposed; the Non-equilibrium Symmetry Boundary Condition (NSBC). The idea behind this boundary condition is to utilize the characteristics of the non-equilibrium distribution function to assign values to the incoming populations. A simple gradient based extrapolation technique and a far-field criterion are used to predict the macroscopic fluid variables. To demonstrate the non-reflective behaviour of the NSBC, two different tests have been carried out, examining the capability of the boundary to absorb acoustic waves respectively vortices. The results for both tests show that the amount of reflection generated by the NSBC is nearly zero.

1 Introduction

In many fluid dynamics applications the region of interest comprises of only a small subdomain in space and time. When modelling such applications using numerical methods, ideally, one would like to isolate this region, as to minimize computational expenses and to allow for a sufficiently fine grained discretization. Isolating this region often requires advanced boundary treatment, in which continuity of the flow field is assumed. In other words, the amount of energy being reflected at the boundary has to be zero. This is where the so-called *Non-Reflective Boundary Conditions* (NRBC's) come into practise. Focusing on compressible flow solvers like LBM, the NRBC's can be divided into two different groups. The first group, known as the *Characteristic Boundary Conditions* (CBC's), aims at canceling out reflections by suppressing any incoming waves. The second group, the *Absorbing Layer* (AL) approach, uses a layer of several nodes thick to absorb any outgoing waves. Over the years, various efforts have been made to model such NRBC's with

R. Euser (✉) · C. Vuik
Delft Institute of Applied Mathematics, Delft, The Netherlands
e-mail: c.vuik@tudelft.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_39

LBM. We list some important studies regarding these efforts, where we remark that all references can be found in our extended paper [1]. In 2006 Chikatamarla et al. proposed Grad's approximation for missing data, in which incoming populations are assigned values based on a low-dimensional sub-manifold in distribution function space. Shortly thereafter Kam et al. published about the use of NRBC's for aeroacoustics simulations, in which he compares several boundary treatments based on extrapolation, filtering and absorbing layers. By the end of 2008 Izquierdo and Fueyo proposed an LBM formulation for the Characteristic Boundary Condition (CBC), based on the one-dimensional (LODI) characteristics of the Euler equations and their extension to the Navier-Stokes equivalent (NS-CBC). Following the work of Hu, Najafi-Yazdi and Mongeau developed a direction independent AL-NRBC based on the Perfectly Matched Layer (PML) approach. In 2013 Schlaffer presented an extensive research on NRBC's, in which he introduced the Impedence Boundary Condition (IBC). To continue with the CBC developments, Heubes et al. proposed a linear combination between Thompson's boundary conditions and the LODI relations. Comparing the approaches of Izquierdo and Heubes, Puig-Arànega et al. found that the LODI equations become inappropriate when the dimensionality of the flow increases. Consequently, Jung et al. developed a two-dimensional generalization of the CBC, by recovering the transverse and viscous terms in the characteristics analysis. Extending on the above approaches, Wissocq et al. were able to improve the numerical stability of the CBC at high Reynolds numbers by taking advantage of a regularized collision scheme. In this study a new type of boundary concept is proposed to approximate non-reflective flow behavior at the boundary. The idea of this concept is to utilize the characteristics of the non-equilibrium distribution function to assign values to the incoming populations. A simple gradient extrapolation technique coupled with a far-field reference vector is used to predict the macroscopic fluid variables.

2 The Boltzmann Transport Equation

Based on kinetic theory, the Boltzmann Transport Equation (BTE) (1) describes the statistical behavior of molecular motion inside a system by using a seven-dimensional *probability density function* f, also referred to a *particle distribution function* (PDF) or simply *distribution function* when using the concept of *fictitious particles*:

$$\frac{\partial f}{\partial t} + \xi_\alpha \frac{\partial f}{\partial x_\alpha} + \frac{F_\alpha}{\rho}\frac{\partial f}{\partial \xi_\alpha} = \Omega(f) \tag{1}$$

where f is a function of time (t) space ($\mathbf{x}$) and *velocity space* ($\boldsymbol{\xi}$). Whenever a medium relaxes towards steady state, the solution of the BTE becomes the *equilibrium distribution function* f^{eq} (EDF):

$$f^{\text{eq}}(\rho, \mathbf{u}, \theta, \boldsymbol{\xi}) = \frac{\rho}{(2\pi\theta)^{d/2}} e^{-\frac{|\boldsymbol{\xi}-\mathbf{u}|^2}{2\theta}} \tag{2}$$

where all quantities are non-dimensional; ρ is the density and θ is the temperature, equal to RT/u_0, in which R and u_0 are respectively the gas constant and characteristic velocity; $\mathbf{u}$ is the macroscopic velocity of the medium and d the number of spatial dimensions. A key component of Eq. (1) is the *collision operator* $\Omega(f)$, which represents all possible ways in which particles can collide with one another:

$$\Omega(f) = -\frac{1}{\tau}(f - f^{\text{eq}}) \tag{3}$$

where τ is the *relaxation time*, a direct function of the transport coefficients of a medium, such as viscosity and heat diffusivity. The *macroscopic moments* like *mass density* (4) and *momentum density* (5) can be obtained by integrating the moments of f respectively f^{eq} over the d-dimensional velocity space:

$$\rho = \int f \, d^d\xi \quad = \int f^{\text{eq}} \, d^d\xi \tag{4}$$

$$\rho u_\alpha = \int \xi_\alpha f \, d^d\xi = \int \xi_\alpha f^{\text{eq}} \, d^d\xi \tag{5}$$

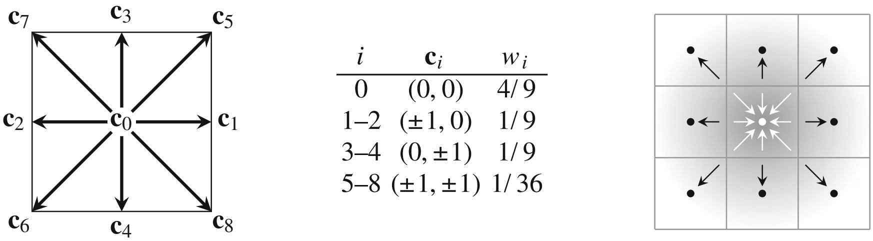
3 The Lattice Boltzmann Method

Based on the Lattice Boltzmann equation (LBE) (6), the Lattice Boltzmann method (LBM) is a direct discretization of the BTE in both time, space and velocity space:

$$f_i(\mathbf{x} + \mathbf{c}_i \Delta t, t + \Delta t) - f_i(\mathbf{x}, t) = \Omega_i(\mathbf{x}, t) \tag{6}$$

The method constructs numerical approximations by iteratively *streaming* and *colliding* discrete distribution functions f_i, confined by the discrete velocities $\mathbf{c}_i$ of a *lattice* (Fig. 1). By introducing compact notation for (6) and substituting the collision operator (3), the LBE can be rewritten as:

$$f_i^* = f_i - \frac{\Delta t}{\tau}\left(f_i - f_i^{\text{eq}}\right) \tag{7}$$



i	$\mathbf{c}_i$	w_i
0	$(0, 0)$	$4/9$
1–2	$(\pm 1, 0)$	$1/9$
3–4	$(0, \pm 1)$	$1/9$
5–8	$(\pm 1, \pm 1)$	$1/36$

Fig. 1 D2Q9 model—Lattice configuration (left) and exchange between lattices (right)

where f_i^* are the discrete *post-collision* distribution functions and Δt is the discrete time step. The discrete equilibrium distribution function f_i^{eq} is given by:

$$f_i^{\text{eq}} = w_i \rho \left(1 + \frac{\mathbf{u} \cdot \mathbf{c}_i}{c_s^2} + \frac{(\mathbf{u} \cdot \mathbf{c}_i)^2}{2c_s^4} - \frac{\mathbf{u} \cdot \mathbf{u}}{2c_s^2} \right) \tag{8}$$

where $c_s = \frac{1}{\sqrt{3}}$ is the LBM speed of sound and w_i are the discrete *weights*, associated with the lattice velocities $\mathbf{c}_i$.

4 The Multiple-Relaxation-Time Collision Model

To increase the accuracy and stability of the current solution, the so-called Multiple-relaxation-time (MRT) collision model has been added. By relaxing the *velocity moments* of $\mathbf{f}$ at different rates, rather than relaxing $\mathbf{f}$ itself at a single rate, the MRT collision model is capable of modelling a large range of Reynolds numbers. Similar to (7) the MRT LBE is given by:

$$\mathbf{f}^* = \mathbf{f} - \mathbf{M}^{-1}\mathbf{S} \left(\mathbf{m} - \mathbf{m}^{\text{eq}} \right) \Delta t \tag{9}$$

where $\mathbf{m}$ and $\mathbf{m}^{\text{eq}}$ are respectively the *velocity moments* and the *equilibrium velocity moments*:

$$\mathbf{m} = \mathbf{M} \cdot \mathbf{f} \qquad \mathbf{m}^{\text{eq}} = \mathbf{M} \cdot \mathbf{f}^{\text{eq}} \tag{10}$$

The quantity $\mathbf{M}$ is a $Q \times Q$ *transformation matrix*, whose entries can be found by constraining the moments of $\mathbf{m}$. Following the Gram-Schmidt (GS) procedure [1], $\mathbf{M}$ can be formed by constructing a set of *mutually orthogonal vectors*, each corresponding to a certain moment of $\mathbf{f}$. The quantity $\mathbf{S}$ in (9) represents the *relaxation matrix* and is used to relax the different velocity moments. In the case of the GS approach, this matrix has the following diagonal form:

$$\mathbf{S} = \mathrm{diag}\left(C_\rho, \omega_e, \omega_\epsilon, C_{j_x}, \omega_q, C_{j_y}, \omega_q, \omega_\nu, \omega_\nu\right) \tag{11}$$

where ω_e and ω_ϵ are the energy relaxation rates; ω_q is the relaxation rate for the energy flux and ω_ν is the viscous relaxation rate. The constants C_ρ, C_{j_x} and C_{j_y} represent the conserved quantities and can be assigned any value.

5 The Non-equilibrium Symmetry Boundary

Based upon the approximately symmetrical shape of the non-equilibrium distribution function f^{neq}, a new type of non-reflective boundary condition (NRBC) has been constructed, known as the Non-equilibrium Symmetry Boundary Condition (NSBC). The key behind the NSBC is the approximation that the discrete non-equilibrium populations f_i^{neq} are assumed to be equal to their anti-symmetric counterparts $f_{\bar{i}}^{\mathrm{neq}}$:

$$f_i^{\mathrm{neq}} = f_{\bar{i}}^{\mathrm{neq}} \tag{12}$$

As a result the incoming populations f_{in} at the boundary nodes can be calculated using the non-equilibrium contributions of the outgoing populations f_{out}:

$$f_{\mathrm{in}} = f_{\mathrm{in}}^{\mathrm{eq}} + f_{\mathrm{out}}^{\mathrm{neq}} \tag{13}$$

As this approach requires the equilibrium populations $f_{\mathrm{in}}^{\mathrm{eq}}$ to be computed first, correct values for the macroscopic fluid vector $\mathbf{m} = (\rho, \mathbf{u})$ need to be predicted in advance. Although there exist various approaches to accomplish this [1], good results were obtained by simply taking the gradient of $\mathbf{m}$ along the normal $\mathbf{n}$ of the boundary, multiplied by the coefficient γ, a relaxation parameter used to minimize the amount of reflection. As for the D2Q9 model $\gamma = 0.6$ was found to give the best results:

$$\mathbf{m}_p = \mathbf{m} - \gamma\, \partial_{\mathbf{n}}\mathbf{m} \tag{14}$$

To allow for the predicted fluid vector $\mathbf{m}_p$ to convergence towards a certain reference fluid vector, a so-called *far-field flow criterion* is introduced, yielding:

$$\mathbf{m}_c = (1 - \beta)\, \mathbf{m}_p + \beta \mathbf{m}_0 \tag{15}$$

where the coefficient β is the far-field factor and $\mathbf{m}_0$ the reference fluid vector. After all boundary populations have been assigned they are corrected by *rescaling* them with respect to ρ_c and *shifting* them with respect to $\mathbf{u}_c$, as to guarantee conservation of the macroscopic moments:

$$\tilde{f}_i = f_i - w_i \left[\Delta\rho + \mathbf{c}_i \Delta(\rho\mathbf{u})\right] \qquad \text{with:} \begin{cases} \Delta\rho & = \sum_i f_i - \rho_c \\ \Delta(\rho\mathbf{u}) & = \sum_i \mathbf{c}_i f_i - \rho_c \mathbf{u}_c \end{cases} \tag{16}$$

After the correction has been performed, the standard collision procedure can be carried out, in which there is no distinction between the boundary and the internal fluid.

6 Test Case 1: Propagation of Acoustic Waves

Acoustic waves, also known as *sound waves*, are characterised by local pressure variations propagating at a certain speed c_s through a medium. When considering this medium to be a fluid with negligible viscosity, the propagation of such waves is governed by the *ideal wave equation*:

$$\nabla^2 s = \frac{1}{c_s^2} \partial_t^2 s \tag{17}$$

A possible solution of (17) is the one-dimensional Gaussian *plane wave* given by:

$$p(x,t) = \rho_0 c_s^2 \left[1 + s(x,t)\right] \tag{18}$$

$$u(x,t) = \mp c_s s(x,t) \tag{19}$$

$$s(x,t) = \frac{\sqrt{e}\zeta}{\rho_0 \lambda} \left(x \pm c_s t\right) e^{-\frac{(x \pm c_s t)^2}{2\lambda^2}} \tag{20}$$

where p is the *total wave pressure*, u the *wave velocity*, s the *condensation*, ζ the *wave amplitude* and λ a steepness factor. To examine the capability of the NSBC to absorb such a wave, a two-dimensional square domain of fluid is considered, containing an initially inhomogeneous distribution of density and velocity, representing a plane wave. To observe the behaviour of the NSBC under different angles of incidence, the wave is configured to approach the boundary under an angle of 60° with respect to the horizontal axis. The domain consists of 128×128 lattice units and is fully bounded by NSBC's ($\gamma = 0.6$ and $f_f = 0.0$). The wave properties are set to $\rho_0 = 1.0$, $\zeta = 0.01$ and $\lambda = l/32$. To approximate Eq. (17), the fluid viscosity is assumed to be zero (e.g. $\tau = 0.5$). The MRT relaxation rates have been chosen as $\omega_e = \omega_\epsilon = \omega_q = 1.9$. The pressure results of Fig. 2 show that the reflectivity of the NSBC is nearly zero.

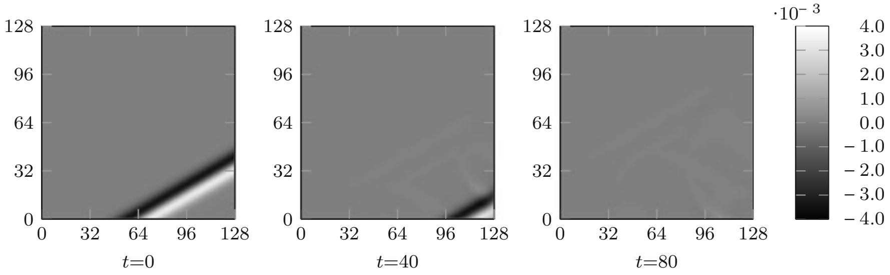


Fig. 2 Test case 1: Propagation of a plane wave under an angle—Pressure results

7 Test Case 2: Convected Vortex

Formed in stirred fluids, vortices are a major component in many flow applications. Due to their characteristics and complex interaction with the surrounding fluid, absorption of vortices can be challenging. To examine the capability of the NSBC in this area, a two-dimensional Lamb-Oseen vortex [1] is convected towards the right boundary of a square domain with a grid size of 128 × 128 lattice units. The vortex is initialized by introducing a local perturbation of the flow field according to:

$$u = u_0 - \beta u_0 \frac{(y - y_0)}{R_c} e^{-\frac{r^2}{2R_c}} \tag{21}$$

$$v = \beta u_0 \frac{(x - x_0)}{R_c} e^{-\frac{r^2}{2R_c}} \tag{22}$$

$$\rho = \left[1 - \frac{(\beta u_0)^2}{2C_v} e^{-\frac{r^2}{2}}\right]^{\frac{1}{\gamma - 1}} \tag{23}$$

$$r = (x - x_0)^2 + (y - y_0)^2 \tag{24}$$

where $u_0 = 0.1$ is the reference velocity, $\beta = 0.5$ a coefficient and $R_c = 20$ the vortex radius (All quantities are in lattice units). The *gas constant* γ and the *volumetric heat capacity* C_v are defined by:

$$\gamma = \frac{d + 2}{d} \qquad C_v = \frac{d}{2} c_s^2 \tag{25}$$

where d is the number of spatial dimensions. The Reynolds number equals $\text{Re} = 10^3$ and is based on u_0 and the size of the computational domain. Concerning the domain boundaries; a Dirichlet velocity boundary [1] is defined at the left and an NSBC with $\gamma = 0.6$ and $f_f = 0.0$ is defined at the right; the bottom and top boundaries are assumed to be periodic. The MRT relaxation rates are $\omega_e = \omega_\epsilon = 1.4$

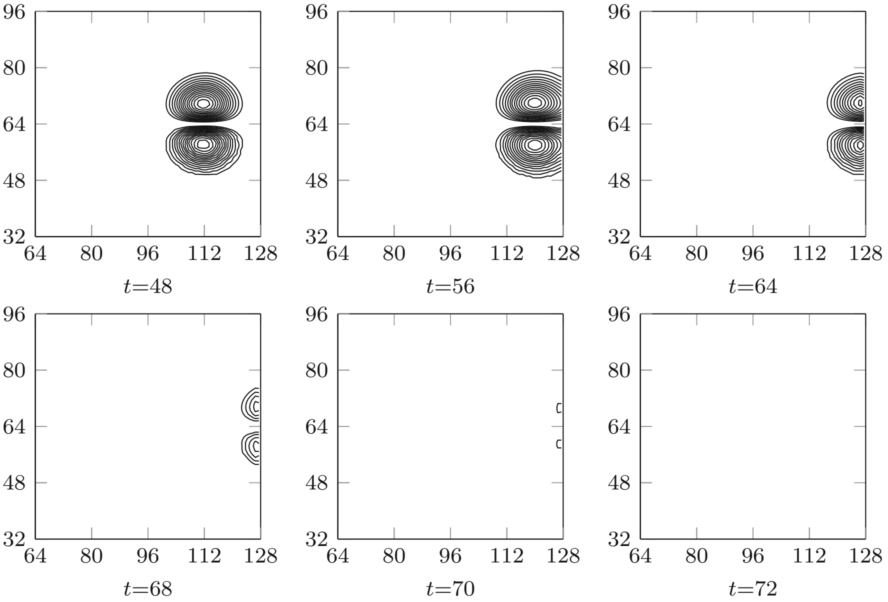


Fig. 3 Test case 5: Convected vortex—Isovalues of longitudinal velocity

and $\omega_q = 1.2$. The results from Fig. 3 show that the vortex is fully absorbed by the boundary.

8 Summary

A new type of non-reflective boundary formulation (NSBC) is proposed, based on the approximately symmetrical shape of the non-equilibrium distribution function. In this formulation, the incoming populations at a boundary node are assigned the non-equilibrium contributions of the outgoing populations. The equilibrium contributions of these incoming populations are computed using a predicted macroscopic fluid vector, determined by a simple gradient based extrapolation method. Additionally, a far-field flow criterion can be applied to this fluid vector, to allow for convergence towards a certain reference value. After all populations have been assigned, they are rescaled and shifted with respect to the fluid vector, as to satisfy conservation of the macroscopic moments. To examine the non-reflectiveness of proposed boundary condition, two different tests have been carried out. In the first test the capability of the NSBC to absorb acoustic waves has been studied. Results show that the reflections are nearly zero, even when considering a large angle of incidence. As a second test the absorption of a convected vortex has been modelled. Isovalues of the longitudinal velocity indicate that the vortex is completely absorbed by the boundary. To summarize, the NSBC has found to be an interesting alternative for modelling non-reflective boundaries. However further investigations are needed to determine the validity of present boundary formulation.

Reference

1. Euser, R., Vuik, C.: A non-reflective boundary condition for LBM based on the assumption of non-equilibrium symmetry. ArXiv. 2005.08674 (2020)

ALE Space-Time Discontinuous Galerkin Method for the Interaction of Compressible Flow with Linear and Nonlinear Dynamic Elasticity and Applications to Vocal Fold Vibrations

Miloslav Feistauer, Monika Balázsová, and Jaromír Horáček

Abstract The paper deals with the discontinuous Galerkin method (DGM) for the solution of compressible Navier-Stokes equations in the ALE form in time-dependent domains combined with the solution of linear and nonlinear dynamic elasticity. The developed methods are oriented to fluid-structure interaction (FSI), particularly to the simulation of air flow in a time-dependent domain representing vocal tract and vocal folds vibrations. We compare results obtained with the aid of linear and nonlinear elasticity models. The results show that it is more adequate to use the nonlinear elasticity St. Venant-Kirchhoff model in contrast to the linear elasticity model.

1 Compressible Navier-Stokes Problem in a Time Dependent Domain and Dynamic Elasticity Problem

1.1 Compressible Flow

We are concerned with the problem of compressible flow in a time-dependent bounded domain $\Omega_t \subset \mathbb{R}^2$ with $t \in [0, T]$. The boundary of Ω_t is formed by three disjoint parts: $\partial\Omega_t = \Gamma_I \cup \Gamma_O \cup \Gamma_{W_t}$, where Γ_I is the inlet, Γ_O is the outlet and Γ_{W_t} represents impermeable time-dependent walls.

M. Feistauer (✉) · M. Balázsová
Charles University, Faculty of Mathematics and Physics, Praha 8, Czech Republic
e-mail: feist@karlin.mff.cuni.cz

J. Horáček
Institute of Thermomechanics, The Academy of Sciences of the Czech Republic, Praha 8, Czech Republic
e-mail: jaromir@it.cas.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_40

The time dependence of the domain Ω_t is taken into account with the aid of the *Arbitrary Lagrangian-Eulerian* (ALE) method. It is based on a regular one-to-one ALE mapping of the reference configuration Ω_0 onto the current configuration Ω_t : $\mathcal{A}_t$: $\bar{\Omega}_0 \longrightarrow \bar{\Omega}_t$. Further, we define the domain velocity $\tilde{z}(\boldsymbol{X}, t) = \frac{\partial}{\partial t}\mathcal{A}_t(\boldsymbol{X}), \quad t \in [0, T]\,, \ \boldsymbol{X} \in \Omega_0, z(\boldsymbol{x}, t) = \tilde{z}(\mathcal{A}_t^{-1}(\boldsymbol{x}), t), \ t \in [0, T]\,, \ \boldsymbol{x} \in \Omega_t$ and the ALE derivative of the state vector function $\boldsymbol{w} = \boldsymbol{w}(\boldsymbol{x}, t)$ defined for $\boldsymbol{x} \in \Omega_t$ and $t \in [0, T]$: $\frac{D^{\mathcal{A}}}{Dt}\boldsymbol{w}(\boldsymbol{x}, t) = \frac{\partial \tilde{\boldsymbol{w}}}{\partial t}(\boldsymbol{X}, t)$, where $\tilde{\boldsymbol{w}}(\boldsymbol{X}, t) = \boldsymbol{w}(\mathcal{A}_t(\boldsymbol{X}), t), \ \boldsymbol{X} \in \Omega_0, \ \boldsymbol{x} = \mathcal{A}_t(\boldsymbol{X})$. Then the continuity equation, the Navier-Stokes equations and the energy equation can be written in the ALE form

$$\frac{D^{\mathcal{A}}\boldsymbol{w}}{Dt} + \sum_{s=1}^{2} \frac{\partial \boldsymbol{g}_s(\boldsymbol{w})}{\partial x_s} + \boldsymbol{w}\mathrm{div}z = \sum_{s=1}^{2} \frac{\partial \boldsymbol{R}_s(\boldsymbol{w}, \nabla \boldsymbol{w})}{\partial x_s}, \tag{1}$$

where $\boldsymbol{w} = (\rho, \rho v_1, \rho v_2, E)^T \in \mathbb{R}^4$, $\boldsymbol{g}_s(\boldsymbol{w}) = \boldsymbol{f}_s(\boldsymbol{w}) - z_s\boldsymbol{w}$, $\boldsymbol{f}_s(\boldsymbol{w}) = (\rho v_s, \rho v_1 v_s + \delta_{1s}p, \rho v_2 v_s + \delta_{2s}p, (E+p)v_s)^T$, $\boldsymbol{R}_s(\boldsymbol{w}, \nabla\boldsymbol{w}) = (0, \tau_{s1}^V, \tau_{s2}^V, \tau_{s1}^V v_1 + \tau_{s2}^V v_2 + k\frac{\partial\theta}{\partial x_s})^T$, $s = 1, 2$, $\tau_{ij}^V = \lambda\delta_{ij}\mathrm{div}\boldsymbol{v} + 2\mu d_{ij}(\boldsymbol{v}), d_{ij}(\boldsymbol{v}) = \frac{1}{2}\left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i}\right)$, $i, j = 1, 2$. We have $\boldsymbol{R}_s(\boldsymbol{w}, \nabla\boldsymbol{w}) = \sum_{k=1}^{2} \mathbb{K}_{s,k}(\boldsymbol{w})\frac{\partial \boldsymbol{w}}{\partial x_k}$, where $\mathbb{K}_{s,k}(\boldsymbol{w})$ are 4×4 matrices depending on $\boldsymbol{w}$, and $\boldsymbol{f}_s(\boldsymbol{w}) = \mathring{A}(\boldsymbol{w})\boldsymbol{w}$ with $\mathring{A}(\boldsymbol{w}) = D\boldsymbol{f}_s(\boldsymbol{w})/\boldsymbol{w}$, see [2]. The following notation is used: ρ—fluid density, p—pressure, E—total energy, $\boldsymbol{v} = (v_1, v_2)$—velocity vector, θ—absolute temperature, $c_v > 0$—specific heat at constant volume, $\gamma > 1$—Poisson adiabatic constant, $\mu > 0, \lambda = -2\mu/3$—viscosity coefficients, $k > 0$—heat conduction coefficient, τ_{ij}^V—components of the viscous part of the stress tensor. System (1) is completed by the thermodynamical relations $p = (\gamma - 1)\left(E - \rho\frac{|\boldsymbol{v}|^2}{2}\right), \quad \theta = \frac{1}{c_v}\left(\frac{E}{\rho} - \frac{|\boldsymbol{v}|^2}{2}\right)$ and equipped with the initial condition $\boldsymbol{w}(\boldsymbol{x}, 0) = \boldsymbol{w}^0(\boldsymbol{x}), \ \boldsymbol{x} \in \Omega_0$ and the boundary conditions:
$\rho = \rho_D, \quad \boldsymbol{v} = \boldsymbol{v}_D, \quad \sum_{j=1}^{2}\left(\sum_{i=1}^{2} \tau_{ij}^V n_i\right) v_j + k\frac{\partial\theta}{\partial \boldsymbol{n}} = 0$ on the inlet Γ_I,
$\boldsymbol{v} = z_D(t) =$ velocity of a moving wall, $\quad \frac{\partial\theta}{\partial\boldsymbol{n}} = 0$, on the moving wall Γ_{W_t},
$\sum_{j=1}^{2} \tau_{ij}^V n_j = 0, \quad \frac{\partial\theta}{\partial\boldsymbol{n}} = 0, \quad i = 1, 2$, on the outlet Γ_O,
with prescribed data $\rho_D, \ \boldsymbol{v}_D, \ z_D$. By $\boldsymbol{n}$ we denote the unit outer normal.

1.2 Dynamic Elasticity

We assume that an elastic body is represented by a bounded domain $\Omega^b \subset \mathbb{R}^2$ with boundary $\partial\Omega^b = \Gamma_D^b \cup \Gamma_N^b$. Let $T > 0$. We seek a displacement function $\boldsymbol{u} : \Omega^b \times [0, T] \to \mathbb{R}^2$ such that

$$\rho\frac{\partial^2 \boldsymbol{u}}{\partial t^2} + c_M\rho\frac{\partial \boldsymbol{u}}{\partial t} - \mathrm{div}\boldsymbol{P}(\boldsymbol{F}) = \boldsymbol{f} \qquad \text{in } \Omega^b \times [0, T], \tag{2}$$

$$\boldsymbol{u} = \boldsymbol{u}_D \quad \text{in } \Gamma_D^b \times [0, T], \tag{3}$$

$$\boldsymbol{P}(\boldsymbol{F}) \cdot \boldsymbol{n} = \boldsymbol{g}_N \quad \text{in } \Gamma_N^b \times [0, T], \tag{4}$$

$$\boldsymbol{u}(\cdot, 0) = \boldsymbol{u}_0, \quad \frac{\partial \boldsymbol{u}}{\partial t}(\cdot, 0) = \boldsymbol{y}_0 \qquad \text{in } \Omega^b. \tag{5}$$

Here $\boldsymbol{f}$ is outer volume force, $\rho > 0$ is material density, $\boldsymbol{P}$ denotes stress tensor and the quantity $\boldsymbol{F}$ depends on $\boldsymbol{u}$ as shown further. The expression $c_M \rho \frac{\partial \boldsymbol{u}}{\partial t}$ with $c_M > 0$ represents structural damping.

In case of linear elasticity the stress tensor depends linearly on the strain tensor $\boldsymbol{e}(\boldsymbol{u}) = (\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^T)/2$ according to the relation $\boldsymbol{P}(\boldsymbol{F}) = \boldsymbol{\sigma}(\boldsymbol{u}) = \lambda^b \operatorname{tr}(\boldsymbol{e}(\boldsymbol{u}))\mathbb{I} + 2\mu^b \boldsymbol{e}(\boldsymbol{u})$. Here λ^b and μ^b are the Lamé parameters that can be expressed with the aid of the Young modulus E and the Poisson ratio ν: $\lambda^b = \frac{E\nu}{(1+\nu)(1-2\nu)}, \quad \mu^b = \frac{E}{2(1+\nu)}$.

In the case of nonlinear model we introduce the deformation mapping $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$, deformation gradient (i.e., the Jacobian matrix of the deformation mapping $\boldsymbol{\varphi}$) $\boldsymbol{F} := \nabla \boldsymbol{\varphi}(\boldsymbol{x})$, the Jacobian of the deformation $J = \det \boldsymbol{F} > 0$ and the Green strain tensor $\boldsymbol{E} = (E_{ij})_{i,j=1}^2 = \boldsymbol{e} + \boldsymbol{E}^*$ with components

$$E_{ij} = \underbrace{\frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right)}_{e_{ij}-\text{linear part}} + \underbrace{\frac{1}{2}\sum_{k=1}^{2} \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}}_{E^*_{ij}-\text{nonlinear part}} . \tag{6}$$

In the case of a nonlinear material we consider the St. Venant-Kirchhoff model with the stress tensor defined by the following relations:

$$\boldsymbol{\Sigma} = \lambda^b \operatorname{tr}(\boldsymbol{E})\boldsymbol{I} + 2\mu^b \boldsymbol{E}, \quad \boldsymbol{P}(\boldsymbol{F}) = \boldsymbol{F}\boldsymbol{\Sigma}, \tag{7}$$

where $\boldsymbol{\Sigma}$ is the second Piola-Kirchhoff stress tensor. Writing $\boldsymbol{\Sigma}(\boldsymbol{u}) = (\Sigma_{ij})_{i,j=1}^2$, we get

$$\Sigma_{ij} = \lambda^b \left(\sum_{l=1}^{2} \frac{\partial u_l}{\partial x_l} + \frac{1}{2}\sum_{l=1}^{2}\sum_{k=1}^{2}\left(\frac{\partial u_k}{\partial x_l}\right)^2\right)\delta_{ij} + \mu^b \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} + \sum_{k=1}^{2} \frac{\partial u_k}{\partial x_i}\frac{\partial u_k}{\partial x_j}\right). \tag{8}$$

For a detailed description we can refer the reader to the monograph [1].

1.3 Transmission Conditions

On the common boundary between fluid and structure

$\tilde{\Gamma}_{Wt} = \left\{\boldsymbol{x} \in \mathbb{R}^2;\ \boldsymbol{x} = \boldsymbol{X} + \boldsymbol{u}(\boldsymbol{X}, t),\ \boldsymbol{X} \in \Gamma_N^b\right\}$ we consider interface conditions representing the continuity of the normal stress and velocity:

(a) linear elasticity:

$$\sum_{j=1}^{2}\sigma_{ij}^{b}(\boldsymbol{X})n_j(\boldsymbol{X})=\sum_{j=1}^{2}\tau_{ij}^{f}(\boldsymbol{x})n_j(\boldsymbol{X}),\ i=1,2,\boldsymbol{v}(\boldsymbol{x},t)=\frac{\partial \boldsymbol{u}(\boldsymbol{X},t)}{\partial t},$$

(b) nonlinear elasticity:

$$\boldsymbol{P}(\boldsymbol{F}(\boldsymbol{u}(\boldsymbol{X},t)))\boldsymbol{n}(\boldsymbol{x})=\boldsymbol{\tau}^{f}(\boldsymbol{x},t)\mathrm{Cof}(\boldsymbol{F}(\boldsymbol{u}(\boldsymbol{X},t)))\boldsymbol{n}(\boldsymbol{x}),\boldsymbol{v}(\boldsymbol{x},t)=\frac{\partial \boldsymbol{u}(\boldsymbol{X},t)}{\partial t}.$$

Here $\boldsymbol{\tau}^f=\{\tau_{ij}^f\}_{i,j=1}^2$ is the stress tensor of the fluid.

2 Discretization

In both flow and elasticity problems we assume that the domains Ω_t and Ω^b are polygonal.

2.1 Discretization of the Flow Problem

The flow problem is discretized by the space-time discontinuous Galerkin method (STDGM). By $\mathcal{T}_{ht}$ we denote a triangulation of the domain Ω_{ht} with standard properties. By $\mathcal{F}_{ht}$, $\mathcal{F}_{ht}^B$, $\mathcal{F}_{ht}^I$ we denote the systems of all faces of all elements $K\in\mathcal{T}_{ht}$, boundary faces and inner faces, respectively. Further, we introduce the set of "Dirichlet" boundary faces $\mathcal{F}_{ht}^D=\{\Gamma\in\mathcal{F}_{ht}^B$; a Dirichlet condition is prescribed on $\Gamma\}$. Each face Γ is associated with a unit normal $\boldsymbol{n}_\Gamma$, which has the same orientation as the outer normal on $\Gamma\in\mathcal{F}_{ht}^B$. We set h_Γ = length of Γ. The symbol $\langle\cdot\rangle$ denotes the mean value from the sides of $\Gamma\in\mathcal{F}_{ht}^I$, and $[\cdot]$ denotes the jump on Γ.

For the space-time discretization we introduce a partition $0=t_0<t_1<\ldots<t_M=T$ of the time interval $[0,T]$ and denote $I_m=(t_{m-1},t_m)$, $\tau_m=t_m-t_{m-1}$, for $m=1,\ldots,M$. Then we define the space $\mathrm{SS}_{ht}^r=\{v;\,v|K\in P_r(K)\ \forall K\in\mathcal{T}_{ht}\}^4$ and the approximate solution is sought in the space

$$\boldsymbol{S}_{h\tau}^{rq}=\Big\{\phi\ ;\ \phi|_{I_m}=\sum_{i=0}^{q}\zeta_i\phi_i,\ \text{where}\ \phi_i\in\boldsymbol{S}_{ht}^r,\ \zeta_i\in P^q(I_m)\Big\}^2, \qquad (9)$$

where integers $r,q\geq 1$, $P_r(K)$ denotes the space of all polynomials on K of degree $\leq r$ and $P^q(I_m)$ denotes the space of all polynomials in t on I_m of degree $\leq q$. For $\boldsymbol{\varphi}\in\boldsymbol{S}_{h\tau}^{rq}$ we set $\boldsymbol{\varphi}_m^{\pm}=\boldsymbol{\varphi}(t_m^{\pm})=\lim_{t\to t_m\pm}\boldsymbol{\varphi}(t)$, $\{\boldsymbol{\varphi}\}_m=\boldsymbol{\varphi}_m^{+}-\boldsymbol{\varphi}_m^{-}$. The initial state $\boldsymbol{w}_{h\tau}(0-) \in S_{h0}^{p}$ is defined as the $L^2(\Omega_{h0})$-projection of $\boldsymbol{w}^0$ on $\boldsymbol{S}_{h0}^{r}$. Moreover, we introduce the prolongation $\overline{\boldsymbol{w}}_{h\tau}(t)$ of $\boldsymbol{w}_{h\tau}|_{I_{m-1}}$ on the interval I_m. By $(\cdot,\cdot)_t$ we denote the $L^2(\Omega_{ht})$-scalar product.

The discrete problem is based on the use of the following forms defined for $\overline{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h \in \mathrm{SS}_{ht}^{r}$ (see, e.g. [2]):

$$\hat{a}_h(\overline{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h, t) = \sum_{K\in\mathcal{T}_{ht}} \int_K \sum_{s=1}^{2}\sum_{k=1}^{2} \mathbb{K}_{s,k}(\overline{\boldsymbol{w}}_h)\, \frac{\partial \boldsymbol{w}_h}{\partial x_k} \cdot \frac{\partial \boldsymbol{\varphi}_h}{\partial x_s}\,\mathrm{d}\boldsymbol{x} \tag{10}$$

$$- \sum_{\Gamma\in\mathcal{F}_{ht}^{I}} \int_\Gamma \sum_{s=1}^{2} \left\langle \sum_{k=1}^{2} \mathbb{K}_{s,k}(\overline{\boldsymbol{w}}_h) \frac{\partial \boldsymbol{w}_h}{\partial x_k} \right\rangle (\boldsymbol{n}_\Gamma)_s \cdot [\boldsymbol{\varphi}_h]\,\mathrm{d}S$$

$$- \sum_{\Gamma\in\mathcal{F}_{ht}^{D}} \int_\Gamma \sum_{s=1}^{2}\sum_{k=1}^{2} \mathbb{K}_{s,k}(\overline{\boldsymbol{w}}_h) \frac{\partial \boldsymbol{w}_h}{\partial x_k} (\boldsymbol{n}_\Gamma)_s \cdot \boldsymbol{\varphi}_h\,\mathrm{d}S,$$

$$J_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h, t) \tag{11}$$

$$= \sum_{\Gamma\in\mathcal{F}_{ht}^{I}} \int_\Gamma \frac{\mu C_W}{h_\Gamma} [\boldsymbol{w}_h]\cdot[\boldsymbol{\varphi}_h]\,\mathrm{d}S + \sum_{\Gamma\in\mathcal{F}_{ht}^{D}} \int_\Gamma \frac{\mu C_W}{h_\Gamma} \boldsymbol{w}_h \cdot \boldsymbol{\varphi}_h\,\mathrm{d}S,$$

$$d_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h, t) = \sum_{K\in\mathcal{T}_{ht}} \int_K (\boldsymbol{w}_h \cdot \boldsymbol{\varphi}_h)\,\mathrm{div}\boldsymbol{z}\,\mathrm{d}x, \tag{12}$$

$$\hat{b}_h(\overline{\boldsymbol{w}}_h, \boldsymbol{w}_h, \boldsymbol{\varphi}_h, t) \tag{13}$$

$$= - \sum_{K\in\mathcal{T}_{ht_{k+1}}} \int_K \sum_{s=1}^{2} (\mathring{\mathrm{A}}_s(\overline{\boldsymbol{w}}_h(x)) - z_s(x))\mathbb{I})\boldsymbol{w}_h(x))\cdot \frac{\partial \boldsymbol{\varphi}_h(x)}{\partial x_s} dx$$

$$+ \sum_{\Gamma\in\mathcal{F}_{ht}^{I}} \int_\Gamma \Big(\mathbb{P}_g^{+}\big(\langle\overline{\boldsymbol{w}}_h\rangle_\Gamma, \boldsymbol{n}_\Gamma\big)\boldsymbol{w}_h^{(L)} + \mathbb{P}_g^{-}\big(\langle\overline{\boldsymbol{w}}_h\rangle_\Gamma, \boldsymbol{n}_\Gamma\big)\boldsymbol{w}_h^{(R)}\Big)\cdot [\boldsymbol{\varphi}_h]\,\mathrm{d}S$$

$$+ \sum_{\Gamma\in\mathcal{F}_{ht}^{B}} \int_\Gamma \Big(\mathbb{P}_g^{+}\big(\langle\overline{\boldsymbol{w}}_h\rangle_\Gamma, \boldsymbol{n}_\Gamma\big)\boldsymbol{w}_h^{(L)} + \mathbb{P}_g^{-}\big(\langle\overline{\boldsymbol{w}}_h\rangle_\Gamma, \boldsymbol{n}_\Gamma\big)\overline{\boldsymbol{w}}_h^{(R)}\Big)\cdot \boldsymbol{\varphi}_h\,\mathrm{d}S,$$

$$\ell_h(\boldsymbol{w}_h, \boldsymbol{\varphi}_h, t) = \sum_{\Gamma\in\mathcal{F}_{ht}^{D}} \int_\Gamma \frac{\mu C_W}{h_\Gamma} \boldsymbol{w}_B \cdot \boldsymbol{\varphi}_h\,\mathrm{d}S, \quad (\phi, \psi)_{\Omega_t} = \int_{\Omega_t} \phi\psi\,\mathrm{d}x. \tag{14}$$

Here $C_W > 0$ is a sufficiently large constant. The symbols $\mathbb{P}_g^+(\boldsymbol{w}, \boldsymbol{n})$ and $\mathbb{P}_g^-(\boldsymbol{w}, \boldsymbol{n})$ denote the "positive" and "negative" parts of the matrix $\mathbb{P}_g(\boldsymbol{w}, \boldsymbol{n}) = \sum_{s=1}^2 (\mathring{\mathrm{A}}_s(\boldsymbol{w}) - z_s \mathbb{I}) n_s$ defined, e.g., in [2]. The boundary state $\boldsymbol{w}_B$ is defined on the basis of the prescribed Dirichlet boundary conditions and extrapolation.

Now the *space-time DG approximate solution* is defined as a function $\boldsymbol{w}_{h\tau} \in \S_{h\tau}^{rq}$ satisfying the following relation for $m = 1, \dots, M$:

$$\int_{I_m} \left(\left(\frac{D^{\mathcal{A}} \boldsymbol{w}_{h\tau}}{Dt}(t), \boldsymbol{\varphi}_{h\tau} \right)_{\Omega_t} + \hat{a}_h(\overline{\boldsymbol{w}}_{h\tau}, \boldsymbol{w}_{h\tau}, \boldsymbol{\varphi}_{h\tau}, t) \right) \mathrm{d}t \tag{15}$$

$$+ \int_{I_m} \left(\hat{b}_h(\overline{\boldsymbol{w}}_{h\tau}, \boldsymbol{w}_{h\tau}, \boldsymbol{\varphi}_{h\tau}, t) + J_h(\boldsymbol{w}_{h\tau}, \boldsymbol{\varphi}_{h\tau}, t) + d_h(\boldsymbol{w}_{h\tau}, \boldsymbol{\varphi}_{h\tau}, t) \right) \mathrm{d}t$$

$$+ (\{\boldsymbol{w}_{h\tau}\}_{m-1}, \boldsymbol{\varphi}_{h\tau}(t_{m-1}+)) = \int_{I_m} \ell_h(\boldsymbol{w}_{hD}, \boldsymbol{\varphi}_{h\tau}, t) \,\mathrm{d}t, \quad \forall \boldsymbol{\varphi}_{h\tau} \in \mathbf{S}_{h\tau}^{rq}.$$

The function $\overline{\boldsymbol{w}}_{h\tau}$ is a prolongation of $\boldsymbol{w}_{h\tau}|_{I_{m-1}}$ to the time interval I_m.

It can be seen that this relation is equivalent to a linear algebraic system on every time interval I_m, $m = 1, \dots, M$.

2.2 Discretization of the Elasticity Problem

Because of the discretization of the elasticity problem we consider the displacement $\boldsymbol{u}$ and introduce the deformation velocity $\boldsymbol{y}$. The basic system (2) is split into two systems of first-order in time:

$$\rho^b \frac{\partial \boldsymbol{y}}{\partial t} + c_M \rho^b \boldsymbol{y} - \mathrm{div} \boldsymbol{P}(\boldsymbol{F}) = \boldsymbol{f}, \quad \frac{\partial \boldsymbol{u}}{\partial t} - \boldsymbol{y} = 0 \quad \text{in } \Omega^b \times [0, T], \tag{16}$$

with boundary and initial conditions (3)–(5)

Now we proceed in a similar way as in the flow problem. In the domain Ω^b we construct a triangulation $\mathcal{T}_h$. The approximate solution at every time instant $t \in [0, T]$ will be sought in the finite-dimensional space

$$\boldsymbol{S}_{hs} = \left\{ v \in L^2(\Omega^b);\, v|_K \in P_s(K),\, K \in \mathcal{T}_h \right\}^2, \tag{17}$$

where $s > 0$ is an integer. By $\mathcal{F}_h$ we denote the system of all faces of all elements $K \in \mathcal{T}_h^b$ and distinguish there sets of boundary, "Dirichlet", "Neumann" and inner faces: $\mathcal{F}_h^{Bb} = \left\{ \Gamma \in \mathcal{F}_h^b;\, \Gamma \subset \partial\Omega^b \right\}$, $\mathcal{F}_h^{Db} = \left\{ \Gamma \in \mathcal{F}_h;\, \Gamma \subset \Gamma_D^b \right\}$, $\mathcal{F}_h^{Nb} = \left\{ \Gamma \in \mathcal{F}_h^b;\, \Gamma \subset \Gamma_N^b \right\}$ and $\mathcal{F}_h^{Ib} = \mathcal{F}_h \backslash \mathcal{F}_h^{Bb}$.

Elasticity forms:

$$a_h^b(\boldsymbol{u},\boldsymbol{\varphi}) = \sum_{K\in\mathcal{T}_h^b}\int_K \boldsymbol{P}\left(\boldsymbol{F}\right) : \nabla\boldsymbol{\varphi}\,\mathrm{d}x - \sum_{\Gamma\in\mathcal{F}_h^{Ib}}\int_\Gamma \left(\langle\boldsymbol{P}\left(\boldsymbol{F}\right)\rangle\boldsymbol{n}\right)\cdot[\boldsymbol{\varphi}]\,\mathrm{d}S \tag{18}$$

$$- \sum_{\Gamma\in\mathcal{F}_h^{Db}}\int_\Gamma \left(\boldsymbol{P}\left(\boldsymbol{F}\right)\boldsymbol{n}\right)\cdot\boldsymbol{\varphi}\,\mathrm{d}S,$$

$$J_h^b(\boldsymbol{u},\boldsymbol{\varphi}) = \sum_{\Gamma\in\mathcal{F}_h^I}\int_\Gamma \frac{C_W^b}{h_\Gamma}\,[\boldsymbol{u}]\cdot[\boldsymbol{\varphi}]\,\mathrm{d}S + \sum_{\Gamma\in\mathcal{F}_h^D}\int_\Gamma \frac{C_W}{h_\Gamma}\boldsymbol{u}\cdot\boldsymbol{\varphi}\,\mathrm{d}S, \tag{19}$$

$$\ell_h^b(\boldsymbol{\varphi})(t) = \sum_{K\in\mathcal{T}_h}\int_K \boldsymbol{f}(t)\cdot\boldsymbol{\varphi}\,\mathrm{d}x + \sum_{\Gamma\in\mathcal{F}_h^N}\int_\Gamma \boldsymbol{g}_N(t)\cdot\boldsymbol{\varphi}\,\mathrm{d}S \tag{20}$$

$$A_h^b = a_h^b + J_h^b, \quad (\boldsymbol{u},\boldsymbol{\varphi})_{\Omega^b} = \int_{\Omega^b}\boldsymbol{u}\cdot\boldsymbol{\varphi}\,\mathrm{d}x. \tag{21}$$

The time discretization is carried out with the use of a backward difference formula (BDF) $\frac{\partial\boldsymbol{u}}{\partial t}(t_m) \approx \frac{D_{\mathrm{appr}}\boldsymbol{u}_h^m}{Dt} = \alpha_0\,\boldsymbol{u}_h^m + \sum_{\ell=1}^q \alpha_\ell\boldsymbol{u}_h^{m-\ell}$. Similar formula is used for the approximation of $\partial\boldsymbol{y}/\partial t$. The coefficients α_ℓ depend on time steps $\tau_m, \tau_{m-1}, \ldots$. See, e.g. Tables 8.2 and 8.3 in [2].

Now we come to the complete BDF-DG discrete problem: Find $\boldsymbol{u}_h^m$, $\boldsymbol{z}_h^m \in \boldsymbol{S}_{hs}$ such that for all $\boldsymbol{\varphi}_h \in \boldsymbol{S}_{hs},\ m = 1,\ldots,M$,

$$\text{(a)}\left(\rho\frac{D_{\mathrm{appr}}\boldsymbol{z}_h^m}{Dt},\boldsymbol{\varphi}_h\right)_{\Omega^b} + c_M\left(\rho\,\boldsymbol{y}_h^m,\boldsymbol{\varphi}_h\right)_{\Omega^b} + A_h^b\left(\boldsymbol{u}_h^m,\boldsymbol{\varphi}_h\right) = \ell_h^b(\boldsymbol{\varphi}_h)\,(t_m), \tag{22}$$

$$\text{(b)}\left(\frac{D_{\mathrm{appr}}\boldsymbol{u}_h^m}{Dt},\,\boldsymbol{\varphi}_h\right)_{\Omega^b} - \left(\boldsymbol{y}_h^m,\boldsymbol{\varphi}_h\right)_{\Omega^b} = 0, \tag{23}$$

$$\text{(c)}\left(\boldsymbol{u}_h^0,\boldsymbol{\varphi}_h\right)_{\Omega^b} = \left(\boldsymbol{u}_0,\boldsymbol{\varphi}_h\right)_{\Omega^b},\quad \left(\boldsymbol{y}_h^0,\boldsymbol{\varphi}_h\right)_{\Omega^b} = \left(\boldsymbol{y}_0,\boldsymbol{\varphi}_h\right)_{\Omega^b}. \tag{24}$$

Nonlinear discrete problems are solved by the Newton method. Linear systems are solved by the direct solver UMFPACK or iterative method GMRES.

3 Numerical Experiments

Our goal is to apply the developed method to the simulation of flow-induced vibrations of vocal folds excited by airflow coming from a model of trachea, through the glottis region to the vocal tract model ended in ambient air. The geometry of the domain occupied by the fluid and vocal folds is shown in Fig. 1. Moreover we add to this geometry a semicircle subdomain with a radius 3.0 cm as an outlet Γ_O.

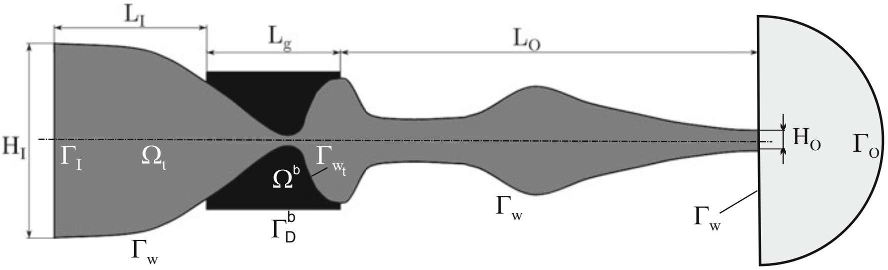
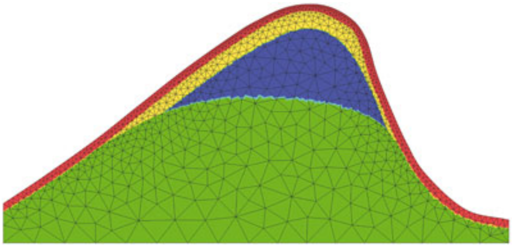


Fig. 1 Geometry of the computational domain at time $t = 0$ and the description of its size: $L_I = 20.0\,\text{mm}$, $L_g = 17.5\,\text{mm}$, $L_O = 55.0\,\text{mm}$, $H_I = 25.5\,\text{mm}$, $H_O = 2.76\,\text{mm}$



E^b [kPa]	μ^b [-]
12	0.4
8	0.4
1	0.495
100	0.4

Fig. 2 Nonhomogeneous model of vocal folds—values of Young modulus E^b and Poisson ratio μ^b

The fluid flow problem is computed on the triangulation with 17,652 elements. Further, for the fluid flow problem the following data are used: magnitude of the inlet velocity $v_{in} = 4{,}\mathrm{m\,s^{-1}}$, dynamic viscosity $\mu = 1.80 \cdot 10^{-5}\,\mathrm{kg\,m^{-1}\,s^{-1}}$, inlet density $\rho_{in} = 1.225\,\mathrm{kg\,m^{-3}}$, outlet pressure $p_{out} = 97{,}611\,\mathrm{Pa}$, Reynolds number $Re = \rho_{in} v_{in} H_I / \mu = 6941.7$, $\kappa = 2.428 \cdot 10^{-2}\,\mathrm{kg\,m\,s^{-3}\,K^{-1}}$, $c_v = 721.428\,\mathrm{m^2\,s^{-2}\,K^{-1}}$, $\gamma = 1.4$. We use the polynomial approximation of degree 2 in space and degree 1 in time. We employ the penalization constant $C_W = 500$ for inner faces and $C_W = 5000$ for boundary edges. The time step τ is set to $10^{-6}\,\mathrm{s}$. In the elasticity problem we consider the St. Venant-Kirchhoff model. We set $\rho^b = 1040\,\mathrm{kg\,m^{-3}}$. The triangulation has 5118 elements. The division of the domain into four regions with different material characteristics is illustrated in Fig. 2 with the material characteristics ordered from the lower layer to the upper one. The penalization constant $C_W^b = 4 \cdot 10^6$, the BDF method of order 2 and piecewise linear approximation in space are used.

Figure 3 shows velocity field in the glottal region at two time instants of the vocal folds self-oscillation. In these time instants different jet declination behind the channel constriction, i.e. the Coanda effect can be observed.

There is a question if it is possible to use a linear elasticity model or if it is necessary to apply a nonlinear model. It is tested with the ratio R, computed from the strain tensor, defined by $R := \frac{\|\boldsymbol{e}\|}{\|\boldsymbol{E}\|} = \frac{\|\boldsymbol{e}\|}{\|\boldsymbol{e}+\boldsymbol{E}^*\|}$ (see (6)). If $R \approx 1$, then the

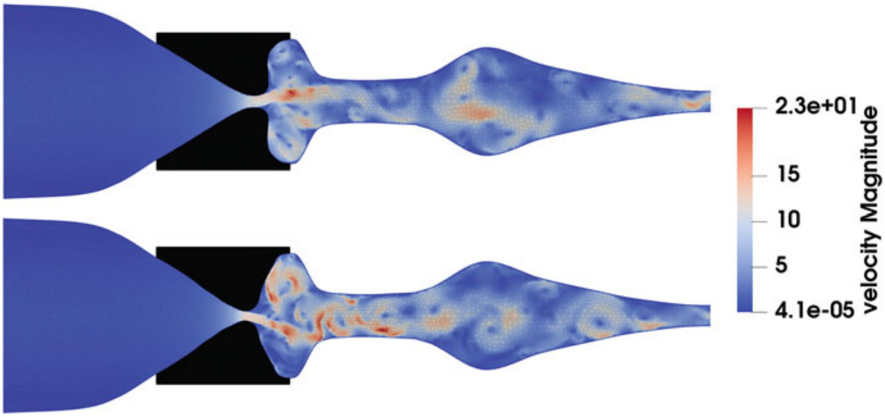


Fig. 3 Velocity field in the glottal region at two time instants of the vocal folds self-oscillation

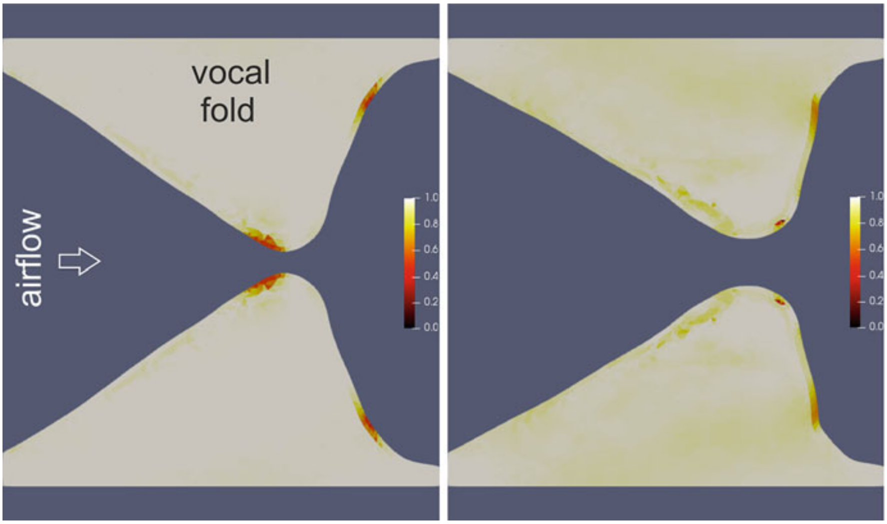


Fig. 4 Deformation of vocal folds in dependence on time and the ratios of the norms at two different time instants

nonlinear part of the strain tensor has no influence to the computation (the linear elasticity model is sufficient), but if $R \approx 0$, then the nonlinear part strongly takes effect and it is necessary to use a nonlinear elasticity model. Figure 4 shows the deformation of the vocal folds at 2 time instants for a maximal and minimal glottal gap during vocal folds oscillations. The ratio $R \approx 1$ is depicted by grey and case $R \approx 0$ by dark red color. It can be seen that nonlinear part of the strain tensor takes effect in elements near to the boundary. Therefore, to correctly capture deformations of the vocal folds, it is necessary to use a nonlinear model of elasticity.

Acknowledgments This research was supported under the grant of the Czech Science Foundation No. 19-04477S. The research of M. Balázsová was also supported by the Charles University, project SVV-2017-260455.

References

1. P. G. Ciarlet, Mathematical Elasticity, Volume I, Three-Dimensional Elasticity, Volume 20 of Studies in Mathematics and its Applications, Elsevier Science Publishers B.V., Amsterdam (1988)
2. V. Dolejší, M. Feistauer, Discontinuous Galerkin method – Analysis and applications to compressible flow, Springer, Cham (2015)

Efficient Solvers for a Stabilized Three-Field Mixed Formulation of Poroelasticity

Massimiliano Ferronato, Matteo Frigo, Nicola Castelletto, and Joshua A. White

Abstract We focus on a three-field (displacement-velocity-pressure) stabilized mixed method for poroelasticity based on piecewise trilinear (Q_1), lowest order Raviart-Thomas (RT_0), and piecewise constant (P_0) approximations for displacement, Darcy's velocity and fluid pore pressure, respectively. Since the selected discrete spaces do not intrinsically satisfy the inf-sup condition in the undrained/incompressible limit, we propose a stabilization strategy based on local pressure jumps. Then, we focus on the efficient solution of the stabilized formulation by a block preconditioned Krylov method. Robustness and efficiency of the proposed approach are demonstrated in two sets of numerical experiments.

1 Introduction

The strong form of the poroelasticity initial/boundary value problem on the domain $\Omega \subset \mathbb{R}^d$ ($d = 2, 3$) and in the time interval $I = (t_0, t_{\max})$ can be stated as follows [1]: given appropriate boundary and initial conditions, find the displacement vector $\boldsymbol{u}$, Darcy's velocity $\boldsymbol{q}$, and the pore pressure p such that

$$\operatorname{div}\left(\mathbb{C}_{dr} : \operatorname{sym}(\operatorname{grad} \boldsymbol{u}) - bp\mathbf{1}\right) = \mathbf{0} \qquad \text{on } \Omega \times I, \tag{1a}$$

$$\mu\boldsymbol{\kappa}^{-1} \cdot \boldsymbol{q} + \operatorname{grad} p = \mathbf{0} \qquad \text{on } \Omega \times I, \tag{1b}$$

$$b \operatorname{div} \dot{\boldsymbol{u}} + S_\epsilon \dot{p} + \operatorname{div} \boldsymbol{q} = f \qquad \text{on } \Omega \times I, \tag{1c}$$

M. Ferronato (✉) · M. Frigo
University of Padova, Department ICEA, Padova, Italy
e-mail: massimiliano.ferronato@unipd.it; matteo.frigo.3@phd.unipd.it

N. Castelletto · J. A. White
Atmospheric, Earth and Energy Division, Lawrence Livermore National Laboratory, Livermore, CA, USA
e-mail: castelletto1@llnl.gov; jawhite@llnl.gov

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_41

where $\mathbb{C}_{dr}$ is the rank-four elasticity tensor, b the Biot coefficient, and $\mathbf{1}$ the rank-two identity tensor; μ and $\boldsymbol{\kappa}$ are the fluid viscosity and the rank-two permeability tensor, respectively; S_ϵ is the constrained specific storage coefficient, i.e. the reciprocal of Biot's modulus, and f the volumetric source term. For additional details about the formulation and its well-posedness, see [2]. By using a mixed Q_1-RT_0-P_0 discretization in space and backward Euler time-marching scheme, the discrete solution for each time instant t is obtained by solving the block linear system [3]:

$$\mathbf{A}\mathbf{x} = \mathbf{b} \quad \text{with} \quad \mathbf{A} = \begin{bmatrix} K & 0 & -Q \\ 0 & A & -B \\ Q^T & \Delta t B^T & P \end{bmatrix}, \quad \mathbf{x} = \begin{bmatrix} \mathbf{u} \\ \mathbf{q} \\ \mathbf{p} \end{bmatrix}, \quad \mathbf{b} = \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_q \\ \mathbf{f}_p \end{bmatrix}, \tag{2}$$

where $\mathbf{u}$, $\mathbf{q}$ and $\mathbf{p}$ denote the vectors of displacement, Darcy's velocity, and pressure unknowns, respectively, at the current time, and Δt is the time-step size. The blocks K, A and P are the classical small displacement stiffness, (scaled) velocity mass, and (scaled) pressure mass matrix, respectively, with Q and B the rectangular coupling blocks. K and A are symmetric positive definite (SPD), while P is diagonal with non-negative entries. For more details about the linear system and the solution approaches, see [3, 4] and references therein.

The Q_1-RT_0-P_0 discretization is a popular and effective choice, but it is unstable in the limit of undrained conditions with incompressible solid and fluid phases [5]. In this paper, we focus on the stabilization of problem (1) by a technique based on the macroelement concept, known as Local Pressure Jump (LPJ) stabilization. This strategy was originally introduced in [6] for stabilizing the Q_1-P_0 discretization of the Stokes equations and recently used in multiphase poromechanics [7]. Two sets of numerical experiments, testing the effectiveness of the stabilization, conclude the paper.

2 Local Pressure Jump Stabilization

In presence of undrained conditions ($\mathbf{q} = \mathbf{0}$), either for low permeability, $\boldsymbol{\kappa} \rightarrow \mathbf{0}$, or small time-step, $\Delta t \rightarrow 0$, and incompressible fluid and solid grain ($S_\epsilon \rightarrow 0$), the system (2) reduces to:

$$\mathbf{B}\mathbf{y} = \mathbf{c} \quad \text{with} \quad \mathbf{B} = \begin{bmatrix} K & -Q \\ Q & 0 \end{bmatrix}, \quad \mathbf{y} = \begin{bmatrix} \mathbf{u} \\ \mathbf{p} \end{bmatrix}, \quad \mathbf{c} = \begin{bmatrix} \mathbf{f}_u \\ \mathbf{f}_p \end{bmatrix}, \tag{3}$$

which resembles the Q_1-P_0 discretization of the Stokes equations where spurious modes can appear in the pressure solution. The LPJ stabilization relies on the macroelement construction, which is an elegant theoretical framework used to prove the inf-sup condition in a saddle-point problem. The grid elements are grouped in macroelements, composed in our case by either four quadrilaterals ($d = 2$) or eight hexahedra ($d = 3$). Let us denote with $\mathcal{M}_h$ the macroelement grid, with $M \in \mathcal{M}_h$, Γ_M^∂ and Γ_M the generic macroelement, its external and internal faces, respectively. Let $\mathbf{B}^*$ be the restriction of $\mathbf{B}$ in (3) to the macroelement M with null-displacement Dirichlet conditions on Γ_M^∂ and S^* the related Schur complement. The basic idea is to ensure the discrete solvability condition, i.e., $\ker S^* = \{\mathbf{1}\}$, on every macroelement. The inf-sup condition holds true for any grid constructed by patching together these stable macroelements. The local Schur complement reads $S^* = Q^{*T} K^{*-1} Q^*$, where Q^* and K^* are the $d \times 2^d$ and $d \times d$ macroelement restrictions of Q and K. Then, the size of $\ker S^*$ is at least equal to $2^d - d$. The idea is to add a new term P_s^* to the mass balance equation such that the new macroelement system with $\mathbf{B}_s^*$ is consistent, i.e., $\ker S_s^* = \{\mathbf{1}\}$, with:

$$\mathbf{B}_s^* = \begin{bmatrix} K^* & -Q^* \\ Q^{*T} & P_s^* \end{bmatrix}, \qquad S_s^* = P_s^* + Q^{*T} K^{*-1} Q^*. \tag{4}$$

The LPJ stabilization matrix P_s^* is given by the area-weighted inter-element pressure jumps [6]:

$$[P_s^*]_{ij} = \sum_{M \in \mathcal{M}_h} \sum_{e \in \Gamma_M} \beta h \int_e [\![\chi_i]\!]_e [\![\chi_j]\!]_e \, \mathrm{d}\Gamma,$$

with $\{\chi_i, \chi_j\}$ ranging over the element-wise constant basis for the pressure space, $[\![\cdot]\!]_e$ the jump across the face e, h the mesh parameter (defined locally), and β a stabilization parameter. From a physical viewpoint, the term P_s^* relaxes the incompressibility constraint by introducing a fictitious flux through the inner faces of each macroelement. Such fluxes counterbalance the spurious pressure modes lying in the kernel of S^* and the mass-conservation is still guaranteed at the macroelement level. Note also that these fluxes are effective only in undrained condition, while they become irrelevant when $\mathbf{q} \neq \mathbf{0}$.

One last point concerns the choice of the stabilization parameter β. An optimal candidate can be guessed from the eigenspectrum of S^*. Following the ideas in [8], we obtain $\beta = (b/2)^2/(2G + \lambda)$ for $d = 2$, where λ and G are the Lamé parameter and the shear modulus, respectively. For $d = 3$, a recent analysis [7], based on minimizing the conditioning number of S^*, suggests setting $\beta = (3b)^2/(32(4G + \lambda))$.

2.1 Linear Solver

By introducing the LPJ stabilization, the global block system (2) becomes:

$$\mathbf{A}_s\mathbf{x} = \mathbf{b} \quad \text{with} \quad \mathbf{A}_s = \begin{bmatrix} K & 0 & -Q \\ 0 & A & -B \\ Q^T & \Delta t B^T & P + P_s \end{bmatrix}, \tag{5}$$

with P_s composed by the $m = \dim(\mathcal{M}_h)$ blocks P_s^*. Recalling that the pattern of the block B provides the face-to-element connections, i.e., $[B]_{ij} \neq 0$ means that the face i belongs to the element j, it is possible to see that the sparsity pattern of P_s^* is the same as $B^{*T} B^*$, being B^* the restriction of B on the macroelement M with no-flow conditions on Γ_M^∂. Hence, the sparsity pattern of P_s is a subset of the sparsity pattern of $B^T B$. This property allows for the straightforward use of the Block Triangular Preconditioner (BTP) developed in [3] for the system (5):

$$\mathbf{M}_s^{-1} = \begin{bmatrix} M_K^{-1} & 0 & 0 \\ 0 & M_A^{-1} & 0 \\ -M_S^{-1} Q^T M_K^{-1} & -\Delta t M_S^{-1} B^T M_A^{-1} & M_S^{-1} \end{bmatrix}, \tag{6}$$

with $S = P_s + P + S_K + \Delta t S_A$, and M_K^{-1}, M_A^{-1} and M_S^{-1} inner preconditioners for K, A and S, respectively. S_K is the diagonal fixed-stress matrix approximating $Q^T K^{-1} Q$, while $S_A = B^T \tilde{A}^{-1} B$ with $\tilde{A}$ a diagonal spectrally equivalent approximation of A. Since P is diagonal, the sparsity pattern of S is still that of $B^T B$ independently of the presence of P_s.

3 Numerical Results

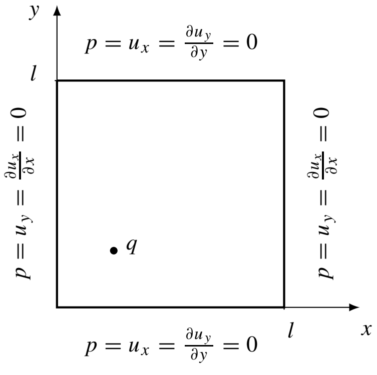
Two test cases are used to investigate the effectiveness of the proposed stabilization and its influence on the linear solver: (1) Barry-Mercer's problem [9], a 2D benchmark of linear poroelasticity particularly suitable for validation purposes; (2) an impermeable cantilever beam, used to analyze the efficiency of the stabilization in both eliminating spurious oscillations and improving the linear solver performance.

3.1 Barry-Mercer's Problem

The problem domain and the physical parameters are provided in Table 1. A periodic source term is set at point q of a square domain with homogeneous boundary conditions. For the details and the analytical solution, the reader can refer to [9].

Figure 1 shows the convergence behavior of the L_2-norm of the error for the pressure field, which is linear as expected. Figure 2 compares the unstabilized and stabilized formulations for a uniform discretization with spacing $h = 1/16$. The unstabilized model presents the classical checkerboard oscillations in the pressure

Table 1 Barry-Mercer's problem: (a) domain sketch and (b) physical parameters

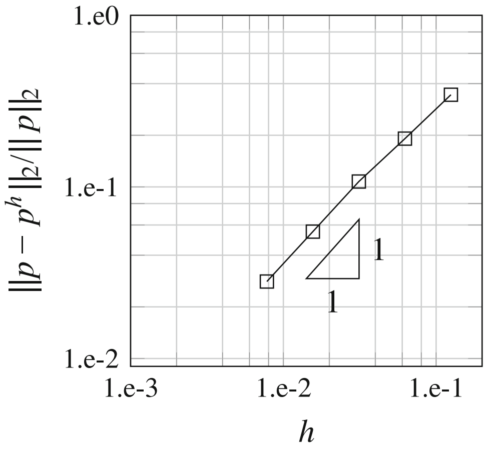

y
$p = u_x = \frac{\partial u_y}{\partial y} = 0$
l
$p = u_y = \frac{\partial u_x}{\partial x} = 0$
• q
$p = u_y = \frac{\partial u_x}{\partial x} = 0$
$p = u_x = \frac{\partial u_y}{\partial y} = 0$
l
x

(a)

Quantity	Value	Unit
Young's modulus (E)	10^5	Pa
Poisson's ratio (ν)	0.1	–
Biot's coefficient (b)	1.0	–
Constrained specific storage (S_ϵ)	0	Pa
Isotropic permeability (κ)	10^{-9}	m^2
Fluid viscosity (μ)	10^{-3}	$\frac{\mathrm{Pa}}{\mathrm{s}}$
Domain size x-y (l)	1.0	m

(b)

Fig. 1 Barry-Mercer's problem: convergence in the L_2 norm of the pressure solution



behavior. The oscillations disappear with the proposed stabilization with no loss of accuracy. The same results can be observed in Fig. 3 along three vertical profiles close to the source point.

3.2 Cantilever Problem

A porous cantilever beam is considered, with the same physical properties as Barry-Mercer's problem (Table 1). The domain is the unit square or cube for the 2-D and 3-D case, respectively. No-flow boundary conditions along all sides are imposed, with the displacements fixed along the left edge and a uniform load applied at the top. Figure 4 shows the pressure solution obtained with a grid spacing $h = 1/10$. In the unstabilized formulation, checkerboard oscillations arise close to the left

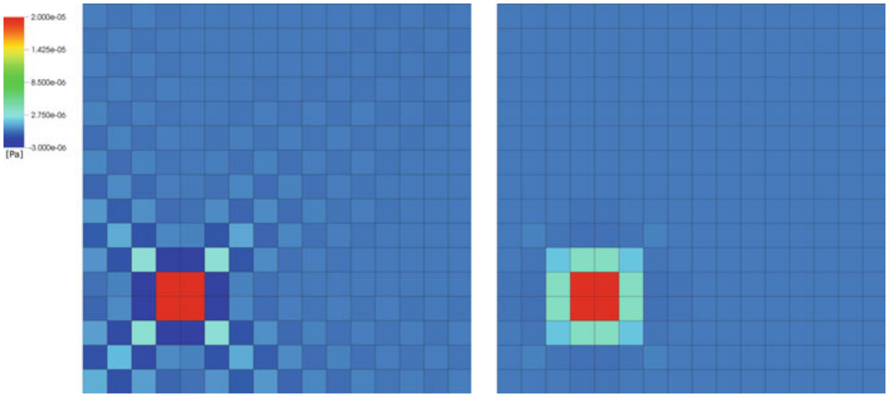


Fig. 2 Barry-Mercer's problem: pressure solution for the unstabilized and stabilized formulations

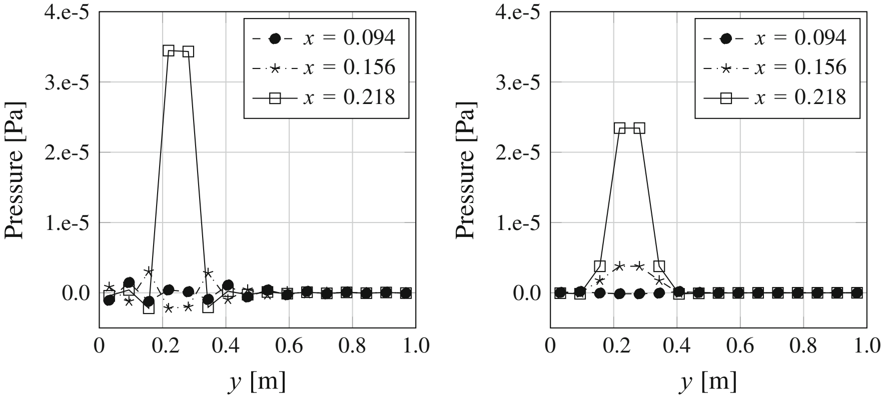


Fig. 3 Barry-Mercer's problem ($h = 1/16$): pressure along the y-direction for the unstabilized (left) and stabilized (right) formulations

constrained edge. As in the previous test case, the proposed stabilization eliminates the spurious pressure modes. This behavior can be better observed along the three vertical profiles provided in Fig. 5.

Finally, we analyze the effects of the stabilization procedure on the linear solver. In order to emphasize the role of the approximations introduced in the Schur complement S, the inner preconditioners M_K, M_A and M_S are applied via a nested direct solver, using the Separate Displacement Component for M_K. Table 2 provides the iteration count for different time-step and grid sizes. For $\Delta t = 0.1\,\text{s}$, the two formulations give essentially the same outcome. Indeed, when the conditions are far from the incompressible/undrained limit the effect of the stabilization vanishes as to both the solution accuracy and the solver performance. On the other hand, with a small time-step size, e.g., $\Delta t = 0.00001\,\text{s}$, the preconditioned

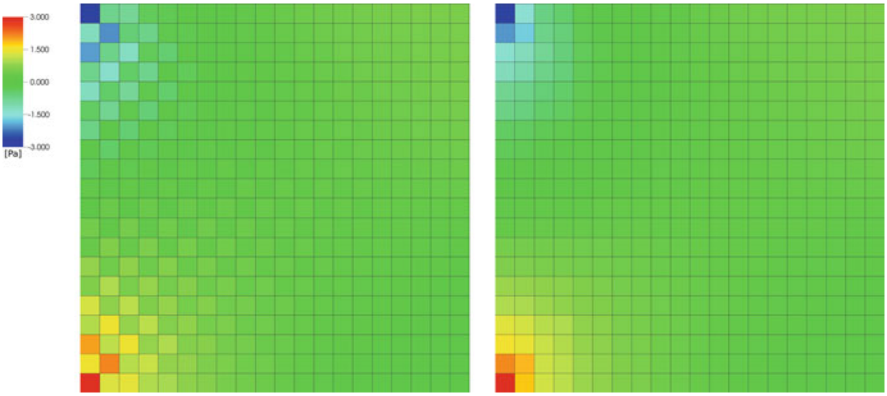


Fig. 4 2D Cantilever beam: pressure solution for the unstabilized and stabilized formulations

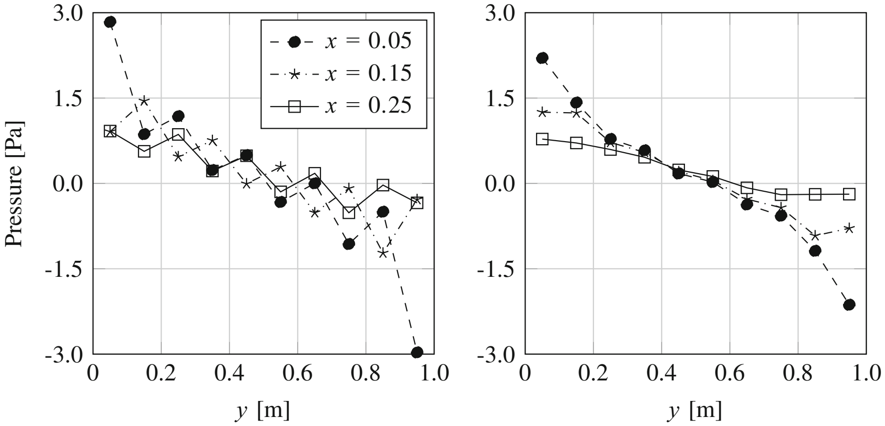


Fig. 5 2D Cantilever beam: pressure solution along the y-direction for the unstabilized (left) and stabilized (right) formulations

Table 2 3D Cantilever beam: iteration count for (a) $\Delta t = 10^{-1}$ s and (b) $\Delta t = 10^{-5}$ s

$1/h$	# cells	No stab.	Stab.
10	1000	47	47
20	8000	52	52
40	64,000	55	55

(a)

$1/h$	# cells	No stab.	Stab.
10	1000	116	49
20	8000	267	57
40	64,000	231	63

(b)

Krylov method behavior can significantly differ between the two formulations. An important degradation in the linear solver performance is observed when using the unstable formulation, as a consequence of the presence of near-singular modes. In the stabilized formulation, such an issue is completely removed and the iteration counts prove also quite stable with the grid size h.

4 Conclusion

In this work, we have introduced a stabilized formulation for three-field Q_1-RT_0-P_0 coupled poromechanics. This stabilization is obtained from the LPJ technique originally advanced for Stokes' problems. The LPJ stabilization method turns out to be effective not only in fulfilling the inf-sup condition on the approximation spaces, but also in improving the linear solver performance. In particular, efficient algorithms already developed for the classical three-field formulation, such as the BTP approach [3], can be used straightforwardly, with just a slight and inexpensive modification in the approximate Schur complement computation. The effectiveness of the proposed stabilization has been investigated in two test cases, showing its capability of eliminating spurious pressure oscillations in undrained conditions, preserving the solution accuracy and convergence, and accelerating the solver convergence.

Acknowledgments Funding was provided by TOTAL S.A. through the FC-MAELSTROM project. Portions of this work were performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07-NA27344.

References

1. Biot, M. A.: General theory of three-dimensional consolidation. J. Appl. Phys. (1941) doi: https://doi.org/10.1063/1.1712886
2. Lipnikov, K.: Numerical Methods for the Biot Model in Poroelasticity. PhD thesis, University of Houston (2002)
3. Castelletto, N., White, J. A. and Ferronato, M.: Scalable algorithms for three-field mixed finite element coupled poromechanics. J. Comput. Phys. (2016) doi: https://doi.org/10.1016/j.jcp.2016.09.063
4. Frigo, M., Castelletto, N. and Ferronato, M.: A Relaxed Physical Factorization Preconditioner for Mixed Finite Element Coupled Poromechanics. SIAM J. on Sci. Comp. (2019) doi: https://doi.org/10.1137/18M120645X
5. Rodrigo, C., Hu, X., Ohm, P., Adler, J.H., Gaspar, F.J. and Zikatanov, L.T.: New stabilized discretizations for poroelasticity and the Stokes' equations. Comp. Methods Appl. Mech. Engrg. (2018) doi: https://doi.org/10.1016/j.cma.2018.07.003
6. Silvester, D. and Kechkar, N.: Stabilised bilinear-constant velocity-pressure finite elements for the conjugate gradient solution of the Stokes problem. Comput. Methods Appl. Mech. Engrg. (1990) doi: https://doi.org/10.1016/0045-7825(90)90095-4

7. Camargo, J. T., White, J. A. and Borja, R. I.: A macroelement stabilization for multiphase poromechanics. Comput. Geosci. (2020) doi: https://doi.org/10.1007/s10596-020-09964-3
8. Silvester, D., Elman, H. and Wathen, A.: Finite Elements and Fast Iterative Solvers: With Applications in Incompressible Fluid Dynamics. Oxford University Press (2005)
9. Barry, S. I. and Mercer, G. N.: Exact Solutions for Two-Dimensional Time-Dependent Flow and Deformation Within a Poroelastic Medium. J. Appl. Mech. (1999) doi: https://doi.org/10.1115/1.2791080

Time-Dependent Two-Dimensional Fourth-Order Problems: Optimal Convergence

J.-P. Croisille and D. Fishelov

Abstract Here we present a new approach for the analysis of high-order compact schemes for the clamped plate problem. A similar model is the Navier-Stokes equation in streamfunction formulation. In our book "Navier-Stokes Equations in Planar Domains", Imperial College Press, 2013, we have suggested fourth-order compact schemes for the Navier-Stokes equations. The same type of schemes may be applied to the clamped plate problem. For these methods the truncation error is only of first-order at near-boundary points, but is of fourth order at interior points. It is proven that the rate of convergence is actually four, thus the error tends to zero as $O(h^4)$.

1 Introduction

The 2D incompressible Navier-Stokes (NS) equations $\partial_t(\Delta\psi) + (\nabla^{\perp}\psi)\cdot\nabla(\Delta\psi) = \nu\Delta^2\psi$, where ψ is the streamfunction, play an important role in various areas of physics. In [2] we suggested fourth-order compact schemes for the NS problem, including important foundations for their error analysis.

In Sect. 2 we analyze the error for the two-dimensional problem $\partial_t u + \Delta^2 u = f$- the time-dependent clamped plate problem. This is related to the time dependent Navier-Stokes equations since both equations include the biharmonic operator. We prove that even though the truncation error is only $O(h)$ at near boundary points, the scheme is fourth-order accurate and the error is $O(h^4)$, where h is the mesh size. Similar situations occur also for the high-order finite difference schemes suggested in [1] and [6].

J.-P. Croisille
Department of Mathematics, University de Lorraine, Metz, France
e-mail: jean-pierre.croisille@univ-lorraine.fr

D. Fishelov (✉)
Department of Mathematics, Afeka Tel-Aviv Academic College of Engineering, Tel-Aviv, Israel
e-mail: daliaf@afeka.ac.il

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_42

2 The Equation $\partial_t u + \Delta^2 u = f$

Consider the fourth-order partial differential problem

$$\begin{aligned}
&\partial_t u + \Delta^2 u = f(x, y, t), \quad (x, y) \in (0, 1) \times (0, 1), \qquad t > 0,\\
&u(0, y, t) = u(1, y, t) = 0, \quad u_x(0, y, t) = u_x(1, y, t) = 0, \quad 0 \le y \le 1,\\
&u(x, 0, t) = u(x, 1, t) = 0, \quad u_y(x, 0, t) = u_y(x, 1, t) = 0, \quad 0 \le x \le 1,\\
&u(x, y, 0) = g(x, y), \quad (x, y) \in [0, 1] \times [0, 1].
\end{aligned} \tag{1}$$

In order to approximate the solution of Eq. (1), we lay out a uniform grid $(x_j, y_k) = \left(\frac{j}{N}, \frac{k}{N}\right),\ j, k = 0, 1, \ldots, N$. Let $\mathfrak{f}(t)$ be the evaluation of f at the grid points. Then, we define a grid function $\mathfrak{v}_{j,k}(t)$, which serves as an approximation of $u(x_j, y_k, t)$ for $j, k = 0, \ldots, N$, to be the solution of

$$\begin{aligned}
&\partial_t \mathfrak{v}_{j,k}(t) + \tilde{\Delta}_h^2 \mathfrak{v}_{j,k}(t) = \mathfrak{f}_{j,k}(t), \quad j, k = 1, \ldots, N-1,\\
&\mathfrak{v}_{0,k}(t) = \mathfrak{v}_{N,k}(t) = 0, \quad (\mathfrak{v}_x)_{0,k}(t) = (\mathfrak{v}_x)_{N,k}(t) = 0, \quad k = 0, \ldots, N,\\
&\mathfrak{v}_{j,0}(t) = \mathfrak{v}_{j,N}(t) = 0, \quad (\mathfrak{v}_y)_{j,0}(t) = (\mathfrak{v}_y)_{j,N}(t) = 0, \quad j = 0, \ldots, N,\\
&\mathfrak{v}_{j,k}(0) = g_{j,k}, \quad j, k = 0, \ldots, N.
\end{aligned} \tag{2}$$

Here

$$\tilde{\Delta}_h^2 = \delta_x^4 + \delta_y^4 + 2[\delta_x^2 \delta_y^2 - \frac{h^2}{12}(\delta_x^4 \delta_y^2 + \delta_y^4 \delta_x^2)], \tag{3}$$

where, for $j,\ k = 1, \ldots, N-1$,

$$\begin{aligned}
(\delta_x^4 \mathfrak{v})_{j,k} &= \tfrac{12}{h^2}(\delta_x \mathfrak{v}_x - \delta_x^2 \mathfrak{v})_{j,k},\\
(\delta_y^4 \mathfrak{v})_{j,k} &= \tfrac{12}{h^2}(\delta_y \mathfrak{v}_y - \delta_y^2 \mathfrak{v})_{j,k},
\end{aligned} \tag{4}$$

$$\begin{aligned}
(\sigma_x \mathfrak{v}_x)_{j,k} &= (\delta_x \mathfrak{v})_{j,k},\\
(\sigma_y \mathfrak{v}_y)_{j,k} &= (\delta_y \mathfrak{v})_{j,k},
\end{aligned} \tag{5}$$

$$\begin{aligned}
(\sigma_x \mathfrak{w})_{j,k} &= \tfrac{1}{6}(\mathfrak{w}_{j-1,k} + 4\mathfrak{w}_{j,k} + \mathfrak{w}_{j+1,k}),\\
(\sigma_y \mathfrak{w})_{j,k} &= \tfrac{1}{6}(\mathfrak{w}_{j,k-1} + 4\mathfrak{w}_{j,k} + \mathfrak{w}_{j,k+1}).
\end{aligned} \tag{6}$$

Thus, the approximated solution satisfies

$$\partial_t \mathfrak{v}_{j,k}(t) + \tilde{\Delta}_h^2 \mathfrak{v}_{j,k}(t) = \mathfrak{f}_{j,k}(t), \quad j, k = 1, \ldots, N-1. \tag{7}$$

Let $u^*(t)$ be the evaluation of u on the grid points at time t. Then,

$$\partial_t u^*_{j,k}(t) + \tilde{\Delta}_h^2 u^*_{j,k}(t) = \mathfrak{f}_{j,k}(t) - \mathfrak{r}_{j,k}(t) \quad j, k = 1, \ldots, N-1, \tag{8}$$

where $\mathfrak{r}(t)$ is the truncation error. By Taylor expansions, if u has continuous derivatives up to order 8, the components of the truncation error $\mathfrak{r}$ for all t may be written as (see [2], Proposition 10.8)

$$\begin{aligned} &\mathfrak{r}_{j,k} = O(h^4) \quad j,k = 2,\dots,N-2, \\ &\mathfrak{r}_{1,k} = O(h), \quad \mathfrak{r}_{N-1,k} = O(h), \quad k = 1,\dots,N \\ &\mathfrak{r}_{j,1} = O(h), \quad \mathfrak{r}_{j,N-1} = O(h), \quad j = 1,\dots,N. \end{aligned} \tag{9}$$

Define the error $\mathfrak{e}(t) = \mathfrak{v}(t) - u^*(t)$. Then, by subtracting (8) from (7), we have

$$\partial_t \mathfrak{e}(t) + \tilde{\Delta}_h^2 \mathfrak{e}(t) = \mathfrak{r}(t). \tag{10}$$

The following Optimal Convergence Theorem holds (see [2, 4, 5]).

Theorem 1 (One-Dimensional Case) *Suppose that the vector* $\tau \in \mathbb{R}^{(N-1)}$, *containing the truncation errors, satisfies*

$$\tau_1 = O(h) \quad \tau_j = O(h^4),\ j = 2,\dots,N-2, \quad \tau_{N-1} = O(h). \tag{11}$$

Then, the operator δ_x^{-4}, *operating on* τ *satisfy*

$$\max_{1\le j\le N-1} |(\delta_x^{-4}\tau)_j| \le Ch^4, \quad \textit{where } C \textit{ does not depend on } N. \tag{12}$$

We relate the grid function $\mathfrak{v}_{j,k}$, $j,k = 1,\dots,N-1$ with the column vector

$$V = \left[\mathfrak{v}_{1,1},\dots,\mathfrak{v}_{N-1,1},\mathfrak{v}_{1,2},\dots\mathfrak{v}_{N-1,2},\dots,\mathfrak{v}_{1,N-1},\dots,\mathfrak{v}_{N-1,N-1}\right]^T \in \mathbb{R}^{(N-1)^2}. \tag{13}$$

The bottom ordering of vector $V \in \mathbb{R}^{(N-1)^2}$ is obtained by letting the index j vary first while keeping k fixed, then vary the index k (see [3]). Then, we relate the two-dimensional finite difference operators with matrix operators of size $(N-1)\times(N-1)$ for $N \ge 2$, acting on a vector V. Most of those operators are obtained as Kronecker products of $(N-1)\times(N-1)$ matrices. Recall that the Kronecker product of the matrices $G \in \mathbb{M}_{m,n}$ and $H \in \mathbb{M}_{p,q}$ is the matrix $G \otimes H \in \mathbb{M}_{mp,nq}$ defined by

$$G \otimes H = \begin{bmatrix} g_{1,1}H & g_{1,2}H & \dots & g_{1,n}H \\ \dots & & & \\ \dots & & & \\ g_{m,1}H & g_{m,2}H & \dots & g_{m,n}H \end{bmatrix}. \tag{14}$$

Let the matrix B represent the biharmonic discrete operator in one dimension and the matrix D represent $-\delta_x^2$ (or $-\delta_y^2$) in one dimension. Then, $I \otimes B$ and $B \otimes I$

represent the biharmonic operators δ_x^4 and δ_y^4, respectively. Similarly, $I \otimes D$ and $D \otimes I$ represents the operator $-\delta_x^2$ and $-\delta_y^2$, respectively. In addition,

$$R(t) = \left[\mathfrak{r}_{1,1}, \dots, \mathfrak{r}_{N-1,1}, \mathfrak{r}_{1,2}, \dots, \mathfrak{r}_{N-1,2}, \dots, \mathfrak{r}_{1,N-1}, \dots, \mathfrak{r}_{N-1,N-1}\right]^T \in \mathbb{R}^{(N-1)^2} \tag{15}$$

is related to the truncation error. Therefore, inequality (12) may be written in vector notation as follows.

Corollary 1 *Let $R(t) = R^{(1)}(t) + R^{(2)}(t) \in \mathbb{R}^{(N-1)^2}$, where*

$$R^{(1)}(t) = \left[\mathfrak{r}_{1,1}, 0, \dots, 0, \mathfrak{r}_{N-1,1}, \mathfrak{r}_{1,2}, \dots, \mathfrak{r}_{N-1,2}, \dots, \mathfrak{r}_{1,N-1}, 0, \dots, 0, \mathfrak{r}_{N-1,N-1}\right]^T, \tag{16}$$

$$R^{(2)}(t) = \left[0, \mathfrak{r}_{2,1}, .., \mathfrak{r}_{N-2,1}, 0, 0, \dots, 0, \dots, 0, \dots, 0, 0, \mathfrak{r}_{2,N-1} \dots, \mathfrak{r}_{N-2,N-1}, 0\right]^T. \tag{17}$$

Then,

$$\max_{1 \le m \le (N-1)^2} |((I \otimes B^{-1}) R^{(1)}(t))_m| \le Ch^4, \quad 0 < t < T, \tag{18}$$

where $I \otimes B^{-1}$ represents the operator δ_x^{-4}, and

$$\max_{1 \le m \le (N-1)^2} |((B^{-1} \otimes I) R^{(2)}(t))_m| \le Ch^4, \quad 0 < t < T, \tag{19}$$

where $(B^{-1} \otimes I)$ represents the operator δ_y^{-4}.

Proof We may write (16) and (17) as $R^{(1)} = \left[R_1^{(1)}; \dots; R_{N-1}^{(1)}\right]$ and $R^{(2)} = \left[R_1^{(2)}; \dots; R_{N-1}^{(2)}\right]$, respectively, where

$$\begin{aligned}
&R_1^{(1)} = [\mathfrak{r}_{1,1}, 0, \dots, 0, \mathfrak{r}_{N-1,1}]^T, && R_1^{(2)} = [0, \mathfrak{r}_{2,1}, \dots, \mathfrak{r}_{N-2,1}, 0]^T, \\
&R_j^{(1)} = [\mathfrak{r}_{1,j}, \dots, \mathfrak{r}_{N-1,j}]^T, \ j = 2, \dots, N-2, && R_j^{(2)} = [0, \dots, 0]^T, \ j = 2, \dots, N-2, \\
&R_{N-1}^{(1)} = [\mathfrak{r}_{1,N-1}, 0, \dots, 0, \mathfrak{r}_{N-1,N-1}]^T. && R_{N-1}^{(2)} = [0, \mathfrak{r}_{2,N-1}, \dots, \mathfrak{r}_{N-2,N-1}, 0]^T.
\end{aligned} \tag{20}$$

Using the definition of a Kronecker product, we have

$$I \otimes B = \begin{bmatrix} B & \underline{0} \dots\dots & \underline{0} \\ \underline{0} & B \ \underline{0} \ \dots & \underline{0} \\ \dots & & \\ \underline{0} & \underline{0} \dots \underline{0} & B \end{bmatrix}, \qquad (I \otimes B)^{-1} = \begin{bmatrix} B^{-1} & \underline{0} & \dots\dots & \underline{0} \\ \underline{0} & B^{-1} & \underline{0} \ \dots & \underline{0} \\ \dots & & & \\ \underline{0} & \underline{0} & \dots \ \underline{0} & B^{-1} \end{bmatrix}. \tag{21}$$

Therefore, $(I \otimes B^{-1})R(t) = \left[B^{-1}R_1^{(1)}(t), B^{-1}R_2^{(1)}(t), \ldots, B^{-1}R_{N-2}^{(1)}(t), B^{-1}R_{N-1}^{(1)}(t)\right]^T$. By the optimal convergence theorem

$$\max_{1 \le m \le (N-1)^2} |((I \otimes B^{-1})R^{(1)}(t))_m| \le Ch^4, \quad 0 < t < T. \tag{22}$$

Hence (18) holds. By a similar proof (19) holds. □

Theorem 2 *Suppose the solution $u(x, y, t)$ to the system (1) has derivatives up to order* 8 *with respect to x and y, then the error $\mathfrak{e}(t)$ is bounded by*

$$|\mathfrak{e}(t)|_h \le Ch^4, \quad 0 < t < T, \tag{23}$$

where $|\mathfrak{e}(t)|_h = \sqrt{\sum_{j=1}^{N-1} \sum_{k=1}^{N-1} h^2 |\mathfrak{e}_{j,k}(t)|^2}$ and C depends only on $u_0(x, y)$ and T.

Proof Define $E(t)$ as the vector containing the components of the error at time t

$$E = \left[\mathfrak{e}_{1,1}, \ldots \mathfrak{e}_{N-1,1}, \mathfrak{e}_{1,2}, \ldots \mathfrak{e}_{N-1,2}, \ldots, \mathfrak{e}_{1,N-1}, \ldots \mathfrak{e}_{N-1,N-1}\right]^T \in \mathbb{R}^{(N-1)^2}. \tag{24}$$

The operator $\tilde{\Delta}_h^2$ may be represented by the matrix A of size $(N-1)^2 \times (N-1)^2$ (see [3]), where

$$A = I \otimes B + B \otimes I + 2\left[(I \otimes D)(D \otimes I) + \frac{h^2}{12}(I \otimes D)(B \otimes I) + \frac{h^2}{12}(D \otimes I)(I \otimes B)\right]. \tag{25}$$

Hence, A is a symmetric positive definite matrix. In vector notation Eq. (10) may be written as $\partial_t E(t) + A\ E(t) = R(t)$. Multiplying both sides of the last equation by e^{At}, we have $\partial_t(e^{At}\ E(t)) = e^{At}\ R(t)$. Integrating the last equation for ρ from 0 to t and multiplying by e^{-At}, we have

$$E(t) = \int_0^t e^{-A(t-\rho)} R(\rho) d\rho. \tag{26}$$

Multiplying $R(\rho)$ from the left by AA^{-1} yields

$$E(t) = \int_0^t [e^{-A(t-\rho)}\ A]\ \ [A^{-1}R(\rho)]d\rho = \int_0^t [e^{-A(t-\rho)}\ A]\ \ [A^{-1}(R^{(1)}(\rho) + R^{(2)}(\rho))]d\rho, \tag{27}$$

where $R^{(1)}$ and $R^{(2)}$ are defined in (16) and (17) (see also (20)). We decompose $E(t)$ in the sum $E(t) = E^{(1)}(t) + E^{(2)}(t)$, where

$$E^{(1)} = \int_0^t [e^{-A(t-\rho)}\ A]\ \ [A^{-1}\ R^{(1)}(\rho)]d\rho,\ \ E^{(2)} = \int_0^t [e^{-A(t-\rho)}\ A]\ \ [A^{-1}\ R^{(2)}(\rho)]d\rho. \tag{28}$$

We show that $\|E^{(1)}\|_2 \leq Ch^3$ and $\|E^{(2)}\|_2 \leq Ch^3$. Using (25), then for the term $E^{(1)}$ we decompose A as follows. $A = (I \otimes B)Q_1$, where Q_1 is defined by

$$\begin{aligned} Q_1 = I \otimes I + (I \otimes B)^{-1}(B \otimes I) + 2(I \otimes B)^{-1} \\ \big[(I \otimes D)(D \otimes I) + \frac{h^2}{12}(I \otimes D)(B \otimes I) + \frac{h^2}{12}(D \otimes I)(I \otimes B)\big]. \end{aligned} \tag{29}$$

Using (25), then for the term $E^{(2)}$ we decompose A as follows. $A = (B \otimes I)Q_2$, where Q_2 is defined by

$$\begin{aligned} Q_2 = I \otimes I + (B \otimes I)^{-1}(I \otimes B) + 2(B \otimes I)^{-1} \\ \big[(I \otimes D)(D \otimes I) + \frac{h^2}{12}(I \otimes D)(B \otimes I) + \frac{h^2}{12}(D \otimes I)(I \otimes B)\big]. \end{aligned} \tag{30}$$

Therefore,

$$\begin{aligned} E^{(1)}(t) = \int_0^t [e^{-A(t-\rho)}\ A]\ \ Q_1^{-1}\ [(I \otimes B)^{-1} R^{(1)}(\rho)]d\rho \\ E^{(2)}(t) = \int_0^t [e^{-A(t-\rho)}\ A]\ \ Q_2^{-1}\ [(I \otimes B)^{-1} R^{(2)}(\rho)]d\rho. \end{aligned} \tag{31}$$

First we consider $\|E^{(1)}(t)\|_2$. Expanding on $Q_1^{-1}\ [(I \otimes B)^{-1}R^{(1)}(\rho)]$, we prove that the norm of Q_1^{-1} is bounded from above. Note that (since Q_1^{-1} and Q_1 are not necessarily symmetric matrices),

$$\|Q_1^{-1}\|_2 = \sqrt{\max_{1 \leq k \leq (N-1)^2} |\lambda_k((Q_1^{-1})^T Q_1^{-1})|}. \tag{32}$$

We show that the eigenvalues of $(Q_1^{-1})^T Q_1^{-1}$ are positive and bounded from above by 1. Alternatively, we show that eigenvalues of $Q_1^T Q_1$ are bounded from below by 1. We may decompose Q_1 as a sum $Q_1 = K_1 + K_2$, where

$$\begin{aligned} &K_1 = I \otimes I + (I \otimes B)^{-1}(B \otimes I) \\ &K_2 = 2(I \otimes B)^{-1}\big[(I \otimes D)(D \otimes I) + \tfrac{h^2}{2}(I \otimes D)(B \otimes I) + \tfrac{h^2}{2}(D \otimes I)(I \otimes B)\big]. \end{aligned} \tag{33}$$

Thus, $Q_1^T\ Q_1 = (K_1 + K_2)^T(K_1 + K_2) = K_1^T K_1 + (K_1^T K_2 + K_2^T K_1) + K_2^T K_2$. The matrix K_1 is decomposed as a sum of the two positive definite matrices $K_1 =$

$P_1 + P_2$, where $P_1 = I \otimes I, \quad P_2 = (I \otimes B)^{-1}(B \otimes I)$. Note that P_1 and P_2 are symmetric positive-definite matrices. Therefore, the matrix $K_1^T K_1$ may be written as

$$K_1^T K_1 = I \otimes I + 2P_2 + P_2^2. \tag{34}$$

Thus, $K_1^T K_1$ is a sum of a symmetric positive definite matrix $I \otimes I$ and a symmetric positive definite matrix $2P_2 + P_2^2$. Since all the eigenvalues of $I \otimes I$ are 1, then all the eigenvalues of $K_1^T K_1$ are greater than 1. Now we consider the matrix $K_1^T K_2 + K_2^T K_1$, which is a symmetric matrix. We show that its eigenvalues are positive. First, the matrix K_1 is symmetric positive definite. Next, the matrix K_2 is a product of two symmetric positive definite matrices S and T, where

$$S = 2(I \otimes B)^{-1}, \ T = (I \otimes D)(D \otimes I) + \tfrac{h^2}{2}(I \otimes D)(B \otimes I) + \tfrac{h^2}{2}(D \otimes I)(I \otimes B). \tag{35}$$

Thus,

$$K_2 = ST = ST^{1/2}T^{1/2} = T^{-1/2}T^{1/2}ST^{1/2}T^{1/2} = T^{-1/2}(T^{1/2}ST^{1/2})T^{1/2}. \tag{36}$$

Therefore, K_2 is similar to a positive definite matrix, thus its eigenvalues are positive. Since K_1^T and K_2 are positive definite matrices, then by a similar argument as in (35)–(36), the eigenvalues of $K_1^T K_2$ are positive. Similarly, the eigenvalues of $K_2^T K_1$ are also positive. Therefore, the matrix $K_1^T K_2 + K_2^T K_1$ is symmetric, having positive eigenvalues. Consider now the symmetric matrix $K_2^T K_2$. We have shown that the eigenvalues of K_2 are positive, therefore so are the eigenvalues of $K_2^T K_2$. Hence, all the eigenvalues of $Q_1^T Q_1$ are greater than 1. As a result, all the eigenvalues of $(Q_1^{-1})^T Q_1^{-1}$ are smaller than 1. Hence,

$$\|Q_1^{-1}\|_2 = \sqrt{\max_{1 \le k \le (N-1)^2} |\lambda_k\big((Q_1^{-1})^T Q_1^{-1}\big)|} \le 1. \tag{37}$$

are symmetric positive definite matrices. Similarly, $\|Q_2^{-1}\|_2 \le 1$. We continue with bounding $E^{(1)}(t)$. The matrix $e^{-A(t-\rho)}A$ may be diagonalized by a unitary matrix Z, which is independent of $t - \rho$ containing the normalized eigenvectors of the symmetric matrix A. Thus,

$$e^{-A(t-\rho)}A = Z\ \Lambda(t-\rho)\ Z^T, \tag{38}$$

where $\Lambda(\rho) = \mathrm{diag}(e^{-\lambda_1(t-\rho)}\lambda_1, \dots, e^{-\lambda_{(N-1)^2}(t-\rho)}\lambda_{(N-1)^2})$ and $\lambda_1, \dots, \lambda_{(N-1)^2}$ are the eigenvalues of A. Since Z is independent of $t-\rho$, we obtain from (31) and (38)

$$E^{(1)}(t) = Z \int_0^t \Lambda(t-\rho) Z^T \; Q_1^{-1} \; [(I \otimes B)^{-1} R^{(1)}(\rho)] d\rho. \tag{39}$$

We consider now the component i (for $i = 1, \dots, (N-1)^2$) of the vector $E^{(1)}(t)$.

$$E_i^{(1)}(t) = \textstyle\sum_{k=1}^{(N-1)^2} Z_{ik} \int_0^t \Lambda_{k,k}(t-\rho) \; (Z^T \; Q_1^{-1} \; (I \otimes B)^{-1} \; R^{(1)}(\rho))_k d\rho. \tag{40}$$

Expanding on $(Z^T \; Q_1^{-1} \; (I \otimes B)^{-1} \; R^{(1)}(\rho))_k$, we have

$$\begin{aligned}(Z^T \; Q_1^{-1} \; (I \otimes B)^{-1} \; R^{(1)}(\rho))_k &= \textstyle\sum_{l=1}^{(N-1)^2} (Z^T \; Q_1^{-1})_{kl} \; ((I \otimes B)^{-1} \; R^{(1)}(\rho))_l \\ &= \textstyle\sum_{l=1}^{(N-1)^2} (Z^T \; Q_1^{-1})_{kl} \sum_{m=1}^{(N-1)^2} (I \otimes B)^{-1}_{lm} \; R_m^{(1)}(\rho).\end{aligned} \tag{41}$$

$$E_i^{(1)}(t) = \textstyle\sum_{k=1}^{(N-1)^2} Z_{ik} \sum_{l=1}^{(N-1)^2} (Z^T \; Q_1^{-1})_{kl} \sum_{m=1}^{(N-1)^2} (I \otimes B)^{-1}_{lm} \int_0^t \Lambda_{k,k}(t-\rho) \; R_m^{(1)}(\rho) d\rho. \tag{42}$$

Since $\Lambda_{k,k}(t-\rho) = e^{-\lambda_k(t-\rho)}\lambda_k$ and $e^{-\lambda_k(t-\rho)}\lambda_k \geq 0$, we have (by the extended mean-value theorem for integration)

$$\begin{aligned}E_i^{(1)}(t) &= \textstyle\sum_{k=1}^{(N-1)^2} Z_{ik} \sum_{l=1}^{(N-1)^2} (Z^T \; Q_1^{-1})_{kl} \sum_{m=1}^{(N-1)^2} (I \otimes B)^{-1}_{lm} \big[\int_0^t e^{-\lambda_k(t-\rho)} \; \lambda_k d\rho\big] \; R_m^{(1)}(\rho_{m,k}) \\ &= \textstyle\sum_{k=1}^{(N-1)^2} Z_{ik} \big[1 - e^{-\lambda_k t}\big] \; \sum_{l=1}^{(N-1)^2} (Z^T \; Q_1^{-1})_{kl} \; \sum_{m=1}^{(N-1)^2} (I \otimes B)^{-1}_{lm} \; R_m^{(1)}(\rho_{m,k}),\end{aligned} \tag{43}$$

where $0 \leq \rho_{m,k} \leq t$.

Let $L^{(k)} = [R_1(\rho_{1,k}), R_2(\rho_{2,k}), \dots, R_{(N-1)^2}(\rho_{(N-1)^2,k})]^T$. Using (16), we have

$$L^{(k)} = [O(h), 0, \dots, 0, O(h), O(h), O(h^4), \dots, O(h^4), O(h), \dots, O(h), 0, \dots, 0, O(h)]^T. \tag{44}$$

Define $V^{(k)} = (I \otimes B)^{-1} L^{(k)}$. Then, Eq. (43) may be written as

$$E_i(t) = \textstyle\sum_{k=1}^{(N-1)^2} Z_{ik} \big[1 - e^{-\lambda_k t}\big] \; \sum_{l=1}^{(N-1)^2} (Z^T \; Q_1^{-1})_{kl} \; V_l^{(k)}. \tag{45}$$

By the Corollary 1, Eq. (18), we have

$$|V_l^{(k)}| = |\sum_{m=1}^{(N-1)^2} ((I \otimes B)^{-1})_{lm} L_m^{(k)}| \leq Ch^4, \quad 0 < t < T, \tag{46}$$

where C is independent of N. Define the vector W by $W_l = \max_{k=1,\dots,(N-1)^2} |V_l^{(k)}|$. By Eq. (46) the L_2 norm of the vector W is bounded by

$$\|W\|_2 \leq Ch^3. \tag{47}$$

Define $D_1 = \mathrm{diag}(1 - e^{-\lambda_1 t}, \dots, 1 - e^{-\lambda_{(N-1)^2} t})$. Therefore, Eq. (45) yields

$$\|E^{(1)}(t)\|_2 \leq \|Z\|_2 \|D_1\|_2 \ \|Z^T\|_2 \ \|Q_1^{-1}\|_2 \ \|W\|_2. \tag{48}$$

Since $Z^T = Z^{-1}$ and by Eq. (37), we have $\|Z\|_2 = \|Z^T\|_2 = 1, \quad \|Q_1^{-1}\|_2 \leq 1$. We show now that $\|D_1\|_2 \leq C$. Since the eigenvalues A are positive, we have $\|D_1\|_2 = \max_{1 \leq j \leq (N-1)^2} |1 - e^{-\lambda_j t}| \leq 1$. We conclude from (48) and (47) that $\|E^{(1)}(t)\|_2 \leq Ch^3$. Similarly, $\|E^{(2)}(t)\|_2 \leq Ch^3$. Therefore, for $|\mathfrak{e}(t)|_h = \sqrt{\sum_{j=1}^{N-1} \sum_{k=1}^{N-1} h^2 |\mathfrak{e}_{j,k}|^2}$, we have $|\mathfrak{e}(t)|_h \leq Ch^4, \quad 0 < t < T$, which concludes the proof. □

3 Numerical Results

Consider the equation $u_t + \Delta^2 u = f$ with the exact solution $u = e^{-t}(1 - x^2)^2(1 - y^2)^2$ on $[-1, 1]$, $t > 0$, where $f(x, t)$ is chosen so that u is the solution of the differential equation above (Table 1).

Table 1 Compact scheme for $u_t + \Delta^2 u = f$ with exact solution: $u = e^{-t}(1 - x^2)^2(1 - y^2)^2$ on $[-1, 1]$, $t > 0$. We present $|e|_h$ the error in u, and $|e_x|_h$ the error in u_x in the l_2 norm at $t = 0.25$

Mesh	$N = 8$	Rate	$N = 16$	Rate	$N = 32$	Rate	$N = 64$
$\|e\|_h$	1.0819(−4)	3.91	7.2142(−6)	4.00	4.5152(−7)	4.00	2.8221(−8)
$\|e_x\|_h$	1.8773(−6)	3.97	1.2001(−5)	4.01	7.4422(−7)	4.00	4.6480(−8)

Acknowledgments The authors would like thank Professor Matania Ben-Artzi from the Hebrew University, for his comments and insights concerning the matter of this work.

References

1. S. Abarbanel, A. Ditkowski and B. Gustafsson, "On Error Bounds of Finite Difference Approximations to Partial Differential Equations Temporal Behavior and Rate of Convergence", J. Sci. Comput., **15** (2000), pp. 79–116.
2. M. Ben-Artzi, J.-P. Croisille and D. Fishelov, *"Navier-Stokes Equations in Planar Domains"*, Imperial College Press, 2013.
3. M. Ben-Artzi, J.-P. Croisille and D. Fishelov, "A fast direct solver for the biharmonic problem in a rectangular grid", SIAM Journal on Scientific Computing, **31** (2008), pp. 303–333.
4. M. Ben-Artzi and G. Katriel, "Spline functions, the biharmonic operator and approximate eigenvalues", Numer. Mathematik, **141** (2019), pp. 839–879.
5. D. Fishelov, M. Ben-Artzi and J.-P. Croisille, "Recent advances in the study of a fourth-order compact scheme for the one-dimensional biharmonic equation", J. Sci. Comput., **53** (2012), pp. 55–79.
6. S. Wang and G. Kreiss, "Convergence of Summation-by-Parts Finite Difference Methods for the Wave Equation", J. Sci. Comput., **71** (2017), pp. 219–245.

Accurate Numerical Eigenstates of the Gross-Pitaevskii Equation

Bo Gervang and Christian Bach

Abstract We consider a bosonic gas of N bosons. Hartree-Fock approximation allows for a product wave function of single particle solutions $\Psi(\boldsymbol{x_i})$

$$\Psi(x_1, x_2, \cdots, x_N) = \prod_i^N \Psi(\boldsymbol{x}_i)$$

Using a pseudo-potential to account for the condensate self-interaction, the Hamiltonian is found to be

$$H = \sum_{i=1}^N \left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial \mathbf{x}_i^2} + V(\mathbf{x}_i) \right) + \sum_{i<j} \frac{4\pi \hbar^2 a_s}{m} \delta(\mathbf{x}_i - \mathbf{x}_j),$$

In this setting m is the mass of the particles, a_s is the scattering length of the bosons and $\hbar = \frac{h}{2\pi}$. If all single particle solutions satisfy the governing equation, we arrive at

$$\underbrace{\left(-\frac{\hbar^2}{2m} \frac{\partial^2}{\partial \mathbf{x}^2} + V(\mathbf{x}) + \gamma |\psi(\mathbf{x})|^2 \right)}_{H_{\text{GPE}}[\psi](\mathbf{x})} \psi(\mathbf{x}) = \mu \psi(\mathbf{x}), \tag{1}$$

where μ is the chemical potential. Equation (1) is the non-linear Gross-Pitaevskii equation and $\gamma = \frac{4\pi\hbar^2 a_s}{m}$. We use a spectral element method to discretise (1). To compute the eigenstates $\{\psi(\mathbf{x})\}$ of the nonlinear Hamiltonian H_{GPE}, we use two different methods the first is an iterative eigenstate solver and in the second we use a constrained Newton method.

B. Gervang (✉) · C. Bach
Aarhus University, Department of Engineering, Aarhus C, Denmark

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_43

1 Introduction

The first experimental realisation of a Bose-Einstein condensate (BEC) took place in 1995 and then an increased attention in the experimental as well as in the theoretical physics community has taken place. The non-linear Schrodinger equation can well describe the properties of BECs. Specifically we will use the form of the non-linear Schrodinger equation denoted as the Gross-Pitaevskii equation, PGE. In this description, the BEC is treated as a non-uniform, interacting Bose gas at zero temperature. The term "interacting" refers in the GPE-description to at least two-particle interactions, which are modelled in a mean-field approximation and gives rise to a non-linear term.

In general it is not possible to solve the GPE using analytical methods, however in one dimension it is possible to use the Thomas-Fermi approximation to obtain analytical solutions of solitons [4]. We are interested in numerical solution of the eigenstates of the PGEs.

In the interaction of quantum matter with gravity, excited states are observed. In ultra cold neutrons in a gravitational trap excited states have been investigated in order to test Newton's inverse-square law at small distances [6]. Another physical system available for investigating the quantum-gravity regime for a wide range of parameters is the Bose-Einstein condensates under microgravity conditions [5].

Similar experiments with ultra cold atoms in a Gravito-Optical Surface Trap, GOST, with a small and variable gravitational acceleration can be performed so that the density profile of the quantum states related to various energy levels can be measured with better resolution. The Airy functions were used to solve the eigenvalue problem for the schrodinger equations in a GOST [1].

In order to be able to describe the eigenstates for a BEC, we are solving the eigenvalue problem for the nonlinear Schrodinger equations, which in our case is the Gross-Pitaevskii equation (GPE).

2 The Mathematical Model

To describe the dynamics of a BEC subject to two-particle interactions, given by the non-linear term $g_s|\Psi(\mathbf{x},t)|^2$ and to an external potential V_{ext} we use the time-dependent GPE,

$$i\hbar\partial_t\Psi(\mathbf{x},t) = \Big(-\frac{\hbar}{2m}\Delta + V_{ext}(\mathbf{x},t) + g_s|\Psi(\mathbf{x},t)|^2\Big)\Psi(\mathbf{x},t), \tag{2}$$

where $\Psi(\mathbf{x}), \mathbf{x} = (x,y,z)$ is normalised to the total number of particles $N = \int_\Omega |\Psi(\mathbf{x})|^2 d^3x$. The GPE is valid for dilute condensates obeying the diluteness criterion that the s-wave scattering length a and the average density of the gas $\bar{n}$ must fulfill $\bar{n}|a|^3 \ll 1$. The nonlinearity parameter g_s is determined by the scattering length via $g_s = \frac{4\pi\hbar^2 a}{m}$, where m is the mass of the atom. Furthermore, the scattering length can acquire both signs having magnitudes of some nanometres. We will only consider repulsive two-particle interactions, which implies $g_s > 0$. The function $\Psi(\mathbf{x}, t)$ is the wave function or state function of the condensate.

To obtain a time independent solution we use the ansatz, $\Psi(\mathbf{x}, t) = \Psi(\mathbf{x}) exp(-i\mu t/\hbar)$ from which we obtain

$$\mu\Psi(\mathbf{x}) = \Big(-\frac{\hbar}{2m}\Delta + V_{ext}(\mathbf{x}) + g_s|\Psi(\mathbf{x})|^2\Big)\Psi(\mathbf{x}), \tag{3}$$

where μ is the chemical potential. Usually, the external potential $V_{ext}(\mathbf{x})$ models a trap in order to spatially confine the condensate, but can also account for an external perturbation on the system. We assume that the potential is bounded from below and we take $V_{ext}(\mathbf{x}) > 0$.

The stationary GPE can be derived from an action

$$A[\Psi; \mu] = F[\Psi] - 1/2\mu N[\Psi], \tag{4}$$

where the free energy is given by

$$F[\Psi] = \int_\Omega \Big(\frac{\hbar}{2m}(\nabla\Psi(\mathbf{x}))^2 + 1/2V_{ext}(\mathbf{x})\Psi^2(\mathbf{x}) + \frac{g_s}{4}\Psi^4(\mathbf{x})\Big)d^3x, \tag{5}$$

and we assume that Ψ is real.

From now on we restrict our attention to one-dimensional problems. In the following we scale coordinates and normalise the wave function according to

$$x \longrightarrow Lx, \qquad \Psi \longrightarrow \sqrt{N}\Psi/L^{3/2},$$

where $\Psi(x)$ is normalised to 1 and we obtain the normalised stationary GPE

$$\Big(-\frac{d^2}{dx^2} + \bar{V}_{ext}(x) + \gamma\Psi^2(x)\Big)\Psi(x) = \varepsilon\Psi(x), \tag{6}$$

where the dimensionless parameters are

$$\bar{V}_{ext}(x) = \frac{2mL^2 V_{ext}(x)}{\hbar^2}, \quad \gamma = \frac{2mNg_s}{L\hbar^2}, \quad \varepsilon = \frac{2m\mu L^2}{\hbar^2}.$$

The length scale L is arbitrary and can be chosen of convenience. It is observed that the non-linearity parameter γ depends on L. We also note that we don't restrict ourselves to wave functions that are normalised to one. Instead we are searching for solutions that are not normalised for a given pair of γ, ε. From (7) it is observed that each found solution can be normalised to one by adjusting the non-linearity parameter γ.

2.1 Discretisation

The expansion of the solution in space is chosen to be Lagrange polynomials defined on a Gauss-Lobatto quadrature point grid. Lagrange polynomials in one dimension are given by

$$h_i(x) = \prod_{j=0, j\neq i}^{n} \frac{x - x_j}{x_i - x_j}.$$

The Lagrange polynomials h_i form a complete orthogonal set. Another thing we will exploit is their cardinality $h_i(x_j) = \delta_{ij}$. Using a pseudo spectral discretisation, solutions are on the form

$$\Psi(x) = \sum_{i}^{n} \Psi_i h_i(x)$$

Since the Lagrange polynomials can easily be differentiated we also can obtain the discrete nabla and Laplace operators of the unknown. The discrete algebraic system can now be assembled, see e.g. [2].

3 The Picard Iteration Scheme

If the PGE hadn't contained the non-linear coupling term we could have solved the problem using a standard eigendecomposition method. Unfortunately, the non-linear coupling term prevents us from using a standard method, but by introducing a Picard iteration scheme we can solve the problem in a three step procedure.

First Step
Solve the discrete system of

$$\left(-\frac{d^2}{dx^2} + \bar{V}_{ext}(x) + \gamma(\Psi^2(x))^n\right)\hat{\Psi}(x) = \varepsilon\hat{\Psi}(x), \tag{7}$$

where $(\Psi(x))^n$ is known from the previous iteration.

Second Step
Perform the eigendecomposition.

Third Step
Update Ψ as: $(\Psi)^{n+1} = \Psi^n * (1-\beta) + \hat{\Psi} * \beta$, where β is an under-relaxation factor of the order 0.01.

Iterate the above three steps until convergence has been reached.

4 Results for the Picard Method

In Fig. 1 we plot the probability function ($|\Psi|^2$) over the domain $L \in [-10, 10]$ for the ground state and the first seven excited states. The start guess is shown in blue and the final solution is shown in red.

In Fig. 2 we plot the iteration history for the solution of the eight states. It is observed that it takes approximately 1000 iterations for the ground state and the first excited state. For the remaining excited states it takes more than 3500 iterations. Even though it is possible to obtain the solutions using a Picard iteration method the iteration count is prohibitive large for practical applications. In the next section we will discuss the constrained Newton method, which is a faster method to obtain the solutions.

5 The Constrained Newton Method

Another approach is to use a Newton method that is a gradient method that follow a descent direction until a local minimum of the action is reached. Solutions can be understood as critical points of some action A, see (4). The type of critical point related to a solution is determined by eigenvalues of the Hessian evaluated at the critical point:

- If all eigenvalues are positive then the critical point is a local minimum of A
- If all eigenvalues are negative then we have a local maximum
- If we have a finite number of negative eigenvalues and all other values are positive then we have a horse saddle. In this situation the number of negative eigenvalues is the number of linear independent descent directions at this critical point
- When the Hessian is degenerate at a critical point we have a monkey saddle

In Fig. 3 we show an example of a horse saddle and a monkey saddle.

Finding critical points at certain saddle types depend on educated guesses. In order to have a straight forward method it is necessary to confine the search to a sub-manifold in the underlying function space. Previous we defined the action as

$$A[\Psi; \mu] = F[\Psi] - 1/2\mu N[\Psi]. \tag{8}$$

Using the principle of least action we know that if Ψ is a critical point of the action A then the gradient of A vanishes and hence Ψ is a solution of the GPE. Using the gradient of the action

$$\nabla A[\psi; \mu] = \Big(- \frac{d^2}{dx^2} + (V_{ext} - \mu) + \gamma(\psi)^2 \Big)\psi(x),$$

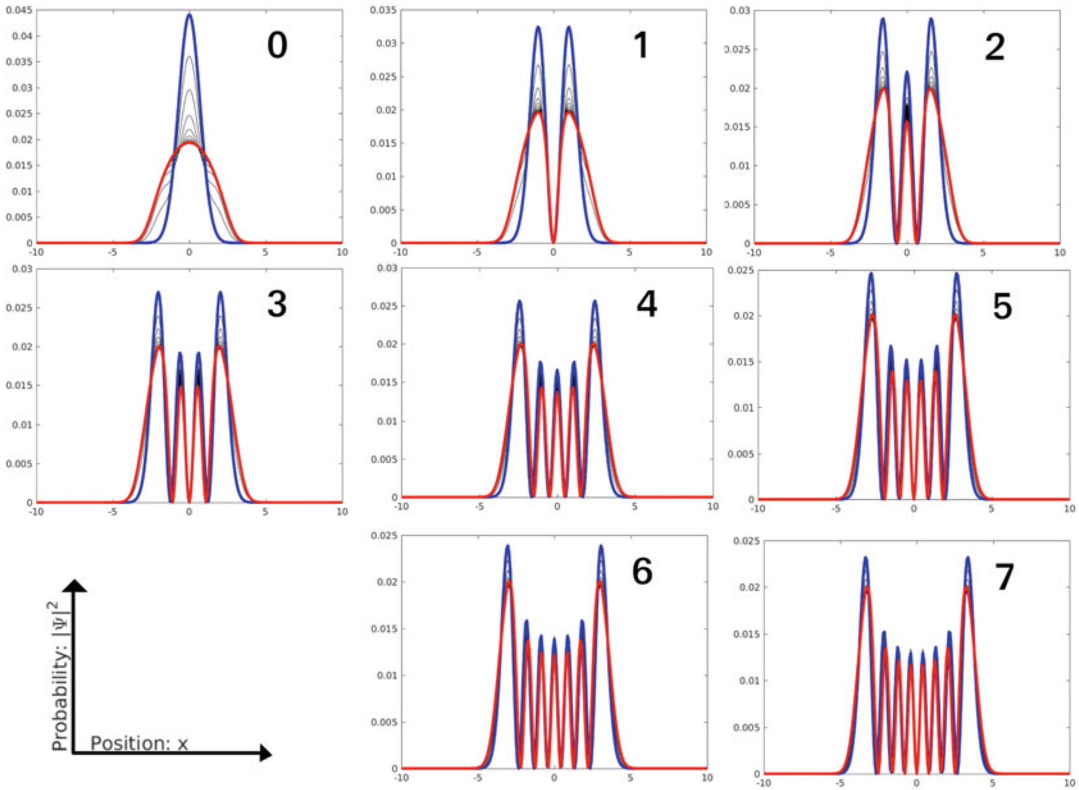


Fig. 1 Evolution of the square of the wave-function for the ground state and the first seven excited states in the domain $L \in [-10, 10]$

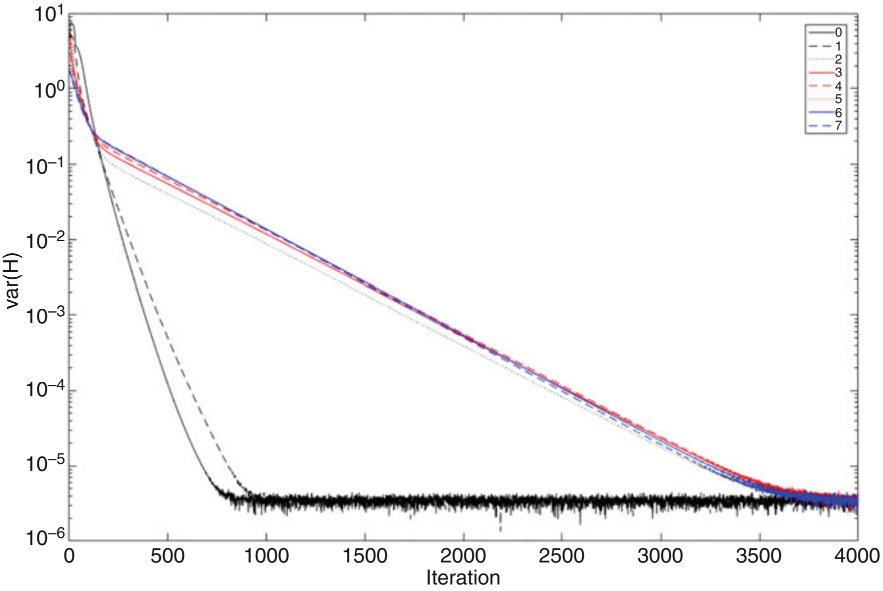


Fig. 2 Variance of the Hamiltonian with respect to the found eigenstates versus iteration step

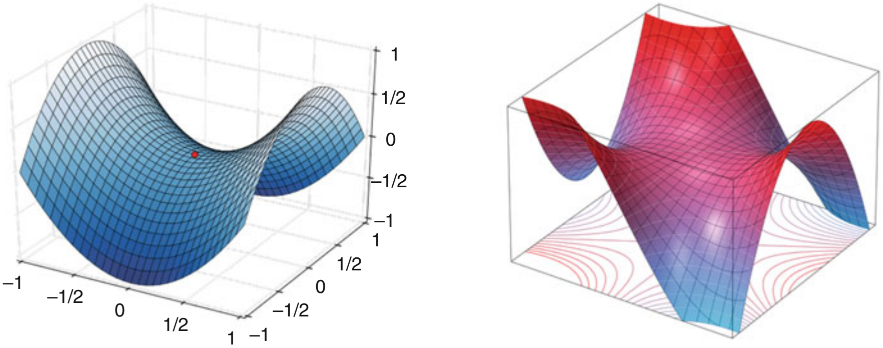


Fig. 3 An example of a horse and a monkey saddle

we can use a discrete Newton method given by Marojevic et al. [3]

$$\psi^{k+1} = \psi^k - \tau \mathcal{H}^{-1} \nabla A[\psi^k; \mu],$$

where $\mathcal{H}^{-1}$ could be the inverse of the Hessian or another preconditioning operator. Unfortunately, in the non-linear case the Newton method is only capable of finding Morse index zero solutions (number of negative eigenvalues). Finding higher Morse index solutions for strong non-linearities can in general not be handled by a standard Newton method. It is, in general, very useful to confine the search for a solutions to a manifold where the critical point lies in a local minimum on this manifold so that classical Newton methods are able to find such solutions. A possible method is [3]

$$\begin{aligned}\langle \nabla A[r_s^k \psi_{n,s} + q_s^k \phi_{n,s}^k; \mu_{n,s}], \psi_{n,s}\rangle = 0,\\ \langle \nabla A[r_s^k \psi_{n,s} + q_s^k \phi_{n,s}^k; \mu_{n,s}], \phi_{n,s}^k\rangle = 0.\end{aligned} \tag{9}$$

We solve for the two unknown r and q. n is the quantum number and k and s are counter variables. $\phi_{n,s}^k$ can be viewed as a correction to the solution $\psi_{n,s}$ for the previous eigenvalue.

Remark 1 We are working with a fixed eigenvalue $\mu_{n,s}$, which is increased by a value $\Delta\mu$ after a solution is found for the current $\mu_{n,s}$, thus $\mu_{n,s+1} = \mu_{n,s} + \Delta\mu$. The non-linearity $\gamma_{n,s}$ is determined as a function of this chosen $\mu_{n,s}$.

Remark 2 The solution found in this way is not normalized to one. Therefore we normalize it and readjust $\gamma_{n,s}$ according to the particle number N.

Remark 3 The search direction in each Newton step is

$$\begin{aligned}d^k = \mathcal{H}^{-1}\nabla A[\phi_{n,s}^k ::; \mu_{n,s}] = \Big(- \frac{d^2}{dx^2} + (V_{ext} - \mu_{n,s}) +\\ 3\gamma_{n,s}(\phi_{n,s}^k)^2\Big)^{-1} \nabla A[\phi_{n,s}^k; \mu_{n,s}].\end{aligned}$$

This assures that together with (9) we don't leave the subspace with the same nodal structure.

6 Results for the Constrained Newton Method

In Fig. 4 we show the results of the constrained Newton method. The wave functions are plotted over the domain $L \in [-10, 10]$.

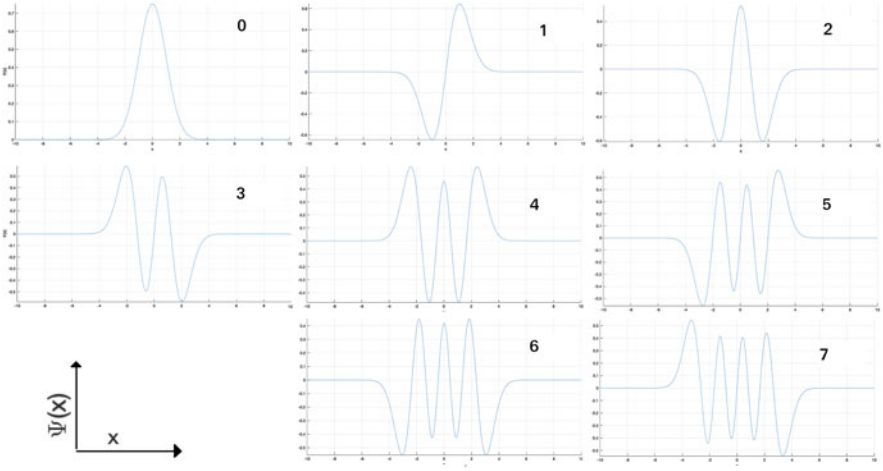


Fig. 4 The first eight wave functions obtained by the constrained Newton method

7 Conclusion

The Gross-Pitaevskii equation has been solve using two different numerical techniques. The first used an eigendecomposition coupled with a Picard iteration scheme and the second used a constrained Newton method. With both methods we could obtain the solution, but the constrained Newton method was more stable and converged much faster.

References

1. Landau, L., Lifshitz, E.: Quantum Mechanics, Non-Relativistic Theory. Pergamon Press, Oxford (1987)
2. Canuto, C., Hussaini, M.Y., Quarteroni, A., Zang, T.A.: Spectral Methods in Fluid Dynamics. Springer, Berlin Heidelberg (1988)
3. Marojevic, Z., Goklu, E., Lammerzahl, C.: Energy eigenfunctions of the 1D Gross-Pitaevskii equation. Comp. Phys. Comm. **184**, 1920–1930 (2013)
4. Nicolin, A., Carretero-Gonzalez, R.: Nonlinear dynamics of Bose-condensed gases by means of a Gaussian variational approach. Phys. A. **184 (24)**, 6032–6044 (2008)
5. van Zoest, T., Gaaloul, N., Singh, Y., Ahlers, H. et al: Bose-Einstein Condensation in Microgravity. Science. **328 (5985)**, 1540–1543 (2010)
6. Abele, H., Baessler, S., Westphal, A.: Quantum states of neutrons in the gravitational field and the limits for non-Newtonian interaction in the range between 1 μm and 10 μm. In: Giulini, D., Kiefer, C., Lammerzahl, C. (Eds.) Lecture Notes in Physics, vol 631, pp 355–366. Springer, Heidelberg (2003).

Basic Machine Learning Approaches for the Acceleration of PDE Simulations and Realization in the FEAT3 Software

Hannes Ruelmann, Markus Geveler, Dirk Ribbrock, Peter Zajac, and Stefan Turek

Abstract In this paper we present a holistic software approach based on the FEAT3 software for solving multidimensional PDEs with the Finite Element Method that is built for a maximum of performance, scalability, maintainability and extensibility. We introduce basic paradigms how modern computational hardware architectures such as GPUs are exploited in a numerically scalable fashion. We show, how the framework is extended to make even the most recent advances on the hardware market accessible to the framework, exemplified by the ubiquitous trend to customize chips for Machine Learning. We can demonstrate that for a numerically challenging model problem, artificial neural networks can be used while preserving a classical simulation solution pipeline through the incorporation of a neural network preconditioner in the linear solver.

1 Introduction

Multidimensional PDE-based simulation is one of the most important yet also most challenging tasks of our time concerning computational resources both hardware- and software-side. Here the Finite Element Method (FEM) is proven to be a superior tool on complex geometries that however needs sophisticated software approaches to be feasible for production in academia and industry: Local performance on each core/CPU or device has to be provided by exploiting the underlying hardware. Numerical scale-up has to be addressed through the design and implementation of advanced parallelisation techniques. Both aspects have to be taken into account in a holistic software framework design that also provides maintainability and extensibility.

H. Ruelmann · M. Geveler (✉) · D. Ribbrock · P. Zajac · S. Turek
TU Dortmund University, Dortmund, Germany
e-mail: hannes.ruelmann@math.tu-dortmund.de; markus.geveler@math.tu-dortmund.de; dirk.ribbrock@math.tu-dortmund.de; peter.zajac@math.tu-dortmund.de; stefan.turek@math.tu-dortmund.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_44

In this paper we offer insight into the third generation of the Finite Element Analysis Toolbox software family (FEAT3), developed at TU Dortmund University. In order to address the aforementioned aspects, we introduce the paradigms of *Hardware-oriented Numerics* and *Unconventional High Performance Computing* (UCHPC) in the context of performance, scalability and maintainability in Sect. 2.

Under the aspect of extensibility, we demonstrate how the framework is catching up with modern hardware trends: Machine Learning (ML) opens a variety of options to support traditional methods of the numerical treatment of solving PDEs particularly for application-oriented CFD simulations with new algorithms on the software level. Any such approach provides access to modern hardware, since chip vendors tailor their designs to ML techniques to satisfy the upcoming and rapidly growing AI-market. In this paper we demonstrate, how artificial neural networks can assist in solving PDEs and how this is implemented in the FEAT3 framework in Sect. 3.

2 FEAT3: Unconventional High Performance Finite Elements

2.1 Trends in Modern Hardware and Green HPC

In the last two decades, it became clear that the continuously increasing single-core speed, which was driven by Moore's law, will stagnate at some point. In consequence, hardware vendors are switching their focus to parallelism, both in the sense of supercomputer clusters as well as various forms of specialized *many-core* hardware accelerators such as general-purpose Graphics Processing Units or Tensor Processing Units, where the latter are custom-designed chips tailored to Machine Learning applications. From a programmer's point of view, these specialized hardware units differ from ordinary CPUs and their built-in vector extensions in the sense that one cannot simply utilize them by enabling a compiler switch. Instead, new specialized algorithms, which efficiently exploit the underlying hardware's strengths, have to be designed and implemented by using third-party libraries. In the context of scientific computing, the concept, where the available hardware determines what algorithms are run on it, can be labeled as Unconventional High Performance Computing.

Another aspect, which is slowly (but steadily) gaining attention, is the continuously increasing energy consumption of supercomputers and the subsequent need for improved energy efficiency. As a consequence, the Green500 list[1] has been launched in 2007 as a means to benchmark the energy efficiency of supercomputers measured in Flops per Watt rather than the total compute power measured in Flops that is used in the famous Top500 list. Again, special accelerator hardware as well as

[1] See https://www.top500.org/green500.

hardware primarily designed for mobile/embedded systems, which are designed to work with a limited battery power supply, play a key role when it comes to achieving high energy efficiency on modern supercomputers.

2.2 Finite Elements and the Need for Speed

The finite element method has proven to be a powerful numerical tool for solving PDEs arising from various scientific fields. Researchers in the academia value the FEM due to its underlying variational formulation, which also forms the backbone of a rigorous theory for the analysis of PDEs, as well as the properties that can be derived from this close relationship between mathematical theory and practical implementability. One notable example is the large set of methods that can be used for error estimation and error control. However, simulations of realistic problems—especially those arising from the industry—result in problem sizes, which are often several orders of magnitude larger than what the academia is typically dealing with and therefore require a different focus in software design and implementation. In consequence, any simulation toolkit that aims to tackle large problems needs to be capable of utilizing modern large-scale parallel hardware beyond the usual small-scale workstation setup, which makes parallel programming, hardware-efficient optimizations and performance engineering indispensable.

2.3 Fast Linear Solvers Based on Geometric Multigrid Methods

One major component of any FEM simulation, which often dominates the overall runtime, is the solution of (non-)linear systems of equations (LSEs). In addition to the usual direct factorization solvers and iterative Krylov subspace methods, the FEM also allows for more specialized solvers that take the underlying discretization into account. The most prominent class of such specialized linear solvers is the class of geometric multigrid methods (GMG), which is one of the few solvers that can solve many LSEs arising from the FEM in linear runtime, thus making it an ideal candidate for large-scale simulations. The GMG is an iterative method which (in its simplest form) solves the LSE by recursively restricting the system onto a coarser mesh, solving the LSE on the coarser mesh, and then projecting the coarse solution back onto the original LSE and post-processing this coarse solution by a *smoother*. The smoother is typically the most costly part of the GMG and its convergence properties are a crucial ingredient for obtaining the h-independent convergence (and thus linear runtime) of the GMG, see e.g. [5, 6].

2.4 FEAT3: FEM+GMG Meets UCHPC

To tackle the above mentioned challenges that come with modern unconventional hardware, we have been implementing the *Finite Element Analysis Toolbox 3* (FEAT3) software package, which is a modular template-based parallel FEM+GMG framework written in C++11. FEAT3 utilizes MPI to implement parallelization paradigms for large-scale supercomputer clusters based on finite element domain decomposition, which support both simple data-parallel algorithms as well as more powerful geometric multigrid solvers based on the concept of *scalable recursive clustering* (ScaRC), see [8, 10].

FEAT3 supports 2D and 3D triangular, quadrilateral, tetrahedral and hexahedral unstructured meshes as well as structured meshes, and is currently being extended to support PDEs on manifolds. A large variety of finite element ansatz spaces have already been implemented, including (but not limited to) the standard conforming Lagrangian elements up to third order, the non-conforming Crouzeix-Raviart and Rannacher-Turek elements as well as a few higher order elements like the Argyris element, see e.g. [11].

FEAT3 can assemble the arising LSEs in various floating point formats, including the standard IEEE-754 single and double precision formats as well as the quadruple precision format offered by the *libquadmath*, which is part of the GCC's standard library set. We also currently experimenting with various third-party libraries, which offer simulated support for low-precision data types like the half precision or the competing *bfloat16* format, which is used by many modern TPUs.

Sparse matrices can not only be assembled as generic unstructured matrices in the well-known *compressed sparse rows* (CSR) format, but also in various special matrix formats including banded or stencil-based matrices. In the case of PDE systems with multiple variables, e.g. the (incompressible) Navier-Stokes equations with velocity and pressure variables, FEAT3 additionally offers various forms of nested *meta*-matrix and -vector class templates. These templates allow for an almost arbitrary mixing of the *array-of-structures* and *structure-of-arrays* data blocking concepts, which play an important role in the design of flexible and efficient data structures.

As mentioned before, one primary challenge in the context of node-level performance engineering is the development of specialized algorithms which are suitable to utilize the underlying hardware efficiently and unfortunately many of these algorithms cannot be hidden behind an opaque back end which serves as a simple hardware abstraction layer. Based on the experiences we have gained with our previous software packages, see [7, 8], as well as several specialized benchmarking projects, see [9], we have realized that it is often necessary to access low-level hardware API functions throughout the whole simulation code directly and this often competes with the desire to provide an easy-to-use abstract high-level interface, which is what most other academic FEM software packages prioritize. FEAT3, on the other hand, has been designed from ground up to support unconventional hardware (including GPUs and TPUs) in numerical research software applications, especially by offering low-level access to all underlying data structures and algorithms, thus making it an ideal testing ground in early development stages. This focus on specialized hardware support via transparent class templates is a major distinguishing feature of FEAT3.

It is important to mention that *all* third-party libraries (including CUDA and MPI) are supported in an *opt-in* fashion, i.e. FEAT3 can be compiled and used in a *naked* build mode (with reduced functionality) without any other dependencies than the C++ standard library. This ensures that FEAT3 can be easily ported to new hardware and operating systems other the usual Linux/Unix ecosystems, even if one or more third-party libraries cannot be compiled on these platforms, which allows us to easily exploit a broad range of hardware from PowerPC Clusters over Windows desktop machines to embedded ARM systems. FEAT3 has already been tested successfully on low-energy systems running on solar-powered battery power supplies, see [3, 4].

The build system for FEAT3 is based on the popular *CMake* system along with a small set of scripts written in the *Python* programming language, which help to enhance the capabilities of CMake to support various build settings via a custom user-controlled build-id system. Our build system also includes a basic test system based on the test driver of CMake, which is executed nightly on our Linux compute servers as well as our university's cluster LiDO3.[2] The correctness of most core classes of the FEAT3 kernel is ensured by a set of *unit-tests*, which test individual classes and their member functions in an isolated testing environment. In addition, the test system also contains a basic set of more complex *regression test* applications, which help to determine whether changes to the kernel classes have changed the behaviour of code that is composed of many interacting classes, which therefore cannot be tested by isolated unit-tests. These nightly tests are compiled with a set of different compilers and different build configurations to continuously ensure the C++11 standard conformity and to detect unexpected changes in code behavior induced by platform changes or compiler bugs. This unit test system is complemented by several specialized benchmarking projects, e.g. the CFD Benchmarking Project [2].

The source code of FEAT3 is released under the GPL3 open source license and is publicly available in the form of a git repository, which can be accessed from the FEATFLOW website.[3]

[2] See https://www.lido.tu-dortmund.de/cms/en/home/index.html.

[3] See http://www.featflow.de/en/software/feat3.html.

3 A Concise Machine Learning Framework to Accelerate Linear Solvers

3.1 Poisson Problem and Anisotropies

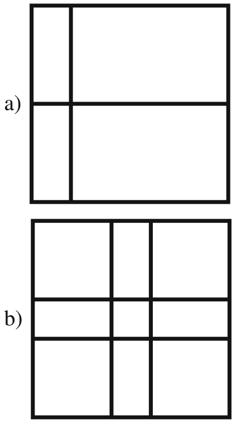
When solving the incompressible Navier-Stokes equation with global Multilevel Pressure Schur Complement techniques, the so called Pressure Poisson problem is dominant regarding calculation time [2]. For the sake of simplicity we therefore choose the Poisson equation to be our model problem, which reads:

Find $u : \Omega \to \mathbb{R}$ such that

$$-\Delta u = f \quad \text{in } \Omega, \qquad u = 0 \quad \text{on } \partial\Omega \tag{1}$$

and discretize it with the Finite Element method.

For the unit square $\Omega = (0, 1)^2$ domain the standard quadrilateral triangulation results in h-independent convergence of the multigrid method with a fixed number of smoothing basis iterations for different smoother. By introducing some anisotropies to the initial mesh, see Fig. 1, the convergence commences to be dependent on the grid size. Merely the ILU method with the Reverse Cuthill-McKee renumbering algorithm maintains the h-independence for the directional anisotropy (Fig. 1a). In case of aspect ratios in both direction (Fig. 1b) renumbering has no further effect and the ILU even with renumbering as well as the other smoothing methods will lead to more multigrid iterations for finer meshes.



lvl	dofs	Jac (0.5)	GS (1.0)	SPAI-1(1.0)	ILU-0 (0.5)	ILU-RCMK
10	2,100,225	109	33	24	26	7
9	525,825	106	32	23	25	7
8	131,841	103	30	22	24	7
7	32,960	98	28	21	23	7
6	8,240	90	25	19	21	6
5	2,060	78	22	17	18	6
4	515	55	16	13	12	6

lvl	dofs	Jac (0.5)	GS (0.7)	SPAI-1(1.0)	ILU-0 (0.7)
9	2,362,369	654	370	140	102
8	591,361	619	350	130	95
7	148,225	562	319	118	85
6	37,249	486	289	103	75
5	9,409	377	218	80	57
4	2,401	258	166	51	34
3	625	175	96	31	20

Fig. 1 Left: coarse grid; right: Number of multigrid V-cycles for different smoothers with eight pre- and post-smoothing steps each. Damping parameter in brackets. Aspect ratio: (**a**) 1:10, (**b**) 1:20

3.2 Approximate Inverses with Neural Network

In this approach we use a neural network prototype trained on function regression to map the FEM system matrix to its corresponding inverse and thus get a beneficial approximation of that inverse which we can use as smoother in multigrid methods or as preconditioner. The structure of the neural network is important to yield strong approximate inverses which are able to smooth the system or lead to converging methods when used e.g. in a Richardson iteration solver. On the other hand it provides a large design space in which we can keep balance between calculation time and accuracy. In [1] we show that fully-connected feed forward multilayer perceptrons are able to extrapolate coefficient matrices which are suitable SPAI-like preconditioners within defect correction methods. In this paper we expand the working system to anisotropic meshes.

To avoid storage problems for larger matrices we use the online-learning method plus only feed the non-zero entries of the system matrix to the neural network. The approximate inverse can be filtered to a sparse matrix thus the assembly of the preconditioner is one pass of the neural network in addition to the application, which is a sparse-matrix-vector-multiplication. With sophisticated matrix formats this performs in parallel and efficiently on modern hardware accelerators. This perfectly couples with FEAT3, which is also used to generate the training data tensor. We randomly shift the inner nodes in order to get a training dataset with the system matrices and associated inverses. This procedure bases on r-adaption techniques we want to use during the CFD simulation, with which the node shift is used to minimize the error.

3.3 Neural Networks for Anisotropic Meshes

To measure the quality of the approximate inverse out of neural networks we use the modified Richardson iteration and compare the number of solver iterations with the conventional damped Jacobi as well as the Gauß-Seidel method. Figure 2 displays the results for different aspect ratios on disturbed meshes. For higher anisotropies, e.g. 1:10 (see Fig. 2b), the common methods collapse and the Jacobi method reaches the maximum iteration number. The low number of iterations even for finer meshes

a)

dim	Jac (0.7)	GS (1.0)	NN
25	422	147	26
121	1955	683	39
529	8622	3017	64

b)

dim	Jac (0.7)	GS (1.0)	NN
25	1101	385	22
121	5036	1762	32
529	10000	7939	37

Fig. 2 Number of iterations for damped Jacobi (Jac), Gauß-Seidel and Richardson iteration with neural networks. Aspect ratio 1:3 (left) and 1:10 (right)

gives strong evidence that neural networks can generate valuable preconditioners. The number of iterations raise just slightly for different refinements and the behavior depends on the training parameter of the neural network.

4 Conclusion and Future Work

We showed that it is possible to combine methods of Machine Learning, which empowers several scientific fields and industry, with the numerical treatment of solving PDEs. Regarding real world CFD simulation, large problem sizes and difficulties like anisotropies arise. The presented Finite Element Analysis Toolbox 3 is specially tailored to solve such problems with respect to performance, hardware efficiency as well as a high accuracy. FEAT3 offers the ability to future-oriented UCHPC along with a easy-to-use framework for academic researchers. On the one hand Machine Learning fits perfectly into this gap of using modern hardware, since chip vendors specially adjust their portfolio to satisfy the fast-growing AI-market. And we demonstrated in a small test scenario, that the designed Machine Learning-based preconditioner with a Richardson iteration as solver maintained low numbers of iteration even for anisotropies with such a high aspect ratio that common solvers fail on the other hand. This is an evidence that Machine Learning techniques can perfectly empower and amplify current numerical PDE solving methods.

One of the most important enhancements in future work will be the extension of the neural network to real applications including large problem sizes.

References

1. Ruelmann, H., Geveler, M., Turek, S.: On the Prospects of Using Machine Learning for the Numerical Simulation of PDEs: Training Neural Networks to Assemble Approximate Inverses, ECCOMAS Newsletter June 2018, pp. 27–32, 2018.
2. Turek, S.: Efficient Solvers for Incompressible Flow Problems: An Algorithmic and Computational Approach, vol. 6. Springer, 1999
3. Geveler, M., Ribbrock, D., Ruelmann, H., Donner, D., Höppke, C., Schneider, D., Tomaschewski, D., Turek, S.: The ICARUS white paper: A scalable, energy–efficient, solar–powered HPC center based on low power GPUs, UcHPC'16 at Euro-Par'16, Grenoble, 2016
4. Geveler, M., Reuter, B., Aizinger, V., Göddeke, D., Turek, S.: Energy efficiency of the simulation of three-dimensional coastal ocean circulation on modern commodity and mobile processors-A case study based on the Haswell and Cortex-A15 microarchitectures, LNCS, ISC'16, Computer Science-Research and Development, 1–10, Workshop on Energy-Aware HPC, Springer, doi: https://doi.org/10.1007/s00450-016-0324-5, 2016
5. M. Geveler, D. Ribbrock, D. Goeddeke, P. Zajac, S. Turek: Efficient Finite Element Geometric Multigrid Solvers for Unstructured Grids on Graphics Processing Units; in P. Ivanyi, B.H.V. Topping, (Editors), "Proceedings of the Second International Conference on Parallel, Distributed, Grid and Cloud Computing for Engineering", Civil-Comp Press, Stirlingshire, UK, Paper 22, doi: https://doi.org/10.4203/ccp.95.22, 2011

6. M. Geveler, D. Ribbrock, D. Goddeke, P. Zajac, S. Turek: Towards a complete FEM-based simulation toolkit on GPUs: Unstructured grid finite element geometric multigrid solvers with strong smoothers based on sparse approximate inverses; Computers and Fluids, Vol 80, 2013, pp. 327–332, doi: https://doi.org/10.1016/j.compfluid.2012.01.025
7. S. Turek, D. Göddeke, C. Becker, S.H.M. Buijssen, H. Wobker: FEAST – realization of hardware-oriented numerics for HPC simulations with finite elements; Concurrency and Computation: Practice and Experience, 2010, Volume 22, Issue 16, doi: https://doi.org/10.1002/cpe.1584
8. D. Göddeke: Fast and Accurate Finite-Element Multigrid Solvers for PDE Simulations on GPU Clusters; PhD thesis, Lehrstuhl für angewandte Mathematik und Numerik, Fakultät für Mathematik, Technische Universität Dortmund, 2010, doi: https://doi.org/10.17877/DE290R-8758
9. D. van Dyk, M. Geveler, S. Mallach, D. Ribbrock, D. Göddeke, C. Gutwenger: HONEI: A collection of libraries for numerical computations targeting multiple processor architectures; Computer Physics Communications, Volume 180, Issue 12, 2009, pp. 2534–2543, doi: https://doi.org/10.1016/j.cpc.2009.04.018
10. S. Turek, C. Becker, S. Kilian: Hardware-oriented numerics and concepts for PDE software; Future Generation Computer Systems 22 (2006) 217–238, doi: https://doi.org/10.1016/j.future.2003.09.007
11. P.G. Ciarlet: The Finite Element Method for Elliptic Problems; North-Holland, 1978, doi: https://doi.org/10.1137/1.9780898719208

Deflated Preconditioned Conjugate Gradients for Nonlinear Diffusion Image Enhancement

Xiujie Shan and Martin van Gijzen

Abstract Nonlinear diffusion equations have been successfully used for image enhancement by reducing the noise in the image while protecting the edges. In discretized form, the denoising requires the solution of a sequence of linear systems. The underlying system matrices stem from a discrete diffusion operator with large jumps in the diffusion coefficients. As a result these matrices can be very ill-conditioned, which leads to slow convergence for iterative methods such as the Conjugate Gradient method. To speed-up the convergence we use deflation and preconditioning. The deflation vectors are defined by a decomposition of the image. The resulting numerical method is easy to implement and matrix-free. We evaluate the performance of the method on a simulated image and on a measured low-field MR image for various types of deflation vectors.

1 Introduction

Many people have benefited from the development of the MRI scanner. However, MRI scanners are expensive and therefore unaffordable for many people in low-income countries. Thus developing a simple and affordable MRI system is urgently needed. The research described in this paper is part of the work to develop a low-field MRI machine for imaging the head of small infants to detect hydrocephalus, a disease that affects many newborns in Africa. A Halbach-array of permanent magnets was designed, optimized, and built [3, 6] to replace the expensive super-conducting magnets that are used in conventional MRI systems. This simpler and inexpensive hardware yields more noisy images, which requires the use of denoising processing for medicine practice.

X. Shan
Delft University of Technology, Harbin Institute of Technology, Harbin, China

M. van Gijzen (✉)
Delft University of Technology, Delft, Netherlands
e-mail: M.B.vanGijzen@tudelft.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_45

The diffusion filtering method interprets pixel intensities as a physical quantity that spreads by a diffusion process in the image [2]. The most simple diffusion model for image denoising is standard heat diffusion. The solution of the model is equivalent to a Gaussian low-pass filter, which is also considered to be the filter in signal processing. The major drawback of this model is that it diffuses edges as well as noise. To overcome this, the constant diffusion coefficient is replaced by a coefficient that depends on the image gradient. This idea was first proposed in [7] by Perona and Malik. The resulting PM-model is given by:

$$\begin{aligned} &\frac{\partial u}{\partial t} = \nabla \cdot (c(\|\nabla u\|)\nabla u) \quad \text{in} \quad \Omega \times (0, T), \\ &u(x, 0) = f \quad \text{in} \quad \Omega, \\ &\frac{\partial u}{\partial \mathbf{n}} = 0 \quad \text{on} \quad \partial\Omega \times (0, T), \end{aligned} \tag{1}$$

where Ω is the image domain, T is the stopping time, u is the pixel value (which is complex for MR images), f is the noisy image and c is a nonnegative monotonically decreasing function with $c(0) = 1$ and $c(+\infty) \to 0$. Because of the ill-posedness of the PM model, Catté et al. [1] have introduced a regularization method that makes the problem well-posed.

In this paper, we consider the following diffusion coefficient which was originally proposed in [7], modified with the technique in [1] to make the problem well-posed:

$$c(\|\nabla u\|) = e^{-(\|G_\sigma * \nabla u\|/K)^2}. \tag{2}$$

In this equation, G_σ is a Gaussian with standard deviation σ and K is a damping parameter.

We discretize Eq. (1) in space using the standard finite different method, see e.g. [1]. We use implicit Euler to discretize in time and take the diffusion coefficient corresponding the previous time step to linearize the equation. In every time step, we have to solve a large and sparse linear system

$$Au = b \tag{3}$$

where A is symmetric and positive definite. For such systems, the conjugate gradient (CG) method [4] is the method of choice. A classical result for the convergence of CG is that after k iterations the error is bounded by

$$\|u - u_k\|_A \leq 2\|u - u_0\|_A \left(\frac{\sqrt{\kappa} - 1}{\sqrt{\kappa} + 1}\right)^k \tag{4}$$

where $\kappa = \lambda_n/\lambda_1$ is the spectral condition number and the A-norm of u is given by $\|u\|_A = (u^T Au)^{1/2}$. The convergence is slow when the condition number κ is very large. One way to improve this is to solve the preconditioned system $M^{-1}Au = M^{-1}b$, where M is a matrix that resembles the matrix A. To further speed up the convergence, one can use a deflation technique to map isolated extreme eigenvalues to zero, effectively removing them from the system. Nicolaides [5] chooses deflation vectors that correspond to subdomains: entries of the deflation vector are one for the nodes in its subdomain and others are zero. In [10], subdomain deflation is applied to Poisson problems with strong contrasts in the coefficient which results in a strong improvement of the convergence. This has motivated us to apply this technique to our problem. To define subdomains, we segment the image. Thresholding, region growing, and small patches are used for segmentation, leading to different ways to define the deflation vectors.

The structure of our paper is as follows. Section 2 describes the deflated and preconditioned CG method and we give three choices of the deflation vectors. The influence of preconditioner for the systems is also investigated by analyzing the eigenvalues in Sect. 2. Section 3 gives numerical experiments for the simulated Shepp-Logan image [9] and for a measured Shepp-Logan image. The comparison of the different deflation vectors is presented in Sect. 3 using a simulated and a measured Shepp-Logan image. We end with conclusions in Sect. 4.

2 PCG Methods with Subdomain Deflation

Deflation has been successfully applied to speed up the convergence of the Preconditioned Conjugate Gradient method (PCG) for a number of problem with strong variations in coefficients [8, 10]. The main idea [5] of this DPCG method is summarized below.

2.1 DPCG

The idea of deflation is to split the solution into two parts, one in the range of the deflation subspace $\mathcal{R}(Z)$ and one in its complement. In order to achieve this, we define the projector P by

$$P = I - AZ(Z^TAZ)^{-1}Z^T, \quad Z \in \mathbb{R}^{n\times m} \tag{5}$$

where $Z = [z_1\ z_2\ \cdots\ z_m]$ is the deflation matrix, which we assume to be of full rank. I is the identity matrix. Since $u = (I - P^T)u + P^Tu$ we have

$$(I - P^T)u = Z(Z^TAZ)^{-1}Z^TAu = ZA_c^{-1}Z^Tb \tag{6}$$

where $A_c = Z^T AZ$. Equation (6) is easy to calculate, we only need to calculate $P^T u$. Using $AP^T = PA$, we can solve the deflated system

$$PA\tilde{u} = Pb \tag{7}$$

for $\tilde{u}$ using the PCG method and then multiplying $\tilde{u}$ by P^T to obtain $P^T u$.

A common choice for the matrix Z, first proposed in [5], is based on a decomposition of the domain Ω. Decomposing domain Ω into m nonoverlapping subdomains Ω_i, $i = 1, 2, \cdots, m$, we choose vectors z_i for $i \in \{1, 2, \ldots, m\}$ such that $z_i = 1$ on $\overline{\Omega_i}$ and $z_i = 0$ on Ω_j, $j \neq i$, $j \in \{1, 2, \ldots, m\}$. With this special choice of Z, the technique for solving the system is referred to subdomain deflation.

We now give the DPCG algorithm for solving the system (3) as follows. Since the pixels values correspond to MR images they are complex valued. For this reason we have to take complex inner products. We therefore use conjugate transpose H instead of normal transpose T in the algorithm. The preconditioning matrix is denoted by M.

DPCG Algorithm

```
$A_c = Z^H AZ$
$P = I - AZ(A_c)^{-1} Z^H$
$r_0 = Pb - PAu_0$
$k = 0$
    while $r_k \neq 0$ do
        Solve $z_k = M^{-1} r_k$
        $k = k + 1$
        if $k = 1$ then
            $p_1 = z_0$
        else
            $\beta_k = r_{k-1}^H z_{k-1}/(r_{k-2}^H z_{k-2})$
            $p_k = z_{k-1} + \beta_k p_{k-1}$
        end if
        $\alpha_k = r_{k-1}^H z_{k-1}/(p_k^H PAp_k)$
        $\tilde{u}_k = \tilde{u}_{k-1} + \alpha_k p_k$
        $r_k = r_{k-1} - \alpha_k PAp_k$
    end while
    $u = Z(A_c)^{-1} Z^H b + P^H \tilde{u}_k$.
```

2.2 Three Different Choices for the Deflation Vectors

We use DPCG to solve Eq. (3). We construct the matrix Z by segmenting the image into small images in three different ways: using thresholding, region growing, and same size patches. For the thresholding and region growing method, we expect that by choosing the interface at edges in the image, i.e., at the location of the jumps in the coefficients, the convergence of the iteration method can be improved. The third technique of same size patches corresponds to the method described in [5]. It does not make use of the image structure, but has the advantage that it is easy to implement. Below we describe the segmentation methods in more detail.

Thresholding

The thresholding method is frequently used for image segmentation. It is a simple and effective segmentation method for images with different intensities [2]. Assuming that the intensity values of image $|f|$ are between 0 and 1, we divide $[0, 1]$ into subintervals I_k. The image is segmented by dividing it into (not necessarily connected) regions with pixel intensities in the same subinterval.

Region Growing

Region growing (RG) segments the image into connected regions with pixel intensities in the same subinterval. To this end, neighbouring pixels are examined, starting from an initial seed point, to determine whether the pixel neighbors should be added to the same region based on a growing condition. The region growing condition we use is as follows: let $|f(i_0, j_0)| \in I_k$ and pixel (i, j) be a neighbour of (i_0, j_0). Then if $|f(i, j)| \in I_k$, the two pixels belong to the same region.

Region Growing

```
Divide the interval $I = [0, 1]$ into parts $I_k$, $k = 1, \dots, s$
**for** $k = 1 : s$ **do**
    **while** *stack is empty* **do**
        1 Search image sequentially, find the first pixel $(i_0, j_0)$ that belongs to
          $I_k$ that does not belong to a segment and set $(i_0, j_0)$ to be seed point.
        2 For all neighbour pixels $(i, j)$ of $(i_0, j_0)$
        **if** $(i, j)$ *is not visited and satisfies the region growing condition* **then**
            Add pixel $(i, j)$ to the stack.
        **end if**
        3 Take a new pixel from the stack and return it to step 2 as $(i_0, j_0)$
    **end while**
**end for**
```

Same Size Patches

The square domain Ω (image) with resolution $m \times n$ is segmented into $s \times r$ subdomains of the same size (patches), where m/s, n/r are integers.

2.3 *Preconditioner*

Subdomain deflation works well if the system matrix contains a few small eigenvalues. In order to achieve this, deflation has to be combined with a suitable preconditioning technique. A simple preconditioner that can achieve this is diagonal scaling. This is illustrated in Fig. 1. The left panel shows the spectrum of the unpreconditioned matrix for the simulated Shepp-Logan image considered in numerical experiments. The right panel shows the spectrum of the preconditioned matrix. Clearly, diagonal scaling maps most eigenvalues to values close to one, with the exception of a few eigenvalues that are mapped to small values.

3 Experimental Results

In this section, we evaluate our method on two images: a simulated Shepp-Logan phantom (128 × 128) and a measured low-field MR image (128 × 128). Comparisons between CG, PCG, DPCG with the three different deflation methods are presented. Our results correspond to one time step of implicit Euler. For the time step, we take $\tau = 0.06$ and the damping parameter in the diffusion coefficient is $K = 3$. For the CG, PCG and DPCG iterations, initial gues is $u^0 = 0$ and as

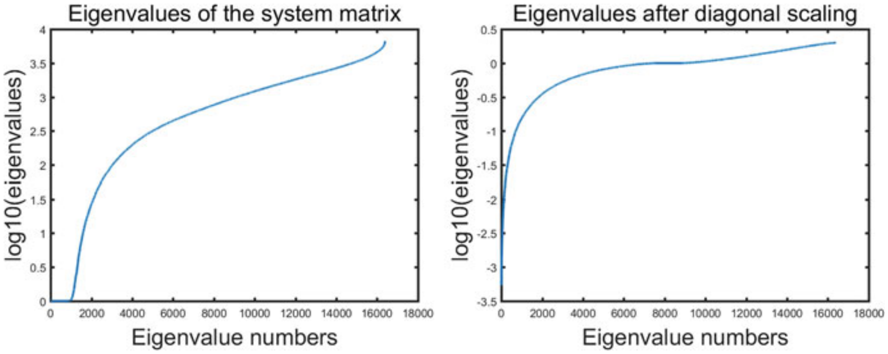


Fig. 1 From left to right: eigenvalues of the system matrix and eigenvalues after diagonal scaling. The eigenvalues are displayed in the log scale

convergence criterion we use $\|r_k\| \leq tol \cdot \|r_0\|$ with $tol = 10^{-5}$. All numerical experiments are carried out using Matlab R2016b on a standard laptop computer.

3.1 Simulated Shepp-Logan Image

Simulated Shepp-Logan image degraded with Gaussian noise with zero mean and variance 0.005 has been tested. The denoising results of the diffusion model are given in Fig. 2. We only show the denoising result of CG because all denoising results based on different numerical algorithms are the same (as they should be). Table 1 shows that CG and PCG need more iterations to converge than the deflated methods. Region growing based DPCG takes more time because of the clustering algorithm. Compared to thresholding segmentation, region growing seems to be more sensitive to noise.

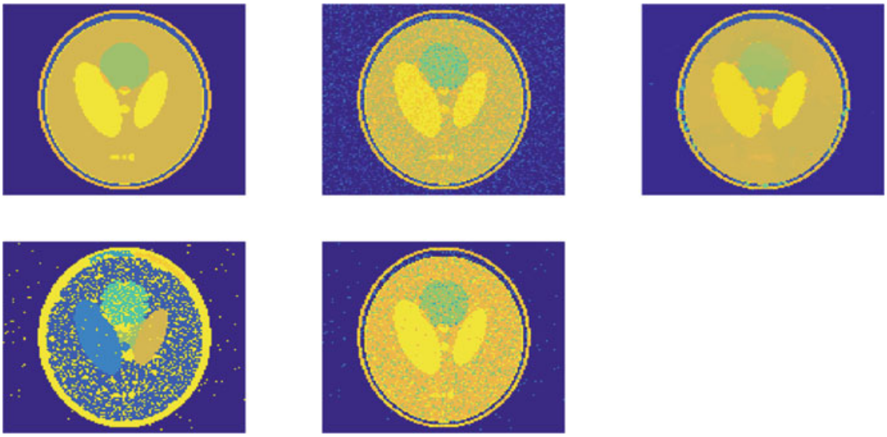


Fig. 2 First row from left to right: original image, noisy image and CG result. Second row from left to right: segmentation (Region growing) and segmentation (Thresholding)

Table 1 Comparisons for simulated Shepp-Logan

Methods	CG	PCG	RG-DPCG	Patches-DPCG(8^2)	Thres-DPCG
Iterations	455	349	230	212	212
Time [a](s)	0.27	0.20	1.21	0.34	0.16

[a] Timings are obtained using Matlab's `cputime` routine. These include the time to segment the image and to construct the deflation matrix

3.2 Measured Shepp-Logan Image

In this section, we test our algorithms on an image of 128×128 pixels acquired with the low-field MRI system described in [6]. Results of this Shepp-Logan image are given in Figs. 3 and 4.

From Fig. 3, we know that the diffusion model achieves a good result for denoising. However, due to the strong noise, segmentation of region growing and thresholding result in many small regions. We observe in Fig. 4 that patches-DPCG achieves the fastest convergence. In the above experiments, we use 4^2 patches to construct the deflation vectors. In Table 2, we investigate how the number of DPCG iterations and solution times depend on the number of patches. The number of iterations is reduced considerably for the three Patches-DPCG methods compared to standard CG and PCG. For this example, 4^2 patches yields the fastest solution time.

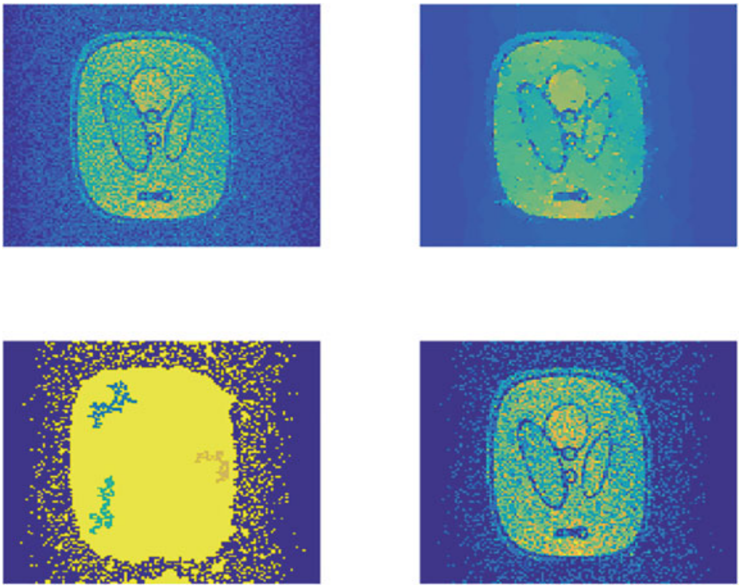


Fig. 3 First row from left to right: original image, DPCG (RG). Second row from left to right: segmentation (RG) and segmentation (Thresholding)

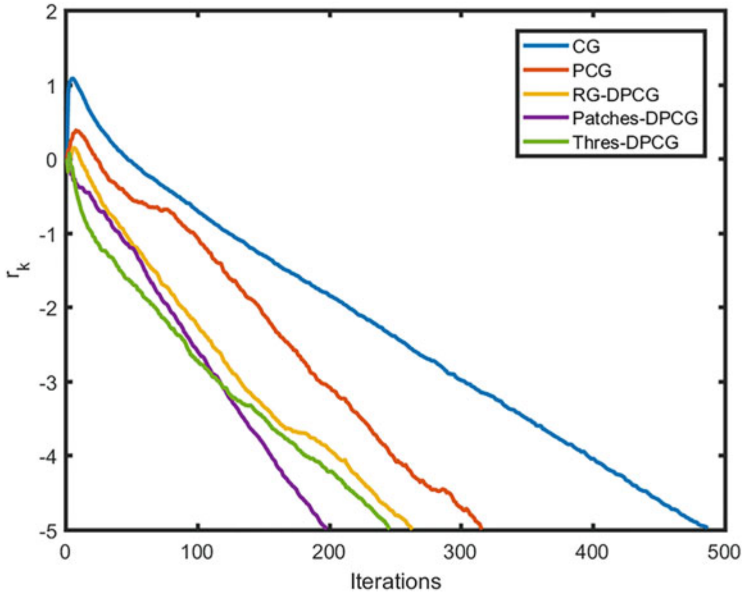


Fig. 4 Residual r_k for the measured MRI Shepp-Logan phantom image

Table 2 Different patches-DPCG methods, results for measured Shepp-Logan

Methods	CG	PCG	Patches-DPCG(4^2)	Patches-DPCG(8^2)	Patches-DPCG(16^2)
Iterations	487	316	198	188	162
Time (s)	0.35	0.25	0.23	0.37	0.88

4 Conclusions

We studied the DPCG method to solve the diffusion equation for image denoising. We used three different ways to construct the deflation vectors. The algorithm is tested on a simulated and a measured image. The deflation method works well for image denoising and the DPCG method converges faster than CG and PCG. Comparing the patch-based DPCG with region growing and thresholding-DPCG, we conclude that patches-DPCG achieves the best convergence and is not sensitive to noise.

Acknowledgments The authors thank the LUMC team for providing the low-field MR image. This work is partly funded by NWO WOTRO under grant W07.303.101 and by the Chinese Scholarship Council.

References

1. F. Catté, P-L. Lions, J-M. Morel, and T. Coll. Image Selective Smoothing and Edge Detection by Nonlinear Diffusion. *SIAM Journal on Numerical Analysis*, 29(1):182–193, 1992.
2. T. Chan and J. Shen. *Image Processing and Analysis: Variational, PDE, Wavelet, and Stochastic Methods*. Society for Industrial and Applied Mathematics Philadelphia, PA, USA, 2005.
3. M.L. de Leeuw den Bouter, M.B. van Gijzen, and R.F. Remis. Conjugate Gradient Variants for L_p-regularized Image Reconstruction in Low-field MRI. *SN Applied Sciences*, 2019.
4. M.R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J. Research Nat. Bur. Standards*, 49:409–436, 1952.
5. R.A. Nicolaides. Deflation of Conjugate Gradients with Applications to Boundary Value Problems. *SIAM Journal on Numerical Analysis*, 24(2):355–365, 1987.
6. T. O'Reilly, W.M. Teeuwisse, and A.G. Webb. Three-dimensional MRI in a Homogenous 27 cm Diameter Bore Halbach Array Magnet. *Journal of Magnetic Resonance*, 307:106578, 08 2019.
7. P. Perona and J. Malik. Scale-space and Edge Detection Using Anisotropic Diffusion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(7):629–639, 1990.
8. G. Rohit, D. Lukarski, M.B. van Gijzen, and C. Vuik. Evaluation of the Deflated Preconditioned CG method to solve Bubbly and Porous Media Flow Problems on GPU and CPU. *International Journal for Numerical Methods in Fluids*, 80:666–683, 09 2015.
9. L.A. Shepp and B.F. Logan. The Fourier Reconstruction of a Head Section. *IEEE Transactions on Nuclear Science*, 21(3):21–43, 1974.
10. C. Vuik, A. Segal, and J.A. Meijerink. An Efficient Preconditioned CG Method for the Solution of a Class of Layered Problems with Extreme Contrasts in the Coefficients. *Journal of Computational Physics*, 152(1):385–403, 1999.

Coupled Flow and Mechanics in a 3D Porous Media with Line Sources

Nadia S. Taki and Ingeborg G. Gjerde

Abstract In this paper, we consider the numerical approximation of the quasi-static, linear Biot model in a 3D domain Ω when the right-hand side of the flow equation is concentrated on a 1D line source δ_Λ. This model is of interest in the context of medicine, where it can be used to model flow and deformation through vascularized tissue. The model itself is challenging to approximate as the line source induces the pressure and flux solution to be singular. To overcome this, we here combine two methods: (1) a fixed-stress splitting scheme to decouple the flow and mechanics equations and (2) a singularity removal method for the pressure and flux variables. The singularity removal is based on a splitting of the solution into a lower regularity term capturing the solution singularities and a higher regularity term denoted the remainder. With this in hand, the flow equations can now be reformulated so that they are posed with respect to the remainder terms. The reformulated system is then approximated using the fixed-stress splitting scheme. We conclude by showing the results for a test case simulating flow through vascularized tissue. Here, the numerical method is found to converge optimally using lowest-order elements for the spatial discretization.

1 Introduction

The coupling of mechanics and flow in porous media is relevant for a wide range of applications, occurring for instance in geophysics [1, 21] and medicine [16, 20]. We put forward a model relevant for simulating perfusion, i.e., blood flow, and

N. S. Taki
University of Stuttgart, Institute for Modelling Hydraulic and Environmental Systems, Stuttgart, Germany
e-mail: nadia.skoglund@iws.uni-stuttgart.de

I. G. Gjerde (✉)
University of Bergen, Department of Mathematics, Bergen, Norway
e-mail: ingeborg.gjerde@uib.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_46

deformation in vascularized tissue. This problem is of high interest in the context of medicine, as clinical measurements of perfusion provide important indicators for e.g. Alzheimer's disease [4, 13], stroke [18] and cancer [8]. Moreover, both the tissue and blood vessels are elastic, and these properties constitute another valuable clinical indicator. Vascular compliance, as one example, is reduced in cases of vascular dementia, but not in cases Alzheimer's disease [6].

We consider the fully coupled quasi-static, linear Biot system [2], modeling a poroelastic media when the source term in the flow equation is concentrated on Λ. Let $\Omega \subset \mathbb{R}^3$ denote a bounded open 3D domain with smooth boundary $\partial\Omega$ and $\Lambda = \sum_{i=1}^{n} \Lambda_i$ a collection of straight line segments $\Lambda_i \subset \mathbb{R} \subset \Omega$ embedded in Ω. The equations on the space-time domain $\Omega \times (0, T)$ read:

Find $(\mathbf{u}, p, \mathbf{w})$ such that:

$$-\nabla \cdot [2\,\mu\,\boldsymbol{\varepsilon}(\mathbf{u}) + \lambda(\nabla \cdot \mathbf{u})\,\mathbf{I}] + \alpha\nabla p = \mathbf{f}, \tag{1a}$$

$$\partial_t\left(\frac{p}{M} + \alpha\nabla \cdot \mathbf{u}\right) + \nabla \cdot \mathbf{w} = \psi + f\delta_\Lambda, \tag{1b}$$

$$\kappa^{-1}\,\mathbf{w} + \nabla p = \rho_f \mathbf{g}, \tag{1c}$$

where $\mathbf{u}$ denotes the displacement, $\boldsymbol{\varepsilon}(\mathbf{u}) = \frac{1}{2}(\nabla\mathbf{u} + \nabla\mathbf{u}^T)$ the (linear) strain tensor, p the pressure, $\mathbf{w}$ the Darcy's flux, α the Biot coefficient, κ the permeability tensor divided by the fluid viscosity, ρ_f the fluid density, $\mathbf{g}$ the gravity vector, M the Biot modulus, μ and λ are Lamé parameters, $\mathbf{f} \in L^2(\Omega)$ the contribution from body forces and a source term $\psi \in L^2(\Omega)$. We assume $M, \alpha, \mu, \lambda \in L^\infty(\Omega)$ to be strictly positive and uniformly bounded and $\rho_f \in \mathbb{R}$, $\mathbf{g} \in \mathbb{R}^3$. Additionally, $\kappa \in W^{2,\infty}(\Omega)$ is assumed scalar-valued, as required by the singularity removal method. For simplicity, we use homogeneous boundary conditions $\mathbf{u} = \mathbf{0}$, $p = 0$ on $\partial\Omega \times [0, T]$ and initial conditions $\mathbf{u} = \mathbf{u}_0$, $p = p_0$ in $\Omega \times \{0\}$.

The system (1a)–(1c) is made non-standard by a generalized Dirac line source δ_Λ of intensity f in the right-hand side of (1b). The line source is defined mathematically as

$$\int_\Omega f\delta_\Lambda v \,\mathrm{d}\Omega = \sum_{j=1}^{m} \int_{\Lambda_j} f(s_j, t)v(s_j)\,\mathrm{d}S \quad \forall v \in C^0(\bar{\Omega}).$$

Physically, it is introduced to model the mass exchange between the vascular network and the surrounding tissue. This exchange occurs through the capillary blood vessels, which have radii ranging from 5 to 10 μm. These blood vessels are too small to be captured as 3D objects in a mesh; instead, they are typically reduced to being one-dimensional line segments, see e.g. [5, 9, 14, 15, 22]. The system (1a)–(1c) would then be on the same form as the one considered in [16], with the exception that the exchange term is here concentrated on the 1D vascular network.

We take $f \in C^0(\Omega)$ and focus our attention on the challenges introduced by the line source δ_Λ. The line source induces p and $\mathbf{w}$ to be singular, i.e., they both diverge to infinity on Λ. Consequently, one has $p \in L^2(\Omega)$ but $\mathbf{w} \notin (L^2(\Omega))^3$; the solution is then not regular enough to fit the analytic framework of [3, 19]. They both prove global convergence for the fixed-stress splitting scheme applied to Biot's equations. Moreover, these singularities are expensive to resolve numerically, making the solution highly challenging to approximate.

In order to tackle this issue, we here combine two strategies: (1) a fixed-stress splitting scheme that decouples the mechanics equation (1a) from the flow equations (1b)–(1c), and (2) a singularity removal method for the flow equations. For an introduction to the fixed-stress splitting scheme, we refer to the works of Mikelić et al. [19] and Both et al. [3]; for an introduction to the singularity removal method, we refer to our earlier work [10, 11].

2 Mathematical Model and Discretization

In this section, we begin by introducing a splitting method that decomposes p and $\mathbf{w}$ into higher and lower regularity terms. Here, the lower regularity terms are given explicitly. The model (1a)–(1c) can be reformulated so that it is given with respect to the higher regularity terms; we refer to this as the singularity removal method. Next, we show how this model can readily be approximated by the fixed-stress splitting scheme.

2.1 Singularity Removal Method

For the sake of notational simplicity, we assume κ to be constant; a spatially varying κ could be handled as shown in [11, Sect. 3.3]. Let $\mathbf{a}_i, \mathbf{b}_i$ denote the endpoints of the line segment Λ_i. From [11, Sect. 3.2], we have a function G defined as

$$G(\boldsymbol{x}) = \sum_{i=1}^{n} \frac{1}{4\pi} \ln\left(\frac{r_{b,i} + L_i + \boldsymbol{\gamma_i} \cdot (\boldsymbol{a}_i - \boldsymbol{x})}{r_{a,i} + \boldsymbol{\gamma}_i \cdot (\boldsymbol{a}_i - \boldsymbol{x})}\right), \tag{2}$$

with $r_{a,i} = \|\boldsymbol{x} - \boldsymbol{a}_i\|$, $r_{b,i} = \|\boldsymbol{x} - \boldsymbol{b}_i\|$, $L_i = \|\boldsymbol{b}_i - \boldsymbol{a}_i\|$ and $\boldsymbol{\gamma}_i = \frac{\boldsymbol{b}_i - \boldsymbol{a}_i}{L_i}$ as the normalized tangent vector of Λ_i. Centrally, this function solves $-\Delta G = \delta_\Lambda$ in the appropriate weak sense; i.e., we have:

$$-\int_\Omega \Delta G \, v \, \mathrm{d}\Omega = \int_\Lambda v \, \mathrm{d}S \quad \forall v \in C^0(\bar{\Omega}).$$

Having this function in hand, we next formulate the following splitting ansatz:

$$p = p_s + p_r, \quad \mathbf{w} = \mathbf{w}_s + \mathbf{w}_r, \tag{3}$$

where $p_s = \frac{fG(\mathbf{x})}{\kappa}$ and $\mathbf{w}_s = -\kappa \nabla p_s$. The terms p_s and $\mathbf{w}_s$ capture the singular part of the solution, and are explicitly given via the function G. This allows p_r and $\mathbf{w}_r$ to enjoy higher regularity and improved approximation properties. Assume for the moment that the solution $\mathbf{u}$ is given. Inserting the splitting (3) into (1b)–(1c) one finds the following reformulated flow equation:

Find $(p_r, \mathbf{w}_r)$ such that:

$$\partial_t \left(\frac{p_r}{M} + \alpha \nabla \cdot \mathbf{u} \right) + \nabla \cdot \mathbf{w}_r = \psi_r, \tag{4a}$$

$$\kappa^{-1}\, \mathbf{w}_r + \nabla p_r = \rho_f \mathbf{g}, \tag{4b}$$

where $\psi_r = \psi - \frac{\partial_t p_s}{M} + G\Delta f + 2\nabla G \cdot \nabla f$. Here, (4b) is straightforward to obtain. For (4a), we used that

$$\begin{aligned} \partial_t \left(\frac{p_r}{M} + \alpha \nabla \cdot \mathbf{u} \right) + \nabla \cdot \mathbf{w}_r &= \psi + f\delta_\Lambda - \frac{\partial_t p_s}{M} - \nabla \cdot \mathbf{w}_s \\ &= \psi + f\delta_\Lambda - \frac{\partial_t p_s}{M} + \nabla \cdot (\kappa \nabla \frac{fG}{\kappa}) \\ &= \psi - \frac{\partial_t p_s}{M} + 2\nabla f \cdot \nabla G + (\Delta f)G. \end{aligned}$$

In the last line we used the product rule to obtain $\nabla \cdot (\kappa \nabla \frac{fG}{\kappa}) = \Delta(fG) = f\Delta G + 2\nabla f \cdot \nabla G + (\Delta f)G$ along with the relation $f\Delta G = -f\delta_\Lambda$.

The value of the reformulation lies in the fact that ψ_r can now be expected to belong to $L^2(\Omega)$. To see this, note that $\psi \in L^2(\Omega)$ by assumption. $G \in L^2(\Omega)$ can be shown by straightforward calculation; it follows that $p_s \in L^2(\Omega)$. Finally, one can show that $\nabla G \cdot \nabla f \in L^2(\Omega)$; for verification of this, we refer to the calculations in [10, Sect. 4.2] along with the embedding $f \in C^0(\Omega) \subset H^1(\Omega)$.

Let now (1a) and (4a)–(4b) denote the reformulated Biot equation. As $\psi_r \in L^2(\Omega)$, this system fits the analytic framework of [3].

2.2 Fixed-Stress Splitting Scheme

Next, we show how the reformulated Biot equations (1a) and (4a)–(4b) can be approximated via the fixed-stress splitting scheme from [3]. Let $\mathcal{T}_h$ be the triangularization of the domain Ω with mesh size h. We let $0 = t^0 < \cdots < t^N = T$ be a partition of the time interval $(0, T)$ with $N \in \mathbb{N}^*$ and define a constant time step size $\tau = t^{k+1} - t^k := T/N$ for $k \geq 0$. To discretize the system, we employ backward Euler for time and a finite element method for space. The solutions are approximated with linear piecewise polynomials, constant piecewise polynomials and lowest-order Raviart-Thomas spaces for the displacement, pressure and flux, respectively. The discrete spaces are given by:

$$\begin{aligned}
\mathbf{V}_h &= \{\, \mathbf{v}_h \in [\, H_0^1(\Omega)\,]^3 \mid \forall K \in \mathcal{T}_h,\ \mathbf{v}_{h|K} \in [\,\mathbb{P}_1\,]^3\},\\
Q_h &= \{\, q_h \in L^2(\Omega) \mid \forall K \in \mathcal{T}_h,\ q_{h|K} \in \mathbb{P}_0\},\\
\mathbf{Z}_h &= \{\mathbf{z}_h \in H(\mathrm{div};\Omega) \mid \forall K \in \mathcal{T}_h,\ \mathbf{z}_{h|K}(\boldsymbol{x}) = \boldsymbol{\eta} + \xi \boldsymbol{x},\ \boldsymbol{\eta} \in \mathbb{R}^3, \xi \in \mathbb{R}\},
\end{aligned}$$

with $\mathbb{P}_1$ and $\mathbb{P}_0$ as the linear and constant piecewise polynomials.

Take now $\langle\cdot,\cdot\rangle$ to be the $L^2(\Omega)$-inner product and $(\mathbf{u}_h^0, p_h^0, \mathbf{w}_h^0) \in \mathbf{V}_h \times Q_h \times \mathbf{Z}_h$ to be the initial values of the solution. We assume the solution of the displacement, pressure and flux is known for the previous time step. The time-discretization of (1a) and (4a)–(4b) then reads:

Given $(\mathbf{u}_h^{n-1}, p_{r,h}^{n-1}, \mathbf{w}_{r,h}^{n-1}) \in \mathbf{V}_h \times Q_h \times \mathbf{Z}_h$, find $(\mathbf{u}_h^n, p_{r,h}^n, \mathbf{w}_{r,h}^n) \in \mathbf{V}_h \times Q_h \times \mathbf{Z}_h$ such that

$$\begin{aligned}
\langle 2\mu\boldsymbol{\varepsilon}(\mathbf{u}_h^n), \boldsymbol{\varepsilon}(\mathbf{v}_h)\rangle + \langle \lambda(\nabla\cdot\mathbf{u}_h^n), \nabla\cdot\mathbf{v}_h\rangle - \langle \alpha p_h^n, \nabla\cdot\mathbf{v}_h\rangle &= \langle \mathbf{f}^n, \mathbf{v}_h\rangle,\\
\left\langle \frac{1}{M} p_{r,h}^n, q_h\right\rangle + \langle \alpha\nabla\cdot\mathbf{u}_h^n, q_h\rangle + \tau\langle \nabla\cdot\mathbf{w}_{r,h}^n, q_h\rangle &= \tau\langle \psi_r^n, q_h\rangle + \left\langle \frac{1}{M} p_{r,h}^{n-1}, q_h\right\rangle\\
&\quad + \langle \alpha\nabla\cdot\mathbf{u}_h^{n-1}, q_h\rangle,\\
\langle \kappa^{-1}\mathbf{w}_{r,h}^n, \mathbf{z}_h\rangle - \langle p_{r,h}^n, \nabla\cdot\mathbf{z}_h\rangle &= \langle \rho_f\mathbf{g}, \mathbf{z}_h\rangle,
\end{aligned}$$

for all $(\mathbf{v}_h, q_h, \mathbf{z}_h) \in \mathbf{V}_h \times Q_h \times \mathbf{Z}_h$.

The idea of the fixed-stress splitting scheme is to decouple the flow and mechanics equation while keeping an artificial volumetric stress $\sigma_\beta = \sigma_0 + K_{dr}\nabla\cdot\mathbf{u} - \alpha p$ constant. Here, $K_{dr} \in L^\infty(\Omega)$ is referred to as the drained bulk modulus. We consider the theoretically optimal tuning parameter $\beta_{FS} = \alpha^2/K_{dr}$ with $K_{dr} = \frac{d}{2}(\mu+\lambda)$ [3].

We define a sequence $(\mathbf{u}_h^{n,i},\ p_{r,h}^{n,i},\ \mathbf{w}_{r,h}^{n,i})$, $i \geq 0$. Let i denote the current iteration step and $i-1$ denote the previous iteration step. Then initialize $\mathbf{u}$, p_r and $\mathbf{w}_r$ by $\mathbf{u}_h^{n,0} = \mathbf{u}_h^{n-1}$, $p_{r,h}^{n,0} = p_{r,h}^{n-1}$ and $\mathbf{w}_{r,h}^{n,0} = \mathbf{w}_{r,h}^{n-1}$, respectively. The algorithm iterates until a stopping criterion is reached. The full scheme reads:

Step 1 Given $(\mathbf{u}_h^{n,i-1},\ p_{r,h}^{n,i-1},\ \mathbf{w}_{r,h}^{n,i-1}) \in \mathbf{V}_h \times Q_h \times \mathbf{Z}_h$. Find $(p_h^{n,i},\ \mathbf{w}_h^{n,i}) \in Q_h \times \mathbf{Z}_h$ such that $\forall\,(q_h,\ \mathbf{z}_h) \in Q_h \times \mathbf{Z}_h$:

$$\begin{aligned}
\left\langle \left(\frac{1}{M} + \beta_{FS}\right) p_{r,h}^{n,i}, q_h\right\rangle + \tau\langle \nabla\cdot\mathbf{w}_{r,h}^{n,i}, q_h\rangle &= \tau\langle \psi_r^n, q_h\rangle + \left\langle \frac{1}{M} p_{r,h}^{n-1}, q_h\right\rangle \qquad (6)\\
&\quad + \langle \alpha\nabla\cdot\mathbf{u}_h^{n-1}, q_h\rangle + \langle \beta_{FS}\, p_{r,h}^{n,i-1}, q_h\rangle\\
&\quad - \langle \alpha\nabla\cdot\mathbf{u}_h^{n,i-1}, q_h\rangle,
\end{aligned}$$

$$\langle \kappa^{-1}\mathbf{w}^{n,i}_{r,h}, \mathbf{z}_h\rangle - \langle p^{n,i}_{r,h}, \nabla\cdot\mathbf{z}_h\rangle = \langle \rho_f \mathbf{g}, \mathbf{z}_h\rangle. \tag{7}$$

Step 2 Update the full pressure and flux solutions: $p_h^{n,i} = p^n_{s,h} + p^{n,i}_{r,h}$ and $\mathbf{w}_h^{n,i} = \mathbf{w}^n_{s,h} + \mathbf{w}^{n,i}_{r,h}$.

Step 3 Given $p_h^{n,i} \in Q_h$. Find $\mathbf{u}_h^{n,i} \in \mathbf{V}_h$, such that $\forall\, \mathbf{v}_h \in \mathbf{V}_h$:

$$\langle 2\mu\boldsymbol{\varepsilon}(\mathbf{u}_h^{n,i}), \boldsymbol{\varepsilon}(\mathbf{v}_h)\rangle + \langle \lambda(\nabla\cdot\mathbf{u}_h^{n,i}), \nabla\cdot\mathbf{v}_h\rangle = \langle \alpha p_h^{n,i}, \nabla\cdot\mathbf{v}_h\rangle + \langle \mathbf{f}^n, \mathbf{v}_h\rangle. \tag{8}$$

3 Numerical Results

In this section, we provide numerical convergence results for a test case using parameters relevant for flow through vascularized tissue. Let the medium of consideration be an isotropic, homogeneous porous medium and κ a positive scalar quantity. We let $(\mathbf{p}^i_{r,h}, \mathbf{w}^i_{r,h}, \mathbf{u}^i_h)$ be the solutions at iteration step i and $(\mathbf{p}^{i-1}_{r,h},\ \mathbf{w}^{i-1}_{r,h},\ \mathbf{u}^{i-1}_h)$ the solutions at the previous iteration step $i-1$. The procedure stops when reaching the following criterion:

$$\left\| (\mathbf{p}^i_{r,h},\ \mathbf{w}^i_{r,h},\ \mathbf{u}^i_h) - (\mathbf{p}^{i-1}_{r,h},\ \mathbf{w}^{i-1}_{r,h},\ \mathbf{u}^{i-1}_h) \right\| \le \epsilon_a + \epsilon_r \left\| (\mathbf{p}^i_{r,h},\ \mathbf{w}^i_{r,h},\ \mathbf{u}^i_h) \right\|,$$

where $\epsilon_a, \epsilon_r > 0$ are given tolerances (see Table 1).

Let $\Omega = \{(0,1)\times(0,1)\times(0,1) \subset \mathbb{R}^3\}$ be a cube discretized by $1/h \times 1/h \times 1/h$ tetrahedrons. The numerical results are obtained by the fixed-stress splitting scheme proposed in Sect. 2.2 and the programming platform FEniCS [17]. Convergence is tested against the following analytic solutions

Table 1 Material parameters used to solve the Biot's equations with lower dimensional source terms (1a)–(1c) in Sect. 3. There is a wide rage of parameters used in literature. The ones represented here are a sample of representative parameters

Symbol	Quantity	Value	Reference
κ	Permeability divided by the fluid viscosity	1.57e−2 mm^2 mPa^{-1} s^{-1}	[23, Table 1]
E	Tuning parameter	1.5e6 mPa	[16, Table 6]
M	Biot modulus	3.9e7 mPa	[12, Table 2]
α	Biot coefficient	1.0	
ν	Poisson's ratio	0.2	
$\mathbf{g}$	Gravitational vector	$\mathbf{0}$ mm s^{-2}	[7]
T	Final time	1.0 s	
τ	Time step	0.1 s	
ϵ_a	Absolute error tolerance	1e−6	
ϵ_r	Relative error tolerance	1e−6	

Table 2 Errors and convergence rates obtained solving (6)–(8) with analytical solutions found in this section. For reference, optimal convergence rates are listed in the bottom row

h	$\|p_a - p_h\|_{L^2(\Omega)}$	$\|\mathbf{w}_a - \mathbf{w}_h\|_{L^2(\Omega)}$	$\|\mathbf{u}_a - \mathbf{u}_h\|_{L^2(\Omega)}$
1/8	1.2e−01	7.2e−03	5.9e−04
1/16	6.3e−02	3.5e−03	1.5e−04
1/32	3.1e−02	1.7e−03	3.7e−05
Rate	1.0	1.0	2.0
Optimal	1.0	1.0	2.0

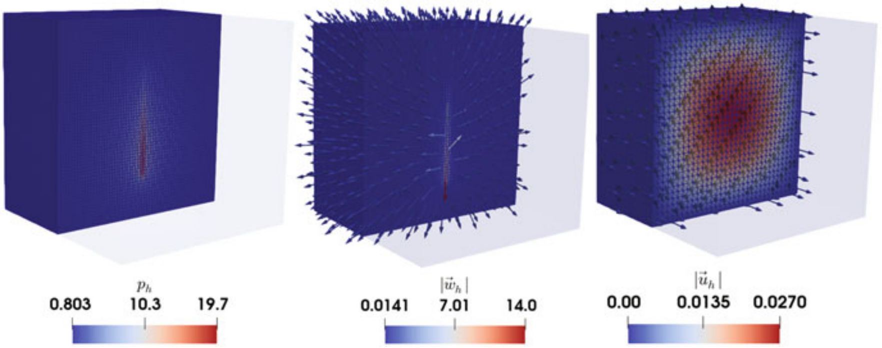


Fig. 1 Left: Plot of the reconstructed pressure. Middle: Magnitude of the full flux. Right: Magnitude of the displacement. All plots are numerical solutions obtained by the fixed-stress splitting scheme (6)–(8) with one line source

$$p_r = \frac{1}{4\pi\kappa} f(t)(r_a - r_b), \qquad \mathbf{w}_r = -\kappa \nabla p_r, \qquad \mathbf{u} = tx(1-x)y(1-y)z(1-z)[1\ 1\ 1]^T,$$

where $f(t) = \sin(t)$ is a pulsative intensity function. We selected two points $\boldsymbol{a} = [0.5\ \ 0.8\ \ 0.5]^T$ and $\boldsymbol{b} = [0.5\ \ 0.2\ \ 0.5]^T$ to describe the line segment. Then computed the solutions using mixed-finite element formulations for the correction terms p_r and $\mathbf{w}_r$, and the solution displacement $\mathbf{u}$ is calculated with the conformal finite element formulation.

Table 2 shows the error and convergence rates obtained using the parameters listed in Table 1. The singularity removal based fixed-stress splitting scheme is seen to converge optimally for each variable $\mathbf{u}_h$, p_h, and $\mathbf{w}_h$. The plots for this problem are illustrated in Fig. 1. The figure includes the plot of the full pressure, the magnitude of the full flux and the magnitude of the displacement, accordingly.

References

1. Bause, M., Radu, F.A., Köcher, U.: Space-time finite element approximation of the Biot poroelasticity system with iterative coupling. Computer Methods in Applied Mechanics and Engineering **320**, 745–768 (2017)
2. Biot, M.A.: General theory of three–dimensional consolidation. Journal of Applied Physics **12**(2), 155–164 (1941)
3. Both, J.W., Borregales, M., Kumar, K., Nordbotten, J.M., Radu, F.A.: Robust fixed stress splitting for Biot's equations in heterogeneous media. Applied Mathematics Letters **68**, 101–108 (2017)
4. Chen, Y., Wolk, D., Reddin, J., Korczykowski, M., Martinez, P., Musiek, E., Newberg, A., Julin, P., Arnold, S., Greenberg, J., Detre, J.: Voxel-level comparison of arterial spin-labeled perfusion MRI and FDG-PET in Alzheimer disease. Neurology **77**(22), 1977–1985 (2011)
5. D'Angelo, C., Quarteroni, A.: On the coupling of 1D and 3D diffusion-reaction equations: Application to tissue perfusion problems. Mathematical Models and Methods in Applied Sciences **18**(8), 1481–1504 (2008)
6. Dhoat, S., Ali, K., Bulpitt, C.J., Rajkumar, C.: Vascular compliance is reduced in vascular dementia and not in alzheimer's disease. Age and Ageing **37**(6), 653–659 (2008)
7. Formaggia, L., Quarteroni, A., Veneziani, A.: Cardiovascular Mathematics: Modeling and Simulation of the Circulatory System, vol. 1 (2009)
8. Gillies, R.J., Schomack, P.A., Secomb, T.W., Raghunand, N.: Causes and effects of heterogeneous perfusion in tumors. Neoplasia **1**(3), 197–207 (1999)
9. Gjerde, I., Kumar, K., Nordbotten, J.M.: A singularity removal method for coupled 1d-3d flow models (2018). ArXiv:1812.03055 [math.AP]
10. Gjerde, I., Kumar, K., Nordbotten, J.M.: A mixed approach to the poisson problem with line sources (2019). ArXiv:1910.11785 [math.AP]
11. Gjerde, I., Kumar, K., Nordbotten, J.M., Wohlmuth, B.: Splitting method for elliptic equations with line sources. ESAIM: M2AN **53**(5) (2019)
12. Guo, L., Vardakis, J., Lassila, T., Mitolo, M., Ravikumar, N., Chou, D., Lange, M., Sarrami-Foroushani, A., Tully, B., Taylor, Z., Varma, S., Venneri, A., Frangi, A., Ventikos, Y.: Subject-specific multiporoelastic model for exploring the risk factors associated with the early stages of alzheimer's disease. Interface Focus **8**(1) (2017)
13. Iturria-Medina, Y., Sotero, R.C., Toussaint, P.J., Mateos-Pérez, J.M., Evans, A.C.: Early role of vascular dysregulation on late-onset alzheimer's disease based on multifactorial data-driven analysis. Nature Communications **7**(1) (2016)
14. Köppl, T., Vidotto, E., Wohlmuth, B., Zunino, P.: Mathematical modeling, analysis and numerical approximation of second-order elliptic problems with inclusions. Mathematical Models and Methods in Applied Sciences **28**(05), 953–978 (2018)
15. Laurino, F., Zunino, P.: Derivation and analysis of coupled PDEs on manifolds with high dimensionality gap arising from topological model reduction. ESAIM: Mathematical Modelling and Numerical Analysis **53**(6), 2047–2080 (2019)
16. Lee, J., Piersanti, E., Mardal, K.A., Rognes, M.: A mixed finite element method for nearly incompressible multiple-network poroelasticity. SIAM Journal on Scientific Computing **41**(2), A722–A747 (2019)
17. Logg, A., Mardal, K.A., Wells, G.N., et al.: Automated Solution of Differential Equations by the Finite Element Method. Springer (2012)
18. Markus, H.S.: Cerebral perfusion and stroke. Journal of Neurology, Neurosurgery & Psychiatry **75**(3), 353–361 (2004)
19. Mikelić, A., Wheeler, M.F.: Convergence of iterative coupling for coupled flow and geomechanics. Computational Geosciences **17**(3), 455–461 (2013)
20. Nagashima, T., Tamaki, N., Matsumoto, S., Horwitz, B., Seguchi, Y.: Biomechanics of hydrocephalus: a new theoretical model. Neurosurgery **21**(6), 898–904 (1987)

21. Storvik, E., Both, J.W., Kumar, K., Nordbotten, J.M., Radu, F.A.: On the optimization of the fixed-stress splitting for Biot's equations. International Journal for Numerical Methods in Engineering **120**(2), 179–194 (2019)
22. Vidotto, E., Koch, T., Köppl, T., Helmig, R., Wohlmuth, B.: Hybrid models for simulating blood flow in microvascular networks. Multiscale Modeling & Simulation **17**(3), 1076–1102 (2019)
23. Weiss, C.: Finite element analysis for model parameters distributed on a hierarchy of geometric simplices. GEOPHYSICS **82**(4), 1–52 (2017)

A Semismooth Newton Solution of the Steady-State Non-isothermal Bingham Flow with Temperature Dependent Nonlocal Parameters

Sergio González-Andrade

Abstract In this paper, we discuss the numerical solution of the non-isothermal steady-state Bingham flow considering that the viscosity and the yield limit variate with temperature. In the present contribution, we focus on the asymptotic limit case of high thermal conductivity. In this case, the energy equation collapses into an implicit energy equation which involves the viscosity and the yield stress functions, while the temperature becomes a constant solution for this equation. Once we obtain the coupled limit system of this energy equation and the classical Bingham variational inequality of the second kind, we propose a mixed formulation for the resulting limit variational inequality and a finite element discretization for the resulting system of PDEs. Next, we develop a semismooth Newton algorithm for the coupled flow model. Finally, we carry on several numerical experiments for validating our method.

1 Problem Statement

We are concerned with the non-isothermal flow of a Bingham fluid, considering temperature dependent parameters. Bingham fluids are materials characterized by the existence of a yield stress. This means that the material behaves like a viscoplastic fluid if the stress tensor overpasses a given constant (yield stress or plasticity threshold), and it behaves like a rigid solid if the stress is beneath this threshold. In this contribution, we focus on the numerical solution of the steady flow of these materials, considering that the viscosity and the yield stress depend on temperature.

Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, be an open and bounded set with Lipschitz boundary $\partial\Omega$. Let us assume that there exist $\Gamma, \Gamma_0 \subset \partial\Omega$, such that $|\Gamma_0| > 0$ and $\partial\Omega = \Gamma \uplus \Gamma_0$.

S. González-Andrade (✉)
Research Center on Mathematical Modeling (MODEMAT), Department of Mathematics - Escuela Politécnica Nacional del Ecuador, Quito, Ecuador
e-mail: sergio.gonzalez@epn.edu.ec

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_47

The governing equations for this phenomenon are

$$\left.\begin{array}{lr}
(\mathbf{u}\cdot\nabla)\mathbf{u}-\boldsymbol{\nabla}\cdot\boldsymbol{\tau}(\theta)+\nabla p=\mathbf{f}, & \text{in }\Omega\\
\nabla\cdot\mathbf{u}=0, & \text{in }\Omega\\
\left.\begin{array}{ll}\boldsymbol{\tau}(\theta)=\mu(\theta)\mathcal{E}\mathbf{u}+g(\theta)\frac{\mathcal{E}\mathbf{u}}{|\mathcal{E}\mathbf{u}|} & \text{if } \mathcal{E}\mathbf{u}\neq 0,\\ |\boldsymbol{\tau}(\theta)|\leq g(\theta) & \text{if } \mathcal{E}\mathbf{u}=0.\end{array}\right\} & \text{in }\Omega\\
\mathbf{u}\cdot\nabla\theta-\kappa\Delta\theta=\boldsymbol{\tau}(\theta):\mathcal{E}\mathbf{u}-\alpha\theta, & \text{in }\Omega,\\
\mathbf{u}=0, & \text{on }\partial\Omega,\\
\frac{\partial\theta}{\partial\mathbf{n}}=0 & \text{in }\Gamma_0,\\
\kappa\frac{\partial\theta}{\partial\mathbf{n}}+\beta\theta=0 & \text{in }\Gamma.
\end{array}\right. \tag{$\mathcal{P}$}$$

Here, $\kappa > 0$ stands for the thermal conductivity and $\mathbf{f}$ stands for the external body forces. We allow the existence of a possible external heat source proportional to θ, if $\alpha > 0$. We assume the classical non-slip boundary conditions for $\mathbf{u}$ and Robin conditions for the energy equation [1, 3].

The variational formulation of system ($\mathcal{P}$) corresponds to the problem: find $(\mathbf{u},\theta)\in V\times W^{1,q}(\Omega)$, for $1<q<n/(n-1)$, such that [1]

$$\begin{array}{l}
\int_\Omega[\mu(\theta)\mathcal{E}\mathbf{u}-(\mathbf{u}\cdot\nabla)\mathbf{u}]:\mathcal{E}\mathbf{u}\,dx+\int_\Omega g(\theta)|\mathcal{E}\mathbf{v}|\,dx-\int_\Omega g(\theta)|\mathcal{E}\mathbf{u}|\,dx\\
\qquad\qquad\geq\int_\Omega\mathbf{f}\cdot(\mathbf{v}-\mathbf{u})\,dx,\ \ \forall\mathbf{v}\in V,\\
\kappa\int_\Omega\nabla\theta\cdot\nabla\phi\,dx+\int_\Omega(\mathbf{u}\cdot\nabla\theta)\phi\,dx+\alpha\int_\Omega\theta\phi\,dx+\beta\int_\Gamma\theta\phi\,ds\\
\quad=\int_\Omega\left[\mu(\theta)|\mathcal{E}\mathbf{u}|^2+g(\theta)|\mathcal{E}\mathbf{u}|\right]\phi\,dx,\ \ \forall\phi\in W^{1,q'}(\Omega).
\end{array} \tag{$\mathcal{VP}$}$$

Here, $V:=\{\mathbf{v}\in\mathbf{H}^1(\Omega)\ :\ \nabla\cdot\mathbf{v}=0\text{ in }\Omega\text{ and }\mathbf{v}=0\text{ on }\Gamma_0\}$ and $\mathbf{f}\in V'$. The term $\boldsymbol{\tau}(\theta):\mathcal{E}\mathbf{u}$ corresponds to the dissipated energy, and it makes sense only if $\mathcal{E}\mathbf{u}\neq 0$. Thus, we use the corresponding form of $\boldsymbol{\tau}(\theta)$ in the energy equation. Furthermore, the associated integral term in the variational formulation is well posed since for $1<q<n/(n-1)$ we have that $q'>n$, which implies that $W^{1,q'}(\Omega)$ is continuously embedded in $L^\infty(\Omega)$.

Existence of solutions for ($\mathcal{VP}$) has been established in [1], by assuming that $\alpha,\beta\geq 0,\alpha+\beta>0$, $\mu\in C(\mathbb{R})$, $g\in C(\mathbb{R})$, and that there exist μ_0, μ_1, g_0 and g_1 such that

$$0<\mu_0\leq\mu(t)\leq\mu_1,\ \ \text{and}\ \ 0\leq g_0\leq g(t)\leq g_1,\ \forall t\in\mathbb{R}. \tag{1}$$

Based on the Houska model (see [4]), we consider the following piecewise linear functions

$$\mu(t)=\max(\mu_0,\min(\mu_1,\ell_\mu(t)))\ \ \text{and}\ \ g(t)=\max(g_0,\min(g_1,\ell_g(t))), \tag{2}$$

where ℓ_μ and ℓ_g represent a line segment in the interval $[0,1]$ for μ and g, respectively (see Fig. 1). These functions satisfy the assumption (1), which implies

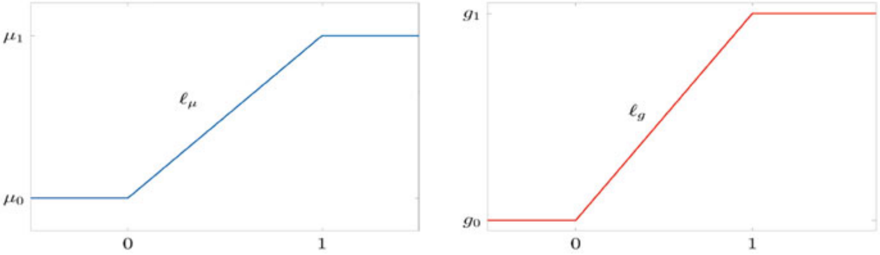


Fig. 1 Model functions for the temperature dependent viscosity μ and yield stress g, respectively

that existence of solutions for the problem ($\mathcal{VP}$) is guaranteed. However, $\mu(\theta)$ and $g(\theta)$ are clearly non differentiable.

Usually, the dissipation term is neglected and the coupling between the velocity and the temperature lies only on the convective term $\mathbf{u} \cdot \nabla\theta$. In several applications, however, the dissipated energy needs to be analyzed in order to fully understand the phenomenon. This is the case of geophysical flows or fluids in food industry.

In this contribution, we will approach the analysis of the coupled model in the limit case $\kappa \to \infty$. This limit model can be associated with the so called super fluids. These materials are supposed to have infinite thermal conductivity, which means that any volume of the fluid, no matter how large, will always be precisely the same temperature throughout.

In order to obtain the limit problem, we start by recalling that system ($\mathcal{VP}$) has a solution $(\mathbf{u}_\kappa, \theta_\kappa) \in V \times W^{1,q}(\Omega)$, corresponding to each $\kappa > 0$. Next, in [1, Th. 2.2] it is proven that under Assumption (1), the sequence $(\mathbf{u}_\kappa, \theta_\kappa)$ converges, as $\kappa \to \infty$, to a couple $(\mathbf{u}, \Theta) \in V \times \mathbb{R}$, solution of

$$\begin{aligned} \mu(\Theta) \int_\Omega \mathcal{E}\mathbf{u} : \mathcal{E}(\mathbf{v}-\mathbf{u})\, dx - \int_\Omega (\mathbf{u}\cdot\nabla)\mathbf{u} : \mathcal{E}(\mathbf{v}-\mathbf{u})\, dx + g(\Theta) \int_\Omega |\mathcal{E}\mathbf{v}|\, dx \\ -g(\Theta) \int_\Omega |\mathcal{E}\mathbf{u}|\, dx \geq \int_\Omega \mathbf{f}\cdot(\mathbf{v}-\mathbf{u})\, dx, \ \ \forall \mathbf{v} \in V, \\ (\alpha\, meas(\Omega) + \beta\, meas(\Gamma))\Theta = \mu(\Theta) \int_\Omega |\mathcal{E}\mathbf{u}|^2\, dx + g(\Theta) \int_\Omega |\mathcal{E}\mathbf{u}|\, dx. \end{aligned} \tag{$\mathcal{NVP}$}$$

System ($\mathcal{NVP}$) corresponds to the steady-state Bingham flow with nonlocal temperature dependent parameters [1, 5]. The second equation in the system above follows from the energy equation in ($\mathcal{VP}$), taking $\phi = 1$, and considering the following limits, which are obtained in [1, Sec. 4.2, pp. 156]

$$\nabla\theta_\kappa \to 0, \ \text{ in } L^q(\Omega) \ \text{ and } \ \theta_\kappa \to \Theta = constant, \ \ \text{in } W^{1,q}(\Omega), \ \ \text{as } \kappa \to \infty.$$

2 Regularization and Numerical Approach

In this section, we focus on the numerical solution of system ($\mathcal{NVP}$) by a semismooth Newton algorithm. The main difficulty concerning the numerical solution of ($\mathcal{NVP}$) is given by the uncertainty regarding the yielded and unyielded regions. In fact, in most cases, there is no a priori knowledge about the regions in which the material behaves like a fluid (yielded regions) or as a rigid solid (unyielded regions). The variational approach analyses this model as a free boundary problem, being the free boundary the phase separating the yielded from the unyielded regions. As a consequence, the flow is modelled by a variational inequality instead of a PDE. In the present contribution, we propose a regularization approach based on the so called Huber local regularization. Indeed, the idea lies in the fact that the variational inequality in ($\mathcal{NVP}$) constitutes a first order optimality condition for an optimization problem involving the nondifferentiable term $\int_\Omega |\mathcal{E}\mathbf{u}|\,dx$ (see [2]). Thus, we replace this term by a local approximation given by the expression $\int_\Omega \Upsilon(\mathcal{E}\mathbf{u})\,dx$, where

$$\mathbb{R}^{d\times d} \ni \mathbf{p} \mapsto \Upsilon(\mathbf{p}) = \begin{cases} g(\Theta)|\mathbf{p}| - \frac{g(\Theta)}{2\gamma}, & \text{if } |\mathbf{p}| \geq g(\Theta)/\gamma \\ \frac{\gamma}{2}|\mathbf{p}|, & \text{if } |\mathbf{p}| < g(\Theta)/\gamma. \end{cases}$$

This function is a local regularization of the Frobenius norm and gives us a local regularization of $\int_\Omega |\mathcal{E}\mathbf{u}|\,dx$. In [2], this approach has been discussed in great detail for the isothermal flow.

In the present case, Θ is a constant solution of a real nonlinear equation, which is defined by the two nondifferentiable functions $\mu(\Theta)$ and $g(\Theta)$. This fact allows us to extend the analysis in [2], based on the Fenchel duality theory and the de Rahm's theorem, to guarantee the existence of a tensor function $\mathbf{q} \in \mathbb{L}^{2\times 2}(\Omega)$ and a function $p \in L^2_0(\Omega)$ such that the following system has a unique solution, for each $\gamma > 0$

$$\begin{array}{lr} \mu(\Theta)\int_\Omega \mathcal{E}\mathbf{u}_\gamma : \mathcal{E}\mathbf{v}\,dx + \int_\Omega \mathbf{q} : \mathcal{E}\mathbf{v}\,dx + \int_\Omega p\,\nabla\cdot\mathbf{v}\,dx = \int_\Omega \mathbf{f}\cdot\mathbf{v}\,dx, & \forall \mathbf{v} \in \mathbf{H}^1_0(\Omega), \\ \int_\Omega r\,\nabla\cdot\mathbf{u}_\gamma\,dx = 0, & \forall r \in L^2_0(\Omega), \\ \max(g(\Theta), \gamma|\mathcal{E}\mathbf{u}_\gamma|)\mathbf{q} - \gamma g(\Theta)\mathcal{E}\mathbf{u}_\gamma, & \text{a.e. in } \Omega, \\ (\alpha\, meas(\Omega) + \beta\, meas(\Gamma))\Theta = \mu(\Theta)\int_\Omega |\mathcal{E}\mathbf{u}_\gamma|^2\,dx + g(\Theta)\int_\Omega |\mathcal{E}\mathbf{u}_\gamma|\,dx. & \end{array} \tag{3}$$

Further, we can extend the analysis in [2, Th. 4.1] to the present case, so we can state that $\mathbf{u}_\gamma \to \mathbf{u}$ in $\mathbf{H}^1_0(\Omega)$, as $\gamma \to \infty$, for each $\Theta \in \mathbb{R}$.

2.1 Discretization and Semismooth Newton Algorithm

The form of μ and g, the presence of the L^1-norm of $\mathcal{E}\mathbf{u}_\gamma$ in the fourth equation in (3) and the characterization of $\mathbf{q}$ in the third equation in (3) imply that the coupled system (3) is not differentiable.

Fortunately, the max and min functions, as well as the Frobenius norm are Newton or slantly differentiable in finite dimension spaces [6]. Following this fact, we propose a first order finite element discretization for the system (3), using the so called (cross-grid $\mathbb{P}_1$)-$\mathbb{Q}_0$ elements. It is known that these elements are LBB-stable and lead to a direct relation between the discrete primal and dual variables [2].

We start by constructing the finite dimension spaces $\mathbf{V}^h \subset \mathbf{H}^1_0(\Omega)$, $\mathbf{W}^h \subset \mathbb{L}^{2\times 2}(\Omega)$ and $Q^h \subset L^2_0(\Omega)$, with $\dim(\mathbf{V}^h) = n$, $\dim(\mathbf{W}^h) = m$ and $\dim(Q^h) = s$. Next, by using the classical Galerkin approach, we obtain the following system, written as an operator equation

$$\Phi(\mathbf{u}_h, \mathbf{q}_h, p_h, \Theta) := \begin{pmatrix} \mu(\Theta)\mathbf{A}^h\mathbf{u}^h + \mathbf{Q}^h\mathbf{q}^h + B^h p^h - \mathbf{f}^h \\ D(\mathbf{m}^h)\,\mathbf{q}^h - g(\Theta)\gamma\mathcal{E}^h\mathbf{u}^h \\ (B^h)^\top\mathbf{u}^h \\ (\omega^h_{\alpha,\beta})\Theta - \mu(\Theta)(\mathbf{u}^h)^\top\mathbf{A}^h\mathbf{u}^h - g(\Theta)\mathbf{K}^\top\widehat{\mathbf{N}}^h(\mathcal{E}^h\mathbf{u}_h) \end{pmatrix} = 0, \tag{4}$$

where $\omega^h_{\alpha,\beta} := \alpha\,|\Omega_h| + \beta\,|\Gamma_h|$. Here, $\mathbf{A}^h$ is the stiffness matrix, while the matrices $\mathbf{Q}^h$ and B^h are obtained in the usual way, from the bilinear forms $\int_\Omega \mathbf{q} : \mathcal{E}\mathbf{v}\,dx$ and $\int_\Omega p\,\nabla\cdot\mathbf{v}\,dx$, respectively. Further, $\mathcal{E}^h$ is a discretized version of the deformation tensor, constructed using the basis functions of $\mathbf{V}^h$ (see [2, Sec. 5]) and $D(\mathbf{m^h}) := \operatorname{diag}(\max\{g(\Theta), \gamma\mathbf{N}^h(\mathcal{E}^h\mathbf{y}^h)\})$. We approximate the value of the Frobenius norm of a discretized tensor, at each triangle in the discretized geometry, by using the following function $\mathbf{N}^h : \mathbb{R}^{4m} \to \mathbb{R}^{4m}$, given by

$$\mathbf{N}^h(\mathbf{q}^h)_i = \mathbf{N}^h(\mathbf{q}^h)_{i+m} = \cdots = \mathbf{N}^h(\mathbf{q}^h)_{i+4m} := |(q_i, q_{i+m}, \ldots, q_{i+4m})^\top|,\ i = 1, \ldots, m.$$

Further, $\widehat{\mathbf{N}}^h : \mathbb{R}^{4m} \to \mathbb{R}^m$ is given by $\widehat{\mathbf{N}}^h(\mathbf{q}^h)_i := |(q_i, q_{i+m}, \ldots, q_{i+4m})^\top|$. Finally, we use a composite midpoint formula to approach the integral $\int_\Omega |\mathcal{E}\mathbf{u}_\gamma|\,dx$. Therefore, $\mathbf{K} \in \mathbb{R}^m$ is a vector whose components are the area of every triangle in the discretization.

Once we have discretized the system (3), we can discuss its generalized differentiability. In particular, we have to calculate the Newton derivative of $D(\mathbf{m^h})$ and the fourth equation in Φ. Let us start by $D(\mathbf{m^h})\,\mathbf{q}^h$. In [2, Sec. 6.1] it is proven that the Newton derivative of this term is given, for any Θ, by $\gamma(\chi_{\mathcal{A}_\mathbf{u}} D(\mathbf{q}^h)\mathbf{N}^h_\mathbf{u}(\mathcal{E}^h\mathbf{u}^h) - g(\Theta))\mathcal{E}^h$, where $\chi_{\mathcal{A}_\mathbf{u}} = D(\mathbf{t}^h)$, with

$$\mathbf{t}^h := \begin{cases} 1 \text{ if } \mathbf{N}(\mathcal{E}\mathbf{u}^h)_i \geq \frac{g(\Theta)}{\gamma} \\ 0 \text{ otherwise.} \end{cases}$$

$\chi_{\mathcal{A}_{\mathbf{u}}}$ is the indicator of the approximated yielded zones ($\mathbf{N}(\mathcal{E}\mathbf{u}^h)_i \geq \frac{g(\Theta)}{\gamma}$), i.e., triangles in which $\chi_{\mathcal{A}_{\mathbf{u}}} = 1$ correspond to yielded regions in the material. Further, $\mathbf{N}_{\mathbf{u}}^h(\mathcal{E}^h\mathbf{u}^h)$ stands for the Jacobian of the discrete norm function $\mathbf{N}^h$, and is given, for a discretized tensor function $\mathbf{q}^h$, by

$$\mathbf{N}_{\mathbf{u}}^h(\mathbf{q}^h) = D(\mathbf{N}^h(\mathbf{q}^h))^{-1} \begin{pmatrix} D(q_1) & D(q_2) & D(q_3) & D(q_4) \\ D(q_1) & D(q_2) & D(q_3) & D(q_4) \\ D(q_1) & D(q_2) & D(q_3) & D(q_4) \\ D(q_1) & D(q_2) & D(q_3) & D(q_4) \end{pmatrix}. \tag{5}$$

Now, we turn to the discretized energy equation. In this case, it is known that the Newton derivatives of $\mu(t)$ and $g(t)$ are given by $M_\mu := \chi_{\mathcal{A}_+^\mu}\chi_{\mathcal{A}_-^\mu}\ell'_\mu(t)$ and $M_g := \chi_{\mathcal{A}_+^g}\chi_{\mathcal{A}_-^g}\ell'_g(t)$, respectively, where

$$\chi_{\mathcal{A}_+^\mu} := \begin{cases} 1 \text{ if } \min(\mu_1, \ell_\mu(t)) \geq \mu_0 \\ 0 \text{ otherwise,} \end{cases} \quad \text{and} \quad \chi_{\mathcal{A}_-^\mu} := \begin{cases} 1 \text{ if } \ell_\mu(t) \leq \mu_1 \\ 0 \text{ otherwise.} \end{cases}$$

$\chi_{\mathcal{A}_+^g}$ and $\chi_{\mathcal{A}_-^g}$ are similarly defined (see [6]).

Summarizing, the SSN step for the operator equation (4) is given by

$$\Psi(\mathbf{u}, \mathbf{q}, p, \Theta)\left(\delta_{\mathbf{u}}, \delta_{\mathbf{q}}, \delta_p, \delta_\Theta\right)^\top = -\Phi(\mathbf{u}_h, \mathbf{q}_h, p_h, \Theta), \tag{6}$$

where the generalized Jacobian is given by

$$\Psi(\mathbf{u}, \mathbf{q}, p, \Theta) := \begin{pmatrix} \mathbf{A}_h & \mathbf{Q}_h & B_h & M_\mu \mathbf{A}_h \mathbf{u}_h \\ \gamma(\chi_{\mathcal{A}_{k+1}}\mathbf{N}_{\mathbf{u}}^h(\mathcal{E}^h\mathbf{u}^h)D(\mathbf{q}^h) - g(\Theta)I)\mathcal{E}^h & D(\mathbf{m}_k^h) & \mathbf{0} & \mathbf{0} \\ (B^h)^\top & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \Psi_{\Theta,\mathbf{u}} & \mathbf{0} & \mathbf{0} & \Psi_{\Theta,\Theta} \end{pmatrix},$$

where $\Psi_{\Theta,\mathbf{u}} := -\mu(\Theta)(\mathbf{u}^h)^\top\mathbf{A}^h - g(\Theta)\mathbf{K}^\top\widehat{\mathbf{N}}_{\mathbf{u}}^h(\mathcal{E}^h\mathbf{u}^h)\mathcal{E}^h$ and $\Psi_{\Theta,\Theta} := \omega_{\alpha,\beta}^h - M_\mu(\mathbf{u}^h)^\top\mathbf{A}^h\mathbf{u}^h - M_g\widehat{\mathbf{N}}^h(\mathcal{E}^h\mathbf{u}_h)$. The matrix $\widehat{\mathbf{N}}_{\mathbf{u}}^h$ is defined analogously as (5).

Theorem 1 *The sequence* $\{\Xi_{k+1} := \Xi_k + \boldsymbol{\delta}_k\}$, *where* $\Xi_k := (\boldsymbol{u}_k, \boldsymbol{q}_{\boldsymbol{k}}, p_k, \Theta_k)\}$ *and* $\boldsymbol{\delta}_k := (\delta_{\boldsymbol{u}}, \delta_{\boldsymbol{q}}, \delta_p, \delta_\Theta)$ *is given by* (6)*, converges superlinearly to the solution of the operator equation* (4)*, provided that* Ξ_0 *is sufficiently close to such solution.*

Proof By extending the results in [2, Lem. 6.3 and 6.4], we can state that the submatrix $\Psi(\mathbf{u}, \mathbf{q}, p, \Theta)(1:3, 1:3)$ is positive definite for a given constant value for Θ. On the other hand, since functions μ and g are Newton differentiable, and since given a vector $\mathbf{u}^h$, the fourth equation in (4) has a unique solution Θ, $\Psi_{\Theta,\mathbf{u}}$ and $\Psi_{\Theta,\Theta}$ does not affect the positive definiteness of the full matrix $\Psi(\mathbf{u}, \mathbf{q}, p, \Theta)$. Therefore, the result follows from [2, Th. 6.5] and the references therein. □

3 Numerical Results

In this section, we present a detailed numerical experiment to show the performance of our numerical approach. We work in the unitary square $\Omega = (0, 1)\times(0, 1)$ and we consider the presence of a body force given by $\mathbf{f}(x_1, x_2) := 300(x_2-0.5, 0.5-x_1)^\top$. This forcing term has non zero curl (curl$\mathbf{f} = -600$), which provokes a rotating movement in the interior of the geometry. We recall that, for the fluid, we consider that non-slip boundary condition holds, i.e., $\mathbf{u} = 0$ on $\partial\Omega$.

For the experiment, we consider $\mu_0 = 1$, $\mu_1 = 2$, $g_0 = 10$ and $g_1 = 15$. Thus, we start with a Bingham fluid with a starting yield limit of 10. In Table 1, we show the reached temperature Θ_f, yield limit g_f, viscosity μ_f and the number of iterations needed by the algorithm, for several values of β with fixed α (up), and with fixed β and changing α (low).

In Fig. 2, the convergence behavior of the algorithm is depicted. There it can be appreciated the superlinear convergence rate, since the residual decays fast in the last iterations.

In Fig. 3, we show the yielded and unyielded regions in the flow for the starting yield limit $g_0 = 10$ and the reached yield limit $g_1 = 11.9753$ in the regime $\alpha = 100$ and $\beta = 0$. The unyielded regions, here depicted as the black regions, corresponds to the portions of the material which are not moving (stagnation zones, which can be seen in the corners) or are moving without plastic deformation i.e. moving as

Table 1 Parameters: $\mu_0 = 1$, $\mu_1 = 2$, $g_0 = 10$, $g_1 = 15$ and $\gamma = 10^3$

$\alpha = 100$	Θ_f	$g_f = g(\Theta_f)$	$\mu_f = \mu(\Theta_f)$	#it.
$\beta = 0$	0.3977	11.9884	1.3977	14
$\beta = 1$	0.3951	11.9753	1.3951	12
$\beta = 10$	0.3731	11.8657	1.3731	14
$\beta = 1$	Θ_f	$g_f = g(\Theta_f)$	$\mu_f = \mu(\Theta_f)$	#it.
$\alpha = 0$	18.8669	15	2	14
$\alpha = 10$	1.7152	15	2	14
$\alpha = 1000$	0.0625	10.3124	1.0625	11

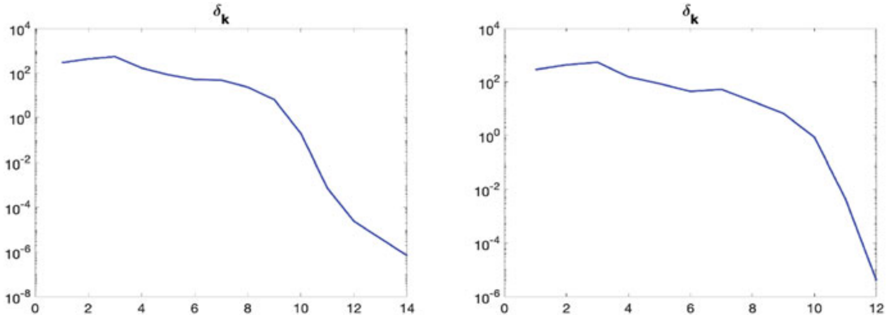


Fig. 2 Convergence behavior of the algorithm for $\alpha = 100$ and $\beta = 0$ (left) and $\beta = 1$ (right). Parameters: $\gamma = 10^3$

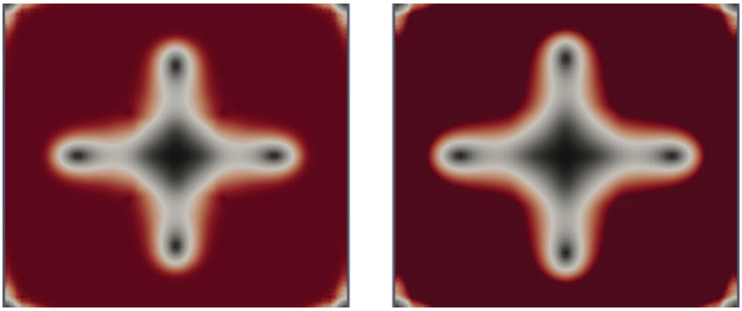


Fig. 3 Yielded (red) and unyielded (black) regions in the flow for $\alpha = 100$ and $g_0 = 0$ (left) and $g_1 = 11.9753$ (right). Parameters: $\gamma = 10^3$

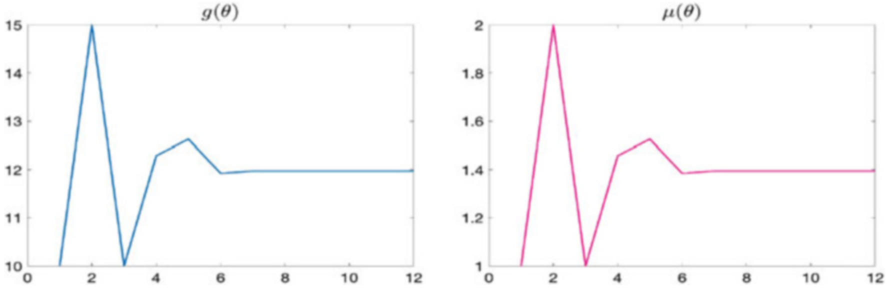


Fig. 4 Evolution of $g(\theta)$ (left) and $\mu(\theta)$ (right) for $\alpha = 100$. Parameters: $\gamma = 10^3$

rigid solids. These last regions are called nucleus, and are the cross-like regions in the center of the geometry. Clearly, as expected, the reached yield limit is bigger, so the material exhibits bigger unyielded zones. Further, these pictures are in good agreement with previous contributions.

Finally, in Fig. 4, we show the evolution of g and μ. We can observe that at first the temperature oscillates, but at the end the functions tend to stabilize, which is in good agreement with the expected behavior of the phenomena, where the temperature stabilizes prior to a change of phase.

Acknowledgments This work was partially supported by the Escuela Politécnica Nacional, under the project PIJ 16-05 “Optimización Matemática del Flujo Dependiente de la Temperatura de Fluidos Viscoplásticos y Elasto-Viscoplásticos”.

References

1. Consiglieri, L. & Rodrigues, J. L.: Steady-state Bingham Flow with Temperature Dependent Nonlocal Parameters and Friction. In: Figueiredo, I. N., Rodrigues, J. F., Santos, L. (eds.) Free Boundary Problems. Theory and Applications, pp. 149–157. Birkhäuser Verlag, Basel (2006).
2. De los Reyes, J. C. & González-Andrade, S.: Numerical Simulation of Two-Dimensional Bingham Fluid Flow by Semismooth Newton Methods. Journal of Computational and Applied Mathematics **235** 11–32 (2010).
3. Duvaut, G. & Lions, J. L.: Transfer de Chaleur dans un Fluide de Bingham dont la Viscosité dépend de la Température. Journal of Functional Analysis **11**, 93–110 (1972).
4. Glowinski, R. & Wachs, A.: On the Numerical Simulation of Viscoplastic Fluid Flow. In R. Glowinski & J. Xu (eds.) Handbook of Numerical Analysis. Numerical Methods for Non-Newtonian Fluids, Vol. XVI, pp 483–718. North-Holland, UK (2011).
5. Messelmi, F., Merouani, B. & Bouzeghaya, F.: Steady-State Thermal Herschel-Bulkley Flow with Tresca's Friction Law. Electronic Journal of Differential Equations **46**, 1–14 (2010).
6. Ulbrich, M.: Semismooth Newton Methods for Operator Equations in Function Spaces. SIAM J. Optim. **13**, 805–841 (2003).

A Sequential Sensor Selection Strategy for Hyper-Parameterized Linear Bayesian Inverse Problems

Nicole Aretz-Nellesen, Peng Chen, Martin A. Grepl, and Karen Veroy

Abstract We consider optimal sensor placement for hyper-parameterized linear Bayesian inverse problems, where the hyper-parameter characterizes nonlinear flexibilities in the forward model, and is considered for a range of possible values. This model variability needs to be taken into account for the experimental design to guarantee that the Bayesian inverse solution is uniformly informative. In this work we link the numerical stability of the maximum a posterior point and A-optimal experimental design to an observability coefficient that directly describes the influence of the chosen sensors. We propose an algorithm that iteratively chooses the sensor locations to improve this coefficient and thereby decrease the eigenvalues of the posterior covariance matrix. This algorithm exploits the structure of the solution manifold in the hyper-parameter domain via a reduced basis surrogate solution for computational efficiency. We illustrate our results with a steady-state thermal conduction problem.

N. Aretz-Nellesen
International Research Training Group Modern Inverse Problems, RWTH Aachen University, Aachen, Germany
e-mail: nellesen@aices.rwth-aachen.de

P. Chen
Oden Institute of Computational Engineering Science, UT Austin, Austin, TX, USA
e-mail: peng@ices.utexas.edu

M. A. Grepl (✉)
Numerical Mathematics (IGPM), RWTH Aachen University, Aachen, Germany
e-mail: grepl@igpm.rwth-aachen.de

K. Veroy
Aachen Institute for Advanced Study in Computational Engineering Science (AICES), RWTH Aachen University, Aachen, Germany
e-mail: veroy@aices.rwth-aachen.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_48

1 Introduction

Mathematical models of physical processes often depend on parameters, such as material properties or source terms, that are known only with some uncertainty. Experimental measurement data can help estimate these parameters and thereby improve the meaningfulness of the model. The Bayesian approach to inverse problems (cf. [7]) yields a (posterior) probability distribution for these parameters that reflects both the prior distribution in the parameters and measurement data.

A major challenge in inverse problems is sensor placement to obtain informative measurement data at restricted experimental cost. There exists a vast optimal experimental design (OED) community focused on different problem types and optimal design criteria. The literature most related to this contribution is the discussion of A-optimality for infinite-dimensional linear Bayesian inverse problems in [1, 2], and the greedy orthogonal matching pursuit algorithm for data assimilation in [4, 6].

In this paper, we consider the optimal placement of sensors to infer a parameter from noisy data in a linear Bayesian inverse problem subject to flexible hyper-parameters. The hyper-parameters characterize variability of the forward model, e.g. variable material properties or geometry, that needs to be taken into account for the sensor placement. For instance, the Bayesian inference problem might need to be solved for multiple data sets with known hyper-parameters, or a most suitable hyper-parameter might need to be sought for fixed data in an "outer loop" optimization. In either case, the same sensors are used within all inference problems, thus necessitating a uniformly "good" choice. The objective of this paper is to provide a sensor selection strategy that follows, uniformly for all hyper-parameters, the A-optimal design criterion of minimizing the trace of the posterior covariance matrix.

In [3], we developed and utilized a numerical stability analysis for parameterized 3D-VAR data assimilation over a linear model correction term to find design criteria for stability-based sensor selection. In this contribution, we first re-interpret these results in the hyper-parameterized linear Bayesian inversion setting, and then show their relation to A-optimal experimental design. This analysis leads to a greedy algorithm that iteratively chooses sensor locations that, under certain assumptions, uniformly decrease the trace of the posterior covariance matrix.

In the upcoming section, we specify our linear forward model, and pose the Bayesian inversion problem in a hyper-parameterized context. We then, in Sect. 3, show the link between its numerical stability, different model coefficients, the eigenvalues of the posterior covariance matrix, and A-optimal experimental design. In Sect. 4 we propose an algorithm to exploit this connection, and present numerical results in Sect. 5 for a thermal conduction problem. We conclude in Sect. 6.

2 A Hyper-Parameterized Bayesian Inverse Problem

We consider a linear Bayesian inverse problem setting for the inference of a finite-dimensional[1] parameter[2] $m \in \mathbb{R}^M$ from noisy data $d \in \mathbb{R}^K$ subject to different hyper-parameters θ that characterize nonlinear (in θ) flexibility in the linear (in m) forward model. Our objective is to find conditions for an observation operator that is uniformly informative for all hyper-parameters. In the following, we specify the forward model and the Bayesian inverse problem, before analysing it in Sect. 3.

Following the Bayesian approach to inverse problems, we consider m to be a random variable, and model our prior belief in its distribution through a non-degenerate Gaussian prior measure $\mu_0 = \mathcal{N}(m_0, \Sigma_0)$ with mean m_0 and symmetric positive-definite (s.p.d.) covariance $\Sigma_0 \in \mathbb{R}^{M \times M}$. We define the inner product $(m_1, m_2)_{\Sigma_0^{-1}} := m_1^T \Sigma_0^{-1} m_2$ and norm $||m_1||^2_{\Sigma_0^{-1}} := (m_1, m_1)_{\Sigma_0^{-1}}$ for $m_1, m_2 \in \mathbb{R}^M$.

For the forward model, let $(\mathcal{U}, (\cdot,\cdot)_{\mathcal{U}})$ be a Hilbert space with induced norm $||u||^2_{\mathcal{U}} := (u, u)_{\mathcal{U}}$, and let $\mathcal{P} \subset \mathbb{R}^p$ be a compact set of possible hyper-parameters. For any $\theta \in \mathcal{P}$, we let $a_\theta : \mathcal{U} \times \mathcal{U} \to \mathbb{R}$ and $b_\theta : \mathbb{R}^M \times \mathcal{U} \to \mathbb{R}$ be non-trivial bilinear forms that are affine[3] and bounded uniformly in θ, with the additional assumption that a_θ is also uniformly coercive.[4] Under these assumptions there exists, for any parameter $m \in \mathbb{R}^M$, a unique, bounded solution to the problem

$$\text{find } u_\theta(m) \in \mathcal{U} \quad \text{s.t.} \quad a_\theta(u_\theta, \psi) = b_\theta(m, \psi) \quad \forall \psi \in \mathcal{U}. \tag{1}$$

We define, for $X \subset \mathbb{R}^M$, the ratios $\underline{\eta}_{X,\theta} := \inf_{m \in X} ||u_\theta(m)||_{\mathcal{U}}/||m||_{\Sigma_0^{-1}} \geq 0$ and $\overline{\eta}_{X,\theta} := \sup_{m \in X} ||u_\theta(m)||_{\mathcal{U}}/||m||_{\Sigma_0^{-1}} < \infty$. Moreover, we define the closed subspace

$$X_\theta := \{m \in \mathbb{R}^M : u_\theta(m) = 0\} = \{m \in \mathbb{R}^M : b_\theta(m, \cdot) = 0\} \subset \mathbb{R}^M \tag{2}$$

of all parameter directions that do not change the state, and let $X_\theta^\perp$ denote its orthogonal complement in the Euclidean inner product. In particular, we have $\underline{\eta}_{X_\theta^\perp,\theta} > 0$.

For our sensors, we consider a library $\mathcal{L} = \{l_k\}_{k=1}^{K_\mathcal{L}}$ of $K_\mathcal{L} < \infty$ sensors $l_k \in \mathcal{U}'$. For a selection $l_{k_1}, \ldots, l_{k_K} \in \mathcal{L}$ of these sensors, we define the observation operator $L = (l_{k_1}, \cdots, l_{k_K})^T : \mathcal{U} \to \mathbb{R}^K$. Measurement data for a parameter

[1] The extension to the infinite-dimensional setting poses additional challenges that will be discussed in a future work.

[2] A general finite-dimensional space can be considered via an affine transformation, c.f. [2, 5].

[3] For conciseness, we refer the reader to [3] for a definition of these properties.

[4] We can readily generalize this setting to non-coercive problems by employing a Petrov-Galerkin formulation. A stability analysis similar to [3] will be explored in a future publication.

$m \in \mathbb{R}^M$ is obtained by applying L to the state $u_\theta(m)$. This gives us the linear, bounded parameter-to-observable map $G_{\theta,L} : \mathbb{R}^M \to \mathbb{R}^K$, $G_{\theta,L}(m) := Lu_\theta(m)$. Our objective is to choose L from $\mathcal{L}$ so that it is approximately A-optimal over $\mathcal{P}$.

For the noise model, we assume to be given an s.p.d. covariance matrix $\Sigma_{\text{noise}} \in \mathbb{R}^{K_\mathcal{L} \times K_\mathcal{L}}$ that describes how the observation noise between all sensors in $\mathcal{L}$ is correlated. The covariance for the sensors in L is then described by the submatrix $\Sigma_L \in \mathbb{R}^{K\times K}$ with $(\Sigma_L)_{i,j} = (\Sigma_{\text{noise}})_{k_i,k_j}$. For fixed L and for data $d_1, d_2 \in \mathbb{R}^K$ we define the inner product $(d_1, d_2)_{\Sigma_L^{-1}} := d_1^T \Sigma_L^{-1} d_2$ and induced norm $||d_1||^2_{\Sigma_L^{-1}} := (d_1, d_1)_{\Sigma_L^{-1}}$. We define $\gamma_L := \sup_{u\in\mathcal{U}} ||Lu||_{\Sigma_L^{-1}}/||u||_\mathcal{U}$ as the norm of L. We model the data to be of the form

$$d = G_{\theta,L}(m) + \eta \quad \text{with Gaussian additive noise} \quad \eta \sim \mathcal{N}(0, \sigma^2\Sigma_L) \tag{3}$$

and scaling parameter $\sigma > 0$. For given data $d \in \mathbb{R}^K$ from an observation operator L, the posterior probability density function of the posterior measure $\mu^{L,d}$ is then given through Bayes' theorem by

$$\pi_{\text{post}}(m|d) \propto \exp\left(-\tfrac{1}{2\sigma^2}||G_{\theta,L}(m) - d||^2_{\Sigma_L^{-1}} - \tfrac{1}{2}||m - m_0||^2_{\Sigma_0^{-1}}\right), \tag{4}$$

where we omit the normalization constant $Z = \int_{\mathbb{R}^M} \exp(-\frac{1}{2\sigma^2}||G_{\theta,L}(m) - d||^2_{\Sigma_L^{-1}}) d\mu_0$. Since $G_{\theta,L}$ is linear, the posterior is a Gaussian (see, e.g., [7, Thm 2.4]), $\mu^{L,d} = \mathcal{N}(m^{\theta,L}_{\text{post}}(d), \Sigma^{\theta,L}_{\text{post}})$, with mean $m^{\theta,L}_{\text{post}}(d) = \Sigma^{\theta,L}_{\text{post}}\left(\frac{1}{\sigma^2} G^*_{\theta,L}\Sigma_L^{-1} d + \Sigma_0^{-1} m_0\right)$ and covariance matrix $\Sigma^{\theta,L}_{\text{post}} = \left(\frac{1}{\sigma^2} G^*_{\theta,L}\Sigma_L^{-1} G_{\theta,L} + \Sigma_0^{-1}\right)^{-1}$.

3 Numerical Stability and A-Optimal Experimental Design

In the following, we first comment on the connection between the numerical stability of the MAP point and the observation operator L. We then link this analysis to A-optimal experimental design.

Since $\mu^{L,d}$ is Gaussian, its mean $m^{\theta,L}_{\text{post}}$ is the maximum a posteriori (MAP) point, and hence the solution to the minimization problem

$$\min_{m\in\mathbb{R}^M} \tfrac{1}{2\sigma^2}||Lu_\theta(m) - d||^2_{\Sigma_L^{-1}} + \tfrac{1}{2}||m - m_0||^2_{\Sigma_0^{-1}}. \tag{5}$$

Through a reformulation as a saddle-point problem, the numerical stability of (5) can be analyzed with respect to θ, L, and σ^2 (c.f. [3] for an analogous analysis). In particular, the difference in the MAP points and states for different $d_1, d_2 \in \mathbb{R}^K$ is bounded by the difference in data. We have, with $\tilde{m}(d) := m_{\text{post}}^{\theta,L}(d)$ for readability,

$$||\tilde{m}(d_1) - \tilde{m}(d_2)||_{\Sigma_0^{-1}}^2 + ||u_\theta(\tilde{m}(d_1)) - u_\theta(\tilde{m}(d_2))||_{\mathcal{U}}^2 \le (C_{\theta,L,\sigma^2})^2 ||d_1 - d_2||_{\Sigma_L^{-1}}^2. \tag{6}$$

The stability coefficient $C_{\theta,L,\sigma^2} > 0$ quantifies the influence of noise on the MAP point. It has the form $C_{\theta,L,\sigma^2} = \gamma_L(1 + \eta^2)/(\sigma^2 + \beta_{\theta,L}^2\eta^2)$, where $\eta = \overline{\eta}_{\mathbb{R}^M,\theta}$ if $\beta_{\theta,L}^2 \le \sigma^2$, and $\eta = \overline{\eta}_{\mathbb{R}^M,\theta}$ otherwise, and $\beta_{\theta,L}$ is the observability coefficient

$$\beta_{\theta,L} := \inf\{||Lu_\theta(m)||_{\Sigma_L^{-1}} :\ ||u_\theta(m)||_{\mathcal{U}} = 1,\ m \in \mathbb{R}^M\}. \tag{7}$$

C_{θ,L,σ^2} decreases in $\beta_{\theta,L}$, and remains bounded for $\sigma^2 \to 0$ iff $\underline{\eta}_{\mathbb{R}^M,\theta} > 0$ and $\beta_{\theta,L} > 0$. Increasing $\beta_{\theta,L}$ can hence help improve robustness of $m_{\text{post}}^{\theta,L}$ against noise.

The goal in A-optimal experimental design is to choose sensors to minimize the trace of the posterior covariance matrix $\Sigma_{\text{post}}^{\theta,L}$. Geometrically, this corresponds to minimizing the mean axis of the uncertainty ellipsoid (c.f. [8]). In the following, we bound the eigenvalues of $\Sigma_{\text{post}}^{\theta,L}$ via $\beta_{\theta,L}$ and $\underline{\eta}_{X_\theta^\perp,\theta}$. These can then be used to choose sensors to decrease the bounds of the eigenvalues, and consequently of the trace.

Utilizing the definitions of $G_{\theta,L}$ and $\underline{\eta}_{m_{\lambda_i},\theta}$ in Sect. 2, and $\beta_{\theta,L}$ in (7), we observe

$$\begin{aligned} m^T G_{\theta,L}^* \Sigma_L^{-1} G_{\theta,L} m = ||Lu_\theta(m)||_{\Sigma_L^{-1}}^2 &\ge \beta_{\theta,L}^2 ||u_\theta(m)||_{\mathcal{U}}^2 = \beta_{\theta,L}^2 ||u_\theta(\Pi_{X_\theta^\perp} m)||_{\mathcal{U}}^2 \\ &\ge \beta_{\theta,L}^2 \underline{\eta}_{X_\theta^\perp,\theta}^2 ||\Pi_{X_\theta^\perp} m||_{\Sigma_0^{-1}}^2 \ge \beta_{\theta,L}^2 \underline{\eta}_{X_\theta^\perp,\theta}^2 C_{\Sigma_0^{-1}}^{-2} ||\Pi_{X_\theta^\perp} m||_{\mathbb{R}^M}^2, \end{aligned} \tag{8}$$

where $\Pi_{X_\theta^\perp}$ is the orthogonal projection onto $X_\theta^\perp$ in the Euclidean inner product and $C_{\Sigma_0^{-1}} := \sup_{m\in\mathbb{R}^M} ||m||_{\mathbb{R}^M}/||m||_{\Sigma_0^{-1}}$ is a norm equivalence constant.

Let $0 < \lambda_1 \le \cdots \le \lambda_M$ be the eigenvalues of the posterior covariance matrix $\Sigma_{\text{post}}^{\theta,L}$, including duplicates. Since $\Sigma_{\text{post}}^{\theta,L}$ is s.p.d., there exists an orthonormal eigenvector basis $(m_{\lambda_i})_{i=1}^M$ of $\mathbb{R}^M$, i.e. $m_{\lambda_i}^T m_{\lambda_j} = \delta_{i,j}$ and $\Sigma_{\text{post}}^{\theta,L} m_{\lambda_i} = \lambda_i m_{\lambda_i}$. With the explicit formula for $\Sigma_{\text{post}}^{\theta,L}$ from Sect. 2, the last equation is equivalent to $\frac{1}{\lambda_i} m_{\lambda_i} = \frac{1}{\sigma^2} G_{\theta,L}^* \Sigma_L^{-1} G_{\theta,L} m_{\lambda_i} + \Sigma_0^{-1} m_{\lambda_i}$. Premultiplying by $m_{\lambda_i}^T$ and inserting (8) yields $\frac{1}{\lambda_i} = \frac{1}{\lambda_i} ||m_{\lambda_i}||_{\mathbb{R}^M}^2 \ge (\frac{1}{\sigma^2} \beta_{\theta,L}^2 \underline{\eta}_{X_\theta^\perp,\theta}^2 C_{\Sigma_0^{-1}}^{-2} ||\Pi_{X_\theta^\perp} m_{\lambda_i}||_{\mathbb{R}^M}^2 + C_{\Sigma_0^{-1}}^{-2})$, and hence

$$\lambda_i \le C_{\Sigma_0^{-1}}^2 / (\tfrac{1}{\sigma^2} \beta_{\theta,L}^2 \underline{\eta}_{X_\theta^\perp,\theta}^2 ||\Pi_{X_\theta^\perp} m_{\lambda_i}||_{\mathbb{R}^M}^2 + 1).$$

Summing over all eigenvalues, including duplicates, we can now bound

$$\mathrm{trace}(\Sigma_{\mathrm{post}}^{\theta,L}) = \textstyle\sum_{i=1}^{M} \lambda_i \leq C^2_{\Sigma_0^{-1}} \sum_{i=1}^{M} (\frac{1}{\sigma^2} \beta^2_{\theta,L} \underline{\eta}^2_{X_\theta^\perp,\theta} ||\Pi_{X_\theta^\perp} m_{\lambda_i}||^2_{\mathbb{R}^M} + 1)^{-1}.$$

Although $||\Pi_{X_\theta^\perp} m_{\lambda_i}||_{\mathbb{R}^M} \geq 0$ is unknown for any individual λ_i, by exploiting that $(m_{\lambda_i})_{i=1}^M$ is an orthonormal basis, it can be shown that $\sum_{i=1}^{M} ||\Pi_{X_\theta^\perp} m_{\lambda_i}||^2_{\mathbb{R}^M} = \dim X_\theta^\perp$. Our strategy is to choose L to increase $\beta_{\theta,L}$; this decreases the bound for each λ_i with $||\Pi_{X_\theta^\perp} m_{\lambda_i}||_{\mathbb{R}^M} > 0$, and hence also the bound for the trace. The coefficient $\beta_{\theta,L}$ becomes more influential the more the data is trusted, i.e. for σ^2 small.

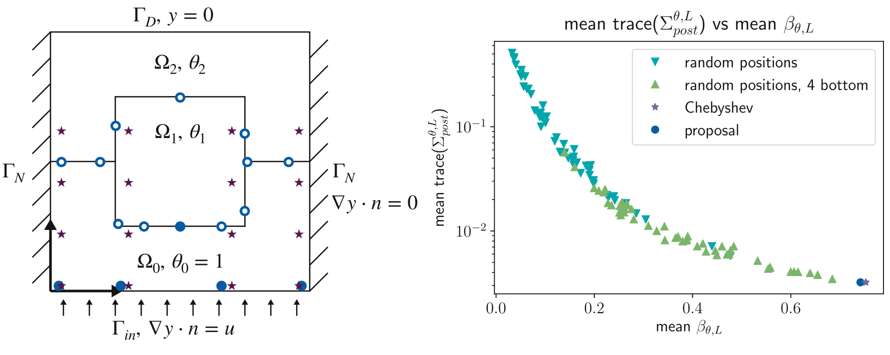
4 Sensor Selection Strategy

Our goal is to choose sensors $\{l_k\}_{k=1}^K$ so that $\beta_{\theta,L}$ is uniformly large over the hyper-parameter domain $\mathcal{P}$. The major challenge for achieving this goal is that evaluating $\beta_{\theta,L}$ for any θ involves solving the forward problem (1) for each basis vector of $\mathbb{R}^M$. We address this problem by approximating the solution $u_\theta(m)$ of (1) with a surrogate reduced basis (RB) solution $u_{\theta,\mathrm{R}}(m)$, that can be computed at a considerably reduced computational cost for a specified accuracy: Suppose for every hyper-parameter $\theta \in \mathcal{P}$ and every parameter $m \in \mathbb{R}^M$, we can compute $u_{\theta,\mathrm{R}}(m) \in \mathcal{U}$ such that $||u_\theta(m) - u_{\theta,\mathrm{R}}(m)||_{\mathcal{U}} \leq \varepsilon_\theta ||u_\theta(m)||_{\mathcal{U}}$ for a relative accuracy $0 \leq \varepsilon_\theta \leq \varepsilon < 1$. It can then be shown analogously to [3, sec. 5.1] that $\beta_{\theta,L} \geq (1 - \varepsilon_\theta)\beta_{\theta,L,\mathrm{R}} - \gamma_L \varepsilon_\theta$, where $\beta_{\theta,L,\mathrm{R}}$ is defined over the surrogate model analogously to (7), and γ_L is the norm of L. The upper bound $\varepsilon < 1$ ensures that $u_{\theta,\mathrm{R}}(m) = 0$ iff $u_\theta(m) = 0$; therefore $\beta_{\theta,L,\mathrm{R}}$ is defined over the same parameter subspace $X_\theta^\perp$ as $\beta_{\theta,L}$. Supposing ε is small enough, we propose to exploit the lower bound of $\beta_{\theta,L}$ by choosing the sensors in L via an iterative greedy approach over $\mathcal{P}$ to increase $\beta_{\theta,L,\mathrm{R}}$, and subsequently $\beta_{\theta,L}$.

Algorithm 1 Stability-based sensor selection

Given: a training set Ξ_{train}, a library $\mathcal{L}$, a target value β_0, a starting parameter $\theta_1 \in \Xi_{\mathrm{train}}$, and an upper limit $K_{\max}$ to the number of sensors.

```
$L \leftarrow \{0\}$, $\beta \leftarrow 0$, $K \leftarrow 0$
while $\beta < \beta_0$ and $K < K_{\max}$ do
    $u_{K+1} \leftarrow \arg\min\{||Lu_{\mathrm{R}}||_{\Sigma_L^{-1}}/||u_{\mathrm{R}}||_{\mathcal{U}} : \; u_{\mathrm{R}} = u_{\theta_{K+1},\mathrm{R}}(m)$ for $m \in \mathbb{R}^M\}$
    choose $l \in \mathcal{L}$ such that $||[L, l]u_{K+1}||_{\Sigma_{[L,l]}^{-1}}$ is maximal
    $L \leftarrow [L, l]$, $K \leftarrow K + 1$
    $\theta_{K+1} \leftarrow \arg\min_{\theta \in \Xi_{\mathrm{train}}} \beta_{\theta,L,\mathrm{R}}$, $\beta \leftarrow \beta_{L,\mathrm{R}}(\theta_{K+1})$
end while
```



(a)

(b)

Fig. 1 (**a**) Domain decomposition of the thermal block problem with boundary conditions. Sensor centres chosen by Algorithm 1 are indicated as circles (filled out for $\beta_{\theta,L}$-criterion), reference Chebyshev centres are marked with stars. (**b**) Mean of the trace of the posterior covariance matrix vs. the mean of $\beta_{\theta,L}$. Mean values are computed over the 1681 hyperparameters in Ξ_{test}

Following the ideas in [4, 6], in each iteration of the loop, the algorithm first chooses (line (3)) the state u_{K+1} which realizes the minimum observability coefficient; it then searches the library $\mathcal{L}$ for the best sensor to observe this state (line (4)), and then extend the observation operator. Line (4) involves first computing $l(u_{K+1})$ (in FE dimension) and then $||[L, l]u_{K+1}||_{\Sigma^{-1}_{[L,l]}}$ (in $\mathcal{O}(K^3)$) for each $l \in \mathcal{L}$. In line (6) the algorithm then finds the hyper-parameter by iterating over $\theta \in \Xi_{\text{train}}$ and computing $\beta_{\theta,L,\text{R}}$ via an eigenvalue problem. Any computation of $\beta_{\theta,L,\text{R}}$ involves solving the RB problem for each basis vector of $\mathbb{R}^M$. The algorithm terminates when either a maximum number of sensors or a target value [5] β_0 has been reached.

The uniform increase of $\beta_{\theta,L,\text{R}}$ over $\mathcal{P}$ relies on the property that extending the observation operator with a sensor does not decrease $\beta_{\theta,L,\text{R}}$ at different hyper-parameters. This property is straightforward to prove for uncorrelated noise, but more involved for the general case. We will explore this aspect in a future publication.

5 Numerical Results

We consider a steady-state heat conduction problem $-\theta\Delta u = 0$ over the unit square Ω. The hyper-parameters $\theta_i \in [0.1, 10]$ specify different thermal conductivities on three subdomains (overview in Fig. 1a). We impose an uncertain inflow boundary

[5] Possibilities for target values are highly dependent on the library $\mathcal{L}$. In practice, β_0 should be chosen by carefully monitoring the changes in β.

condition $u = \sum_{i=0}^{3} m_i p_i$ on Γ_{in}, where p_i is the Legendre polynomial of degree i, and m is distributed as $\mathcal{N}((1, 0, 0, 0)^T, I)$ in $\mathbb{R}^M = \mathbb{R}^4$, with identity matrix I. We refer to [3] for a full description of the model problem and algorithm implementation.

Our library consists of functionals $l_k(u) := \int_\Omega g_k(x)u(x)dx$, where the g_k are Gaussian functions with standard deviation 0.01 and centres in a 97 × 97 regular grid on $[0.02, 0.98]^2$. We model Σ_{noise} as the $\mathcal{U}$-inner-product of the sensors' Riesz representations, and choose $\sigma = 0.01$. The noise correlation at different sensors is then higher the closer they are placed to each other. We apply Algorithm 1 with target value $\beta_0 := 0.5$. In addition to $\beta_{\theta,L}$, we also consider $\tilde{\beta}_{L,\theta_1,\theta_2} := \inf\{||Lu||_{\Sigma_L^{-1}}/||u||_{\mathcal{U}} :\ u = u_{\theta_1}(m_1) + u_{\theta_2}(m_2),\ m_1, m_2 \in \mathbb{R}^M\}$, which distinguishes between different hyper-parameters, and selects sensors for "outer-loop" hyper-parameter estimation (c.f. discussion in [3]). The 16 selected sensors are indicated in Fig. 1a as circles.

We compute the mean trace of the posterior covariance matrix and the mean of $\beta_{\theta,L}$ over a testing set Ξ_{test} of 41 × 41 hyper-parameters located in a regular grid on the logarithmic plane of $\mathcal{P} = (0.1, 10)^2$. For comparison, we repeat the process 50 times with 16 randomly chosen positions, and another 50 times with 16 randomly chosen positions of which at least 4 are placed close to the inflow boundary with $x_2 = 0.02$. Also, we consider another set of 16 centers (indicated as stars in Fig. 1a), where the x_1-locations are chosen in our library closest to the Chebyshev interpolation points for polynomials with degree smaller or equal to 3. These would be the theoretically optimal points (in x_1-direction) for interpolating the Neumann flux. Figure 1b shows the mean trace and the mean $\beta_{\theta,L}$ over Ξ_{test} for the different sensor sets.

We observe that the sensors chosen by Algorithm 1 near the inflow boundary are close to the Chebyshev positions. Both sensor sets have a similarly high mean value for $\beta_{\theta,L}$ and a similarly small trace of the posterior covariance. In contrast, the randomly chosen sensor sets have a larger mean trace and smaller mean $\beta_{\theta,L}$. Here the sets chosen with four centres near the inflow boundary outperform the completely random ones. Altogether, we observe a strong correlation between $\beta_{\theta,L}$ and $\text{trace}(\Sigma_{\text{post}}^{\theta,L})$.

6 Conclusion

In this paper we considered a hyper-parameterized linear Bayesian inverse problem and linked its numerical stability analysis to A-optimal experimental design. This analysis permits the development of an algorithm that iteratively chooses sensor locations from a library to reduce the eigenvalues of the posterior covariance matrix uniformly over the hyper-parameter domain. Future work will extend this analysis to Petrov-Galerkin and time-dependent formulations, as well as application to high-dimensional parameter spaces and nonlinear problems.

Acknowledgments We would like to thank Tan Bui-Thanh, Youssef Marzouk, Francesco Silva, Andrew Stuart, Dariusz Ucinski, and Keyi Wu for very helpful discussions. This work was supported by the Excellence Initiative of the German federal and state governments and the German Research Foundation through Grants GSC 111 and 33849990/GRK2379 (IRTG Modern Inverse Problems).

References

1. Alen Alexanderian, Noemi Petra, Georg Stadler, and Omar Ghattas. A-Optimal Design of Experiments for Infinite-Dimensional Bayesian Linear Inverse Problems with Regularized l0-Sparsification. *SIAM Journal on Scientific Computing*, 36(5):A2122–A2148, 2014.
2. Alen Alexanderian, Philip J Gloor, Omar Ghattas, et al. On Bayesian A-and D-optimal experimental designs in infinite dimensions. *Bayesian Analysis*, 11(3):671–695, 2016.
3. Nicole Aretz-Nellesen, Martin A. Grepl, and Karen Veroy. 3D-VAR for parameterized partial differential equations: a certified reduced basis approach. *Advances in Computational Mathematics*, 2019.
4. Peter Binev, Albert Cohen, Olga Mula, and James Nichols. Greedy algorithms for optimal measurements selection in state estimation using reduced models. *SIAM/ASA Journal on Uncertainty Quantification*, 6(3):1101–1126, 2018.
5. Giuseppe Da Prato. *An introduction to infinite-dimensional analysis*. Springer Science & Business Media, 2006.
6. Yvon Maday, Anthony T. Patera, James D. Penn, and Masayuki Yano. A parameterized-background data-weak approach to variational data assimilation: formulation, analysis, and application to acoustics. *International Journal for Numerical Methods in Engineering*, 102(5):933–965, 2015.
7. Andrew M Stuart. Inverse problems: A Bayesian perspective. *Acta numerica*, 19:451–559, 2010.
8. Dariusz Ucinski. *Optimal measurement methods for distributed parameter system identification*. CRC Press, 2004.

Biomechanical Surrogate Modelling Using Stabilized Vectorial Greedy Kernel Methods

Bernard Haasdonk, Tizian Wenzel, Gabriele Santin, and Syn Schmitt

Abstract Greedy kernel approximation algorithms are successful techniques for sparse and accurate data-based modelling and function approximation. Based on a recent idea of stabilization (Wenzel et al., A novel class of stabilized greedy kernel approximation algorithms: convergence, stability & uniform point distribution. e-prints. arXiv:1911.04352, 2019) of such algorithms in the scalar output case, we here consider the vectorial extension built on VKOGA (Wirtz and Haasdonk, Dolomites Res Notes Approx 6:83–100, 2013. We introduce the so called γ-restricted VKOGA, comment on analytical properties and present numerical evaluation on data from a clinically relevant application, the modelling of the human spine. The experiments show that the new stabilized algorithms result in improved accuracy and stability over the non-stabilized algorithms.

1 Introduction

Kernel methods are used in various fields of machine learning or pattern analysis. They yield efficient and flexible ways to recover functions from data since they can deal with arbitrarily scattered points. The combination of their flexibility with the

B. Haasdonk (✉) · T. Wenzel
Institute for Applied Analysis and Numerical Simulation, University of Stuttgart, Stuttgart, Germany
e-mail: haasdonk@mathematik.uni-stuttgart.de; tizian.wenzel@mathematik.uni-stuttgart.de

G. Santin
Center for Information and Communication Technology, Fondazione Bruno Kessler, Trento, Italy
e-mail: gsantin@fbk.eu

S. Schmitt
Institute for Modelling and Simulation of Biomechanical Systems, University of Stuttgart, Stuttgart, Germany
e-mail: schmitt@simtech.uni-stuttgart.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_49

strong mathematical theory about e.g. existence, convergence, stability make them a nice tool for applications [3, 10].

In this paper we apply a recently introduced idea that has lead to a new class of stabilized greedy kernel algorithms [11], extend it to vectorial function approximation and apply it to a real life setting from research in biomechanics. Some theoretic statements can be extended from the scalar to the vectorial case. All in all these stabilized methods provide further flexibility and are able to efficiently mitigate the problem of having numerical instabilities.

The paper is organized as follows. To begin with we recall in Sect. 2 some basics about kernel interpolation with a focus on greedy kernel approximation and explain the stabilized extension. Section 3 gives background information about our application settings and the use of kernel methods. The following Sect. 4 explains the conducted numerical experiments as well as the practical results. Section 5 concludes with a summary and an outlook.

2 Stabilized VKOGA Algorithm

We start with a nonempty set $\Omega \subset \mathbb{R}^d$. A real-valued kernel is a symmetric function $k : \Omega \times \Omega \to \mathbb{R}$. For arbitrary points $X_N := \{x_1, .., x_N\} \subset \Omega$ the kernel matrix $A \in \mathbb{R}^{N \times N}$ is a symmetric matrix with entries $A_{ij} = k(x_i, x_j)$. If this kernel matrix is positive semi-definite for any set of points $X_N \subset \Omega$, then the kernel is called positive definite. If the kernel matrix is even positive definite for any set of pairwise distinct points, then the kernel is called strictly positive definite. In the following we will focus on this case of strictly positive definite kernels and we refer to the monographs [3, 10] for more details.

For any such kernel there is a unique Hilbert space of functions, namely the native space $(\mathcal{H}_k(\Omega), (\cdot, \cdot)_{\mathcal{H}_k(\Omega)})$, which is a Reproducing Kernel Hilbert Space (RKHS). A popular choice is given by radial basis function kernels, i.e. the kernel can be expressed with the help of some function Φ and a kernel parameter $\epsilon \in \mathbb{R}$ as $k(x, y) = \Phi(\epsilon \|x - y\|)$. Examples are given by the Gaussian kernel $\Phi_{\text{Gauss}}(r) = \exp(-(\epsilon \cdot r)^2)$and the linear Matérn kernel $\Phi(r) = (1 + r) \cdot \exp(-r)$. The decay of the Fourier transform of those radial basis functions is decisive for their properties. The Fourier transform of the Gaussian decays exponentially, whereas the Fourier transform of the linear Matérn decays only algebraically.

In such RKHS the interpolation of functions—or more general data based approximation tasks—can be analyzed. For a given function $f \in \mathcal{H}_k(\Omega)$ and some interpolation points X_N the interpolant s_N is given by the orthogonal projection $\Pi_{V(X_N)}(f)$ of f onto $V(X_N) := \{k(\cdot, x_i), x_i \in X_N\}$ and thus can be expressed as

$$s_N(\cdot) = \Pi_{V(X_N)}(f) = \sum_{i=1}^{N} \alpha_i k(\cdot, x_i), \quad \alpha_i \in \mathbb{R}, 1 \leq i \leq N.$$

In some applications the data is affected by noise, so it does not make sense to interpolate the given values, while it is rather advisable to approximate them while taking some regularization into account. For this one can consider minimizing $\sum_{i=1}^{N} \|f(x_i) - s_N(x_i)\|_2^2 + \lambda \cdot \|s_N\|^2_{\mathcal{H}_k(\Omega)}$ which corresponds to solving the linear system

$$(A + \lambda \cdot I)\alpha = y \tag{1}$$

with $y = (f(x_i))_{i=1}^N$. To measure the interpolation error $\|f - \Pi_{V(X_N)}(f)\|_{L^\infty}$ one can introduce the Power function $P_{X_N} : \Omega \to \mathbb{R}$ as

$$P_N(x) := P_{X_N}(x) = \sup_{0 \neq f \in \mathcal{H}_k(\Omega)} \frac{|f(x) - \Pi_{V(X_N)}(f)(x)|}{\|f\|_{\mathcal{H}_k(\Omega)}}. \tag{2}$$

From this definition we can directly conclude

$$|f(x) - \Pi_{V(X_N)}(f)(x)| \leq P_N(x) \cdot \|f\|_{\mathcal{H}_k(\Omega)}.$$

For the analysis of the kernel interpolants geometric quantities about the distribution of the interpolation points are important. The fill distance h_N and the separation distance q_N are defined as

$$h_N := \sup_{x \in \Omega} \min_{x_i \in X_N} \|x - x_i\|_2, \qquad q_N := \min_{x_i \neq x_j \in X_N} \|x_i - x_j\|_2. \tag{3}$$

A priori it is unclear how to select good interpolation points for a given set of data or some functions. To circumvent this, one applies greedy methods which start with an empty set $X_0 = \{\}$ and iteratively add another interpolation point according to some selection criterion, $X_{N+1} = X_N \cup \{x_{N+1}\}$.

There are three main selection criteria in the literature, namely f-greedy, f/P-greedy and P-greedy [1, 6, 9] which choose the next point from Ω according to some indicator. For the vectorial case [12], for $x \in \Omega$ they are:

1. f-greedy: $\eta_f^{(N)}(x) = \|f(x) - \Pi_{V(X_N)}(f)(x)\|_2$
2. P-greedy: $\eta_P^{(N)}(x) = P_{X_N}(x)$
3. f/P-greedy: $\eta_{f/P}^{(N)}(x) = \|f(x) - \Pi_{V(X_N)}(f)(x)\|_2 / P_{X_N}(x)$.

In order to create a scale of selection criteria which lie in between those known criteria, one introduces a restriction parameter $\gamma \in (0, 1]$ and a restricted set $\Omega_\gamma^{(N)} := \{x \in \Omega, P_N(x) \geq \gamma \cdot \|P_N\|_\infty\}$ and chooses the next interpolation point within $\Omega_\gamma^{(N)}$ according to some standard selection criterion. This works since the Power function is scalar valued and the interpolation points are shared among all dimensions.[1] For $\gamma = 1$ it holds $\Omega_\gamma^{(N)} = \{x \in \Omega,\ P_N(x) = \|P_N\|_\infty\}$, thus we obtain the standard P-greedy algorithm for any selection criterion $\eta^{(N)}(x)$. For $\gamma = 0$ it holds $\Omega_\gamma^{(N)} = \Omega$, thus we obtain the unrestricted algorithm. The naming *restricted* is obviously related to the restriction of the set Ω to $\Omega_\gamma^{(N)}$, the name *stabilized* is motivated by part of the results within [11], which are summarized in Theorem 1. If the maximum within a selection rule is not unique, any point realizing the maximum can be picked.

As an example, the γ-stabilized f-greedy chooses the next point according to

$$x_{N+1} = \underset{x\in\Omega_\gamma^{(N)}}{\arg\max}\, \|f(x) - \Pi_{V(X_N)}(f)(x)\|_2.$$

Several rigorous analytical statements can be derived for this kind of algorithms and we will summarize a few of them in the following Theorem 1. The proofs are straightforward consequences of those which can be found in [11].

Theorem 1 *Assume that $\Omega \subset \mathbb{R}^d$ is a compact domain which satisfies an interior cone condition and has a Lipschitz boundary. Suppose that k is a translational invariant kernel such that its native space is norm equivalent to the Sobolev space $H^\tau(\Omega)$ with $\tau > d/2$. Then any γ-stabilized algorithm applied to a function in $f \in \mathcal{H}_k(\Omega)$ gives a sequence of point sets $X_N \subset \Omega$ such that it holds:*

- *Lower and upper bound on the Power function:*

$$c_P \cdot N^{\frac{1}{2}-\frac{\tau}{d}} \le \|P_N\|_{L^\infty(\Omega)} \le C_P \cdot \gamma^{-2} \cdot N^{\frac{1}{2}-\frac{\tau}{d}}.$$

- *Asymptotic uniform point distribution:*

$$\rho_{X_N} := \frac{h_N}{q_N} \le c \cdot \gamma^{-4} \ \ \forall N \in \mathbb{N}.$$

- *Lower and upper bounds on the smallest eigenvalue:*

$$c_1 \cdot \gamma^{8\tau-4d-4} \cdot N^{1-2\tau/d} \le \lambda_{\min}(X_N) \le c_2 \cdot \gamma^{-4} \cdot N^{1-2\tau/d}.$$

Finally we want to point out that the γ-parameter only affects the choice of points, i.e. it modifies the greedy selection. However, if given points are used the γ parameter does not change the computed interpolant anymore. This is in contrast to the parameters λ and ϵ, which modify also the interpolant if points are given.

From Theorem 1 we can conclude that the product $\lambda_{\min} \cdot \|P_N\|_\infty^2$ is bounded from both sides for $\gamma > 0$. This motivates to take the value of the Power function of

[1] This corresponds to the case of using separable matrix-valued kernels, i.e. $K(x, y) := k(x, y) \cdot I$ where I is the $d \times d$ identity matrix [14].

the previously chosen point as a measure of stability. This will be used in Eq. (4) to implement a stopping criterion based on stability.

3 Application to Spine Modelling

Biomechanical models of the human spine account for the most significant structures which carry the load of daily life. That are mostly the ligaments, the muscles with both passive and active contributions and the intervertebral discs (IVDs), of course. An IVD, in this sense, can be seen as a combination of both the defining structure for the degrees of freedom between two vertebral bodies and the force and rotational moment transducing elements between these two bones, cf. Fig. 1.

Mostly, IVDs are modelled by a linear approximation of forces and rotational moments calculated according to the respective displacements, e.g. [5]. Alternatively, as long as quasi-static movements are studied, very detailed, finite element models of isolated IVDs or a combination of few spinal segments are used [2]. Another approach to model a reduced IVD used a polynomial approximation and showed that the classical linear approximations overestimate actual stiffnesses in the working range [4, 7]. This observation gave rise to the idea of looking into an even more sophisticated mapping of displacements on the input to output forces and rotational moments.

Obviously, kernel modelling seems to be an ideal approach for this need. First, kernel surrogates promise to capture the mapping characteristics well, second, extensions to higher input and output dimensions seem feasible and third, compared to respective detailed finite element models surrogate models evaluate the mapping stunningly fast [13].

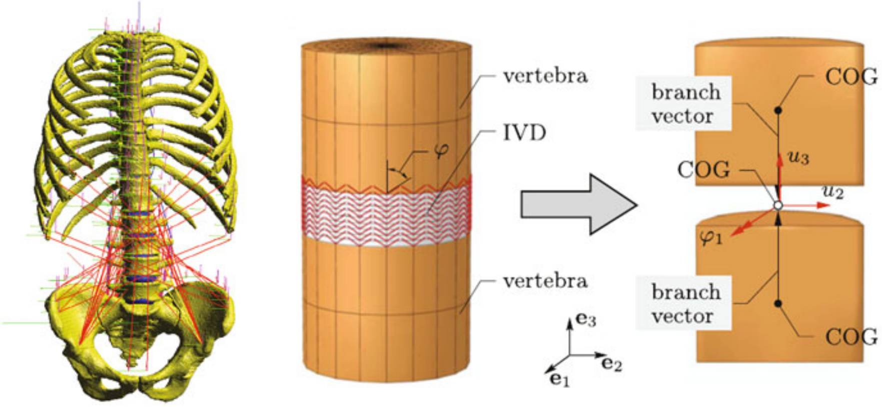


Fig. 1 Visualization of the biomechanical model. On the left the whole spine model is depicted, on the right the modelling scheme of an IVD reduced to a 3-d force/torque element is shown [4]

Assuming symmetry in the sagittal and frontal plane, an input-output relation $f : \mathbb{R}^3 \to \mathbb{R}^3$ is considered and studied, here.

4 Numerical Experiments

The considered dataset consists of 1370 input points in $\mathbb{R}^3$ with corresponding output points in $\mathbb{R}^3$. 1238 points are used for training and validation, the remaining points are used as a test set. No scaling is applied to the data. In order to show the flexibility and thus improved accuracy of the stabilized algorithms on the presented data set, we compute unstabilized approximants, used as base models, as well as stabilized models for the f- and the f/P-greedy algorithm. For both the unstabilized and stabilized models we also use regularization in a second step. The experiments are related to those in [8, 13], however due to different setups they are not identical.

The base models are given by standard kernel surrogates where the point selection is done either with vectorial f-greedy or f/P-greedy. To evaluate good parameters, first of all a 5-fold cross validation is run on 20 logarithmic equally spaced kernel parameters ϵ. The best ϵ value is used for a second step, where the best λ parameter from Eq. (1) is evaluated with help of another 5-fold cross validation. For this we use 20 logarithmic equally spaced values between 10^{-16} and 10^3. As an error measure we use the Root Mean Square Error (RMSE)

$$E_{\text{RMSE}}(s, X, Y) = \left(\frac{1}{|X|} \cdot \sum_{i=1}^{|X|} \|s(x_i) - y_i\|_2^2 \right)^{1/2} .$$

The stabilized models are given by kernel surrogates where the points are selected with help of stabilized vectorial f- or f/P-greedy algorithms. We start by using the same kernel parameter ϵ which was selected for the base model and run instead a 5-fold cross validation on 11 equally spaced stabilization parameters $\gamma \in [0, 1]$. As a second step we evaluate again the best λ parameter with help of a 5-fold cross validation. This procedure keeps the computation time similar to the base model.

The used hyperparameters are summarized in Table 1. For the experiments the linear Matérn kernel is used since it satisfies all the prerequisites of Theorem 1.

The greedy selection algorithms stop either when all points within the training set are selected or if some threshold on the residual or on the Power function is met.

Table 1 Overview of the hyperparameter ranges. The γ values are equally spaced, the others are logarithmically equally spaced

k	$\epsilon_{\min}$	$\epsilon_{\max}$	n_ϵ	$\gamma_{\min}$	$\gamma_{\max}$	n_γ	$\lambda_{\min}$	$\lambda_{\max}$	n_λ
5	10^{-2}	10^1	20	0	1	11	10^{-16}	10^3	20

As a tolerance on the residual we use $\tau_f = 10^{-7}$, that means the selection is stopped if $\max \|s_N(x_i) - y_i\|_2 < \tau_f$ is met. As a tolerance for the Power function we use $\tau_P = 10^{-3}$ and the selection is stopped if

$$P_N(x_{N+1}) < \tau_P. \tag{4}$$

We remark that this last criterion is directly linked to the stability. If points with small Power function value are selected, it means that the interpolation points cluster. We recall that this means in particular that $\lambda_{\min}(X_N)$ is below a certain threshold, making further computations unstable. Moreover, although a thorough discussion on the fine tuning of these thresholds is beyond the scope of this paper, we remark that the chosen values appear to be reasonable in this setting since they are sufficient to achieve the desired accuracy, while avoiding instabilities.

Table 2 lists both the hyperparameters which were selected by the cross-validations and the resulting accuracies of the interpolants. The $E_{\text{max,rel}}$ and the $E_{\text{RMSE,rel}}$ errors are defined according to

$$E_{\text{max,rel}} := \max_{i=1,..,|X|} \|s(x_i) - y_i\|_2 / \|y_i\|_2, \quad E_{\text{RMSE,rel}} \quad := \left(\frac{1}{|X|} \cdot \sum_{i=1}^{|X|} \frac{\|s(x_i) - y_i\|_2^2}{\|y_i\|_2^2} \right)^{1/2} .$$

In the left plot of Fig. 2 the number of selected points during the cross validation are depicted for the f/P-greedy. One can see that increasing the stabilization parameter γ yields more interpolation points. The reason is that the stopping criterion $P_N(x_{N+1}) < \tau_P$ is reached later since the selected points are distributed more uniformly as quantified in Theorem 1 and thus the greedy algorithms run further. We omit plotting results for the f-greedy as they do not differ considerably. Eventually these further interpolation points yield a better interpolation accuracy, which can be seen in Table 2. Especially the maximal relative error $E_{\text{max,rel}}$ and the relative RMSE error $E_{\text{RMSE, rel}}$ are improved. In the right plot of Fig. 2 the error decay for f/P-greedy depending on the number of chosen points during the selection (first step of training) is visualized for the E_{RMSE} error. One can observe that the algorithm stops quite early since the stability stopping criterion (4) is met. Larger stabilization parameters yield a slower drop, however more interpolation points.

Table 2 Overview of the selected hyperparameter and the accuracies of the kernel models

	f-greedy		f/P-greedy	
	Hyperparameters	Results	Hyperparameters	Results
Base	$\epsilon_{\text{base}} = 6.158 \cdot 10^{-2}$	$E_{\max} = 347.22$	$\epsilon_{\text{base}} = 4.281 \cdot 10^{-2}$	$E_{\max} = 4729.19$
	$\gamma_{\text{base}} = 0$	$E_{\text{RMSE}} = 39.58$	$\gamma_{\text{base}} = 0$	$E_{\text{RMSE}} = 1104.12$
	$\lambda_{\text{base}} = 10^{-5}$	$E_{\max,\text{rel}} = 6.95$	$\lambda_{\text{base}} = 10^{-2}$	$E_{\max,\text{rel}} = 116.02$
	$n_{\text{base}} = 142$	$E_{\text{RMSE,rel}} = 9.00 \cdot 10^{-1}$	$n_{\text{base}} = 63$	$E_{\text{RMSE,rel}} = 14.83$
Stabilized	$\epsilon_{\text{stab}} = \epsilon_{\text{base}}$	$E_{\max} = 344.91$	$\epsilon_{\text{stab}} = \epsilon_{\text{base}}$	$E_{\max} = 234.40$
	$\gamma_{\text{stab}} = 0.5$	$E_{\text{RMSE}} = 35.69$	$\gamma_{\text{stab}} = 0.2$	$E_{\text{RMSE}} = 30.22$
	$\lambda_{\text{stab}} = 10^{-5}$	$E_{\max,\text{rel}} = 1.79 \cdot 10^{-1}$	$\lambda_{\text{stab}} = 10^{-15}$	$E_{\max,\text{rel}} = 6.77 \cdot 10^{-1}$
	$n_{\text{stab}} = 690$	$E_{\text{RMSE,rel}} = 2.26 \cdot 10^{-2}$	$n_{\text{stab}} = 358$	$E_{\text{RMSE,rel}} = 7.97 \cdot 10^{-2}$

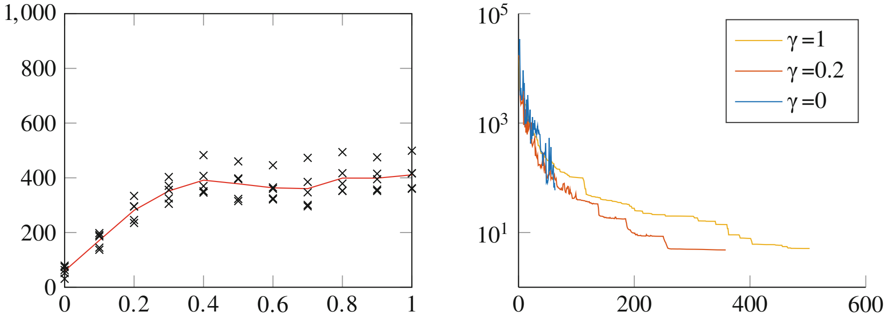


Fig. 2 Left plot: Number of chosen interpolation points (y-axis) during the 5-fold cross validation procedure for f/P-greedy in dependence of the restriction parameter $\gamma \in \{0, 0.1, .., 1\}$ (x-axis). The black crosses indicate the five numbers of chosen points during the validation, the red line describes the mean value of those. Right plot: E_{RMSE} error decay (y-axis) during the training of the f/P-greedy model depending on the number of interpolation points (x-axis) for the unstabilized model ($\gamma = 0$), the stabilized model with validated γ-parameter ($\gamma = 0.2$) and the fully stabilized model ($\gamma = 1$, i.e. P-greedy)

5 Conclusion and Outlook

In this paper a vectorial extension of a recent idea of stabilization of greedy kernel approximation algorithms was introduced and analytical properties were stated. A numerical application was addressed using data that emerge in the simulation of the human spine and the stabilization led to significant improvements in terms of accuracy and stability due to a better point distribution.

A two-step approach was used to combine the stabilization with regularization. In future work we will consider a combined approach of stabilization and regularization and use data with more input and output dimensions. Ultimatively we aim at using real patient data and dataset extension approaches by using invariances and symmetries or the use of invariant kernels.

Acknowledgments The authors acknowledge the funding of the project by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy - EXC 2075 - 390740016.

References

1. De Marchi, S., Schaback, R., Wendland, H.: Near-optimal data-independent point locations for radial basis function interpolation. Adv. Comput. Math. **23**(3), 317–330 (2005). http://dx.doi.org/10.1007/s10444-004-1829-1
2. Dreischarf, M., Zander, T., Shirazi-Adl, A., Puttlitz, C., Adam, C., Chen, C., Goel, V., Kiapour, A., Kim, Y., Labus, K., Little, J., Park, W., Wang, Y., Wilke, H., Rohlmann, A., Schmidt, H.: Comparison of eight published static finite element models of the intact lumbar spine: Predictive power of models improves when combined together. Journal of Biomechanics **47**(8), 1757–1766 (2014). https://doi.org/10.1016/j.jbiomech.2014.04.002

3. Fasshauer, G.E., McCourt, M.: Kernel-Based Approximation Methods Using MATLAB, *Interdisciplinary Mathematical Sciences*, vol. 19. World Scientific Publishing Co. Pte. Ltd., Hackensack, NJ (2015)
4. Karajan, N., Röhrle, O., Ehlers, W., Schmitt, S.: Linking continuous and discrete intervertebral disc models through homogenisation. Biomechanics and Modeling in Mechanobiology **12**(3), 453–66 (2013). https://doi.org/10.1007/s10237-012-0416-5
5. Monteiro, N.M.B., da Silva, M.P.T., Folgado, J.O.M.G., Melancia, J.P.L.: Structural analysis of the intervertebral discs adjacent to an interbody fusion using multibody dynamics and finite element cosimulation. Multibody System Dynamics **25**(2), 245–270 (2011). https://doi.org/10.1007/s11044-010-9226-7
6. Müller, S.: Komplexität und stabilität von kernbasierten rekonstruktionsmethoden (complexity and stability of kernel-based reconstructions). Ph.D. thesis, Fakultät für Mathematik und Informatik, Georg-August-Universität Göttingen (2009). https://ediss.uni-goettingen.de/handle/11858/00-1735-0000-0006-B3BA-E
7. Rupp, T., Ehlers, W., Karajan, N., Günther, M., Schmitt, S.: A forward dynamics simulation of human lumbar spine flexion predicting the load sharing of intervertebral discs, ligaments, and muscles. Biomechanics and Modeling in Mechanobiology **14**(5), 1081–1105 (2015)
8. Santin, G., Haasdonk, B.: Kernel methods for surrogate modelling. Tech. Rep. arXiv:1907.10556, University of Stuttgart (2019). To appear in the MOR Handbook, de Gruyter
9. Schaback, R., Wendland, H.: Adaptive greedy techniques for approximate solution of large RBF systems. Numer. Algorithms **24**(3), 239–254 (2000). http://dx.doi.org/10.1023/A:1019105612985
10. Wendland, H.: Scattered Data Approximation, *Cambridge Monographs on Applied and Computational Mathematics*, vol. 17. Cambridge University Press, Cambridge (2005)
11. Wenzel, T., Santin, G., Haasdonk, B.: A novel class of stabilized greedy kernel approximation algorithms: Convergence, stability & uniform point distribution. arXiv e-prints arXiv:1911.04352 (2019)
12. Wirtz, D., Haasdonk, B.: A vectorial kernel orthogonal greedy algorithm. Dolomites Res. Notes Approx. **6**, 83–100 (2013). http://drna.padovauniversitypress.it/system/files/papers/WirtzHaasdonk-2013-VKO.pdf
13. Wirtz, D., Karajan, N., Haasdonk, B.: Surrogate modeling of multiscale models using kernel methods. International Journal for Numerical Methods in Engineering **101** (2015). https://doi.org/10.1002/nme.4767
14. Wittwar, D., Santin, G., Haasdonk, B.: Interpolation with uncoupled separable matrix-valued kernels. Dolomites Res. Notes Approx. **11**, 23–29 (2018). https://doi.org/10.14658/pupj-drna-2018-3-4. https://drna.padovauniversitypress.it/2018/3/4

Augmented Lagrangian Method for Thin Plates with Signorini Boundaries

Erik Burman, Peter Hansbo, and Mats G. Larson

Abstract We consider C^1-continuous approximations of the Kirchhoff plate problem in combination with a mesh dependent augmented Lagrangian method on a simply supported Signorini boundary.

1 Introduction

To introduce the augmented Lagrangian method we first consider a simple Poisson problem, find u such that

$$-\Delta u = f \text{ in } \Omega, \qquad u = g \text{ on } \Gamma \tag{1}$$

where Ω is a bounded domain with boundary $\Gamma := \partial\Omega$ and exterior unit normal $\boldsymbol{n}$,

The Lagrange multiplier approach to prescribing $u = g$ is to seek stationary points to

$$\mathcal{L}(v, \mu) := \frac{1}{2}a(v, v) - \langle \mu, v - g\rangle_\Gamma - (f, v) \tag{2}$$

E. Burman
Department of Mathematics, University College London, London, UK
e-mail: e.burman@ucl.ac.uk

P. Hansbo (✉)
Department of Mechanical Engineering, Jönköping University, Jönköping, Sweden
e-mail: peter.hansbo@ju.se

M. G. Larson
Department of Mathematics and Mathematical Statistics, Umeå University, Umeå, Sweden
e-mail: mats.larson@math.umu.se

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_50

where

$$(f, v) := \int_\Omega f v \, d\Omega, \; a(u, v) := \int_\Omega \nabla u \cdot \nabla v \, d\Omega \tag{3}$$

and $\langle \cdot, \cdot \rangle_\Gamma$ denotes the $H^{-1/2}/H^{1/2}$-duality pairing. Whenever the arguments are smooth enough we define,

$$\langle \mu, v - g \rangle_\Gamma := \int_\Gamma \mu (v - g) \, ds \tag{4}$$

Stationary points are given by finding $(u, \lambda) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ such that

$$a(u, v) - \langle \lambda, v \rangle_\Gamma = (f, v) \quad \forall v \in H^1(\Omega) \tag{5}$$

$$\langle \mu, u \rangle_\Gamma = \langle \mu, g \rangle_\Gamma \quad \forall \mu \in H^{-1/2}(\Gamma) \tag{6}$$

Formally, the Lagrange multiplier is given by $\lambda = \partial_n u$, where $\partial_n v := \boldsymbol{n} \cdot \nabla v$. In a discretization of this problem, the approximation of the multiplier and the displacement must fulfil an *inf–sup* condition ensuring that the problem will not be overconstrained.

We now augment the Lagrangian by a penalty term and seek stationary points to

$$\mathcal{L}(v, \mu) := \frac{1}{2} a(v, v) - \langle \mu, v - g \rangle_\Gamma + \frac{1}{2} \| \gamma^{1/2} (v - g) \|_\Gamma^2 - (f, v) \tag{7}$$

leading to the problem of finding $(u, \lambda) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$ such that

$$a(u, v) - \langle \lambda, v \rangle_\Gamma + \langle \gamma \, u, v \rangle_\Gamma - \langle \mu, u \rangle_\Gamma = (f, v) + \langle \gamma \, g, v \rangle_\Gamma - \langle \mu, g \rangle_\Gamma \tag{8}$$

for all $(v, \mu) \in H^1(\Omega) \times H^{-1/2}(\Gamma)$. The discretization of this problem requires the same careful balance between approximation spaces for the primal variable and the multiplier as does the standard Lagrange multiplier method. Indeed if we introduce the space

$$V_h := \{ v_h \in H^1(\Omega) : v_h|_K \in \mathbb{P}_k(K), \, \forall K \in \mathcal{T}_h \}, \qquad \text{for } k \geq 1 \tag{9}$$

where $\mathcal{T}_h$ is a conforming quasi-uniform partition of Ω and $\mathbb{P}_k(K)$ denotes the set of polynomials of degree less than or equal to k on the element K for the discretization of u, we must find a multiplier space Λ_h such that the inf-sup condition is satisfied. However, if we seek $u_h \in V_h$ with the discrete multiplier $\lambda_h := \partial_n u_h$, we recover Nitsche's method:

$$a(u_h, v) - \langle \partial_n u_h, v \rangle_\Gamma - \langle \partial_n v, u_h \rangle_\Gamma + \langle \gamma \, u_h, v \rangle_\Gamma = (f, v) + \langle g, \gamma v - \partial_n v \rangle_\Gamma \tag{10}$$

for all $v \in V_h$, with $\mu = \partial_n v$, which is stable with the choice $\gamma = \gamma_0/h$, where h is the local meshsize and γ_0 large enough.

If we alternatively consider a stable discretization $\lambda_h \in \Lambda_h$, the discrete problem can be seen as seeking stationary points $(u_h, \lambda_h) \in V_h \times \Lambda_h$ to the modified Lagrangian

$$\mathcal{L}_h(v, \mu) := \frac{1}{2}a(v, v) + \frac{1}{2}\|\gamma^{1/2}(v - g - \gamma^{-1}\mu)\|_\Gamma^2 - \|\gamma^{-1/2}\mu\|_\Gamma^2 - (f, v)_\Omega \tag{11}$$

which is obtained from $\mathcal{L}$ in (7) by rearranging terms and noting that the discrete multiplier is in $L_2(\Gamma)$.

We now turn to an inequality constraint on the boundary: $u \le g$ on Γ. The corresponding Kuhn–Tucker conditions read:

$$u - g \le 0, \quad \lambda \le 0, \quad \lambda(u - g) = 0 \quad \text{on } \Gamma. \tag{12}$$

These conditions can alternatively be written (cf. [6])

$$\lambda = -\gamma\,[u - g - \gamma^{-1}\,\lambda]_+ \tag{13}$$

where $\gamma \in \mathbb{R}^+$, $[x]_+ = \max(x, 0)$. We can now, following Alart and Curnier [1], define the following discrete augmented Lagrangian

$$\mathcal{L}_h(v, \mu) := \frac{1}{2}a(v, v) + \frac{1}{2}\|\gamma^{1/2}[v - g - \gamma^{-1}\mu]_+\|_\Gamma^2 - \|\gamma^{-1/2}\mu\|_\Gamma^2 - (f, v) \tag{14}$$

The corresponding Euler-Lagrange equations read: find $(u_h, \lambda_h) \in V_h \times \Lambda_h$ such that

$$a(u_h, v) + \langle\gamma\left[u_h - g - \gamma^{-1}\lambda_h\right]_+, v\rangle_\Gamma = (f, v) \quad \forall v \in V_h \tag{15}$$

and

$$\langle\gamma\left[u_h - g - \gamma^{-1}\lambda_h\right]_+ + \lambda_h, \gamma^{-1}\mu\rangle_\Gamma = 0 \quad \forall \mu \in \Lambda_h \tag{16}$$

If $[u_h - g - \gamma^{-1}\lambda_h]_+ = 0$ (no contact) then $\lambda_h = 0$ and if $[u_h - g - \gamma^{-1}\lambda_h]_+ > 0$ (contact) we recover the standard augmented formulation for the imposition of the Dirichlet condition $u = g$. The multiplier approach (15)–(16) using a stable pair $V_h \times \Lambda_h$ was shown to produce approximations of optimal accuracy in [4]. Now set $\lambda_h = \partial_n u_h$, $\mu = \partial_n v$ and seek $u_h \in V_h$ such that

$$a(u_h, v) + \langle\gamma\,[u_h - g - \gamma^{-1}\,\partial_n u_h]_+, v - \gamma^{-1}\,\partial_n v\rangle_\Gamma - \langle\gamma^{-1}\,\partial_n u_h, \partial_n v\rangle_\Gamma = (f, v) \tag{17}$$

for all $v \in V_h$. With the choice $\gamma = \gamma_0/h$ this is the Nitsche method for Signorini problems first proposed in the context of elastic contact by Chouly and Hild [6]. For more information on augmented Lagrangian methods and variants thereof, see [3].

2 The Kirchhoff Plate Model

We now proceed formally to extend the discussion to the Kirchhoff plate model, posed on a domain $\Omega \subset \mathbb{R}^2$ with boundary $\Gamma = \partial\Omega$ and exterior unit normal $\boldsymbol{n}$. We seek an out-of-plane (scalar) displacement u to which we associate the strain (curvature) tensor

$$\boldsymbol{\kappa}(u) := \boldsymbol{\varepsilon}(\nabla u) := \frac{1}{2}\left(\nabla \otimes (\nabla u) + (\nabla u) \otimes \nabla\right) = \nabla \otimes \nabla u \tag{18}$$

and the plate stress (moment) tensor

$$\boldsymbol{M}(u) := \boldsymbol{\sigma}(\nabla u) :=D\left(\boldsymbol{\varepsilon}(\nabla u) + \nu(1-\nu)^{-1}\mathrm{div}\nabla u\,\boldsymbol{I}\right) \tag{19}$$

$$=D\left(\boldsymbol{\kappa}(u) + \nu(1-\nu)^{-1}\Delta u\boldsymbol{I}\right) \tag{20}$$

where

$$D = \frac{Et^3}{12(1+\nu)} \tag{21}$$

with E the Young's modulus, ν the Poisson's ratio, and t the plate thickness. We will use the standard convention that all quantities are positive downwards.

The Kirchhoff equilibrium problem takes the form: given the out-of-plane load (per unit area) f, find the displacement u such that

$$\mathrm{div}\,\mathbf{div}\,\boldsymbol{M}(u) = f \quad \text{in } \Omega \tag{22}$$

where **div** and div denote the divergence of a tensor and a vector field, respectively. We shall first consider a smooth boundary Γ with simply supported boundary conditions

$$u = 0 \quad \text{on } \Gamma, \qquad M_{nn}(u) = 0 \quad \text{on } \Gamma \tag{23}$$

where $M_{ab} = \boldsymbol{a} \cdot \boldsymbol{M} \cdot \boldsymbol{b}$ for $\boldsymbol{a}, \boldsymbol{b} \in \mathbb{R}^2$. Defining the tangent vector on the boundary as $\boldsymbol{t} = (n_2, -n_1)$, multiplying by a test function v and using repeated integration by parts we find that

$$(\operatorname{div} \mathbf{div}\, \boldsymbol{M}(u), v) = (\boldsymbol{M}(u), \boldsymbol{\kappa}(v)) - \langle M_{nn}(u), \partial_n v\rangle_\Gamma \tag{24}$$

$$- \langle M_{nt}(u), \partial_t v\rangle_\Gamma + \langle \boldsymbol{n} \cdot \mathbf{div}\, \boldsymbol{M}(u), v\rangle_\Gamma \tag{25}$$

In the case of a smooth boundary we note that

$$\langle M_{nt}(u), \partial_t v\rangle_\Gamma = -\langle \partial_t M_{nt}(u), v\rangle_\Gamma \tag{26}$$

and by introducing the Kirchhoff shear force $T := \boldsymbol{n} \cdot \mathbf{div}\, \boldsymbol{M} + \partial_t M_{nt}$ we have

$$(\operatorname{div} \mathbf{div}\, \boldsymbol{M}(u), v) = (\boldsymbol{M}(u), \boldsymbol{\kappa}(v)) - \langle M_{nn}(u), \partial_n v\rangle_\Gamma + \langle T(u), v\rangle_\Gamma \tag{27}$$

Taking into account the boundary conditions, the variational problem thus takes the form: find $u \in V = \{v \in H^2(\Omega) : v = 0 \text{ on } \Gamma\}$
such that

$$(\boldsymbol{M}(u), \boldsymbol{\kappa}(v)) = (f, v) \quad \forall v \in V \tag{28}$$

We will next consider the Signorini condition $u \geq g$ on Γ, which corresponds to a case where the plate boundary rests on a rigid foundation but is not fixed to it. Introducing a multiplier representing $T(u)$ we have that

$$\operatorname{div} \mathbf{div}\, \boldsymbol{M}(u) = f \quad \text{in } \Omega \tag{29}$$

$$M_{nn}(u) = 0 \quad \text{on } \Gamma \tag{30}$$

$$T(u) + \lambda = 0 \quad \text{on } \Gamma \tag{31}$$

$$u - g \geq 0 \quad \text{on } \Gamma \tag{32}$$

$$\lambda \leq 0 \quad \text{on } \Gamma \tag{33}$$

$$\lambda(u - g) = 0 \quad \text{on } \Gamma \tag{34}$$

In this case, the Kuhn–Tucker conditions can be rewritten

$$\lambda = -\gamma[g - u - \gamma^{-1}\lambda]_+ \tag{35}$$

Remark 1 (Handling Polygonal Domains) In the case of a domain with piecewise smooth boundaries, so called Kirchhoff corner forces occur in corner points

[9, Chapter 5.5]. This case was considered by Nazarov et al. [8] but with an alternative formulation (the biharmonic operator, leading to quite different boundary conditions). We here assume that Γ consists of smooth connected parts Γ_i with corner intersections at $\boldsymbol{x}_i$. Now (26) has to be modified as follows:

$$\langle M_{nt}(u), \partial_t v\rangle_\Gamma = -\langle \partial_t M_{nt}(u), v\rangle_\Gamma + \sum_i \left(M_{nt}^-(u(\boldsymbol{x}_i) - M_{nt}^+(u(\boldsymbol{x}_i))\right) v(\boldsymbol{x}_i) \tag{36}$$

where $M_{nt}^\pm(u(\boldsymbol{x}_i)) = \lim_{\epsilon\downarrow 0} M_{nt}(u(x_i \pm \epsilon, y_i \pm \epsilon))$, giving rise to (virtual work of) point forces in the corners. Unlike the Kirchhoff shear forces, the corner forces are present whether there is contact or not, and are implemented as contributions to the stiffness matrix.

3 Finite Element Method

We will use C^1-continuous element on meshes $\mathcal{K}_h$ made up of rectangles. On each element $K \in \mathcal{K}_h$ we let Q_3 denote the outer product of cubic polynomials:

$$Q_3 = \left\{ p(x, y) : \ p(x, y) = \sum_{0\le i,j\le 3} c_{ij} x^i y^j \right\}$$

where c_{ij} are constants. The approximation space associated with the Bogner-Fox-Schmit (BFS) element first proposed in [2] is defined by

$$V_h = \left\{ v \in C^1(\Omega) : v|_K \in Q_3, \ \forall K \in \mathcal{K}_h \right\} \tag{37}$$

The shape functions on the BFS element are then made up of outer products of cubic splines, typically used for beam problems. We refer to Zhang [10] for further details on this approximation. Though this element might seem limited in view of it only being defined on rectangular meshes, the recent CutFEM for BFS [5] extends its use to arbitrary geometries.

In analogy with (17) we now pose the following discrete problem: find $(u_h, \lambda_h) \in V_h \times \Lambda_h$, Λ_h to be chosen, such that

$$(\boldsymbol{M}(u_h), \boldsymbol{\kappa}(v)) - \langle \gamma \left[g - u_h - \gamma^{-1}\lambda_h \right]_+, v\rangle_\Gamma = (f, v) \quad \forall v \in V_h \tag{38}$$

and

$$- \langle \gamma \left[g - u_h - \gamma^{-1}\lambda_h \right]_+, \gamma^{-1}\mu\rangle_\Gamma - \langle \gamma^{-1}\lambda_h, \mu\rangle_\Gamma = 0 \quad \forall \mu \in \Lambda_h \tag{39}$$

We next consider replacing λ_h following the ideas of Sect. 1. To this end, we formally set $\lambda_h = -T(u_h)$ and $\mu = -T(v)$ to obtain the problem of finding $u_h \in V_h$ such that

$$(\boldsymbol{M}(u_h), \boldsymbol{\kappa}(v)) - \langle\gamma\,[g - \psi(u_h)]_+\,, \psi(v)\rangle_\Gamma - \langle\gamma^{-1}T(u_h), T(v)\rangle_\Gamma = (f, v) \tag{40}$$

for all $v \in V_h$, where $\psi(w) := w - \gamma^{-1}T(w)$. Setting now $\gamma = \gamma_0/h^3$, stability, existence and uniqueness of the discrete solution can be shown combining the results of [7] and [4]. We leave the details to a forthcoming publication.

4 Numerical Results

We consider a quadratic plate (0, 1) × (0, 1) of thickness $t = 0.1$ and with moduli of elasticity $E = 100$, $\nu = 0.5$. This plate is loaded by a point force of unit strength. The free parameter was chosen as $\gamma_1 = 10^4 D$, where D is given by (21). The maximum displacement on the boundary is set to $g = 0$.

4.1 Point Load in the Center of the Plate

We load the plate with a unit point load at the center. In Fig. 1 we show the computed displacement field with the Signorini boundary indicated by a dotted line. In Fig. 2

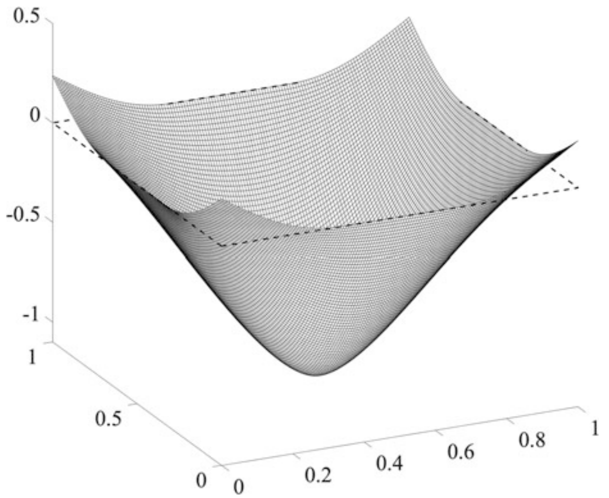


Fig. 1 Elevation of the solution on the finest mesh in a sequence. Point load at (1/2, 1/2)

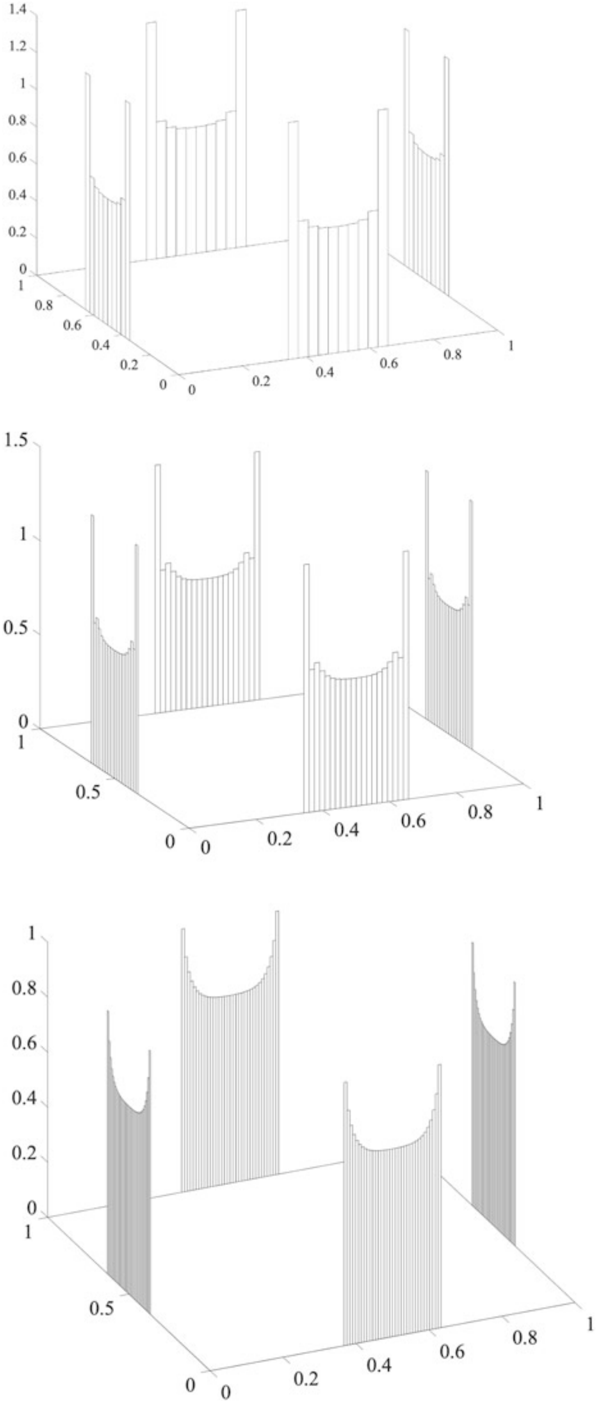


Fig. 2 Kirchhoff shear forces in the contact zone on consecutively refined meshes

we show the computed Kirchhoff shear force in the contact zone (evaluated at the midpoint of each element side) on a sequence of uniformly refined meshes. We note the symmetry of the solution.

4.2 Point Load at (3/4, 3/4)

The same plate is now loaded with unit point load at (3/4, 3/4). In Fig. 3 we show the computed displacement field, again with the Signorini boundary indicated by a dotted line. In Fig. 4 we show the corresponding Kirchhoff shear force in the contact zone. We note the elevation of the shear force close to the first point of contact, similar, but more pronounced, to Fig. 2.

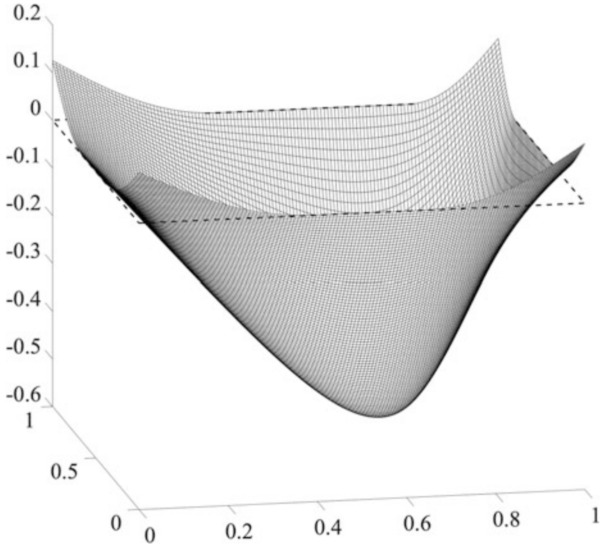
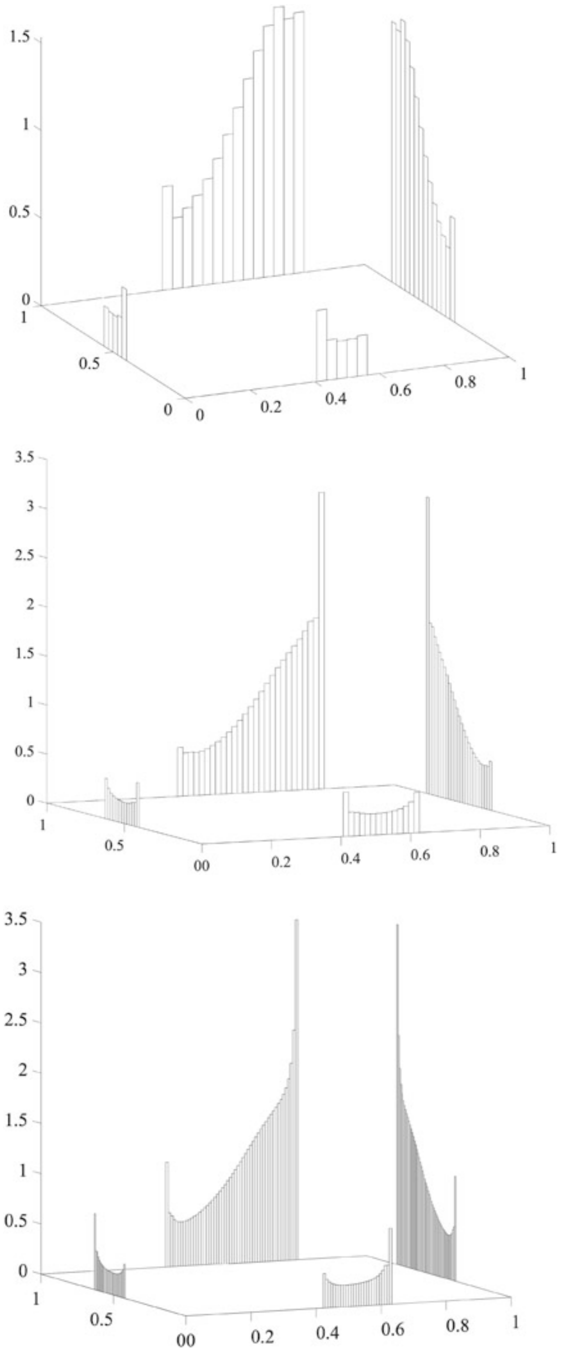


Fig. 3 Elevation of the solution on the finest mesh in a sequence. Point load at (3/4, 3/4)

Fig. 4 Kirchhoff shear forces in the contact zone on consecutively refined meshes



Acknowledgments This research was supported in part by the Swedish Foundation for Strategic Research Grant No. AM13-0029, the Swedish Research Council Grants Nos. 2017-03911 and 2018-05262 , and the Swedish Research Programme Essence. Erik Burman was partially supported by the grant: EP/P01576X/1.

References

1. P. Alart and A. Curnier. A mixed formulation for frictional contact problems prone to Newton like solution methods. *Comput. Methods Appl. Mech. Engrg.*, 92(3):353–375, 1991.
2. F. K. Bogner, R. L. Fox, and L. A. Schmit. The generation of interelement compatible stiffness and mass matrices by the use of interpolation formulae. In *Proc. Conf. Matrix Methods in Struct. Mech., AirForce Inst. of Tech., Wright Patterson AF Base, Ohio*, pages 397–444, 1965.
3. E. Burman and P. Hansbo. Deriving robust unfitted finite element methods from augmented Lagrangian formulations. In *Geometrically unfitted finite element methods and applications*, volume 121 of *Lect. Notes Comput. Sci. Eng.*, pages 1–24. Springer, Cham, 2017.
4. E. Burman, P. Hansbo, and M. G. Larson. Augmented Lagrangian finite element methods for contact problems. *ESAIM Math. Model. Numer. Anal.*, 53(1):173–195, 2019.
5. E. Burman, M. G. Larson, and P. Hansbo. Cut Bogner-Fox-Schmit elements for plates. *Adv. Model. and Simul. in Eng. Sci.*, 7:27, 2020.
6. F. Chouly and P. Hild. A Nitsche-based method for unilateral contact problems: numerical analysis. *SIAM J. Numer. Anal.*, 51(2):1295–1307, 2013.
7. P. Hansbo and M. G. Larson. A discontinuous Galerkin method for the plate equation. *Calcolo*, 39(1):41–59, 2002.
8. S. A. Nazarov, A. Stylianou, and G. Sweers. Hinged and supported plates with corners. *Z. Angew. Math. Phys.*, 63(5):929–960, 2012.
9. V. Slivker. *Mechanics of structural elements: theory and applications*. Springer Science & Business Media, 2006.
10. S. Zhang. On the full C_1-Q_k finite element spaces on rectangles and cuboids. *Adv. Appl. Math. Mech.*, 2(6):701–721, 2010.

A Hybrid High-Order Method for Flow Simulations in Discrete Fracture Networks

Florent Hédin, Géraldine Pichot, and Alexandre Ern

Abstract We are interested in solving flow in large tridimensional Discrete Fracture Networks (DFN) with the hybrid high-order (HHO) method. The objectives of this paper are: (1) to demonstrate the benefit of using a high-order method for computing macroscopic quantities, like the equivalent permeability of fracture rocks; (2) to present the computational efficiency of our C++ software, `NEF++`, which implements the solving of flow in fractures based on the HHO method.

1 The Flow Problem in Fractured Rocks

In fractured rocks, fluid flows mostly within a complex arrangement of fractures, classically modeled as a Discrete Fracture Network (DFN) [1, 2]. In the present reduced model, the fractures, denoted by Ω_f, $f = 1, \ldots, N_f$, are distributed in a three-dimensional domain Ω and are modeled as ellipses whose position and orientation are evaluated from statistical laws given by geological studies [3, 4]. We consider single phase flow problems within these networks of fractures. As we are mainly interested in flow simulations in granite type rocks, a classical assumption is to consider the rock matrix as impervious. Figure 1 presents three examples of DFN in a cubic domain.

Let $\mathbf{x}$ be the local 2D coordinates of fracture Ω_f. Let N be the total number of intersections between fractures, I_k be the k$^{\text{th}}$ intersection, $k = 1, \ldots, N$, and F_k be the set of fractures containing I_k. In each fracture Ω_f, we assume that the governing

F. Hédin · G. Pichot (✉)
Inria, Paris, France
CERMICS, Ecole des Ponts, Marne-la-Vallée, France
e-mail: florent.hedin@inria.fr; geraldine.pichot@inria.fr

A. Ern
CERMICS, Ecole des Ponts, Marne-la-Vallée, France
Inria, Paris, France
e-mail: alexandre.ern@enpc.fr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_51

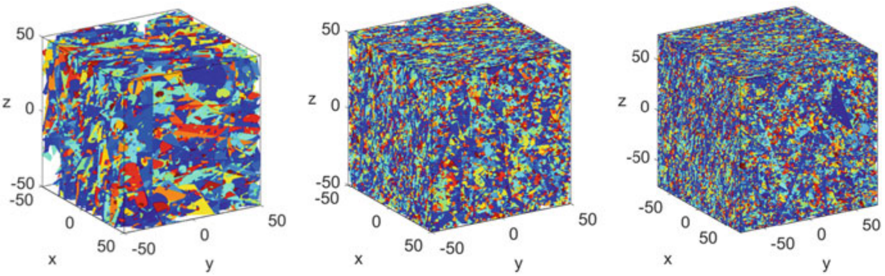


Fig. 1 (left) B1: 19,007 fractures; (center) B2: 152,399 fractures ; (right) B3: 508,338 fractures

equations for the hydraulic head scalar function p and for the flux per unit length function $\mathbf{u}$ are the mass conservation equation and Poiseuille's law [1]:

$$\nabla \cdot \mathbf{u}(\mathbf{x}) = f(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega_f, \tag{1a}$$

$$\mathbf{u}(\mathbf{x}) = -\mathcal{T}(\mathbf{x})\, \nabla p(\mathbf{x}) \quad \text{for } \mathbf{x} \in \Omega_f. \tag{1b}$$

The parameter $\mathcal{T}(\mathbf{x})$ is a given transmissivity field (unit [$\mathrm{m}^2.\mathrm{s}^{-1}$]). The function $f \in L^2(\Omega_f)$ represents the sources/sinks. Additionally, continuity of the hydraulic head and continuity of the transversal flux apply at the intersections between the fractures [2, 5]:

$$p_{k,i} = p_k \quad \text{on } I_k, \forall f \in F_k, \tag{2a}$$

$$\sum_{i \in F_k} \mathbf{u}_{k,f} \cdot \mathbf{n}_{k,f} = 0 \quad \text{on } I_k, \tag{2b}$$

where $p_{k,i}$ is the trace of hydraulic head on I_k in fracture Ω_f, p_k is the unknown hydraulic head on the intersection I_k and $\mathbf{u}_{k,f} \cdot \mathbf{n}_{k,f}$ is the normal flux through I_k coming from fracture Ω_f, with $\mathbf{n}_{k,f}$ the outward normal unit vector of the intersection I_k with respect to the fracture Ω_f. Boundary conditions (BC) on the cube faces are of Dirichlet or Neumann type. For edges that belong to the border of the fractures but not to a cube face, a homogeneous Neumann BC is applied to express the imperviousness of the rock matrix.

2 The HHO Method for Solving Flow in DFN

Several methods have been developed to solve flow in DFN in the recent years as detailed in the survey [6] and the references therein. The methods highly depend on the mesh strategy chosen to mesh the DFN [7, 8]. In our work, we keep the intersections explicitely. It implies a substantial work regarding the development of robust and efficient software in order to be able to mesh efficiently large networks with a good quality mesh [9]. In all our test cases, the mesh is generated with the software BLSURF_FRAC [10, 11] and the planar mesher is BL2D[12]. The data files generated by BLSURF_FRAC follow the description given by Appendix A in [13]. Here we consider the so-called conforming discretizations at the intersections between the ellipses but the software BLSURF_FRAC is also able to generate non-conforming discretizations as well. The advantage of keeping the intersections explicitely in the mesh generation is that it allows to attach unknowns to the edges and then continuity conditions (2) are easier to impose.

Among the methods that attach unknowns to the edges, let us cite the mixed-hybrid finite elements method (MHFEM), for conforming [5, 14–17] or non-conforming [18, 19] discretizations at the intersections. More recently, a hybrid high-order (HHO) method has been developed [20, 21]. HHO is already used in many applications and has been recently used for fracture/matrix coupling [22]. HHO is closely related to Hybridizable Discontinuous Galerkin methods (HDG) [23]. The main advantages of HHO are (1) it allows general meshes (including polytopal cells and nonmatching interfaces), (2) it manages arbitrary polynomial face orders k, (3) it leads solve a linear system with only the unknowns at the edges and the matrix of this system is symmetric positive definite, (4) it delivers approximate solutions converging at order h^{k+1} in the energy norm and h^{k+2} in the L^2-norm (if full elliptic regularity holds) [20, 21]. Moreover, this HHO method is implemented in the open source library, DiSk++ [24], which is a C++ template based library, both in the dimension and also in the finite element shapes. Notice that only the 2D feature of the DiSk++ library is used in this study as the rock matrix is assumed impervious, however the dimensional templating offered by DiSk++ will be very useful for future porous fractured rocks simulations.

3 Computation of the Equivalent Permeability with the HHO Method

The goal of this section is to demonstrate the benefit of using a high-order method for computing upscaled quantities, like the equivalent permeability.

The equivalent permeability tensor is a macroscopic quantity of interest classically used by hydrogeologists for upscaling [25]. Its components can be derived from numerical simulations. Typically, the three diagonal components of the permeability tensor are given by applying permeameter boundary conditions in the directions x, y and z respectively. As we are rather interested in analyzing the performance of the HHO method to compute such macroscopic quantities, we focus here only on a flow in the direction x to derive the x-component of the permeability tensor defined as: $K_x = \dfrac{Q_{in,x}}{L\,\Delta h}$ for a cubic domain of size L, with $Q_{in,x}$ the input flux (units $\mathrm{m}^3\,\mathrm{s}^{-1}$) with respect to permeameter boundary conditions in the direction x.

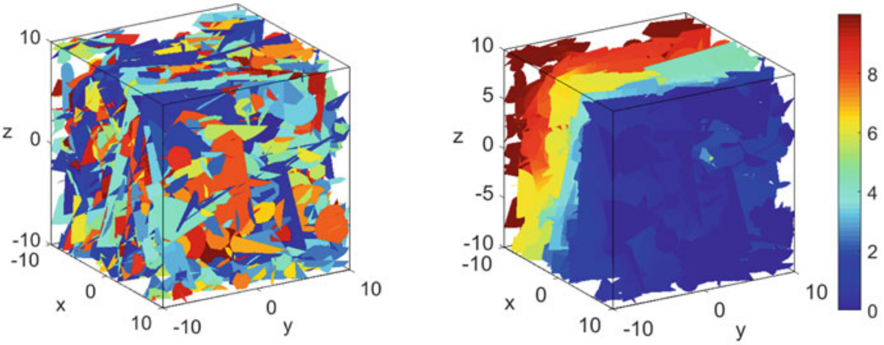


Fig. 2 (left) B0: 1,397 fractures, (right) mean hydraulic head for permeameter BC

Table 1 Test case B0: 1397 fractures, computed values of the x-component of the equivalent permeability K_x with mesh refinement and different numerical methods

	Equivalent permeability K_x			
DFN B0 #edges	MHFEM RT0	HHO, $k = 0$	HHO, $k = 1$	HHO, $k = 2$
137,680	0.090929	0.090929	0.098410	0.099924
528,611	0.096663	0.096663	0.100556	0.101296
2,120,115	0.099615	0.099615	0.101507	0.101892
8,533,221	0.101032	0.101032	0.101943	0.102171
34,299,544	0.101696	0.101696	–	–

We propose to compute the equivalent permeability in the direction x of the small network $B0$ shown on Fig. 2 (left). The domain is a cube of size $L = 20$. This network has 1397 fractures and 2481 intersections. We imposed a permeameter boundary condition in the direction x with a difference of hydraulic head of 10 m between the two opposite cube faces. The mean hydraulic head solution obtained with the MHFEM (Raviart-Thomas 0) for a mesh with 8, 533, 221 edges is shown on Fig. 2 (right). We compute K_x with the MHFEM RT0 and with the HHO method for face polynomial degrees $k = 0$, 1 and 2 and we compare the results. Table 1 presents the values of K_x as the mesh is refined.

The simulations for $k = 1$ and $k = 2$ on the finer mesh are not available yet as they require a lot of computational resources. For the MHFEM RT0 method, the software that is used is a Matlab software, called `NEF-Flow` [11], developed at Inria and CNRS (France), and for the HHO method for DFN, the simulations are performed with the C++ software `NEF++`, developed at Inria and described in more details in the following section.

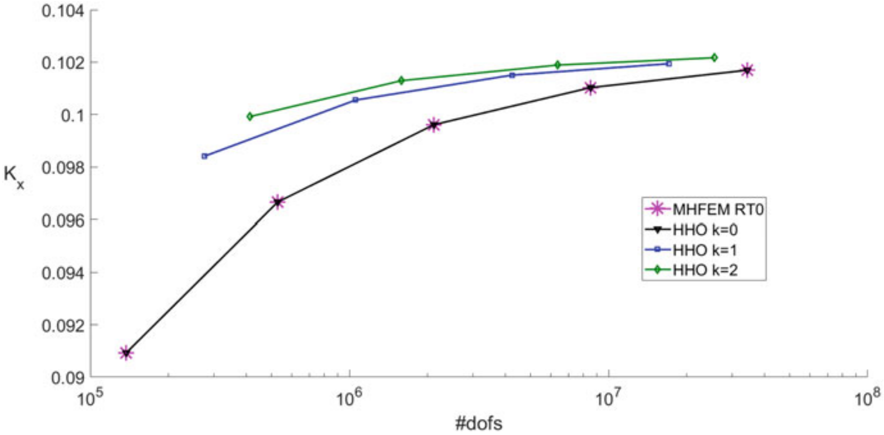


Fig. 3 Test case B0: x-component of the equivalent permeability tensor

Figure 3 presents the equivalent permeability with respect to the number of degrees of freedom (dofs). For the MHFEM RT0 and the HHO method with $k = 0$, the number of dofs are equal to the total number of edges, denoted by n_E. For the HHO method with $k = 1$, the number of dofs are $2\,n_E$. For the HHO method with $k = 2$, the number of dofs are $3\,n_E$. At low order ($k = 0$), the curves for K_x obtained with the HHO method and the MHFEM RT0 method almost superimpose, which was expected as the two methods are very close. The benefits of the increased orders of convergence are clearly seen from Fig. 3 since the equivalent permeability is better approximated by a fixed number of dofs if resorting to a higher-order method. This result shows that the exact solution has enough regularity to take advantage of the computational efficiency delivered by higher-order methods.

4 Performance Obtained with the NEF++ Software

Solving the flow problem (1)–(2) within large 3D DFNs requires robust and efficient software. The goal of this section is to present the C++ software we have developed at Inria, called NEF++, and based on the C++17 standard. NEF++ relies on the Eigen library, which is a C++ template library for linear algebra [26] and on the DiSk++ library for HHO. The linear systems can be solved with direct solvers or with iterative solvers like the preconditioned conjugate gradient or multigrid solvers. In NEF++, the following two direct solvers can be called: Pardiso from Intel MKL library [27] or SuiteSparse [28]. Both solvers support tasks parallelism, either using OpenMP or Intel TBB and support SIMD (Single Instruction, Multiple Data) vectorization.

We propose to solve flow in the three DFN B1, B2 and B3 shown on Fig. 1. We imposed a permeameter boundary condition in the direction x with a difference of hydraulic head of 10 m between the two opposite cube faces. The transmissivity is taken as a constant per fracture and is different from one plane to another. We consider a cubic domain Ω of size $L = 100$ for $B1$ and $B2$ and $L = 150$ for $B3$. The network information about the geometry and the range of transmissivity values are given in Table 2.

For the larger linear systems (for $B1$ and $B2$ with $k = 1$ and $k = 2$ and for $B3$ with $k = 0, 1$ and 2), we are facing with the Intel Pardiso LLT solver some problems that we are currently investigating. On the contrary, we have no problem with the Intel Pardiso LU solver. In the following Tables 3, 4 and 5, the solver information (LLT or LU) will be given for each simulation. Moreover, depending on the requirements in computational resources, the simulations have been run either on a Intel Core i7 6 cores CPU laptop (denoted by IC in the following Tables) or

Table 2 Details about the three DFN test cases $B1$, $B2$ and $B3$

	L	#fractures	#intersections	Range of transmissivity value [$m^2\ s^{-1}$]
B1	100	19,007	28,727	[2.8e−06; 47.4]
B2	100	152,399	302,907	[3.4e−06; 20.33]
B3	150	508,338	1,031,231	[0.3e−05;25.8]

Table 3 Performance obtained with `NEF++` for the HHO method with $k = 0$ on $B1$, $B2$ and $B3$

	#fractures	n_E	Read mesh	Assemb.	Solving	Total time	RAM peak	Solver	Run
B1	19,007	18,494,551	16.1 *s*	53.1 s	50.5 s	2 min	26.89 GiB	LLT	IC
B1	19,007	18,494,551	16.0 *s*	50.4 s	1 min 10 s	2 min17 s	30.59 GiB	LU	IC
B2	152,399	11,054,762	8.6 *s*	30.2 s	37.3 s	1 min 16 s	16.48 GiB	LLT	IC
B3	508,338	12,219,167	10.3 *s*	38.7 s	1 min 17s	2 min 7s	24.84 GiB	LU	IC

Table 4 Performance obtained with `NEF++` for the HHO method with $k = 1$ on $B1$, $B2$ and $B3$

	#fractures	n_E	Read mesh	Assemb.	Solving	Total time	RAM peak	Solver	Run
B1	19,007	18,494,551	20 s	5 min 57 s	3 min 36 s	9 min 53 s	105 GiB	LU	IX
B2	152,399	11,054,762	10 s	3 min 52 s	2 min 24 s	6 min 26 s	64 GiB	LU	IX
B3	508,338	12,219,167	12 s	4 min 42 s	3 min 52 s	8 min 47 s	78 GiB	LU	IX

Table 5 Performance obtained with `NEF++` for the HHO method with $k = 2$ on $B1$, $B2$ and $B3$

	#fractures	n_E	Read mesh	Assemb.	Solving	Total time	RAM peak	Solver	Run
B1	19,007	18,494,551	20 s	13 min 48 s	11 min 4 s	25 min 12 s	204 GiB	LU	IX
B2	152,399	11,054,762	10 s	8 min 39 s	5 min 27 s	14 min 16 s	124 GiB	LU	IX
B3	508,338	12,219,167	12 s	9 min 48 s	12 min 9 s	22 min 9 s	153 GiB	LU	IX

on a cluster node with 4 Intel Xeon E7-8890 processors and 1,024 GiB of RAM (denoted by IX in the following Tables).

Table 3 gives the computational time (for reading the mesh, assembling and solving the linear system) and peak memory of the NEF++ software for $k = 0$ for the three test cases. For the B1 test case, we provide the results obtained with the LLT and LU solvers. With LU, the RAM memory requirements are higher than with LLT, as expected.

Tables 4 and 5 give the computational time (for reading the mesh, assembling and solving the linear system) and peak memory of the NEF++ software for $k = 1$ and $k = 2$ respectively for the three test cases. Despite $B3$ has fewer edges than $B1$, it takes more time to solve the associated linear system with a direct solver as it has more intersections (see Table 2). As shown by Tables 3, 4 and 5, increasing k requires more computation times and memory as the number of dofs increases but the solutions are more accurate, as highlighted in Sect. 3.

5 Conclusion

The results in terms of computational time and accuracy we are currently obtaining with the NEF++ software are very promising to handle in a near future the millions of fractures networks provided by external industrial partners. As the RAM requirements with direct solvers are quite large, we are currently investigating the use of iterative solvers. As emphasized in this paper, the HHO method has a strong potential, also for deriving upscaled quantities owing to its high-order feature. Moreover, as HHO naturally deals with general shape elements, non-conforming discretizations at the intersections between the fractures can be naturally handled in a conforming way. Finally, as a future work, we are interested in using the dimensional templating feature offered by the DiSk++ library to solve flow in porous fractured rocks.

Acknowledgments This research has been partially supported by the LABEX AMIES ANR-10-LBX-0002-01 project. The authors are grateful to the LabCom *fractory* (CNRS, Université de Rennes 1 and Itasca Consultants) for providing the geometry and transmissivity data for all the DFN considered in this paper. The authors warmly thank Nicolas Pignet (EDF) for many fruitful discussions related to DiSk++. Part of the simulations presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine (see https://www.plafrim.fr/).

References

1. V. Martin, J. Jaffré & J. E. Roberts, Modeling fractures and barriers as interfaces for flow in porous media, *SIAM Journal on Scientific Computing*, 26 (5), pp. 1667–1691, 2005.

2. J. Erhel, J. de Dreuzy, and B. Poirriez. Flow simulation in three-dimensional discrete fracture networks. *SIAM Journal on Scientific Computing*, 31(4):2688–2705, 2009.
3. P. Davy, R. Le Goc, C. Darcel, O. Bour, J.-R de Dreuzy & R. Munier, A likely universal model of fracture scaling and its consequence for crustal hydromechanics, *Journal of Geophysical Research: Solid Earth*, 115 (B10), 2010.
4. P. Davy, R. Le Goc & C. Darcel, A model of fracture nucleation, growth and arrest, and consequences for fracture density and scaling, *Journal of Geophysical Research: Solid Earth*, 118 (4), pp. 1393–1407, 2013.
5. J. Maryška, O. Severýn & M. Vohralík, Numerical simulation of fracture flow with a mixed-hybrid FEM stochastic discrete fracture network model, *Computational Geosciences*, 8, pp. 217–234, 2004.
6. A. Fumagalli, E. Keilegavlen, and S. Scialò. Conforming, non-conforming and non-matching discretization couplings in discrete fracture network simulations *J. Comput. Phys.*, 376, pp. 694–71, 2019.
7. J.D Hyman, S. Karra, N. Makedonska, C.W. Gable, S.L Painter, H.S. Viswanathan. dfnWorks: A discrete fracture network framework for modeling subsurface flow and transport. *Computers & Geosciences*, 84, pp. 10–19, 2015.
8. S. Berrone, S. Scialò and F. Vicini. Parallel Meshing, Discretization, and Computation of Flow in Massive Discrete Fracture Networks, *SIAM Journal on Scientific Computing*, 41:4, pp. C317-C338, 2019.
9. P. L. George, H. Borouchaki, F. Alauzet, P. Laug, A. Loseille & L. Maréchal, *Maillage, modélisation géométrique et simulation numérique 2 - Métriques, maillages et adaptation de maillages*, ISTE Editions, 412 pages (2018) (English translation to appear).
10. P. Laug & H. Borouchaki, BLSURF – Mesh Generator for Composite Parametric Surfaces – User's Manual, *Inria Technical Report*, RT-0235, 1999.
11. P. Laug and G. Pichot. Mesh Generation and Flow Simulation in Large Tridimensional Fracture Networks. IMACS Series in Computational and Applied Mathematics vol 22, IMACS, Rome, IT, 2019 - ISSN 10-98-870X https://hal.inria.fr/hal-02102811, 2019.
12. H. Borouchaki, P. Laug, and P. L. George. Parametric surface meshing using a combined advancing-front generalized-delaunay approach. *International Journal for Numerical Methods in Engineering*, 49(1–2):233–259, 2000.
13. J.-R de Dreuzy, G. Pichot, B. Poirriez & J. Erhel, Synthetic benchmark for modeling flow in 3D fractured media, *Computers & Geosciences*, 50, pp. 59–71, 2013.
14. P.A. Raviart & J. Thomas, A mixed finite element method for 2nd order elliptic problems, *Mathematical Aspects of the Finite Element Methods, Lectures Notes in Mathematics*, Springer, Berlin, 606, pp. 292–315, 1977.
15. D. N. Arnold & F. Brezzi, Mixed and nonconforming finite element methods: postprocessing, and error estimates, *ESAIM: M2AN*, 19, pp. 7–32, 1985.
16. F. Brezzi & M. Fortin, *Mixed and Hybrid Finite Element Methods*, Springer-Verlag, New York, 1991.
17. G. Chavent & J. E. Roberts, A unified physical presentation of mixed, mixed-hybrid finite elements and standard finite difference approximations for the determination of velocities in waterflow problems, *Advances in Water Resources*, 14 (6), pp. 329–348, 1991.
18. T. Arbogast, L. C. Cowsar, M. F. Wheeler & I. Yotov, Mixed finite element methods on nonmatching multiblock grids, *SIAM J. Numer. Anal.*, 37 (4), pp. 1295–1315, 2000.
19. G. Pichot, J. Erhel, and J.-R. de Dreuzy. A generalized mixed hybrid mortar method for solving flow in stochastic discrete fracture networks. *SIAM Journal on Scientific Computing*, 34(1):B86–B105, 2012.
20. D.A. Di Pietro, A. Ern, and S. Lemaire. An arbitrary-order and compact-stencil discretization of diffusion on general meshes based on local reconstruction operators. *Computational Methods in Applied Mathematics*, 14(4):461–472, 2014.
21. D.A Di Pietro and A. Ern. A hybrid high-order locking-free method for linear elasticity on general meshes. *Comp. Meth. Appl. Mech. Eng.*, 283, 1–21, 2015.

22. F. Chave, D.A. Di Pietro, and L. Formaggia. A hybrid high-order method for Darcy flows in fractured porous media. *SIAM J. Sci. Comput.*, 40, pp. A1063–A1094, 2018.
23. B. Cockburn, D.A. Di Pietro, and A. Ern. Bridging the hybrid high-order and hybridizable discontinuous galerkin methods. *ESAIM: M2AN*, 50(3):635–650, 2016.
24. M. Cicuttin, D.A. Di Pietro, and A. Ern. Implementation of Discontinuous Skeletal methods on arbitrary-dimensional, polytopal meshes using generic programming. *J. Comput. Appl. Math.*, 344, pp. 852–874, 2018.
25. P. Renard & G. de Marsily, Calculating Equivalent Permeability, *Advances in Water Resources*, 20, pp. 253–278, 1997.
26. Gaël Guennebaud and Benoît Jacob et al., Eigen v3, http://eigen.tuxfamily.org, 2010.
27. https://software.intel.com/en-us/mkl, 2019.
28. http://faculty.cse.tamu.edu/davis/suitesparse.html, 2019.

Fully Algebraic Two-Level Overlapping Schwarz Preconditioners for Elasticity Problems

Alexander Heinlein, Christian Hochmuth, and Axel Klawonn

Abstract Different parallel two-level overlapping Schwarz preconditioners with Generalized Dryja–Smith–Widlund (GDSW) and Reduced dimension GDSW (RGDSW) coarse spaces for elasticity problems are considered. GDSW type coarse spaces can be constructed from the fully assembled system matrix, but they additionally need the index set of the interface of the corresponding nonoverlapping domain decomposition and the null space of the elasticity operator, i.e., the rigid body motions. In this paper, fully algebraic variants, which are constructed solely from the uniquely distributed system matrix, are compared to the classical variants which make use of this additional information; the fully algebraic variants use an approximation of the interface and an incomplete algebraic null space. Nevertheless, the parallel performance of the fully algebraic variants is competitive compared to the classical variants for a stationary homogeneous model problem and a dynamic heterogenous model problem with coefficient jumps in the shear modulus; the largest parallel computations were performed on 4096 MPI (Message Passing Interface) ranks. The parallel implementations are based on the Trilinos package FROSch.

1 Introduction

We consider the solution of large, parallel distributed stationary and dynamic discretized elasticity problems with a moderate Poisson ratio; i.e., we do not consider the nearly incompressible case. As the solver, we use the Generalized

A. Heinlein (✉) · A. Klawonn
Department of Mathematics and Computer Science, University of Cologne, Köln, Germany

Center for Data and Simulation Science, University of Cologne, Köln, Germany
e-mail: alexander.heinlein@uni-koeln.de; http://www.cds.uni-koeln.de

C. Hochmuth
Department of Mathematics and Computer Science, University of Cologne, Köln, Germany

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_52

Minimal Residual (GMRES) method preconditioned by two-level overlapping Schwarz preconditioners with Generalized Dryja–Smith–Widlund (GDSW) [2, 3] and Reduced dimension GDSW (RGDSW) [5, 12] coarse spaces. Even though these preconditioners can be constructed from the fully assembled system matrix, a minimum of geometric information is also needed. In particular, the domain decomposition interface and the null space are used for their construction. Here, we focus on the construction of fully algebraic GDSW type coarse spaces if this information is not available. In particular, we consider the case when the system matrix is uniquely distributed, such that the interface cannot be identified.

Therefore, we will describe a method to approximate the nonoverlapping subdomains, resulting in an approximate interface; cf. [10]. Our parallel implementation is based on the `FROSch` framework [9], which is part of the `ShyLU` package in Trilinos [13]; see [10, 11] for more details on the implementation. To discuss the performance of the fully algebraic approach, we will compare it to the classical GDSW type coarse spaces using all necessary information.

2 Model Problems

The equilibrium equation for an elastic body covering the domain $\Omega = [0, 1]^3$ under the action of a body force f in the time interval $[0, T]$ is

$$\partial_{tt}\mathbf{u} - \operatorname{div}\boldsymbol{\sigma} = f \qquad \text{in } \Omega \times [0, T], \tag{1}$$

with the symmetric Cauchy stress tensor $\boldsymbol{\sigma}$ and the displacement $\mathbf{u}$. We consider a Saint Venant-Kirchhoff material, a hyperelastic material with the linear material law

$$\boldsymbol{\sigma}(\mathbf{E}) = \frac{\lambda}{2}\,(\operatorname{trace}\mathbf{E})^2 + \mu\operatorname{trace}\mathbf{E}I \tag{2}$$

and the nonlinear strain tensor given by $\mathbf{E} := \frac{1}{2}\left(\mathbf{C} - I\right)$, where $\mathbf{C}$ is the right Cauchy-Green tensor. Furthermore, we consider the boundary conditions

$$\begin{aligned} \mathbf{u} &= 0 && \text{on } \partial\Omega_D := \{0\} \times [0, 1]^2, \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= 0 && \text{on } \partial\Omega_N := \partial\Omega \setminus \partial\Omega_D, \end{aligned}$$

and the body force $f = (-20, 0, 0)^T$, for $t < 5 \cdot 10^{-3}$, and $f = 0$, afterwards.

In addition to this, we also consider a stationary problem with $\partial_{tt}\mathbf{u} = 0$, i.e.,

$$\begin{aligned} \operatorname{div}\boldsymbol{\sigma} &= (0, -100, 0)^T && \text{in } \Omega, \\ \mathbf{u} &= 0 && \text{on } \partial\Omega_D := \{0\} \times [0, 1]^2, \\ \boldsymbol{\sigma} \cdot \mathbf{n} &= 0 && \text{on } \partial\Omega_N := \partial\Omega \setminus \partial\Omega_D. \end{aligned} \tag{3}$$

We transform the model problems to their respective variational formulations and discretize them using piecewise linear or quadratic finite elements; we denote the corresponding finite element spaces by $V^h = V^h(\Omega)$. For the time-dependent problem, the resulting semi-discrete problem is further discretized with the Newmark-β method. In particular, we choose the parameters for the fully implicit constant average acceleration method, i.e., $\beta = 1/2$ and $\gamma = 1/4$.

The fully discrete nonlinear equations are linearized using Newton's method. We solve the equation

$$J(u^{(k)})\delta u^{(k+1)} = R(u^{(k)}), \tag{4}$$

for the update $\delta u^{(k+1)}$. Here, $J(u^{(k)})$ and $R(u^{(k)})$ are the Jacobian and the nonlinear residual for the solution $u^{(k)}$, respectively.

2.1 GDSW and RGDSW Preconditioners

We consider the system of linear equations (4) as derived in the previous section. For simplicity, we use the notation $Ax = b$ in this section.

Let Ω be decomposed into nonoverlapping subdomains $\{\Omega_i\}_{i=1}^N$ with typical diameter H and corresponding overlapping subdomains $\left\{\Omega_i'\right\}_{i=1}^N$, resulting from extending the nonoverlapping subdomains by k layers of elements. We define $R_i : V^h \to V_i^h$, $i = 1, \ldots, N$, as the restriction from the global finite element space V^h to the local finite element space $V_i^h := V^h\left(\Omega_i'\right)$ and corresponding prolongation operators R_i^T. In addition to that, we can also define restricted and scaled prolongation operators $\tilde{R}_i^T$; cf., e.g., [1, 4, 7].

Furthermore, let

$$\Gamma := \left\{x \in (\overline{\Omega}_i \cap \overline{\Omega}_j) \setminus \partial\Omega_D : i \neq j, 1 \leq i, j \leq N\right\}$$

be the interface of the nonoverlapping domain decomposition.

The GDSW preconditioner, which was introduced by Dohrmann, Klawonn, and Widlund in [2, 3], is a two-level additive overlapping Schwarz preconditioner with energy minimizing coarse space and exact solvers. Thus, the preconditioner can be written in the form

$$M_{\text{GDSW}}^{-1} = \Phi A_0^{-1} \Phi^T + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i, \tag{5}$$

where $A_i = R_i A R_i^T$. In the second level, we solve the coarse problem matrix $A_0 = \Phi^T A \Phi$. The columns of Φ are the basis functions of the coarse space. To construct the GDSW coarse basis functions, let $R_{\Gamma j}$ be the restriction from Γ

onto the interface component Γ_j. For the GDSW coarse space in three dimensions, the interface components are the vertices, edges, and faces. Then, the values of the GDSW basis functions on Γ read

$$\Phi_\Gamma = \left[R_{\Gamma_1}^T \Phi_{\Gamma_1} \; \ldots \; R_{\Gamma_M}^T \Phi_{\Gamma_M} \right],$$

where the columns of Φ_{Γ_j} are the restrictions of the null space of subdomain Neumann matrices to the interface component Γ_j. For elasticity, the null space consists of the rigid body motions, i.e., the translations and rotations. After partitioning the degrees of freedom into interface (Γ) and interior (I) ones, the matrix A can be written as

$$A = \begin{bmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{bmatrix}$$

and the GDSW coarse basis functions are the discrete harmonic extensions of Φ_Γ into the interior,

$$\Phi = \begin{bmatrix} \Phi_I \\ \Phi_\Gamma \end{bmatrix} = \begin{bmatrix} -A_{II}^{-1} A_{I\Gamma} \Phi_\Gamma \\ \Phi_\Gamma \end{bmatrix}. \tag{6}$$

The RGDSW coarse space is constructed similarly. However, in general, we only obtain one basis function for each vertex, resulting in a much smaller dimension of the coarse space; cf. [5] and, for more details on the parallel implementation in FROSch, [7, 12]. The reduction of the coarse space dimension can also be seen in Table 1. There are several variants of RGDSW coarse spaces, which differ in a scaling of the interface degrees of freedom. Here, we will only consider the most algebraic variant, which is denoted as Option 1 in [5]; cf. [7] for a detailed description of our implementation of Option 1 of the RGDSW coarse space.

In our numerical simulations, we will also employ the recycling strategies presented in [7]. We always reuse the symbolic factorizations from previous time

Table 1 Comparison of coarse matrix sizes for a structured domain decomposition and the approximated subdomain maps for a P1 ($H/h = 21$) and P2 ($H/h = 9$) discretization

	#cores	64	512	4096
GDSW	Rotations	1593	16,149	144,045
	No rotations	837	8589	77,085
	Algebraic P1 disc.	1395	11,355	84,762
	Algebraic P2 disc.	1554	11,466	84,708
RGDSW	Rotations	162	2058	20,250
	No rotations	81	1029	10,125
	Algebraic P1 disc.	93	1065	10,218
	Algebraic P2 disc.	93	1038	10,134

or Newton iterations. Moreover, we reuse the coarse space from previous iterations and, for the dynamic problem, additionally the coarse matrix. Furthermore, as in [7], we always use a scaled first level operator with overlap $\delta = 1h$.

3 Fully Algebraic Construction of GDSW and RGDSW Coarse Spaces

As previously described, the construction of GDSW and RGDSW coarse spaces for elasticity problems requires both the domain decomposition interface and the null space of the operator, i.e., the rigid body motions. Here, we describe how we construct the coarse space if this information is not available.

Algebraic Approximation of the Interface If the distribution of the system matrix is unique, the interface cannot be recovered. Therefore, we will carry out the following process to approximate the nonoverlapping subdomains and hence the interface. Starting from the unique distribution, we first add one layer of elements to each subdomain. The overlap of the resulting domain decomposition now contains the interface but also other finite element nodes. In order to reduce the number of unnecessary nodes, we compare the subdomain ID of the original unique decomposition and the decomposition with one layer of overlap and remove nodes from the overlapping subdomains if the subdomain ID is lower compared to the original decomposition; this process is sketched in [10] and Fig. 1.

Incomplete Null Space The rigid body modes are the translations and rotations of the elastic body. The translations are constant functions which can be constructed without any geometric information. Since we are not able to compute the rotations from the fully assembled matrix and without coordinates of the finite element nodes, we just omit them in the fully algebraic coarse space; see also [11]. For the results

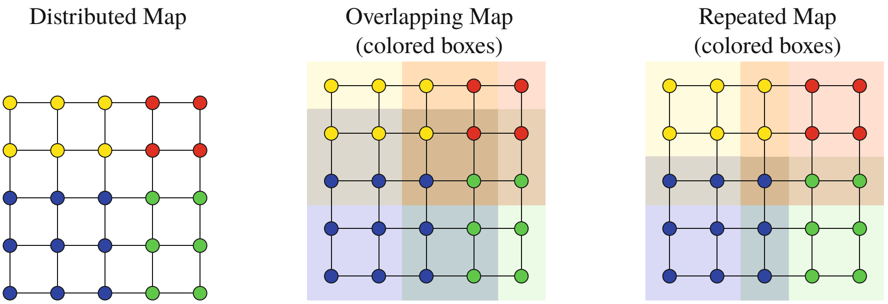


Fig. 1 Sketch of the approximation of the nonoverlapping subdomains and the interface, respectively: uniquely distributed map (left); extension of the uniquely distributed map by one layer of elements resulting in an overlapping map, where the overlap contains the interface (middle); by selection, using the lower subdomain ID, the a map approximating to the nonoverlapping subdomains is constructed (right). Taken from [10]

in Sect. 4, only the number of iterations is negatively affected by omitting rotations from the coarse space but the time to solution actually benefits from the smaller coarse space. Note that, from theory, the rotational basis functions are necessary for numerical scalability. Therefore, we expect that there are problems for which the full coarse space performs better.

4 Numerical Results

In this section, we compare the GDSW and RDSW preconditioners with exact interface maps and full coarse space, GDSW and RGDSW preconditioners with exact interface map but without rotational basis functions, and the fully algebraic variant with approximated interface and without rotational basis functions; for the sake of brevity, we denote the three variants as “rotations”, “no rotations”, and “algebraic”, respectively. As discussed in Sect. 2, we consider a stationary elasticity problem with homogeneous shear modulus of $\mu = 5 \cdot 10^3$ and a dynamic elasticity problem with two material phases; cf. Fig. 2 (left) for a graphical representation of the coefficient distribution of the shear modulus. For both cases, we choose $\nu = 0.4$. For the stationary homogeneous model problem, we use structured grids and structured decompositions into square subdomains, whereas for the dynamic problem, we use a fixed unstructured tetrahedral mesh with roughly 3.3 million elements and 588 k nodes. We use the inexact Newton method of Eisenstat and Walker [6] with a type 2 forcing term until a relative residual of $\epsilon_{\text{nl}} = 10^{-8}$ is achieved. The initial forcing term is $\eta_{\text{init}} = 10^{-3}$ and the maximum and minimum forcing terms are $\eta_{\max} = 10^{-2}$ and $\eta_{\min} = 10^{-8}$, respectively. Therefore, we use a

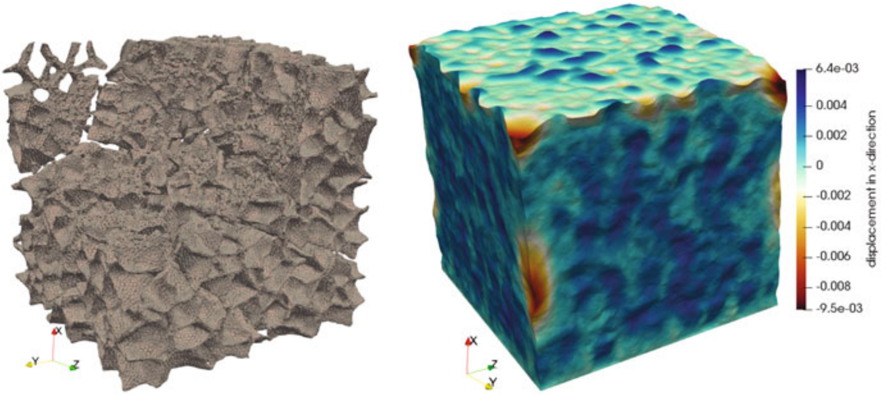


Fig. 2 Left: Slice through elements with high coefficient ($\mu_{\text{high}} = 10^3$) displayed as a wireframe. Low coefficient is $\mu_{\text{low}} = 1$; cf. [8], for a detailed discussion of the foam geometry used for an heterogeneous Poisson problem. Right: Solution of dynamic problem at $T = 10^{-2}$ for $\Delta t = 10^{-3}$ with a warp filter and a 5 times scaling factor

combination of the Trilinos packages Thyra and NOX. Furthermore, NOX is used for a backtracking globalization strategy. In particular, the step length is chosen as 0.5^l with $l = 0, 1, \ldots$ until the Armijo condition is satisfied. All linearized problems are solved with right-preconditioned GMRES with the corresponding GDSW and RGDSW preconditioners and the tolerance for the relative residual error is the forcing term η. All computations were carried out on the supercomputer magnitUDE of the University Duisburg-Essen, Germany.

In Tables 2 and 3, weak scaling results for the stationary model problem with piecewise linear and piecewise quadratic elements are depicted. Although, iteration counts are slightly higher for the RGDSW coarse spaces compared to the respective

Table 2 Stationary problem, discretization P1 ($H/h = 21$), iteration counts are averages over all Newton iterations. All problems were solved in 4 Newton iterations. The three timings are for the setup/solve/total time and are in seconds. All total times are highlighted

Prec.	Type	#cores	64	512	4096
GDSW	Rot.	#its.	17.8	19.0	19.0
		time	35.1 / 7.4 / **42.5**	45.3 / 9.7 / **55.0**	167.1 / 26.1 / **183.2**
	No rot.	#its.	27.3	32.0	35.5
		time	29.3 / 10.6 / **39.9**	32.9 / 13.8 / **46.7**	70.8 / 23.3 / **94.1**
	Algebraic	#its.	32.8	38.5	39.0
		time	39.5 / 13.4 / **52.9**	41.6 / 17.2 / **58.8**	84.3 / 27.3 / **111.6**
RGDSW	Rot.	#its.	20.5	22.5	22.5
		time	28.8 / 8.2 / **37.0**	30.9 / 9.5 / **40.4**	42.0 / 11.7 / **53.7**
	No rot.	#its.	33.0	37.3	39.5
		time	25.2 / 12.4 / **37.6**	26.5 / 14.7 / **41.2**	30.1 / 18.0 / **48.1**
	Algebraic	#its.	40.0	42.0	43.0
		time	27.2 / 15.5 / **42.7**	28.7 / 16.8 / **45.5**	32.9 / 19.6 / **52.5**

Table 3 Stationary problem, discretization P2 ($H/h = 9$), iteration counts are averages over all Newton iterations. All problems were solved in 4 Newton iterations. The three timings are for the setup/solve/total time and are in seconds. All total times are highlighted

Prec.	Type	#cores	64	512	4096
GDSW	Rot.	#its.	16.3	17.3	19.3
		time	40.1 / 5.9 / **46.0**	55.0 / 8.5 / **63.5**	223.3 / 24.4 / **247.7**
	No rot.	#its.	24.5	29.3	32.3
		time	32.5 / 8.4 / **40.9**	38.4 / 11.8 / **46.7**	102.2 / 20.0 / **122.2**
	Algebraic	#its.	57.5	74.8	78.0
		time	42.0 / 20.5 / **62.5**	46.0 / 29.9 / **75.9**	124.8 / 50.5 / **175.3**
RGDSW	Rot.	#its.	18.8	21.3	19.8
		time	27.8 / 6.4 / **34.2**	31.1 / 8.0 / **39.1**	41.3 / 8.9 / **50.2**
	No rot.	#its.	29.0	32.8	35.5
		time	26.2 / 9.4 / **35.6**	27.3 / 11.8 / **39.1**	31.1 / 14.3 / **45.4**
	algebraic	#its.	60.7	78.5	83.0
		time	27.9 / 19.9 / **47.8**	28.7 / 27.9 / **56.6**	34.1 / 33.1 / **67.2**

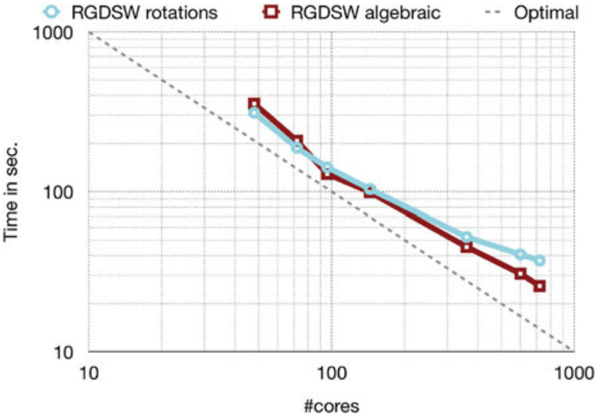


Fig. 3 Strong scaling for dynamic problem up to time $T = 2 \cdot 10^{-2}$ for the foam geometry

GDSW coarse spaces, the total computation time is much smaller for RGDSW due to the lower dimension of the coarse problem. This effect is even stronger for larger numbers of subdomains and cores; cf. Table 1. Furthermore, we observe competitive iteration counts and computing times when using the fully algebraic coarse spaces. In addition to that, the approximation strategy for the interface seems to perform better for piecewise linear than for piecewise quadratic elements.

In Fig. 3, we present strong scaling results from 48 to 720 cores for the dynamic model problem. The reported times are the total times for our preconditioners, i.e., the sum of the times needed for their construction and their applications in GMRES. We solve the problem with $\Delta t = 10^{-3}$ up to a final time $T = 2 \cdot 10^{-2}$ using the RGDSW rotations coarse space and using the RGDSW algebraic coarse space both with matrix recycling; cf. [7]. Here, we observe very good strong scalability results for both variants even though the model problem has coefficient jumps. Again, the fully algebraic variant is competitive.

Acknowledgments The authors gratefully acknowledge financial support by the German Science Foundation (DFG), project no. 214421492 and the computing time granted by the Center for Computational Sciences and Simulation (CCSS) of the University of Duisburg-Essen and provided on the supercomputer magnitUDE (DFG grants INST 20876/209-1 FUGG, INST 20876/243-1 FUGG) at the Zentrum für Informations- und Mediendienste (ZIM). We further thank Jascha Knepper for providing the foam geometry.

References

1. X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM Journal on Scientific Computing*, 21:239–247, 1999.
2. C. R. Dohrmann, A. Klawonn, and O. B. Widlund. Domain decomposition for less regular subdomains: overlapping Schwarz in two dimensions. *SIAM J. Numer. Anal.*, 46(4):2153–2168, 2008.

3. C. R. Dohrmann, A. Klawonn, and O. B. Widlund. A family of energy minimizing coarse spaces for overlapping Schwarz preconditioners. In *Domain decomposition methods in science and engineering XVII*, volume 60 of *Lect. Notes Comput. Sci. Eng.*, pages 247–254. Springer, Berlin, 2008.
4. C. R. Dohrmann and O. B. Widlund. Hybrid domain decomposition algorithms for compressible and almost incompressible elasticity. *Internat. J. Numer. Meth. Engng*, 82(2):157–183, 2010.
5. C. R. Dohrmann and O. B. Widlund. On the design of small coarse spaces for domain decomposition algorithms. *SIAM J. Sci. Comput.*, 39(4):A1466–A1488, 2017.
6. S. C. Eisenstat and H. F. Walker. Choosing the forcing terms in an inexact Newton method. volume 17, pages 16–32. 1996. Special issue on iterative methods in numerical linear algebra (Breckenridge, CO, 1994).
7. A. Heinlein, C. Hochmuth, and A. Klawonn. Reduced dimension GDSW coarse spaces for monolithic Schwarz domain decomposition methods for incompressible fluid flow problems. *International Journal for Numerical Methods in Engineering*, 121(6):1101–1119, 2020.
8. A. Heinlein, A. Klawonn, J. Knepper, and O. Rheinbach. Adaptive GDSW coarse spaces for overlapping Schwarz methods in three dimensions. *SIAM Journal on Scientific Computing*, 41(5):A3045–A3072, 2019.
9. A. Heinlein, A. Klawonn, S. Rajamanickam, and O. Rheinbach. FROSch – a parallel implementation of the GDSW domain decomposition preconditioner in Trilinos. In preparation.
10. A. Heinlein, A. Klawonn, S. Rajamanickam, and O. Rheinbach. FROSch: A Fast and Robust Overlapping Schwarz Domain Decomposition Preconditioner Based on Xpetra in Trilinos. Technical report, Universität zu Köln, November 2018.
11. A. Heinlein, A. Klawonn, and O. Rheinbach. A parallel implementation of a two-level overlapping Schwarz method with energy-minimizing coarse space based on Trilinos. *SIAM J. Sci. Comput.*, 38(6):C713–C747, 2016.
12. A. Heinlein, A. Klawonn, O. Rheinbach, and O. B. Widlund. Improving the parallel performance of overlapping Schwarz methods by using a smaller energy minimizing coarse space. International Conference on Domain Decomposition Methods, pages 383–392. Springer, 2017.
13. M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley. An overview of the Trilinos project. *ACM Trans. Math. Softw.*, 31(3):397–423, 2005.

Stationary Flow Predictions Using Convolutional Neural Networks

Matthias Eichinger, Alexander Heinlein, and Axel Klawonn

Abstract Computational Fluid Dynamics (CFD) simulations are a numerical tool to model and analyze the behavior of fluid flow. However, accurate simulations are generally very costly because they require high grid resolutions. In this paper, an alternative approach for computing flow predictions using Convolutional Neural Networks (CNNs) is described; in particular, a classical CNN as well as the U-Net architecture are used. First, the networks are trained in an expensive offline phase using flow fields computed by CFD simulations. Afterwards, the evaluation of the trained neural networks is very cheap. Here, the focus is on the dependence of the stationary flow in a channel on variations of the shape and the location of an obstacle. CNNs perform very well on validation data, where the averaged error for the best networks is below 3%. In addition to that, they also generalize very well to new data, with an averaged error below 10%.

1 Introduction

Computational Fluid Dynamics (CFD) simulations are a numerical tool to model and analyze the behavior of fluid flow. They are used in a wide range of application areas, such as, e.g., civil and mechanical engineering, meteorology or medical science, a wide range of different fluids and settings. In CFD simulations, the input parameters are classically the material parameters of the fluid, such as density and viscosity, the geometry of the computational domain, and the boundary conditions

M. Eichinger
Department of Mathematics and Computer Science, University of Cologne, Köln, Germany
e-mail: eichingm@smail.uni-koeln.de

A. Heinlein (✉) · A. Klawonn
Department of Mathematics and Computer Science, University of Cologne, Köln, Germany

Center for Data and Simulation Science, University of Cologne, Köln, Germany
e-mail: alexander.heinlein@uni-koeln.de; axel.klawonn@uni-koeln.de; http://www.cds.uni-koeln.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_53

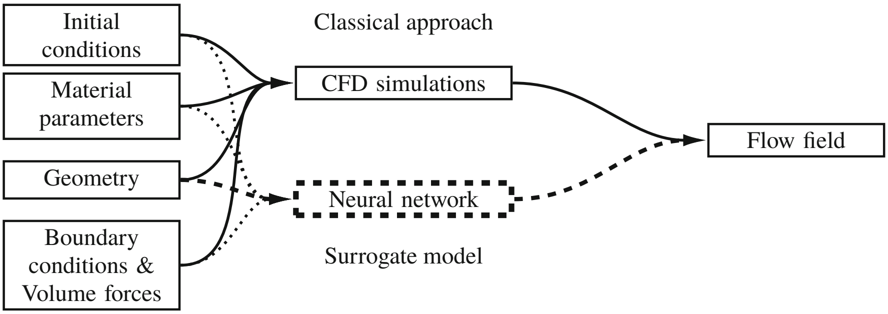


Fig. 1 Our approach is to train a neural network as a surrogate model for CFD simulations. Here, we do not consider varying initial conditions, material parameters, boundary conditions, or volume forces but focus on varying geometries for the computational domain

as well as volume forces. Transient simulations additionally depend on the initial condition. Depending on the underlying model for the fluid flow, the resulting flow field may depend on all these input parameters in a highly nonlinear way. Furthermore, CFD simulations often require a high spatial and temporal resolution in order to obtain accurate results. Therefore, CFD simulations are generally very compute intensive.

The complexity of CFD simulations may be reduced using, e.g., Proper Orthogonal Decomposition (POD), Reduced Basis (RB), or simplified physics methods; these techniques are all Model Order Reduction (MOR) techniques. In this work, we propose a different approach, which can also be regarded as a MOR technique. In particular, we propose to use appropriate neural networks as surrogate models for CFD simulations; cf. Fig. 1. As for MOR techniques, we will have to perform many CFD simulations in advance in a very expensive offline phase. However, the evaluation of the trained model will then be much faster compared to a CFD simulation. Here, we focus on predicting the fluid flow with respect to variations in the geometry of the computational domain. Therefore, we consider a steady flow problem in order to eliminate the time-dependence of the flow field and the dependence of the solution on an initial condition. Furthermore, we keep the material parameters, as well as boundary conditions and volume forces constant.

A different approach for the prediction of fluid flow using neural networks for fixed geometries can be found in, e.g., [9, 10].

Our approach is inspired by the work of Guo, Li, and Iorio [6], where the authors used a Convolutional Neural Network (CNN) to predict the steady flow around obstacles in a channel. In our work, we further extend this approach by using the more complex network architecture of the U-Net, which was introduced in [11], and considering different types of loss functions. In particular, we will compute synthetic training data from CFD simulations using OpenFOAM 5.0 [5] and train CNNs using Keras 2.2.4 [2] with Tensorflow 1.12 [1] backend to approximate the resulting flow fields.

This paper is organized as follows: in Sect. 2, we describe our model problem and the computation of the reference data using CFD simulations. Next, we describe the CNN architectures and the training settings used for our surrogate model in Sect. 3. Section 4 shows the performance of our models on training and validation data as well as the generalization properties for some types of unseen data. Finally, we present a short conclusion and outlook.

2 CFD Simulations

Let us consider computational domains $\Omega_P := [0,6] \times [0,3] \setminus P$, where $P \subset [0,6] \times [0,3]$ is a polygonal star-shaped domain; see Fig. 2.

The stationary flow of an incompressible Newtonian fluid with kinematic viscosity $\nu > 0$ within the computational domain Ω_P is modeled by the steady Navier-Stokes equations,

$$\begin{aligned} -\nu \Delta u + (u \cdot \nabla)\, u + \nabla p &= f \text{ in } \Omega, \\ \nabla \cdot u &= 0 \text{ in } \Omega, \end{aligned} \tag{1}$$

with velocity u and pressure p. Now, let $\partial\Omega_{\text{in}} := 0 \times [0,3]$ and $\partial\Omega_{\text{out}} := 6 \times [0,3]$ be the inlet and outlet, respectively, and $\partial\Omega_{\text{wall}} := ([0,6] \times 0) \cup ([0,6] \times 3) \cup \delta P$ be the remainder of the boundary of Ω. We prescribe

$$\begin{aligned} u &= 3 \text{ on } \partial\Omega_{\text{in}}, \\ \frac{\partial u}{\partial n} - pn &= 0 \text{ on } \partial\Omega_{\text{out}}, \text{ and} \\ u &= 0 \text{ on } \partial\Omega_{\text{wall}} \end{aligned}$$

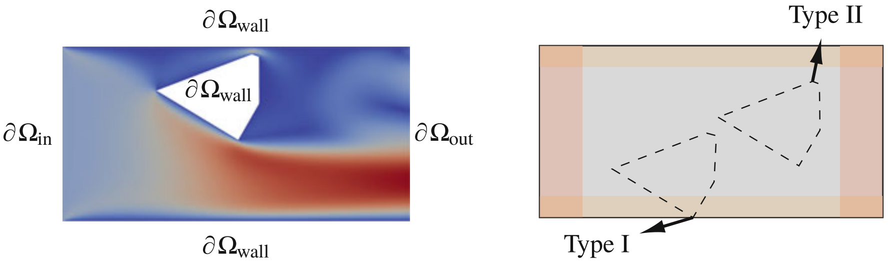

$\partial\Omega_{\text{wall}}$
$\partial\Omega_{\text{wall}}$
$\partial\Omega_{\text{in}}$
$\partial\Omega_{\text{out}}$
$\partial\Omega_{\text{wall}}$
Type II
Type I

Fig. 2 Left: the computational domain is a channel of length 6 and width 3 with a polygonal obstacle. Right: type I obstacles are connected with the bottom wall, type II obstacles have a distance of 0.75 to each part of the boundary (yellow). Both types of obstacles have a distance of 1.5 to the inlet and the outlet (red) and may not cover more than 50% of the cross section of the channel

as boundary conditions; cf. Fig. 2 for an exemplary resulting flow field. Here, n is the outward pointing normal vector.

In order to perform the CFD simulations, we employ the CFD software OpenFOAM 5.0 [5], which is based on the FVM (Finite Volume Method). In particular, we first use SnappyHexMesh to generate compute meshes from STL (Standard Triangle Language) files that describe the polygonal obstacles. Secondly, we compute corresponding stationary flow fields using the SimpleFoam solver.

Since we fix all parameters and boundary conditions, the resulting flow field only depends on the shape and location of the polygonal obstacle P. As shown in Fig. 2, we only consider two different types of obstacles here: obstacles that are connected with the bottom wall and obstacles that are not connected with any part of the boundary of the channel.

3 Surrogate Convolutional Neural Network

To predict fluid flow using neural networks, we fix the structure of the input and output data of our models. Whereas, in numerical CFD simulations, the structure and size of the compute mesh and the solution vector may differ significantly for different configurations, neural networks rely on structured data. Therefore, our approach is to convert both the input, i.e., the description of the obstacle geometry, and the output, i.e., the flow field, to 256 × 128 pixel images; cf. Fig. 3. As input, we either use a binary representation of the geometry, i.e., 0 if the center of a pixel is covered by the obstacle and 1 otherwise, or a Signed Distance Function (SDF) representation, i.e., the value in each pixel is the smallest distance of its center to the boundary of the obstacle multiplied with -1 if the center lies within the obstacle. As output, we interpolate the x and y components of the flow field to 256×128 pixel images.

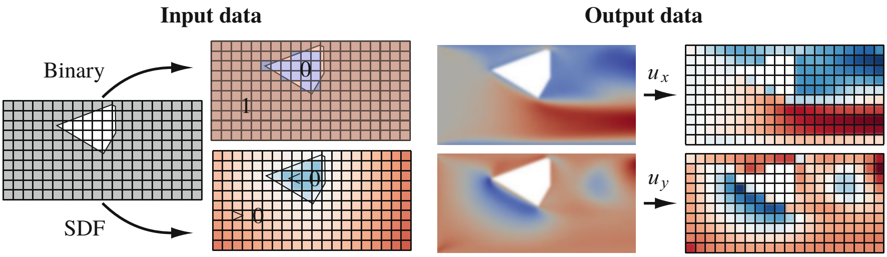


Fig. 3 Generation of structured input data (left) and output data (right) for training the neural networks. As input, we generate a 256 × 128 pixel image with a binary or SDF representation of the obstacle. As output, we interpolate the components u_x and u_y of the flow field on a 256 × 128 pixel image

Due to their good performance on image data, we apply CNNs in order to approximate the nonlinear relation between out input and output data. In particular, we consider the CNN used in [6] as well as the U-Net [11] with both one and two decoder paths and Rectified Linear Unit (ReLU) activation; for more details on neural network, see, e.g., [4, 12].

Furthermore, we consider a total of 100,000 data sets (50,000 type I and 50,000 type II obstacles) consisting of equally many polygons with 3, 4, 5, 6, and 12 edges, respectively; the shapes and sizes of the polygons are randomly chosen under the conditions described in Fig. 2. Out of the 100,000 data sets, we randomly select 90,000 as training data and 10,000 as validation data. We optimize applying a Stochastic Gradient Descent (SGD) method with a batch size of 64 and an adaptive scaling of the learning rate using the Adam (Adaptive moments) [8] algorithm with an initial learning rate $\lambda = 0.001$. We use a maximum of 300 epochs for the training and reduce the learning rate by 20% in case of stagnation for more than 50 epochs. In case of SDF input data, we apply Z-normalization, and in case of binary input data, we use batch normalization [7]. Our implementation uses Keras 2.2.4 [2] with Tensorflow 1.12 [1] backend.

Even though, our neural networks may be very complex with approximately 50 million parameters on average, their evaluation is still much cheaper compared to corresponding CFD simulations; on a single core of an AMD Threadripper 2950X (8 × 3.8 Ghz), the evaluation of our neural network models (less than 0.01 s) was more than two orders of magnitude faster than the average CFD simulation (approximately 50 s). On GPU (Graphics Processing Unit) architectures, the speedup will be even larger. However, the training of the neural networks, which includes the computation of the training data using CFD simulations, may take hours or even days.

We refer to [3] for a detailed discussion of the employed models, our software framework, as well as a more detailed discussion the different types of obstacles and a discussion of techniques for efficient generalization to other types of obstacles. In [3], we will also discuss the speedup of our neural networks compared to the CFD simulations in more detail.

4 Results

In order to measure the performance of our neural networks, we first introduce our error measure. Therefore, let V be the set of all polygons P in the set of validation data and I_P be the set of all non-obstacle pixels for one specific polygonal obstacle P. As the error measure to evaluate our neural network models, we consider the averaged relative error

$$\frac{1}{|V|} \sum_{P \in V} \frac{1}{|I_P|} \sum_{p \in I_P} \frac{\|u_p - \hat{u}_p\|_2}{\|u_p\|_2 + 10^{-4}}, \tag{2}$$

with u_p and $\hat{u}_p$ being the reference velocity, computed in a CFD simulation, and the predicted velocity, computed by evaluating the neural networks, respectively. The term 10^{-4} acts as a regularization in case of very low reference velocities u_p.

We compare the CNN from [6] to the U-Net [11] using one and two decoder paths using binary and SDF input data. Furthermore, we will observe that the performance depends significantly on the loss function used to train the neural network. As the loss function, we compare four different choices, i.e., the Mean Squared Error (MSE), the sum of the MSE and the averaged relative error (2), the Mean Absolute Error (MAE), and the sum of the MAE and the averaged relative error (2).

Performance on the Original Data As can be seen in Figs. 4 and 5, the prediction may be very good but may also show qualitative and quantitative differences to the reference solution. However, if a good combination of the network architecture,

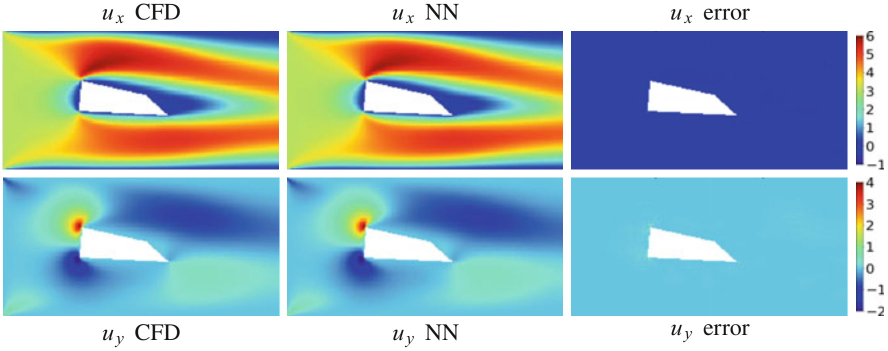


Fig. 4 Comparison of the ground truth flow field (left) computed in a CFD simulation, the prediction by a neural network (middle), and the pointwise error (right) for an example with low averaged relative error (2) is 2%

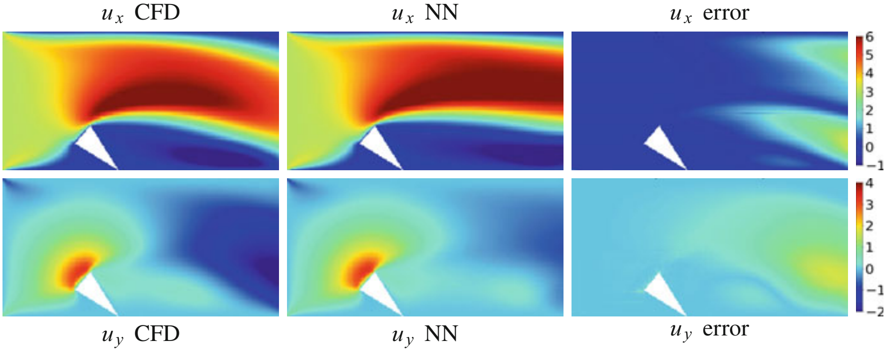


Fig. 5 Comparison of the ground truth flow field (left) computed in a CFD simulation, the prediction by a neural network (middle), and the pointwise error (right) for an example with higher averaged relative error (2) is 31%

Table 1 Comparison of the performance of different CNN models based on the error (2). The best error rates for a given CNN architecture and input type are marked in **bold face**

			CNN [6]			U-Net [11]		
Input	# Dec.	Loss	Total	Type I	Type II	Total	Type I	Type II
SDF	1	MSE	61.16%	110.46%	11.86%	17.04%	29.42%	4.66%
		MSE+(2)	3.97%	3.31%	**4.63%**	2.67%	2.11%	3.23%
		MAE	25.19%	41.52%	8.86%	9.10%	13.89%	4.32%
		MAE+(2)	4.45%	3.84%	5.05%	2.48%	1.87%	**3.10%**
	2	MSE	49.82%	89.12%	10.51%	13.01%	21.59%	4.42%
		MSE+(2)	**3.85%**	**3.05%**	4.64%	**2.43%**	**1.78%**	3.23%
		MAE	45.23%	81.38%	9.08%	5.47%	7.06%	3.89%
		MAE+(2)	4.33%	3.74%	4.91%	2.57%	1.98%	3.17%
Binary	1	MSE	49.78%	88.28%	11.28%	27.15%	49.15%	5.15%
		MSE+(2)	10.12%	11.44%	8.80%	5.49%	6.25%	4.74%
		MAE	39.16%	64.77%	13.54%	15.69%	26.36%	5.02%
		MAE+(2)	10.61%	12.34%	8.87%	**4.48%**	**5.05%**	**3.90%**
	2	MSE	51.34%	91.20%	11.48%	24.00%	43.14%	4.85%
		MSE+(2)	10.03%	11.37%	8.69%	5.56%	6.79%	4.33%
		MAE	37.16%	62.01%	12.32%	21.54%	38.12%	4.96%
		MAE+(2)	**9.53%**	**10.91%**	**8.15%**	6.04%	7.88%	4.20%

the type of input data, the number of decoder paths, and the loss function is used, the average performance of the CNN model over all data is very convincing; see Table 1. Compared to the worst configuration with a total averaged error of 61.16%, the optimal configuration, using the U-Net [11], SDF input, two decoder paths, and MSE+(2) loss function, the total averaged error can be reduced to 2.43%. Using only one decoder path and MAE+(2) loss function yields comparable results but results in a reduction of the number of parameters of the CNN by more than 30%.

The choice of the network architecture and the loss function have the greatest influence on the performance of the model. In particular, the U-Net generally performs much better than the other CNN, and a combination of MSE or MAE with (2) improves then performance significantly compared to only using MSE or MAE.

Generalization Properties In order to investigate the generalization properties, we now only consider the U-Net architecture with one decoder path, and MAE+(2) loss function and only vary the type of input data. This model performed very well for the training and validation data but is more efficient compared to the models with two decoder paths. In Fig. 6, we present the flow field for a circular obstacle, which was not part of the training and validation data. It can be observed that the averaged relative error (2) for this example is only 3%. The very good generalization properties of our model are also apparent in the results in Table 2, which shows the results for several different types of polygonal obstacles which were not part of the training and validation data. The maximum total averaged error is below 10%.

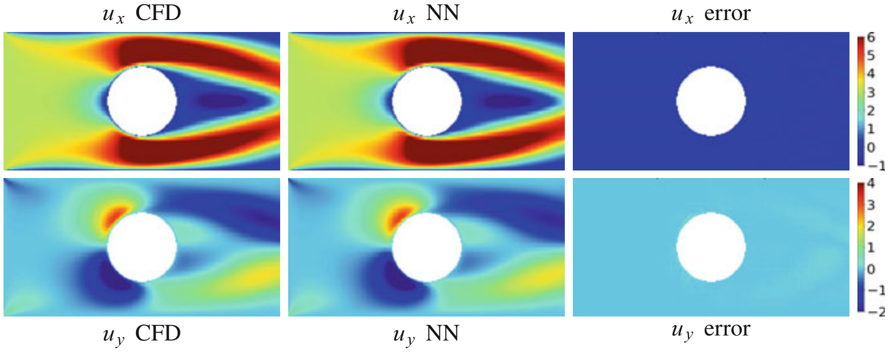


Fig. 6 Generalization properties for the U-Net with one decoder path, MAE+(2) loss function, and SDF input data: comparison the ground truth flow field (left) computed in a CFD simulation, the prediction by the neural network (middle), and the pointwise error (right). Circular obstacles were not part of the training data of the CNN. The averaged relative error (2) is 3%

Table 2 Generalization properties of the U-Net with one decoder path, and MAE+(2) loss function. Error (2) for polygon types which were not in the data set used for training: 1000 polygons (500 type I and 500 type II) for each different number of edges

	SDF input			Binary input		
Polygon # Edges	Total	Type I	Type II	Total	Type I	Type II
7	2.71%	1.89%	3.53%	4.39%	4.61%	4.16%
8	2.82%	1.98%	3.65%	4.67%	4.89%	4.44%
10	3.21%	2.32%	4.10%	5.23%	5.51%	4.94%
15	4.01%	3.16%	4.86%	7.76%	7.85%	6.66%
20	5.08%	4.22%	5.93%	9.70%	10.43%	8.97%

These results confirm that the neural network is able to generalize to other types of polygonal obstacles and is not overfitted to training data.

5 Conclusion

In this work, we have shown that CNNs may serve as efficient surrogate models for CFD simulations. We have focussed on the dependence of the flow field on the geometry of the computational domain, and the extension of this framework to, e.g., varying boundary conditions or material parameters will be future work.

Due to the limited available space, we are not able to discuss, e.g, further generalization techniques for our models or to compare computing times of the CFD simulations and the CNNs in detail. We refer to [3] for a more detailed discussion.

References

1. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. Software available from tensorflow.org.
2. F. Chollet et al. Keras. https://keras.io, 2015.
3. M. Eichinger, A. Heinlein, and A. Klawonn. Flow predictions using convolutional neural networks. In preparation.
4. I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*, volume 1. MIT press Cambridge, 2016.
5. C. J. Greenshields. Openfoam user guide, v5. 0. *OpenFOAM foundation Ltd*, 2017.
6. X. Guo, W. Li, and F. Iorio. Convolutional neural networks for steady flow approximation. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 481–490, New York, NY, USA, 2016. ACM.
7. S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv:1502.03167*, 2015.
8. D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *1412.6980*, 2014.
9. M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part i): Data-driven solutions of nonlinear partial differential equations. *arXiv:1711.10561*, 2017.
10. M. Raissi, P. Perdikaris, and G. E. Karniadakis. Physics informed deep learning (part ii): Data-driven discovery of nonlinear partial differential equations. *arXiv:1711.10566*, 2017.
11. O. Ronneberger, P. Fischer, and T. Brox. U-net: Convolutional networks for biomedical image segmentation. *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, 2015.
12. J. Watt, R. Borhani, and A. K. Katsaggelos. *Machine Learning Refined: Foundations, Algorithms, and Applications*. Cambridge University Press, 2016.

Discontinuous Galerkin Model Order Reduction of Geometrically Parametrized Stokes Equation

Nirav Vasant Shah, Martin Wilfried Hess, and Gianluigi Rozza

Abstract The present work focuses on the geometric parametrization and the reduced order modeling of the Stokes equation. We discuss the concept of a parametrized geometry and its application within a reduced order modeling technique. The full order model is based on the discontinuous Galerkin method with an interior penalty formulation. We introduce the broken Sobolev spaces as well as the weak formulation required for an affine parameter dependency. The operators are transformed from a fixed domain to a parameter dependent domain using the affine parameter dependency. The proper orthogonal decomposition is used to obtain the basis of functions of the reduced order model. By using the Galerkin projection the linear system is projected onto the reduced space. During this process, the offline-online decomposition is used to separate parameter dependent operations from parameter independent operations. Finally this technique is applied to an obstacle test problem.The numerical outcomes presented include experimental error analysis, eigenvalue decay and measurement of online simulation time.

1 Introduction

Discontinuous Galerkin Method (DGM) has shown quite promising results for the elliptic problems [6] as well as for the hyperbolic problems [2]. DGM uses polynomial approximation for sufficient accuracy and allows discontinuity at the interface for greater flexibility. Model Order Reduction (MOR) allows reducing the size of the system by retaining only "dominant" modes. The faster computations obtained by MOR has helped in many query contexts, real time computations and quick transfer of computational results to industrial problems. MOR in combination with geometric parametrization has emerged as an alternative to the shape optimization and has been used in many engineering applications. As evident

N. V. Shah · M. W. Hess (✉) · G. Rozza
Scuola Internazionale Superiore di Studi Avanzati, Trieste, Italy
e-mail: snirav@sissa.it; mhess@sissa.it; grozza@sissa.it

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_54

from above advantages, the application of geometric parametrization and reduced order modeling to discontinuous Galerkin method will remain at the forefront of scientific work. The present work is organized as follow. We first explain the concept of geometric parametrization. Thereafter, the governing equations, broken Sobolev spaces and weak formulation are stated. The affine expansion and Proper Orthogonal Decomposition (POD) are briefly described in the subsequent sections. Finally, an obstacle test problem demonstrates the application of the introduced method with outcomes involving comparison of full order and reduced order model solutions, error analysis and eigenvalue decay.

2 Geometric Parametrization

Let us consider $\Omega = \Omega(\mu) \in \mathbb{R}^d$ as an open bounded domain. The parameter tuple $\mu \in \mathbb{P}$, where $\mathbb{P}$ is the parameter space, completely characterizes the domain. Also, consider a parameter tuple $\bar{\mu} \in \mathbb{P}$, as the known parameter tuple and $\Omega(\bar{\mu})$ as the reference domain, whose configuration is completely known. We divide the domain $\Omega(\mu)$ into n_{su} triangular subdomains such that $\Omega(\mu) = \bigcup_{i=1}^{n_{su}} \Omega_i(\mu)$, $\Omega_i(\mu) \bigcap \Omega_j(\mu) = \emptyset$, for $i \neq j$. The bijective mappings $\boldsymbol{F}_i(\cdot, \mu) : \Omega_i(\bar{\mu}) \to \Omega_i(\mu)$ link the reference subdomains $\Omega_i(\bar{\mu}) \subset \Omega(\bar{\mu})$ and the parametrized subdomains $\Omega_i(\mu) \subset \Omega(\mu)$. We consider here maps, $\boldsymbol{F}_i$, of the form,

$$x = \boldsymbol{F}_i(\hat{x}, \mu) = \boldsymbol{G}_{F,i}(\mu)\hat{x} + c_{F,i}(\mu) \; ;$$

$$\forall x \in \Omega_i(\mu) \; , \; \forall \hat{x} \in \Omega_i(\bar{\mu}) \; , \; \boldsymbol{G}_{F,i}(\mu) \in \mathbb{R}^{d \times d} \; , \; c_{F,i} \in \mathbb{R}^{d \times 1} \; , \; 1 \leq i \leq n_{su} \; .$$

The boundary of $\Omega(\mu)$, that is $\partial\Omega(\mu)$ is divided into a Neumann boundary $\Gamma_N(\mu)$ and a Dirichlet boundary $\Gamma_D(\mu)$ i.e. $\partial\Omega(\mu) = \Gamma_N(\mu) \cup \Gamma_D(\mu)$. The Jacobian matrices $\boldsymbol{G}_{F,i}$ and the translational vectors $c_{F,i}$ depend only on parameter tuple μ. The construction of maps $\{\boldsymbol{F}_i\}_{i=1}^{n_{su}}$ has been explained in literatures such as [4].

3 Discontinuous Galerkin Formulation

The domain Ω is divided into N_{el} number of triangular elements τ_k such that $\Omega = \bigcup_{k=1}^{N_{el}} \tau_k$. The triangulation $\mathcal{T}$ is the set of all triangular elements i.e. $\mathcal{T} = \{\tau_k\}_{k=1}^{N_{el}}$. The internal boundary is denoted by $\Gamma = \bigcup_{k=1}^{N_{el}} \partial\tau_k \backslash \partial\Omega$. $\overrightarrow{n}$ is the outward pointing normal to an edge of element.

The governing equations in strong form can be stated as,

$$\begin{aligned} \text{Stokes equation: } & -\nu\Delta\overrightarrow{u} + \nabla p = \overrightarrow{f} \ , \ \text{in } \Omega \ , \\ \text{Continuity equation: } & \nabla \cdot \overrightarrow{u} = 0 \ , \ \text{in } \Omega \ , \\ \text{Dirichlet condition: } & \overrightarrow{u} = \overrightarrow{u}_D \ , \ \text{on } \Gamma_D \ , \\ \text{Neumann condition: } & -p\overrightarrow{n} + \nu\overrightarrow{n} \cdot \nabla\overrightarrow{u} = \overrightarrow{t} \ , \ \text{on } \Gamma_N \ . \end{aligned} \tag{1}$$

The velocity vector field $\overrightarrow{u}$ and pressure scalar field p are the unknowns. ν is the material property known as kinematic viscosity. Vector $\overrightarrow{f}$ is the external force term or source term. $\overrightarrow{u}_D$ is the Dirichlet velocity and vector $\overrightarrow{t}$ is the Neumann value.

Let us introduce the broken Sobolev space, for any $p \in \mathbb{N}$,

$$H^p(\Omega, \mathcal{T}) = \{v \in L^2(\Omega) \mid v|_{\tau_k} \in H^p(\tau_k) \ , \ \forall \tau_k \in \mathcal{T}\}.$$

We consider finite dimensional subspaces of broken Sobolev spaces (see [2]), that is the spaces of discontinuous piecewise polynomial functions, for the unknowns.

$$\begin{aligned} \text{For velocity: } & \mathbb{V} = \{\overrightarrow{\phi} \in (L^2(\Omega))^d | \ \overrightarrow{\phi}|_{\tau_k} \in (P^D(\tau_k))^d \ , \ \tau_k \in \mathcal{T}\} \ , \\ \text{For pressure: } & \mathbb{Q} = \{\psi \in (L^2(\Omega)) | \ \psi|_{\tau_k} \in (P^{D-1}(\tau_k)) \ , \ \tau_k \in \mathcal{T}\} \ . \end{aligned}$$

Here, $P^D(\tau_k)$ denotes the space of polynomials of degree $D, \ D \geq 2$ over τ_k. It is to be noted that, due to the application of interior penalty (IP) and boundary penalty, the construction of subspace of Sobolev space is not required for imposing Dirichlet boundary condition.

In finite dimensional or discrete system, velocity approximation $\overrightarrow{u}_h(x)$ and pressure approximation $p_h(x)$ at any point $x \in \Omega$ are given by,

$$\overrightarrow{u}_h(x) = \sum_{i=1}^{N_u} \overrightarrow{\phi}_i \hat{u}_i \ , \ p_h(x) = \sum_{i=1}^{N_p} \psi_i \hat{p}_i \ , \tag{2}$$

where $\hat{u}_i$'s and $\hat{p}_i$'s are coefficients of velocity basis functions and pressure basis functions respectively.

We expect that $\overrightarrow{u}_h \rightarrow \overrightarrow{u}$ and $p_h \rightarrow p$ as $N_u \rightarrow \infty$ and $N_p \rightarrow \infty$ respectively. Considering the scope of present work, the convergence analysis will not be discussed here. The readers are advised to refer to [1, 5, 7].

In the subsequent sections, $(\cdot)$, $(\cdot)_{\Gamma_D}$, $(\cdot)_{\Gamma_N}$, $(\cdot)_\Gamma$ represent the L^2 scalar product over $\Omega, \Gamma_D, \Gamma_N, \Gamma$ respectively. The jump operator $[\cdot]$ and the average operator $\{\cdot\}$ are important concepts in the DGM formulation and are required to approximate the numerical flux. We use the jump and average operators as represented in [5].

The weak form of the Stokes equation is given by,

$$a_{IP}(\overrightarrow{u},\overrightarrow{\phi})+b(\overrightarrow{\phi},p)+\Big(\{p\},[\overrightarrow{n}\cdot\overrightarrow{\phi}]\Big)_{\Gamma\cup\Gamma_D}=l_{IP}(\overrightarrow{\phi})\,, \tag{3}$$

$$\begin{aligned}a_{IP}(\overrightarrow{u},\overrightarrow{\phi})&=\Big(\nabla\overrightarrow{u},\nabla\overrightarrow{\phi}\Big)+C_{11}\Big([\overrightarrow{u}],[\overrightarrow{\phi}]\Big)_{\Gamma\cup\Gamma_D}\\&-\nu\Big(\{\nabla\overrightarrow{u}\},[\overrightarrow{n}\otimes\overrightarrow{\phi}]\Big)_{\Gamma\cup\Gamma_D}-\nu\Big([\overrightarrow{n}\otimes\overrightarrow{u}],\{\nabla\overrightarrow{\phi}\}\Big)_{\Gamma\cup\Gamma_D}\,,\end{aligned} \tag{4}$$

$$b(\overrightarrow{\phi},\psi)=-\int_{\Omega}\psi\nabla\cdot\overrightarrow{\phi}\,, \tag{5}$$

$$l_{IP}(\overrightarrow{\phi})=\Big(\overrightarrow{f},\overrightarrow{\phi}\Big)+\Big(\overrightarrow{t},\overrightarrow{\phi}\Big)_{\Gamma_N}+C_{11}\Big(\overrightarrow{u}_D,\overrightarrow{\phi}\Big)_{\Gamma_D}-\Big(\overrightarrow{n}\otimes\overrightarrow{u}_D,\nu\nabla\overrightarrow{\phi}\Big)_{\Gamma_D}\,. \tag{6}$$

The penalty parameter $C_{11}>0$ is an empirical constant to be kept large enough to maintain the coercivity of $a_{IP}(\overrightarrow{u},\overrightarrow{\phi})$ (see [5]).

The weak form of the continuity equation is as follows,

$$b(\overrightarrow{u},\psi)+(\psi,[\overrightarrow{n}\cdot\overrightarrow{u}])_{\Gamma\cup\Gamma_D}=(\psi,\overrightarrow{n}\cdot\overrightarrow{u}_D)_{\Gamma_D}\,. \tag{7}$$

In the discrete form the system of equations can be written as,

$$\underbrace{\begin{pmatrix}\boldsymbol{A}&\boldsymbol{B}\\\boldsymbol{B}^T&\boldsymbol{0}\end{pmatrix}}_{\text{Stiffness matrix}}\underbrace{\begin{pmatrix}U\\P\end{pmatrix}}_{\text{Solution vector}}=\underbrace{\begin{pmatrix}F_1\\F_2\end{pmatrix}}_{\text{Right hand side (Known)}}\,. \tag{8}$$

Here, $\boldsymbol{A}_{ij}=a_{IP}(\overrightarrow{\phi}_i,\overrightarrow{\phi}_j)$, $\boldsymbol{B}_{ij}=b(\overrightarrow{\phi}_i,\psi_j)+\Big(\{\psi_j\},[n\cdot\overrightarrow{\phi}_i]\Big)_{\Gamma\cup\Gamma_D}$, $F_1=l_{IP}(\overrightarrow{\phi}_i)$ and $F_2=\big(\psi_j,\overrightarrow{n}\cdot\overrightarrow{u}_D\big)_{\Gamma_D}$ for $i=1,\ldots,N_u$ and $j=1,\ldots,N_p$. The column vectors U and P are coefficients $\hat{u}_i$'s and $\hat{p}_i$'s respectively (Eq. (2)).

4 Affine Expansion

We evaluate and solve the Stokes equation weak formulation on the reference domain $\Omega(\bar{\mu})$. Given a parameter tuple $\mu\neq\bar{\mu}$, we need to evaluate the linear system of equations (8) on a new domain $\Omega(\mu)$. To accomplish this, we use the affine expansion using linearity of equation and dividing $\Omega(\bar{\mu})$ into triangular subdomains $\Omega_i(\bar{\mu})$, $i=\{1,2,\ldots,n_{su}\}$ as explained earlier in Sect. 2. The affine expansion of operators has been explained in the literatures such as [4]. The bilinear form $a_{IP}(\cdot,\cdot;\mu)$ can be expressed as,

$$a_{IP}(\overrightarrow{u},\overrightarrow{\phi};\mu)=\sum_{i=1}^{i=Q_a}\theta_a^i(\mu)a_{IP}^i(\overrightarrow{u},\overrightarrow{\phi};\bar{\mu})\ , \tag{9}$$

for some finite Q_a and some bilinear forms $\{a_{IP}^i(\cdot,\cdot)\}_{i=1}^{Q_a}$. The bilinear form $a_{IP}(\cdot,\cdot;\bar{\mu})$ is evaluated once on the reference domain $\Omega(\bar{\mu})$. To evaluate the bilinear form $a_{IP}(\cdot,\cdot;\mu)$ on the parametrized domain $\Omega(\mu)$, we use the affine expansion (9). Since the evaluation of scalar terms $\{\theta_a^i(\mu)\}_{i=1}^{Q_a}$ is much faster than the evaluation of bilinear form $a_{IP}(\overrightarrow{u},\overrightarrow{\phi};\mu)$, significant speedup can be obtained with the help of affine expansion. Similar affine expansion can be used for other terms of the weak form (3). In the case of geometric parametrization, the affine expansion is essentially a change of variables [8]. However, it is pertinent to explain two expansions as specific to DGM formulation.

- In order to transfer the terms containing jump and average operator, following approach is used in the present analysis.

$$\left(\{\nabla\overrightarrow{\phi}\},\left[\overrightarrow{n}\otimes\overrightarrow{\phi}\right]\right)=\left(\nabla\overrightarrow{\phi}^+,\overrightarrow{n}^+\otimes\overrightarrow{\phi}^+\right)+\left(\nabla\overrightarrow{\phi}^+,\overrightarrow{n}^-\otimes\overrightarrow{\phi}^-\right)+\left(\nabla\overrightarrow{\phi}^-,\overrightarrow{n}^+\otimes\overrightarrow{\phi}^+\right)+\left(\nabla\overrightarrow{\phi}^-,\overrightarrow{n}^-\otimes\overrightarrow{\phi}^-\right)\ .$$

  Each term on the right hand side of the above equation can be transformed using the affine map.

- The coercivity term $C_{11}\left([\overrightarrow{\phi}],[\overrightarrow{u}]\right)_{\Gamma\cup\Gamma_D}$ is not transformed but used as evaluated on reference domain $\Omega(\bar{\mu})$. The affine transformation is given by,

$$C_{11}\left([\overrightarrow{\phi}(x),\overrightarrow{u}(x)]\right)_{\Gamma(\mu)\cup\Gamma_D(\mu)}=C_{11}\alpha\left([\overrightarrow{\phi}(\boldsymbol{F}(\hat{x})),\overrightarrow{u}(\boldsymbol{F}(\hat{x}))]\right)_{\Gamma(\bar{\mu})\cup\Gamma_D(\bar{\mu})}\ ,$$

$$\alpha=\frac{\text{length of }\ (\Gamma(\mu)\cup\Gamma_D(\mu))}{\text{length of }\ (\Gamma(\bar{\mu})\cup\Gamma_D(\bar{\mu}))}\ ,\ \hat{x}\in\Omega(\bar{\mu})\ ,\ x\in\Omega(\mu)\ .$$

  Since, C_{11} is an empirical coefficient replacing $C_{11}\alpha$ with C_{11} will not change the formulation as long as the coercivity of $a_{IP}(\overrightarrow{u},\overrightarrow{\phi})$ over parameter space $\mathbb{P}$ is maintained.

5 Reduced Basis Method

Snapshot POD exploits the information contained in the snapshots to construct low dimensional reduced basis space which can approximate the solution within desirable accuracy. The offline phase consists of construction of reduced basis space while the online phase consists of computing coefficients of the reduced basis. For detailed explanation about POD-Galerkin method and offline-online decomposition, we refer to [4].

As first step, the DGM solutions based on $\mu_n, n \in \{1, \ldots., n_s\}$ are calculated i.e. n_s snapshots are generated. The velocity snapshots and the pressure snapshots are stored in $\boldsymbol{S}_v \in \mathbb{R}^{N_u \times n_s}$ and $\boldsymbol{S}_p \in \mathbb{R}^{N_p \times n_s}$ respectively. Let us also introduce inner product matrices $\boldsymbol{M}_v \in \mathbb{R}^{N_u \times N_u}$ and $\boldsymbol{M}_p \in \mathbb{R}^{N_p \times N_p}$.

$$\boldsymbol{M}_{v,ij} = \int_\Omega \overrightarrow{\phi}_i \cdot \overrightarrow{\phi}_j + \sum_{k=1}^{N_{el}} \int_{\tau_k} \nabla \overrightarrow{\phi}_i : \nabla \overrightarrow{\phi}_j \ , \ i, j = 1, \ldots, N_u \ ,$$

$$\boldsymbol{M}_{p,ij} = \int_\Omega \psi_i \psi_j \ , \ i, j = 1, \ldots, N_p \ .$$

The dimension of the reduced basis is denoted as N and it is asserted that $N << N_u, \ N < n_s$. Proper Orthogonal Decomposition obtains orthogonal basis for the low dimensional reduced basis space, by using spectral decomposition.

$$\boldsymbol{S}_v^T \boldsymbol{M}_v \boldsymbol{S}_v = \boldsymbol{V} \boldsymbol{\Theta} \boldsymbol{V}^T \ . \tag{10}$$

The columns of $\boldsymbol{V}$ are eigenvectors and Θ has eigenvalues θ_i , $1 \leq i, j \leq n_s$, in sorted order ($\theta_1 \geq \ldots \geq \theta_{n_s}$) such that, $\Theta_{ij} = \theta_i \delta_{ij}$. Eigenvalue decay, the drop in the magnitude of the eigenvalues, provides upper bound for the error between the solution computed by full order model and the solution computed by POD (see [4]).

The projection matrix $\boldsymbol{B}_v \in \mathbb{R}^{N_u \times N}$, used for the projection from the space of full order model to the space of reduced order model, is given by,

$$\boldsymbol{B}_v = \boldsymbol{S}_v \boldsymbol{V} \boldsymbol{\Theta}^{-\frac{1}{2}} \boldsymbol{R} \ , \ \boldsymbol{R} = [\boldsymbol{I}_{N \times N}; \boldsymbol{0}_{(n_s - N) \times N}] \ , \tag{11}$$

where, $\boldsymbol{I}_{N \times N}$ is the identity matrix of size $N \times N$. The reduced basis space $\boldsymbol{B}_p$ can be generated in a similar manner using the pressure snapshots $\boldsymbol{S}_p$ and the inner product matrix $\boldsymbol{M}_p$. Above procedure is performed during the offline phase.

The discrete system of equations is projected onto the reduced basis space by Galerkin projection as,

$$\underbrace{\begin{pmatrix} \boldsymbol{B}_v^T \boldsymbol{A}(\mu)\boldsymbol{B}_v & \boldsymbol{B}_v^T \boldsymbol{B}(\mu)\boldsymbol{B}_p \\ \boldsymbol{B}_p^T \boldsymbol{B}(\mu)^T \boldsymbol{B}_v & \boldsymbol{0} \end{pmatrix}}_{\tilde{K}} \underbrace{\begin{pmatrix} U_N \\ P_N \end{pmatrix}}_{\zeta} = \underbrace{\begin{pmatrix} \boldsymbol{B}_v^T F_1(\mu) \\ \boldsymbol{B}_p^T F_2(\mu) \end{pmatrix}}_{\tilde{F}} . \tag{12}$$

The solution vectors U and P (Eq. (8)) are then computed as $U = \boldsymbol{B}_v U_N$, $P = \boldsymbol{B}_p P_N$. Projection onto the reduced basis space, solution of smaller system of equations and computation of U and P are steps performed during online phase. During the online phase, the matrices $\boldsymbol{A}(\mu)$, $\boldsymbol{B}(\mu)$ and the vectors $F_1(\mu)$, $F_2(\mu)$ are evaluated using affine expansion.

6 A Numerical Example

The numerical experiments were performed using RBmatlab [3, 9]. The reference domain $\Omega(\bar{\mu})$ is the unit square domain $[0, 1] \times [0, 1]$ with triangle having vertices $(0.3, 0)$, $(0.5, 0.3)$, $(0.7, 0)$ as obstacle. The domain $\Omega(\bar{\mu})$ is divided into 9 mutually non-overlapping subdomains. Two geometric parameters, the coordinates of the tip of the obstacle, with reference values collected in parameter tuple $\bar{\mu} = (0.5, 0.3)$ characterize the domain. The x-direction refers to the horizontal direction and the y-direction refers to the vertical direction. The boundary $x = 0$ is a Dirichlet boundary with inflow velocity at point $(0, y)$ as $u = (y(1 - y), 0)$. The boundary $x = 1$ is a Neumann boundary with zero Neumann value i.e. $\overrightarrow{t} = (0, 0)$. Other boundaries are Dirichlet boundary with no slip condition. The source term is $\overrightarrow{f} = (0, 0)$.

The training set contained 100 uniformly distributed random parameters within the $[0.4, 0.6] \times [0.2, 0.4]$. The test set contained 10 uniformly distributed random parameters within the range $[0.4, 0.6] \times [0.2, 0.4]$. For velocity basis function polynomial of degree $D = 2$ and for pressure basis function polynomial of degree $D - 1 = 1$ were used. The number of velocity degrees of freedom and pressure degrees of freedom were $N_u = 4704$ and $N_p = 1176$ respectively.

Figure 1 compares the solutions computed by DGM and Reduced Basis (RB) at parameter value $\mu = (0.47, 0.33)$ with reduced basis of size 10. The drop in error with respect to the increased size of the reduced basis space (Fig. 2) is inline with the expectation based on the eigenvalue decay (Fig. 3). The average speedup was 20.6. Typically, during the offline phase, the full order system was assembled in 35.37 seconds and was solved in 6.74 s. During the online phase, the reduced system was assembled in 2.03 s and was solved in 0.009 s.

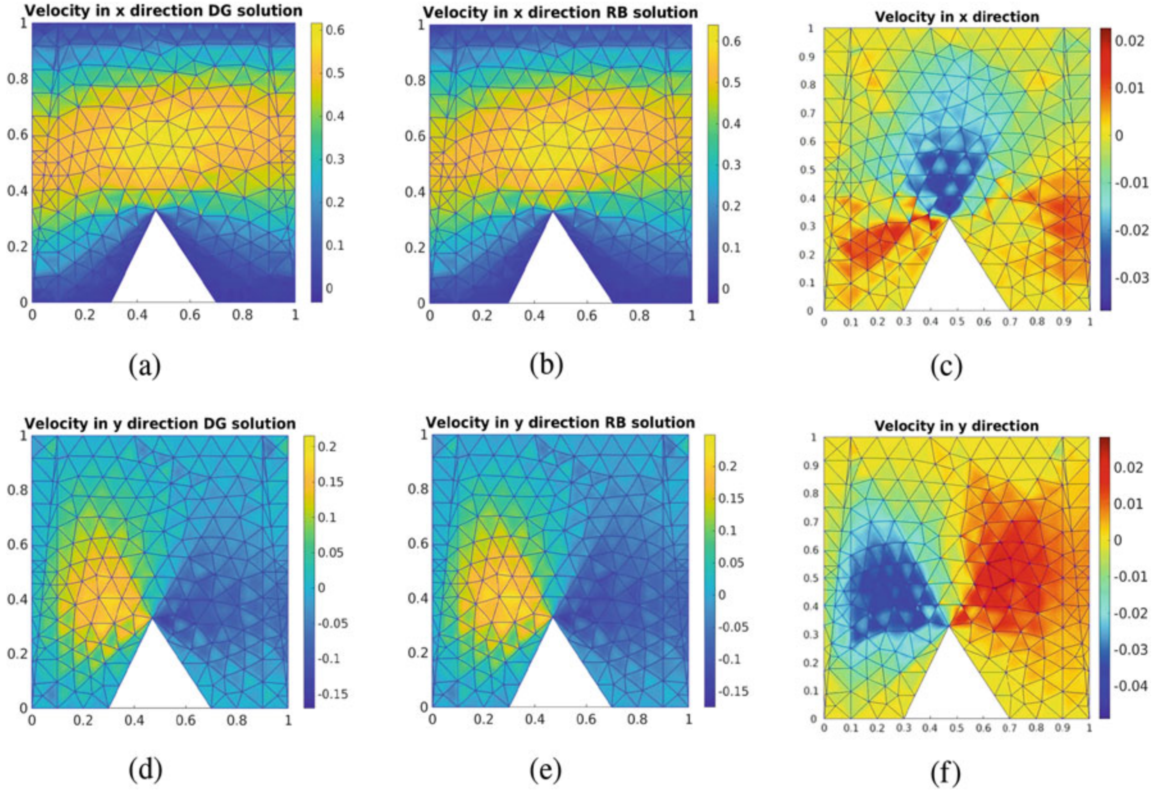
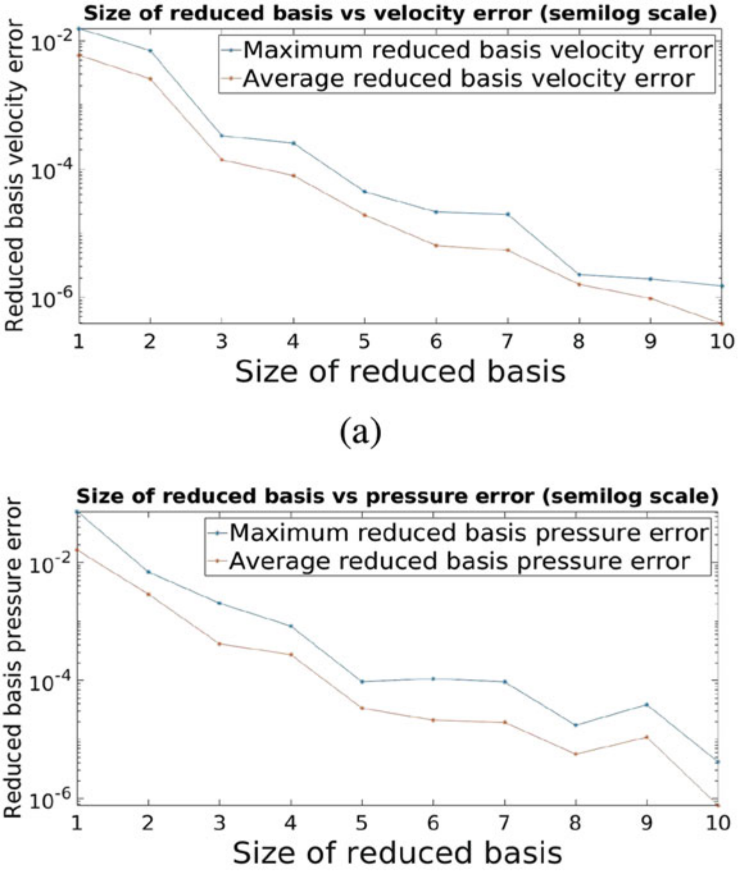


Fig. 1 DGM and RB solution $\mu = (0.47, 0.33)$. (**a**) Velocity x-direction DGM solution. (**b**) Velocity x-direction RB solution. (**c**) x-component of Velocity absolute error $\overrightarrow{u}_h - \overrightarrow{u}_N$. (**d**) Velocity y-direction DGM solution. (**e**) Velocity y-direction RB solution. (**f**) y-component of Velocity absolute error $\overrightarrow{u}_h - \overrightarrow{u}_N$



(a)

(b)

Fig. 2 Size of the reduced basis space vs Relative error. (**a**) Size of the reduced basis space vs. Relative error in velocity with inner product induced by $\boldsymbol{M}_v$. (**b**) Size of the reduced basis space vs. Relative error in pressure with inner product induced by $\boldsymbol{M}_p$

7 Some Concluding Remarks

As demonstrated by the numerical example, proper orthogonal decomposition can accelerate the computations involving geometrically parametrized discontinuous Galerkin interior penalty formulation while maintaining the reliability of solution above minimum acceptable limit. The paper also discussed, the specific issues related to the geometric parametrization and the affine expansion as pertaining to the discontinuous Galerkin interior penalty formulation. We expect the current work to contribute towards exploring further potentials in the field of geometric parametrization and reduced basis approach for the discontinuous Galerkin method.

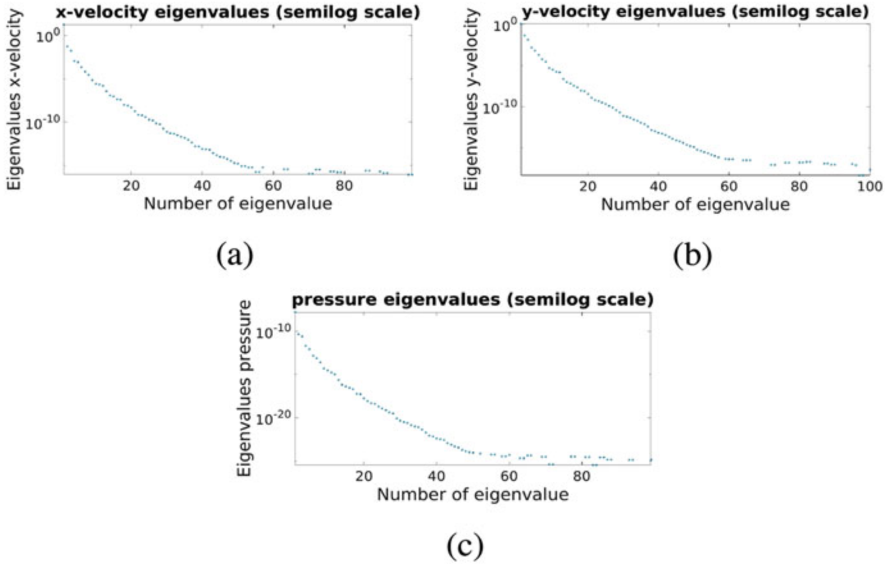


Fig. 3 Eigenvalue decay. (**a**) x-Velocity eigenvalues (semilog scale). (**b**) y-Velocity eigenvalues (semilog scale). (**c**) Pressure eigenvalues (semilog scale)

Acknowledgments This work has been supported by the H2020 ERC Consolidator Grant 2015 AROMA-CFD project 681447 “Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics” and COST Action: TD1307.

References

1. Antonietti PF, Pacciarini P, Quarteroni A (2016) A Discontinuous Galerkin reduced basis element method for elliptic problems. ESAIM: M2AN 50(2):337–360
2. Dolejší V, Feistauer M (2015) Discontinuous Galerkin Method: Analysis and Applications to Compressible Flow. Springer Series in Computational Mathematics, Springer International Publishing
3. Drohmann M, Haasdonk B, Kaulmann S, Ohlberger M (2012) A software framework for reduced basis methods using dune-rb and rbmatlab. In: Dedner A, Flemisch B, Klöfkorn R (eds) Advances in DUNE, Springer Berlin Heidelberg, Berlin, Heidelberg, pp 77–88
4. Hesthaven JS, Rozza G, Stamm B (2015) Certified Reduced Basis Methods for Parametrized Partial Differential Equations, 1st edn. Springer Briefs in Mathematics, Springer, Switzerland
5. Kanschat G, Schoetzau D (2008) Energy norm a posteriori error estimation for divergence-free discontinuous galerkin approximations of the navier-stokes equations. International Journal for Numerical Methods in Fluids 57:1093 – 1113
6. Peraire J, Persson PO (2008) The Compact Discontinuous Galerkin (CDG) method for elliptic problems. SIAM Journal on Scientific Computing 30(4):1806–1824
7. Rivière B (2008) Discontinuous Galerkin Methods for Solving Elliptic and Parabolic Equations: Theory and Implementation. Frontiers in Applied Mathematics, Cambridge University Press

8. Rozza G, Huynh D, Patera A (2007) Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive partial differential equations. Archives of Computational Methods in Engineering 15:1–47, DOI 10.1007/BF03024948
9. Shah NV, Haasdonk B, Hess M, Rozza G (2018) Discontinuous-Galerkin method for direct numerical simulation of the Navier-Stokes equation: Master thesis report. Master's thesis, Universität Stuttgart

An Efficient Numerical Scheme for Fully Coupled Flow and Reactive Transport in Variably Saturated Porous Media Including Dynamic Capillary Effects

Davide Illiano, Iuliu Sorin Pop, and Florin Adrian Radu

Abstract In this paper we study a model for the transport of an external component, e.g., a surfactant, in variably saturated porous media. We discretize the model in time and space by combining a backward Euler method with the linear Galerkin finite elements. The Newton method and the L-Scheme are employed for the linearization and the performance of these schemes is studied numerically. A special focus is set on the effects of dynamic capillarity on the transport equation.

1 Introduction

In this work, we concentrate on efficiently solving reactive transport models in saturated/unsaturated porous media [8, 10]. Such media are observable in the section of the soil closer to the surface where, in the upper part of the domain, we have a coexistence of both water and air phases while, below the water table, the soil becomes fully saturated.

In particular, our model includes dynamic capillarity effects. The capillary pressure is commonly defined as the difference between the pressures of the two phases, in our case, the air and the water. Note that, in the Richards model, the air pressure is set to be equal to zero.

Typically, the capillary pressure is assumed to be a nonlinear decreasing function depending on the water saturation. However, numerous studies are showing that such formulation is often too simplistic and that dynamic effects, due to the changes

D. Illiano (✉) · F. A. Radu
University of Bergen, Bergen, Norway
e-mail: Davide.Illiano@uib.no; Florin.Radu@uib.no

I. S. Pop
University of Bergen, Bergen, Norway

Hasselt University, Hasselt, Belgium
e-mail: sorin.pop@uhasselt.be

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_55

in time of the water phase, should also be included [2, 3, 5, 11, 13]. Based on this, we consider here the system:

$$\begin{aligned} \partial_t\theta - \nabla\cdot\big(K(\theta,\Psi)(\nabla\Psi + \mathbf{e_z})\big) &= \mathbb{S}_1, \\ \Psi + p_{cap}(\theta,c) &= \tau(\theta)\partial_t\theta, \\ \partial_t(\theta c) - \nabla\cdot(D\nabla c - \mathbf{u_w}c) + R(c) &= \mathbb{S}_2. \end{aligned} \tag{1}$$

The first equation is the Richards equation, whereas the second is an ordinary differential equation used to include the non-equilibrium effects in the capillary pressure/water content relation. Equilibrium models are obtained for $\tau = 0$. Furthermore, the third equation is the reactive transport equation. Here, θ is the water content, Ψ the pressure head, c the concentration of the chemical component, K the conductivity, $\mathbf{e_z}$ the unit vector in the direction opposite to gravity, D the diffusion/dispersion coefficient, $\mathbf{u_w}$ the water flux, $R(c)$ the reaction term and finally $\mathbb{S}_1$ and $\mathbb{S}_2$ are any source terms or external forces involved in the process. Note that $\mathbf{u_w} := -K(\theta,\Psi)(\nabla\Psi + \mathbf{e_z})$ where K is a nonlinear function depending on θ and Ψ. In the van Genuchten model [4] one has:

$$K(\theta,\Psi) = \begin{cases} K_s\theta^{\frac{1}{2}}\left[1-\left(1-\theta^{\frac{n}{n-1}}\right)^{\frac{n-1}{n}}\right]^2, & \Psi \le 0 \\ K_s, & \Psi > 0. \end{cases} \tag{2}$$

K_s is the saturated conductivity and n is a soil dependent parameter.

The system (1) is completed by boundary conditions for Ψ and c, and initial conditions for θ and c.

The rest of the paper is organized as follows: in Sect. 2 the equations are discretized and linearized. Section 3 includes a numerical example, based on the literature [6], which allows us to compare the different numerical schemes. Finally, Sect. 4 will conclude this paper with our final remarks.

2 The Numerical Schemes

Applying an Euler implicit time-stepping to (1) gives a sequence of time discrete nonlinear equations. To solve them we apply different linearization schemes: the Newton method, the L-Scheme and a combination of the two [7, 9]. They are compared here thanks to a numerical example inspired by reactive models.

The equations in (1) are fully coupled due to the double dependency of the capillary pressure of both the water content θ and the concentration c. In general, p_{cap} is a function of only θ, e.g., $p_{cap} := 1/\alpha(\theta^{-1/m} - 1)^{1/n}$ as presented in [4]. Anyhow, it has been observed [12] that, if an external component is involved, the surface tension becomes a function of the concentration c and thus, the capillary pressure itself is influenced by this, i.e. $p_{cap} := p_{cap}(\theta, c)$.

In the following, we use the standard notations of functional analysis. The domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2$ or 3, is bounded, open and has a Lipschitz continuous boundary $\partial\Omega$. We denote by $L^2(\Omega)$ the space of real-valued, square-integrable functions defined on Ω and $H^1(\Omega)$ its subspace containing the functions having also the first order weak partial derivatives in $L^2(\Omega)$. $H_0^1(\Omega)$ is the space of functions belonging to $H^1(\Omega)$, having zero values on the boundary $\partial\Omega$. We denote by $< \cdot, \cdot >$ the $L^2(\Omega)$ scalar product and by $\|\cdot\|$ the associated norm. Finally, assume that K is continuous and increasing, $p_{cap} \in C^1\big((0, 1], [0, \infty)\big)$ is decreasing and $\tau \in C^1\big((0, 1], [0, \infty)\big)$.

We now combine the backward Euler method with linear Galerkin finite elements for the discretization of the problem (1). Let $N \in \mathbb{N}$ be a strictly positive natural number, define the time step size $\Delta t = T/N$ and $t_n = n\Delta t$ $(n \in 1, 2, \ldots, N)$. Furthermore, $\mathbb{T}_h$ is a regular decomposition of Ω, $\overline{\Omega} = \underset{\mathbb{T} \in \mathbb{T}_h}{\cup} \mathbb{T}$, into d-dimensional simplices, with h denoting the maximal mesh diameter. The finite element space $V_h \subset H_0^1(\Omega)$ is defined by

$$V_h := \{v_h \in H_0^1(\Omega) \; s.t. \; v_{h|\mathbb{T}} \in \mathbb{P}_1(\mathbb{T}), \; \mathbb{T} \in \mathbb{T}_h\}, \tag{3}$$

where $\mathbb{P}_1(\mathbb{T})$ denotes the space of the afine polynomials on $\mathbb{T}$.

The fully discrete Galerkin formulation of the system (1) can be written as:

Problem P(n) Let $n \geq 1$ be fixed. Given $\Psi_h^{n-1}, \theta_h^{n-1}, c_h^{n-1} \in V_h$, find $\Psi_h^n, \theta_h^n, c_h^n \in V_h$ such that there holds

$$< \theta_h^n - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^n, \Psi_h^n)(\nabla \Psi_h^n + \mathbf{e_z}), \nabla v_{1,h} > = \Delta t < \mathbb{S}_1, v_{1,h} >, \tag{4}$$

$$\Delta t < \Psi_h^n, v_{2,h} > + \Delta t < p_{cap}(\theta_h^n, c_h^n), v_{2,h} > = < \tau(\theta_h^n)(\theta_h^n - \theta_h^{n-1}), v_{2,h} >, \tag{5}$$

and

$$\begin{aligned} < \theta_h^n c_h^n - \theta_h^{n-1} c_h^{n-1}, v_{3,h} > + \Delta t < D \nabla c_h^n + \mathbf{u_w^{n-1}} c_h^n, \nabla v_{3,h} > \\ + \Delta t < R(c_h^n), v_{3,h} > = \Delta t < \mathbb{S}_2, v_{3,h} >, \end{aligned} \tag{6}$$

for all $v_{1,h}, v_{2,h}, v_{3,h} \in V_h$.

Remark 1 We use $\mathbf{u_w^{n}}^{-1} := -K(\theta_h^{n-1}, \Psi_h^{n-1})(\nabla \Psi_h^{n-1} + \mathbf{e_z})$ for the convective term in the transport equation, for simplicity reasons. Nevertheless, all the simulations presented in this paper have also been performed with $\mathbf{u_w^n} := -K(\theta_h^n, \Psi_h^n)(\nabla \Psi_h^n + \mathbf{e_z})$ instead of $\mathbf{u_w^{n}}^{-1}$ and the results were almost identical.

In the following, we propose different solving strategies for the system of equations presented above. These strategies are built on the ones discussed in [7], extending them to the case of dynamic capillary pressure ($\tau(\theta) \neq 0$). They are either a monolithic solver of the full system, or a splitting approach obtained by solving first the flow component, using a previously computed concentration, then updating the transport equation, using the newly computed pressure and water content. In both cases, one has to iterate. Each iteration requires solving a non-linear problem, for which, either the Newton methods or the L-Scheme [7, 9, 10] are considered. These strategies are then named: monolithic-Newton scheme (MON-Newton), monolithic-L-Scheme (MON-LS), nonlinear splitting-Newton (NonLinS-Newton) and nonlinear splitting-L-Scheme (NonLinS-LS). The splitting methods are also known as sequential or segregated approaches.

The index j denotes the iteration index. As a rule, the iterations start with the solution obtained at the previous time step, for example $\Psi^{n,1} := \Psi^{n-1}$. This is not necessary for the L-Scheme, which is globally convergent, but it appears to be a natural choice.

2.1 The Monolithic Newton Method (MON-NEWTON)

The Newton method is a well-known linearization scheme, which is quadratic but only locally convergent. Applying the monolithic Newton method to (4)–(6) leads to

Problem MN(n,j+1) Let $\Psi_h^{n-1}, \theta_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, \theta_h^{n,j} c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that

$$\begin{aligned} < \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > +\Delta t < K(\theta_h^{n,j}, \Psi_h^{n,j})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > \\ +\Delta t < \partial_\theta K(\theta_h^{n,j}, \Psi_h^{n,j})(\nabla(\Psi_h^{n,j}) + \mathbf{e_z})(\theta_h^{n,j+1} - \theta_h^{n,j}), \nabla v_{1,h} > \\ +\Delta t < \partial_\Psi K(\theta_h^{n,j}, \Psi_h^{n,j})(\nabla(\Psi_h^{n,j}) + \mathbf{e_z})(\Psi_h^{n,j+1} - \Psi_h^{n,j}), \nabla v_{1,h} > \\ = \Delta t < \mathbb{S}_1, v_{1,h} >, \end{aligned} \tag{7}$$

$$\begin{aligned} \Delta t < \Psi_h^{n,j+1}, v_{2,h} > +\Delta t < p_{cap}(\theta_h^{n,j}, c_h^{n,j}), v_{2,h} > \\ +\Delta t < \partial_\theta p_{cap}(\theta_h^{n,j}, c_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n,j}), v_{2,h} > +\Delta t < \partial_c p_{cap}(\theta_h^{n,j}, c_h^{n,j}) \\ (c_h^{n,j+1} - c_h^{n,j}), v_{2,h} >= < \tau(\theta_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > \\ + < \partial_\theta \tau(\theta_h^{n,j})(\theta_h^{n,j} - \theta_h^{n-1})(\theta_h^{n,j+1} - \theta_h^{n,j}), v_{2,h} >, \end{aligned} \tag{8}$$

and

$$< \theta_h^{n,j} c_h^{n,j+1} - \theta_h^{n-1} c_h^{n-1}, v_{3,h} > + \Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla v_{3,h} >$$
$$+ \Delta t < R(c_h^{n,j}), v_{3,h} > + \Delta t < \partial_c R(c_h^{n,j})(c_h^{n,j+1} - c_h^{n,j}) >$$
$$= \Delta t < \mathbb{S}_2, v_{3,h} >, \tag{9}$$

hold true for all $v_{1,h}, v_{2,h}, v_{3,h} \in V_h$.

2.2 The Monolithic L-Scheme (MON-LS)

The monolithic L-scheme for solving (4)–(6) reads as

Problem ML(n,j+1) Let $\Psi_h^{n-1}, \theta_h^{n-1}, c^{n-1}, \Psi_h^{n,j}, \theta_h^{n,j} c_h^{n,j} \in V_h$ be given, $L_1^{\Psi}, L_1^{\theta}, L_2, L_3 > 0$, big enough.
Find $\Psi_h^{n,j+1}, \theta_h^{n,j+1}, c_h^{n,j+1} \in V_h$ such that

$$< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > + \Delta t < K(\theta_h^{n,j}, \Psi_h^{n,j})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} >$$
$$+ \Delta t < L_1^{\Psi}(\Psi_h^{n,j+1} - \Psi_h^{n,j}), \nabla v_{1,h} > + \Delta t < L_1^{\theta}(\theta_h^{n,j+1} - \theta_h^{n,j}), \nabla v_{1,h} >$$
$$= \Delta t < \mathbb{S}_1, v_{1,h} >, \tag{10}$$

$$\begin{aligned} \Delta t < \Psi_h^{n,j+1}, v_{2,h} > = & -\Delta t < p_{cap}(\theta_h^{n,j}, c_h^{n,j}), v_{2,h} > \\ + < \tau(\theta_h^{n,j})(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} > & + < L_2(\theta_h^{n,j+1} - \theta_h^{n,j}), v_{2,h} > \end{aligned} \tag{11}$$

and

$$< \theta_h^{n,j} c_h^{n,j+1} - \theta_h^{n-1} c_h^{n-1}, v_{3,h} > + \Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^{n-1}} c_h^{n,j+1}, \nabla v_{3,h} >$$
$$+ \Delta t < R(c_h^{n,j}), v_{3,h} > + < L_3(c_h^{n,j+1} - c_h^{n,j}), v_{3,h} > = \Delta t < \mathbb{S}_3, v_{3,h} >, \tag{12}$$

hold true for all $v_{1,h}, v_{2,h}, v_{3,h} \in V_h$.

The L-Scheme does not involve the computations of derivatives, the linear systems to be solved within each iteration are better conditioned, compared to the ones given by the Newton method [7, 9], and it is globally (linearly) convergent. The convergence of the scheme has been proved, for the equilibrium model ($\tau(\theta) = 0$) in [7], and can be easily extended to the non-equilibrium formulation given by the system (10)–(12).

2.3 The Splitting Approach (NonLinS)

The splitting approach for solving (4)–(6) reads as

Problem S(n,j+1) Let $\Psi_h^{n-1}, \theta^{n-1}, c^{n-1}, \Psi_h^{n,j}, \theta_h^{n,j}, c_h^{n,j} \in V_h$ be given, find $\Psi_h^{n,j+1}, \theta_h^{n,j+1} \in V_h$ such that

$$< \theta_h^{n,j+1} - \theta_h^{n-1}, v_{1,h} > +\Delta t < K(\theta_h^{n,j+1}, \Psi_h^{n,j+1})(\nabla(\Psi_h^{n,j+1}) + \mathbf{e_z}), \nabla v_{1,h} > \\ = \Delta t < \mathbb{S}_1, v_{1,h} >, \tag{13}$$

$$\Delta t < \Psi_h^{n,j+1}, v_{2,h} > +\Delta t < p_{cap}(\theta_h^{n,j+1}, c_h^{n,j}), v_{2,h} > \\ = < \tau(\theta_h^{n,j+1})(\theta_h^{n,j+1} - \theta_h^{n-1}), v_{2,h} >, \tag{14}$$

hold true for all $v_{1,h}, v_{2,h} \in V_h$.

Then, with $\Psi_h^{n,j+1}$ and $\theta_h^{n,j+1}$ obtained from the equations above, find $c_h^{n,j+1} \in V_h$ such that

$$< \theta_h^{n,j+1} c_h^{n,j+1} - \theta_h^{n-1} c_h^{n-1}, v_{3,h} > +\Delta t < D\nabla c_h^{n,j+1} + \mathbf{u_w^n}^{-1} c_h^{n,j+1}, \nabla v_{3,h} > \\ +\Delta t < R(c_h^{n,j+1}), v_{3,h} > = \Delta t < \mathbb{S}_2, v_{3,h} >, \tag{15}$$

holds true for all $v_{3,h} \in V_h$.

The three equations above can be then linearised using either the Newton method (NonLinS-Newton) or the L-Scheme (NonLinS-LScheme).

2.4 The Mixed Linearization Scheme

It has been already observed, for a different set of equations [9], that combining the Newton method and the L-Scheme can improve the convergence of the scheme. The Newton method is quadratically but only locally convergent and it can produce badly conditioned linearized systems. Moreover, the time step is subject to severe restrictions for guaranteeing the convergence of the scheme, and this has also been observed in numerical examples [1, 7, 9].

Contrarily, the L-Scheme is globally convergent and the linear systems to be solved within each iteration are better conditioned, however, it has only a linear rate of convergence.

The mixed formulation, obtained combining the two schemes, appears to be the best approach and shows practically both global and quadratic convergence. The Newton method commonly fails to converge, if the initial guess is too far from the actual solution. Since this guess is usually the solution at the previous time, this can force restriction on the time step. Instead of reducing the time step one can obtain a better approximation of the initial guess, for the Newton method, by performing few L-Scheme iterations. In the numerical simulation here presented, up to 5 iterations were sufficient to reach a good initial guess for the Newton iteration, which ensured its convergence.

3 Numerical Examples

In this section, we use a benchmark problem, from [6], to compare the different linearization schemes and solving algorithms defined above. It describes the recharge of a two-dimensional underground reservoir $\Omega \subset \mathbb{R}^2$, in the interval of time $t \in (0, 3]$. The boundary of the domain and the Dirichlet boundary conditions are defined below.

$$\begin{aligned}
&\Omega = (0,2) \times (0,3), && \Psi(x,y,t) = \begin{cases} -2 + 2.2 * t, & \text{on } \Gamma_{D_1}, t \le 1 \\ 0.2, & \text{on } \Gamma_{D_1}, t > 1 \\ 1 - y, & \text{on } \Gamma_{D_2}, \end{cases} \\
&\Gamma_{D_1} = \{(x,y) \in \partial\Omega | x \in [0,1] \wedge y = 3\}, && \\
&\Gamma_{D_2} = \{(x,y) \in \partial\Omega | x = 2 \wedge y \in [0,1]\}, && \\
&\Gamma_D = \Gamma_{D_1} \cup \Gamma_{D_2}, && c(x,y,t) = \begin{cases} 1, & \text{on } \Gamma_{D_1}, t \le 1 \\ 0, & \text{on } \Gamma_{D_1}, t > 1 \\ 3 - y, & \text{on } \Gamma_{D_2} \cup \Gamma_N. \end{cases} \\
&\Gamma_N = \partial\Omega \setminus \Gamma_D, &&
\end{aligned}$$

Furthermore, no flow conditions are imposed on Γ_N. The initial conditions are given by $\Psi(x, y, 0) := 1 - y, c(x, y, 0) := 3 - y$ and $\theta(x, y, 0) := 0.39$. The capillary pressure is defined as $p_{cap}(\theta, c) := (1 - \theta)^{2.5} + 0.1 * c$, the conductivity is given by (2) and $\tau(\theta) = 1$. Finally, the parameters implemented are: $K_s = 1$, $L_1^{\Psi}, L_1^{\theta}, L_2 = 0.01$, $L_3 = 0.1$ and the iterations stop whenever all the error norms, $\left\|\Psi^{n,j+1} - \Psi^{n,j}\right\|$, $\left\|\theta^{n,j+1} - \theta^{n,j}\right\|$ and $\left\|c^{n,j+1} - c^{n,j}\right\|$, are below 10^{-6}.

We performed the simulations using regular meshes, consisting of squares, with sides $dx = \{1/10, 1/20, 1/40\}$. We considered two fixed time steps $\Delta t = 1/10$ and $\Delta t = 1/50$.

In Fig. 1, we can observe the total numbers of iterations required by the different linearization schemes and solving algorithms. Next to the name of each scheme we report, between parenthesis, which time step Δt has been used.

We can observe, as the Newton method in the monolithic formulation, converges only for coarse meshes, for $\Delta t = 1/10$. For the smaller time step, $\Delta t = 1/50$, it converges for all of the tested meshes. The L-Scheme converges for both time steps, but, since it is linearly convergent, for $\Delta t = 1/50$ would require more iterations than the Newton method.

The results obtained thanks to the mixed formulation are particularly interesting. We can observe that this scheme, both in the monolithic and splitting formulation,

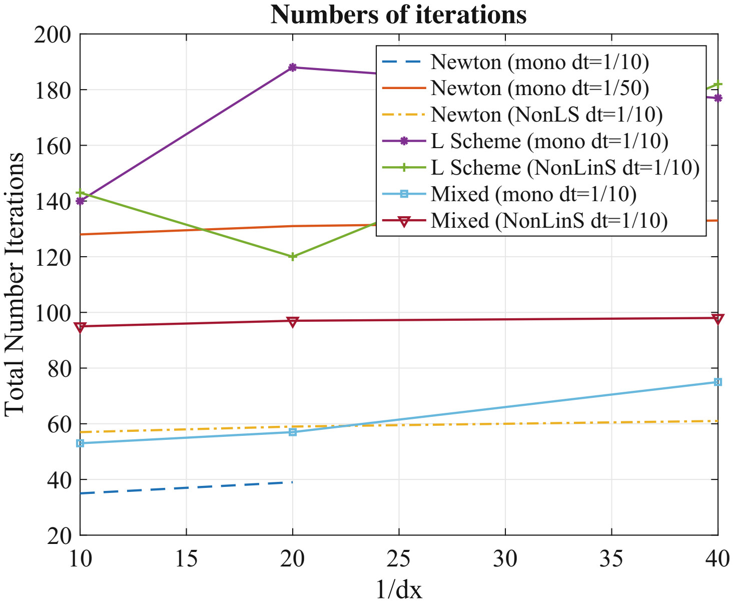


Fig. 1 Total numbers of iterations for different solvers

converges for all the tested meshes also in case of a large time step. Moreover, thanks to the Newton iterations, it appears to be faster than the classical L-Scheme. It is as robust as the L-Scheme and as fast as the Newton method. For more details regarding the mixed scheme, we refer to [9].

4 Conclusions

In this paper, we considered multiphase flow coupled with a one-component reactive transport in variably saturated porous media, including also the dynamic effects in the capillary pressure. The resulting model is nonlinear and for this reason, three different linearization schemes are investigated: the L-Scheme, the Newton method and a combination of the two. We also studied both monolithic solvers and splitting ones.

The tests show that, for this particular set of equations, the best linearization scheme is the one obtained combining the Newton method and the L-Scheme. Such scheme appears to be both quadratically and globally convergent. Finally, this results in a clear reduction of the workload, compared to the classical L-scheme.

Acknowledgments The research of D. Illiano was funded by VISTA, a collaboration between the Norwegian Academy of Science and Letters and Equinor, project number 6367, project name: adaptive model and solver simulation of enhanced oil recovery. The research of I.S. Pop was supported by the Research Foundation-Flanders (FWO), Belgium through the Odysseus programme (project G0G1316N) and Equinor through the Akademia grant.

We thank the members of the *Sintef* research group and in particular to Dr. Olav Moyner for the assistance with the implementation of the numerical examples in *MRST*, the toolbox based on Matlab developed at *Sintef* itself.

References

1. Cao, X., Pop, I.S.: Uniqueness of weak solutions for a pseudo-parabolic equation modeling two phase flow in porous media, Applied Mathematics Letters, Volume 46, Pages 25–30, (2015).
2. Di Carlo, D.: Experimental measurements of saturation overshoot on infiltration, Water Resources Research, Volume 40, Issue 4, (2004).
3. Fucik, R., Mikyska, J., Sakaki, T., Benes, M., Illangasekare, T.H.: Significance of Dynamic Effect in Capillarity during Drainage Experiments in Layered Porous Media, Vadose Zone Journal 9, Volume 3, (2010).
4. van Genuchten, M.: A Closed-form Equation for Predicting the Hydraulic Conductivity of Unsaturated Soils, Soil Science Society of America Journal, Volume 44, Issue 5, Pages 892–898, (1980).
5. Hassanizadeh, S.M., Celia, M.A., Dahle, H.K.: Dynamic Effect in the Capillary Pressure Saturation Relationship and its Impacts on Unsaturated Flow, Vadose Zone Journal, Volume 1, Issue 1, Pages 38–57, (2002).
6. Haverkamp, R., Vauclin, M., Touma, J., Wierenga, P.J., Vachaud, G.: A comparison of numerical simulation models for one-dimensional infiltration, Soil Science Society of America Journal, Volume 41, Pages 285–294, (1977).
7. Illiano, D., Pop, I.S., Radu, F.A.: Iterative schemes for surfactant transport in porous media, arXiv preprint arXiv:1906.00224, (2019).
8. Knabner, P.: Finite element simulation of saturated-unsaturated flow through porous media, LSSC 7, Pages 83–93, (1987).
9. List, F., Radu, F.A.: A study on iterative methods for solving Richards' equation, Computational Geoscience, Volume 20, Issue 2, Pages 341–353, (2016).
10. Pop, I.S., Radu, F.A., Knabner, P.: Mixed finite elements for the Richards' equation: linearization procedure, Journal of computational and applied mathematics, Volume 168, Issue 1, Pages 365–373, (2004).
11. Shubao, T., Lei, G., Shun-li, H., Yang, L.: Dynamic effect of capillary pressure in low permeability reservoirs, Petroleum Exploration and Development, Volume 39, Issue 3, Pages 405–411, (2012).
12. Smith, J., Gillham, R.: Effects of solute concentration-dependent surface tension on unsaturated flow: Laboratory sand column experiments, Water Resource Research, Volume 35, Issue 4, Pages 973–982, (1999).
13. Zhuang, L., van Duijn, C.J., Hassanizadeh, S.M.: The effect of dynamic capillarity in modeling saturation overshoot during infiltration, Vadose Zone Journal, Volume 18, Issue 1, Pages 1–14, (2019).

Multistage Preconditioning for Adaptive Discretization of Porous Media Two-Phase Flow

Birane Kane

Abstract We present a constrained pressure residual (CPR) two-stage preconditioner applied to a discontinuous Galerkin discretization of a two-phase flow in strongly heterogeneous porous media. We consider a fully implicit, locally conservative, higher order discretization on adaptively generated meshes. The implementation is based on the open-source PDE software framework Dune and its PETSc binding.

1 Introduction

The significant geologic complexity involved in multi-phase flow and the treatment of strongly heterogeneous soil properties need efficient preconditioning strategies for fully implicit formulations. Multilevel techniques such as the constrained pressure residual (CPR) two-stage preconditioner allow to exploit the algebraic properties of the Jacobian matrix of the system. The two-stage CPR preconditioner was introduced by Wallis [2, 3] from the previous work of Behie and Vinsome [4] on combinative preconditioners in reservoir engineering. Lacroix et al. [5] combined a first stage preconditioner on the pressure subsystem with Algebraic Multigrid (AMG) and a second stage preconditioner on the full system with ILU-0. The CPR-AMG has proven to be efficient for the simulation of complex problems in reservoir engineering [6–9] and in basin modeling [10]. The CPR impact on h and hp adaptive DG schemes is still not well understood as most of the work with regards to the CPR has so far mainly focused on finite volume methods. To our knowledge this is the first time the CPR-AMG is applied within an adaptive DG discretization framework.

This work is organized as follows: Sect. 2 provides a description of the Jacobian matrix arising from a fully implicit discretization of a two-phase flow problem.

B. Kane (✉)
NORCE Norwegian Research Centre AS, Bergen, Norway
e-mail: birane.kane@norceresearch.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_56

Section 3 sets out the formulation of the CPR-AMG method. Section 4 provides numerical tests implemented within Dune [1].

2 Structure of the Jacobian Matrix

We consider a domain $\Omega \in \mathbb{R}^d$, $d \in \{2, 3\}$. The phases $\alpha = \{w, n\}$ are incompressible and immiscible. Unknown variables are the pressure p_w and the saturation s_n.

$$\begin{aligned} -\nabla \cdot \Big((\lambda_w + \lambda_n)\mathbb{K}\nabla p_w + \lambda_n p_c' \mathbb{K}\nabla s_n - (\rho_w\lambda_w + \rho_n\lambda_n)\mathbb{K}\mathbf{g} \Big) &= q_w + q_n, \\ \phi\frac{\partial s_n}{\partial t} - \nabla \cdot \Big(\lambda_n \mathbb{K}(\nabla p_w - \rho_n \mathbf{g}) \Big) - \nabla \cdot \Big(\lambda_n p_c' \mathbb{K}\nabla s_n \Big) &= q_n. \end{aligned} \tag{1}$$

$\lambda_\alpha := \lambda_\alpha(s_\alpha)$	phase mobility
$\mathbf{g}$	gravity
$\phi > 0$	porosity

$p_c := p_c(s_n)$	capillary-pressure
$\mathbb{K}$	permeability tensor
ρ_α	phase density
q_α	source/sink term

In order to have a complete system we add appropriate boundary and initial conditions. For a more thorough description of the complete system and its DG discretization see [11–14].

The development of effective and robust preconditioning techniques requires to fully understand and exploit the algebraic properties of each individual block of the Jacobian matrix J_G stemming from the fully-implicit and fully-coupled DG discretization of the two-phase flow system (1). Following [11], let $J_G X = b$ be the linear system to solve and $r = b - J_G X$ the residual, where $X = (X_p, X_s)$ is the unknown and $b = (b_p, b_s)^\intercal$ the right-hand side. The Jacobian matrix J_G is expressed as

$$J_G = \begin{pmatrix} J^{pp} & J^{ps} \\ & \\ J^{sp} & J^{ss} \end{pmatrix} = \begin{pmatrix} \frac{\partial G^p}{\partial p} & \frac{\partial G^p}{\partial s} \\ & \\ \frac{\partial G^s}{\partial p} & \frac{\partial G^s}{\partial s} \end{pmatrix}. \tag{2}$$

Here, $J^{pp} \in \mathbb{R}^{n_p \times n_p}$ is the pressure block, $J^{ss} \in \mathbb{R}^{n_s \times n_s}$ is the saturation block. $J^{ps} \in \mathbb{R}^{n_p \times n_s}$ and $J^{sp} \in \mathbb{R}^{n_s \times n_p}$ are the coupling blocks. The term G^p (resp. G^s) denotes the discretization of the first (resp. second) equation of system (1).

We consider in our implementation a dof-based re-ordering of variables where J_G is reformulated as

$$J_G = \begin{pmatrix} (J^{pp})_{1,1} & (J^{ps})_{1,1} & \cdots & (J^{pp})_{1,n_s} & (J^{ps})_{1,n_s} \\ (J^{ss})_{1,1} & (J^{ss})_{1,1} & & (J^{ss})_{1,n_s} & (J^{ss})_{1,n_s} \\ \vdots & & \ddots & & \vdots \\ (J^{pp})_{n_p,1} & (J^{ps})_{n_p,1} & \cdots & (J^{pp})_{n_p,n_s} & (J^{ps})_{n_p,n_s} \\ (J^{ss})_{n_p,1} & (J^{ss})_{n_p,1} & & (J^{ss})_{n_p,n_s} & (J^{ss})_{n_p,n_s} \end{pmatrix}. \tag{3}$$

Above, n_p is the number of dofs for the pressure and n_s is the number of dofs for the saturation. $(J)_{i,j}$ represents the coupling between two dofs.

3 Constrained Pressure Residual Preconditioner

This section provides an extended insight into the structures and the different stages involved in the construction of the CPR preconditioner.

3.1 Method Description

The CPR belongs to the family of two-stage preconditioners, first it extracts and solves a pressure subsystem. The residual associated with this solution is subsequently corrected with an additional preconditioning step that recovers part of the global information contained in the original system. The elliptic feature exhibited by the pressure subsystem allows it to be handled well by multigrid methods. The other equation is usually degenerate parabolic and might be handled by an ILU preconditioner. Figure 1 provides a sketch of the CPR preconditioning.

Definition 1 The general formulation of a two-stage preconditioner is:

$$M_{2st}^{-1} = M_2^{-1}\left[I - \tilde{J}M_1^{-1}\right] + \left(M_1^{-1}\right) \tag{4}$$

where M_1^{-1} (resp. M_2^{-1}) corresponds to the first (resp. second) stage of the preconditioner and the operator $\tilde{J}$ is such that

$$D_1^{-1} J_G D_2^{-1} = \tilde{J} = \begin{pmatrix} \tilde{J}_{pp} & \tilde{J}_{ps} \\ \\ \tilde{J}_{sp} & \tilde{J}_{ss} \end{pmatrix}. \tag{5}$$

Here D_1 and D_2 are decoupling operators, different choices of D_i, $i \in \{1, 2\}$ generate different first stage preconditioners [6]. We provide more details concerning the decoupling operators in the next section.

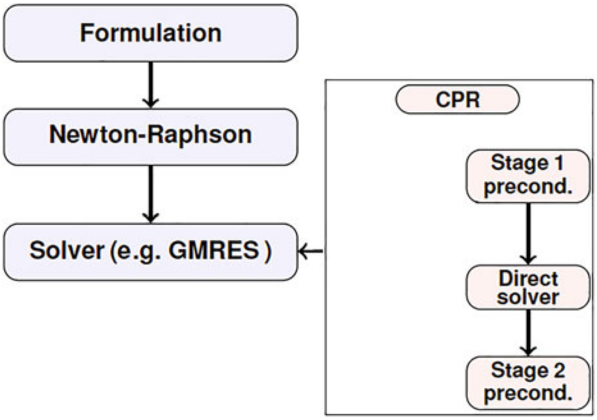


Fig. 1 Sketch of the CPR preconditioning

For the CPR, the first stage in (4) corresponds to

$$M_1^{-1} = C\tilde{J}_{pp}^{-1}C^T, \tag{6}$$

where C^T and C are respectively, restriction and prolongation operators. In particular, C is given by

$$C = \begin{pmatrix} e & & \\ & \ddots & \\ & & e \end{pmatrix} \quad \text{and} \quad e = \begin{pmatrix} 1 \\ 0 \end{pmatrix}.$$

The second stage in (4) is

$$M_2^{-1} = M_{ILU}^{-1}, \tag{7}$$

where M_{ILU}^{-1} is an ILU preconditioner.

CPR Procedure

The CPR preconditioning step $\delta = M_{CPR}^{-1}r$ can be outlined as follows:

1. Weakening of the coupling between the pressure and non pressure blocks:

$$D_1^{-1}J_G = \tilde{J} = \begin{pmatrix} \tilde{J}_{pp} & \tilde{J}_{ps} \\ \tilde{J}_{sp} & \tilde{J}_{ss} \end{pmatrix}; \tag{8}$$

2. Compute the pressure subsystem residual:

$$r_p = C^T D_1^{-1} r; \tag{9}$$

3. Solve the pressure system (e.g. with an AMG preconditioner or as a solver):

$$\tilde{J}_{pp}\delta_p = r_p; \tag{10}$$

4. Expand the pressure solution to the full system:

$$\gamma = C\delta_p = \begin{pmatrix} \delta_p \\ 0 \end{pmatrix}; \tag{11}$$

5. Compute the new residual:

$$\hat{r} = r - \tilde{J}\gamma; \tag{12}$$

6. Prediction and correction step:

$$\delta = M_2^{-1}\hat{r} + \gamma. \tag{13}$$

Here $\delta = (\delta_p, \delta_s)^t$ denotes the correction obtained after the two stages and for the sake of simplicity, we set $D_2^{-1} = I$ for the decoupling step (i.e. (8)).

Remark 1 More robust preconditioners can be formulated with the inclusion of the convective-diffusive block [6],

$$M_{CPR*}^{-1} = M_2^{-1}\left(I - \left(\tilde{J} - M_2\right)\begin{pmatrix} \tilde{J}_{pp}^{-1} & -\tilde{J}_{pp}^{-1}\tilde{J}_{ps}\tilde{J}ss^{-1} \\ 0 & \tilde{J}_{ss}^{-1} \end{pmatrix}\right). \tag{14}$$

3.2 Decoupling Operators

The decoupling introduced in (5) is a very important preprocessing step allowing to weaken the coupling between the pressure and non-pressure blocks while preserving the good algebraic properties for the extracted pressure subsystem [10, and references therein]. The main decoupling strategies usually considered in the literature are the Alternate-Block Factorization (ABF) procedure [15], the Quasi-IMPES procedure [5, 10] and the True-IMPES procedure [16].

Definition 2 Following Bank et al. [15], the ABF method is defined such that

$$D_1 = \begin{pmatrix} D_{pp} & D_{ps} \\ & \\ D_{sp} & D_{ss} \end{pmatrix} = \begin{pmatrix} diag(J_{pp}) & diag(J_{ps}) \\ & \\ diag(J_{sp}) & diag(J_{ss}) \end{pmatrix} \quad \text{and} \quad D_2 = \mathbb{I}.$$

Remark 2 Considering a dof-wise re-ordering, the ABF method corresponds to a simple to block diagonal scaling with

$$D_1 = \begin{pmatrix} (J^{pp})_{1,1} & (J^{ps})_{1,1} & & & \\ (J^{ss})_{1,1} & (J^{ss})_{1,1} & & & \\ & & \ddots & & \\ & & & (J^{pp})_{n_p,n_s} & (J^{ps})_{n_p,n_s} \\ & & & (J^{ss})_{n_p,n_s} & (J^{ss})_{n_p,n_s} \end{pmatrix}. \tag{15}$$

In this work we only focus on the ABF method owing to its structural simplicity and ease of implementation. We might although expect some potential drawbacks because $\tilde{J}_{pp}$ may be "strongly" non-symmetric compared to J_{pp}. It is also important to emphasize the fact that computing the exact inverse of $\tilde{J}_{pp}$ not feasible for large scale settings. It is therefore crucial to calibrate carefully inner and outer tolerances within the nested iterative procedure defined from (5) to (13).

4 CPR Preconditioner Performances

In this section, we analyze the performance of the two stage CPR preconditioner. We consider the 3d heterogeneous model in Fig. 2 introduced in [11]. We use a GMRES PetSc solver with a relative residual norm of 10^{-7} and a Newton tolerance of 5×10^{-7}. The computations are done in serial on a standard Intel workstation. Figure 3 and Table 1 summarize the results of this test case.

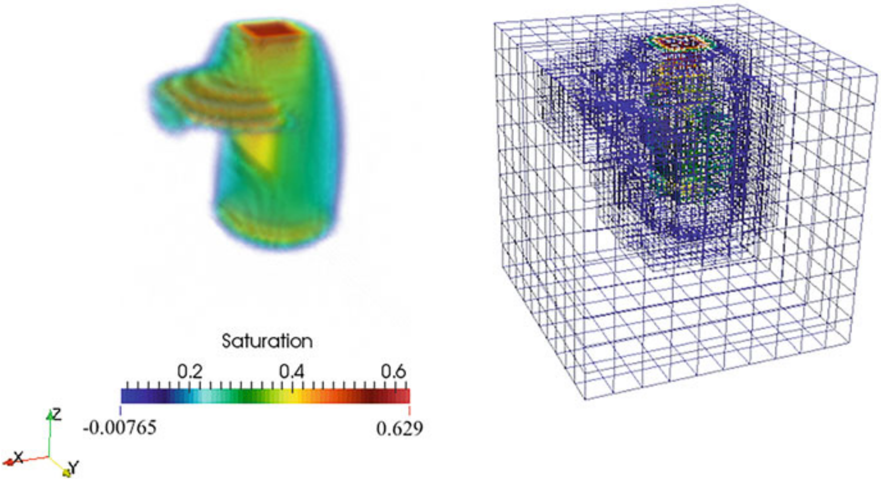


Fig. 2 3d infiltration problem

Fig. 3 Average number of linear iterations per Newton step

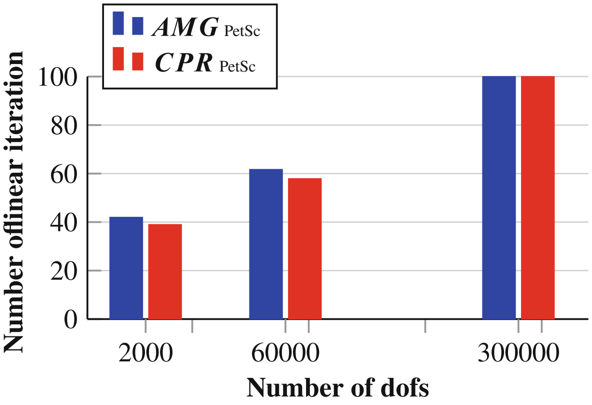
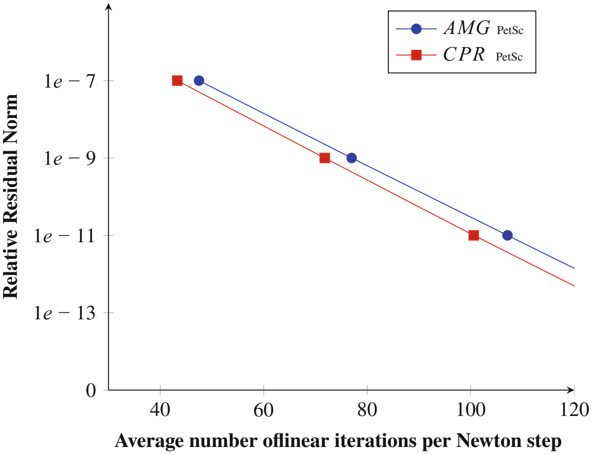


Table 1 Comparison of different preconditioners

Preconditioner	AMGPetSc	CPR PetSc
Avg lin it/Newton	125.4	120.65
Avg assem time/lin it [s]	24.27	24.69
Avg inv time/lin it [s]	8.54	11.22
Total comput time [s]	2856	3075

Fig. 4 Average convergence rates (60,000 dofs, T = 1500 s, Newton tol. 10^{-7})



The performances of the CPR and AMG are quite comparable with respect to the total CPU time. Indeed, the AMG is slightly faster up to 300,000 dofs. The relative residuals with respect to the average number of linear iterations per Newton step are depicted in Fig. 4, it illustrates the typical rate of convergence of the two preconditioners (here the Newton tolerance is 10^{-7}). In order to converge to a residual norm of less than 10^{-13}, AMG is slightly faster than CPR. However, the convergence rate of CPR is better than that of AMG.

5 Conclusion

The performances of the CPR for DG discretizations of porous media multiphase flow are not yet quite satisfactory compared to classical preconditioners such as AMG or ILU. One way to improve the performances of the CPR might consist in loosening the relative residual tolerances for the solution of the pressure subsystem as suggested in [17]. Another alternative consists in implementing more efficient decoupling operators such as the True-Impes and the Quasi-Impes [5, 6, 10, 16].

Acknowledgments The author acknowledges the German Research Foundation through the Cluster of Excellence in Simulation Technology (EXC310) at the University of Stuttgart for support.

References

1. A. Dedner, R. Klöfkorn, M. Nolte, and M. Ohlberger. A Generic Interface for Parallel and Adaptive Scientific Computing: Abstraction Principles and the DUNE-FEM Module. *Computing*, 90(3–4):165–196, 2010. https://doi.org/10.1007/s00607-010-0110-3.
2. JR Wallis et al. Incomplete Gaussian elimination as a preconditioning for generalized conjugate gradient acceleration. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1983.
3. JR Wallis, RP Kendall, TE Little, et al. Constrained residual acceleration of conjugate residual methods. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 1985.
4. A Behie, PKW Vinsome, et al. Block iterative methods for fully implicit reservoir simulation. *Society of Petroleum Engineers Journal*, 22(05):658–668, 1982.
5. Sébastien Lacroix, Yuri V Vassilevski, and Mary F Wheeler. Decoupling preconditioners in the implicit parallel accurate reservoir simulator (ipars). *Numerical linear algebra with applications*, 8(8):537–549, 2001.
6. Klaus Stueben, Tanja Clees, Hector Klie, Bo Lu, Mary Fanett Wheeler, et al. Algebraic multigrid methods (amg) for the efficient solution of fully implicit formulations in reservoir simulation. In *SPE Reservoir Simulation Symposium*. Society of Petroleum Engineers, 2007.
7. Sebastian Gries, Klaus Stüben, Geoffrey L Brown, Dingjun Chen, David A Collins, et al. Preconditioning for efficiently applying algebraic multigrid in fully implicit reservoir simulations. *SPE Journal*, 19(04):726–736, 2014.
8. Larry SK Fung, Ali H Dogru, et al. Parallel unstructured-solver methods for simulation of complex giant reservoirs. *SPE Journal*, 13(04):440–446, 2008.
9. J-M Gratien, J-F Magras, PHILIPPE Quandalle, and OLIVIER Ricois. Introducing a new generation of reservoir simulation software. In *ECMOR IX-9th European Conference on the Mathematics of Oil Recovery*, 2004.
10. Robert Scheichl, R Masson, and J Wendebourg. Decoupling and block preconditioning for sedimentary basin simulations. *Computational Geosciences*, 7(4):295–318, 2003.
11. Birane Kane. Using dune-fem for adaptive higher order discontinuous galerkin methods for strongly heterogenous two-phase flow in porous media. *Archive of Numerical Software*, 5(1), 2017. https://doi.org/10.11588/ans.2017.1.28068.
12. Birane Kane, Robert Klöfkorn, and Christoph Gersbacher. hp–Adaptive Discontinuous Galerkin Methods for Porous Media Flow. In Clément Cancès and Pascal Omnes, editors, *Finite Volumes for Complex Applications VIII - Hyperbolic, Elliptic and Parabolic Problems*,

pages 447–456, Cham, 2017. Springer International Publishing. ISBN 978-3-319-57394-6. https://doi.org/10.1007/978-3-319-57394-6_47.

13. Andreas Dedner, Birane Kane, Robert Klöfkorn, and Martin Nolte. Python framework for hp-adaptive discontinuous galerkin methods for two-phase flow in porous media. *Applied Mathematical Modelling*, 67:179–200, 2019.
14. Birane Kane, Robert Klöfkorn, and Andreas Dedner. Adaptive discontinuous galerkin methods for flow in porous media. In Florin Adrian Radu, Kundan Kumar, Inga Berre, Jan Martin Nordbotten, and Iuliu Sorin Pop, editors, *Numerical Mathematics and Advanced Applications ENUMATH 2017*, pages 367–378, Cham, 2019. Springer International Publishing. ISBN 978-3-319-96415-7.
15. Randolph E Bank, Tony F Chan, William M Coughran, and R Kent Smith. The alternate-block-factorization procedure for systems of partial differential equations. *BIT Numerical Mathematics*, 29(4):938–954, 1989.
16. Matteo Cusini, Alexander A Lukyanov, Jostein Natvig, and Hadi Hajibeygi. Constrained pressure residual multiscale (cpr-ms) method for fully implicit simulation of multiphase flow in porous media. *Journal of Computational Physics*, 299:472–486, 2015.
17. Louis J Durlofsky and Khalid Aziz. Advanced techniques for reservoir simulation and modeling of nonconventional wells. Technical report, Stanford University (US), 2004.

Biorthogonal Boundary Multiwavelets

Fritz Keinert

Abstract The discrete wavelet transform is defined for functions on the entire real line. One way to implement the transform on a finite interval is by using special boundary functions. For orthogonal multiwavelets, this has been studied in previous papers. We describe the generalization of some of these results to biorthogonal multiwavelets.

1 Introduction

The Discrete Wavelet Transform (DWT) is designed to act on infinitely long signals. For signals on a finite interval the algorithm breaks down near the boundaries. This can be dealt with by extending the data by zero padding, extrapolation, symmetry, or other methods, or by using special boundary functions. We concentrate on the latter approach.

The boundary functions can be constructed as linear combinations of scaling functions near the boundary, or from boundary recursion relations. For orthogonal multiwavelets, the relationship between the two approaches was investigated in [1].

We extend these results to the case of biorthogonal multiwavelets. Most of the content of Sects. 2–4 is a relatively straightforward generalization of previous results. We describe some of the new challenges posed by the biorthogonal setting in Sect. 5. The full algorithm and an example are given in Sects. 6 and 7.

F. Keinert (✉)
Iowa State University, Ames, IA, USA
e-mail: keinert@iastate.edu

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_57

2 Biorthogonal Multiwavelets

This section lists a few relevant properties of biorthogonal multiwavelets here, mostly to establish notation. More details can be found in [8].

The multiscaling function $\boldsymbol{\phi}$ and multiwavelet function $\boldsymbol{\psi}$ are function vectors of length r which satisfy *recursion relations*

$$\begin{aligned} \boldsymbol{\phi}(x) &= \sqrt{2}\sum_k H_k \boldsymbol{\phi}(2x-k), \\ \boldsymbol{\psi}(x) &= \sqrt{2}\sum_k G_k \boldsymbol{\phi}(2x-k), \end{aligned} \tag{1}$$

with $r \times r$ coefficient matrices H_k, G_k. In the biorthogonal setting, we have dual functions $\tilde{\boldsymbol{\phi}}$, $\tilde{\boldsymbol{\psi}}$ in addition to the primal $\boldsymbol{\phi}$, $\boldsymbol{\psi}$. These functions satisfy the *biorthogonality relations*

$$\begin{aligned} \langle \boldsymbol{\phi}(x), \tilde{\boldsymbol{\phi}}(x-\ell)\rangle &= \langle \boldsymbol{\psi}(x), \tilde{\boldsymbol{\psi}}(x-\ell)\rangle = \delta_{0\ell} I, \\ \langle \boldsymbol{\phi}(x), \tilde{\boldsymbol{\psi}}(x-\ell)\rangle &= \langle \boldsymbol{\psi}(x), \tilde{\boldsymbol{\phi}}(x-\ell)\rangle = 0 \end{aligned}$$

for all $\ell \in \mathbb{Z}$.

Properties of $\boldsymbol{\phi}$ etc. of practical interest include continuity and approximation order. Continuity can be established by estimating the joint spectral radius of certain matrices [6], or by estimating the Sobolev smoothness [7].

A multiscaling function $\boldsymbol{\phi}$ has *approximation order p* if all polynomials of degree less than p can be expressed locally as linear combinations of integer shifts of $\boldsymbol{\phi}$. The approximation order can be determined from certain sum rules. A minimum approximation order of 1 is a required condition in many theorems.

One decomposition step of the Discrete Wavelet Transform (DWT) can be described by an infinite block banded matrix

$$\tilde{\Delta} = \begin{pmatrix} \ddots & \ddots & \ddots & & & & \\ \ddots & \tilde{T}_{-1} & \tilde{T}_0 & \tilde{T}_1 & \ddots & & \\ & \ddots & \tilde{T}_{-1} & \tilde{T}_0 & \tilde{T}_1 & \ddots & \\ & & \ddots & \tilde{T}_{-1} & \tilde{T}_0 & \tilde{T}_1 & \ddots \\ & & & & \ddots & \ddots & \ddots \end{pmatrix}, \qquad \tilde{T}_k = \begin{pmatrix} \tilde{H}_{2k} & \tilde{H}_{2k+1} \\ \tilde{G}_{2k} & \tilde{G}_{2k+1} \end{pmatrix}. \tag{2}$$

Reconstruction corresponds to multiplication by Δ^*, the transpose of the corresponding matrix formed from the primal recursion coefficients H_k, G_k. We have perfect reconstruction: $\Delta^* \tilde{\Delta} = I$.

3 Refinable Boundary Functions

We will state most results in this paper for the primal functions $\boldsymbol{\phi}$, $\boldsymbol{\psi}$ only, with the understanding that corresponding results also hold for the dual functions. We also state most results only for the left boundary functions, since the notation is easier there.

We make the following assumptions:

- The underlying functions $\boldsymbol{\phi}$, $\boldsymbol{\psi}$, $\tilde{\boldsymbol{\phi}}$, $\tilde{\boldsymbol{\psi}}$ are biorthogonal, continuous, with multiplicity r and approximation order $p \geq 1$, and have recursion coefficients numbered $k = 0, \ldots, N$. If necessary, we shift the subscripts and pad the coefficient sequences with zeros. This condition implies that all functions have support in the interval $[0, N]$.
- The boundary functions have shorter support length of at most $(N-1)$.
- The interval is $[0, M]$, with $M \geq 2(N-1)$. This guarantees that the supports of the left and right endpoint functions do not overlap.
- There are L left and R right endpoint scaling and wavelet functions, grouped together into vectors $\boldsymbol{\phi}^L$ etc. We stress that this means L, R scalar functions, not function vectors, and that L, R are not necessarily multiples of r.

The left endpoint functions are called *refinable* if they satisfy recursion relations of the form

$$\begin{aligned}\boldsymbol{\phi}^L(x) &= \sqrt{2}A\boldsymbol{\phi}^L(2x) + \sqrt{2}\sum_{k=0}^{N-2} B_k\boldsymbol{\phi}(2x-k),\\ \boldsymbol{\psi}^L(x) &= \sqrt{2}E\boldsymbol{\phi}^L(2x) + \sqrt{2}\sum_{k=0}^{N-2} F_k\boldsymbol{\phi}(2x-k).\end{aligned} \tag{3}$$

We will call $\boldsymbol{\phi}^L$ a *regular boundary function* if it is refinable, continuous, and has approximation order at least 1, which implies $\boldsymbol{\phi}^L(0) \neq \mathbf{0}$. For practical applications we are usually interested in regular boundary functions. The continuity and approximation order of the boundary functions can be determined from conditions on the recursion coefficients A and B_k.

The biorthogonality conditions for boundary multiwavelets at the left end are

$$\begin{aligned}\langle \boldsymbol{\phi}^L, \tilde{\boldsymbol{\phi}}^L\rangle &= \langle \boldsymbol{\psi}^L, \tilde{\boldsymbol{\psi}}^L\rangle = I,\\ \langle \boldsymbol{\phi}^L, \tilde{\boldsymbol{\psi}}^L\rangle &= \langle \boldsymbol{\psi}^L, \tilde{\boldsymbol{\phi}}^L\rangle = 0,\\ \langle \boldsymbol{\phi}^L, \tilde{\boldsymbol{\phi}}(x-\ell)\rangle &= \langle \boldsymbol{\phi}^L, \tilde{\boldsymbol{\psi}}(x-\ell)\rangle = 0,\\ \langle \boldsymbol{\psi}^L, \tilde{\boldsymbol{\phi}}(x-\ell)\rangle &= \langle \boldsymbol{\psi}^L, \tilde{\boldsymbol{\psi}}(x-\ell)\rangle = 0.\end{aligned}$$

The first two lines describe biorthogonality among the boundary functions; the remaining two lines describe biorthogonality between boundary and interior functions.

To define the DWT on a finite interval, asssume that $N = 2K + 1$ is odd, by introducing an extra recursion coefficient $H_N = 0$ if necessary. The resulting structure for the DWT decomposition matrix is

$$\tilde{\Delta}_M = \left(\begin{array}{c|ccccccc|c} \tilde{L}_0 & \tilde{L}_1 & \cdots & \tilde{L}_K & 0 & \cdots\cdots & 0 & & 0 \\ \hline 0 & \tilde{T}_0 & \tilde{T}_1 & \cdots & \tilde{T}_K & 0 & \cdots & 0 & 0 \\ \vdots & 0 & \tilde{T}_0 & \tilde{T}_1 & \ddots & \tilde{T}_K & \ddots & \vdots & \vdots \\ \vdots & \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & 0 & \vdots \\ 0 & 0 & \cdots & 0 & \tilde{T}_0 & \tilde{T}_1 & \cdots & \tilde{T}_K & 0 \\ \hline 0 & 0 & \cdots\cdots & 0 & \tilde{R}_0 & \tilde{R}_1 & \cdots & & \tilde{R}_K \end{array} \right). \tag{4}$$

This corresponds to a segment of the infinite matrix $\tilde{\Delta}$ in (2) with some end point modifications. The $\tilde{T}_k$ are as in (2), and

$$\tilde{L}_0 = \begin{pmatrix} \tilde{A} \\ \tilde{E} \end{pmatrix}, \quad \tilde{L}_k = \begin{pmatrix} \tilde{B}_{2k-2} & \tilde{B}_{2k-1} \\ \tilde{F}_{2k-2} & \tilde{F}_{2k-1} \end{pmatrix}, \quad k = 1, \ldots, K.$$

In order to have perfect reconstruction, we require $\Delta_M^* \tilde{\Delta}_M = I$.

By a generalization of the arguments in [1] one can show that with these assumptions, perfect reconstruction is only possible if $L = \tilde{L}$, $R = \tilde{R}$, and $L + R = 2Kr$. See Sect. 5 for more details.

4 Refinable Linear Combinations

One approach that has been used repeatedly is to construct boundary functions from linear combinations of boundary-crossing multiscaling functions [3–5].

The boundary-crossing multiscaling functions are those $\boldsymbol{\phi}(x - k)$ whose support potentially contains 0 or M in its interior. At the left endpoint, these are $k = (-N + 1), \ldots, (-1)$. Since the actual support of $\boldsymbol{\phi}$ could be strictly smaller than $[0, N]$, some of these functions might in fact be in the interior or exterior, especially if the coefficients have been padded with zeros, but that causes no problems.

A linear combination of boundary-crossing basis functions is of the form

$$\boldsymbol{\phi}^L(x) = \sum_{k=-N+1}^{-1} C_k \boldsymbol{\phi}(x - k) \qquad \text{for } x > 0.$$

Constructing boundary functions from linear combinations has the advantage that the functions automatically inherit continuity, and it is easy to preserve approximation orders. However, an arbitrary linear combination is not usually refinable. This is a problem, since the DWT algorithm requires recursion coefficients.

Refinable boundary functions give rise to a DWT algorithm, but continuity and approximation order are not automatic. They can be enforced by premultiplying the coefficients with suitable matrices, but the details get rather complicated.

The best of both worlds is to look for *refinable linear combinations*. If boundary functions are both refinable and linear combinations, they must satisfy

$$\begin{aligned} CV &= AC, \\ CW &= B, \end{aligned} \tag{5}$$

where

$$C = (C_{-N+1}, C_{-N+2}, \cdots, C_{-1}),$$

$$\left(V \middle| W \right) = \left(\begin{array}{cccccc|ccccc} H_{N-1} & H_N & 0 & \cdots & \cdots & 0 & 0 & \cdots\cdots & \cdots & \cdots & 0 \\ H_{N-3} & H_{N-2} & H_{N-1} & H_N & 0 & 0 & \vdots & \ddots\ \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & H_3 & H_4 & \cdots\cdots & H_N & 0 & 0 \\ 0 & \cdots & \cdots & 0 & H_0 & H_1 & H_2 & \cdots\cdots & H_{N-2} & H_{N-1} & H_N \end{array} \right).$$

Equation (5) is a kind of eigenvalue problem, and has only a small number of solutions.

If $\boldsymbol{\phi}$ has approximation order p, then the eigenvalue solutions include the linear combinations required for approximation orders 1, 2, etc. Including them in $\boldsymbol{\phi}^L$ produces boundary functions with approximation order $\min(p, L)$. The solution corresponding to approximation order 1 is the only one with nonzero values at the endpoint, and must always be included.

5 The Biorthogonal Setting

Many of the previous results for orthogonal multiwavelets carry over to the biorthogonal case with very minor changes. In this section, we highlight two places where new results are required.

First, we need to enforce biorthogonality between $\boldsymbol{\phi}^L$ and $\tilde{\boldsymbol{\phi}}^L$. When we construct primal and dual boundary functions using Eq. (5), they are automatically biorthogonal to the interior functions, but not to each other.

In the orthogonal case, this is simply a matter of orthonormalizing the basis functions using a Gram-Schmidt process. In the biorthogonal case, it becomes a nontrivial linear algebra problem.

We observe that if M is any invertible matrix, then $\boldsymbol{\phi}^L$ and $\boldsymbol{\phi}^L_{new} = M\boldsymbol{\phi}^L$ span the same space of boundary functions, and continuity and approximation orders are preserved. We want to multiply $\boldsymbol{\phi}^L$ by a suitable M on the primal side, and $\tilde{\boldsymbol{\phi}}^L$ by some $\tilde{M}$ on the dual side, so that they become biorthogonal. It is not obvious how to find such matrices.

Second, and more importantly, we need a non-orthogonal generalization of the singular value decomposition to prove Lemma 1 below, and for other applications.

One can reduce the DWT matrix (2) to the case of only two coefficients T_0, T_1 by forming blocks. For example, if we have $H_0, \dots, H_7$ and therefore $T_0, \dots, T_3$, we form

$$\text{new}\, T_0 = \begin{pmatrix} T_0 & T_1 & T_2 \\ 0 & T_0 & T_1 \\ 0 & 0 & T_0 \end{pmatrix}, \qquad \text{new}\, T_1 = \begin{pmatrix} T_3 & 0 & 0 \\ T_2 & T_3 & 0 \\ T_1 & T_2 & T_3 \end{pmatrix}. \tag{6}$$

These new coefficients satisfy

$$\begin{aligned} T_0\tilde{T}_0^* + T_1\tilde{T}_1^* &= I, \\ T_0\tilde{T}_1^* = T_1\tilde{T}_0^* &= 0. \end{aligned} \tag{7}$$

One can prove the following lemma:

Lemma 1 *If T_0, T_1, $\tilde{T}_0$, $\tilde{T}_1$ are square matrices of size $2Kr \times 2Kr$ which satisfy relations (7), then there exist nonsingular matrices U, V so that*

$$T_0 = U \begin{pmatrix} I_{\rho_0} & 0 \\ 0 & 0_{\rho_1} \end{pmatrix} V^{-1}, \qquad T_1 = U \begin{pmatrix} 0_{\rho_0} & 0 \\ 0 & I_{\rho_1} \end{pmatrix} V^{-1},$$

$$\tilde{T}_0^* = V \begin{pmatrix} I_{\rho_0} & 0 \\ 0 & 0_{\rho_1} \end{pmatrix} U^{-1}, \qquad \tilde{T}_1^* = V \begin{pmatrix} 0_{\rho_0} & 0 \\ 0 & I_{\rho_1} \end{pmatrix} U^{-1}$$

where $\rho_0 = rank(T_0)$, $\rho_1 = rank(T_1)$, and $\rho_0 + \rho_1 = 2Kr$.

For the case of orthogonal wavelets, this is Lemma 3.1 in [2]. The proof is based on a joint Singular Value Decomposition (SVD) of T_0 and T_1. The proof for the biorthogonal case has not been published yet. It is quite similar to the orthogonal case, but requires a generalization of the SVD.

This result can then be used to prove that $L = \tilde{L} = \rho_1$, $R = \tilde{R} = \rho_0$. It can also be used to derive an algebraic completion algorithm similar to the one in [2].

6 Algorithm

A complete algorithm for constructing biorthogonal boundary wavelets at the left end proceeds as follows:

1. Determine L
2. Find primal and dual solutions of the eigenvalue problem (5)
3. Select L primal and dual boundary functions among the solutions, making sure to include the regular solution
4. Biorthogonalize the boundary functions
5. Extend the coefficient matrices to include the coefficients for the boundary multiwavelet functions $\boldsymbol{\psi}^L$, $\tilde{\boldsymbol{\psi}}^L$.

The last step is another linear algebra problem.

For both theoretical proofs and implementation, the solutions at the right end can be found by reversing the recursion coefficients (which reverses the functions), computing the left endpoint functions, and reversing again.

7 Example

The YCW biorthogonal multiwavelet is described as example 1 in [9]. Both primal and dual scaling functions have an approximation order of 2, and all functions have support in [0, 2]. The multiscaling functions are shown in the top row of Fig. 1.

At the left end, the eigenvalue problem (5) has four solutions, but only two of them correspond to functions nonzero for $x \geq 0$. One of them is regular. Likewise, there are two nonzero dual functions, one of them regular. We find $\rho_1 = 1$, so we only get to choose one left endpoint function, which needs to be the regular solution. These are shown in the middle row of Fig. 1.

At the right end, there are also four solutions, this time all of them nonzero. Since $\rho_0 = 3$ we need three functions at the right end. One choice is shown in the bottom row of Fig. 1.

8 Summary

In two previous papers [1, 2] we developed a strategy and an algorithm for the construction of boundary functions for orthogonal wavelets and multiwavelets on an interval. The feasibility of this approach was demonstrated with several examples. In the present proceedings, we have outlined a generalization of this approach to the biorthogonal case. The biorthogonal setting presents some additional challenges, but the development mostly proceeds along the same lines as before. Further details will be published in the near future.

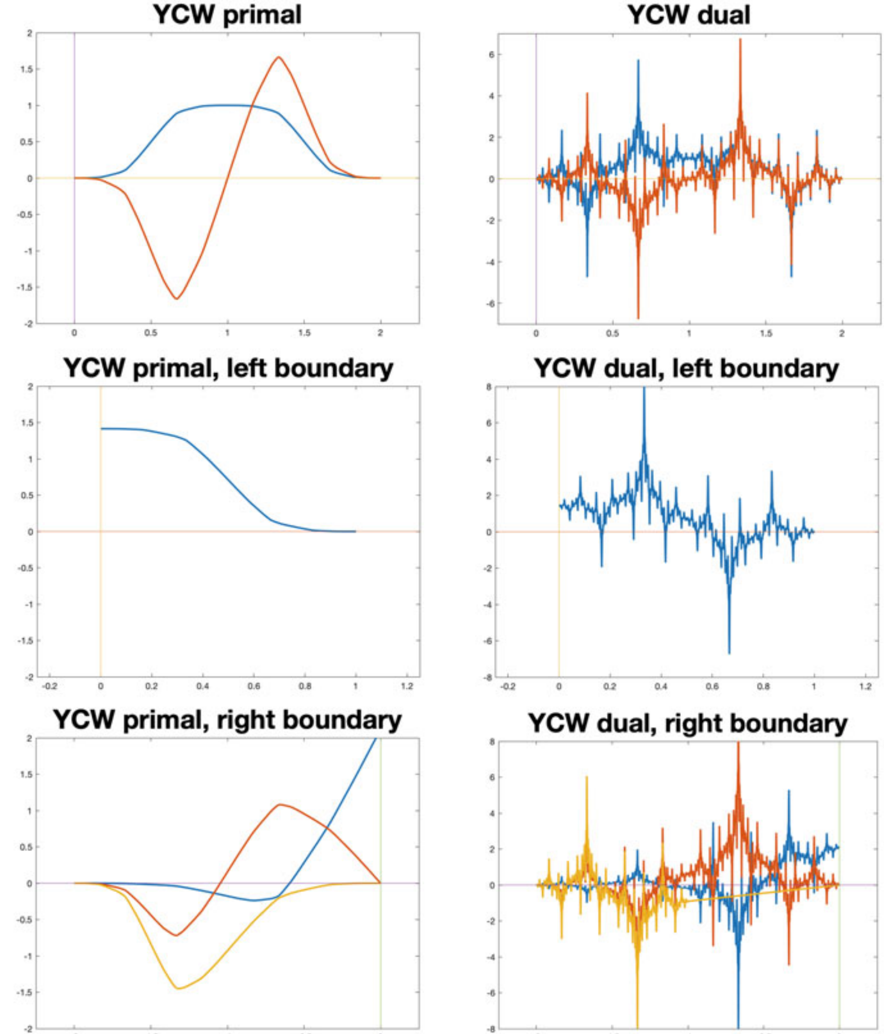


Fig. 1 Top row: The original primal and dual multiscaling functions of the YCW multiwavelet. Middle row: Primal and dual left boundary scaling function, after biorthogonalization. Bottom row: Primal and dual right boundary scaling functions, after biorthogonalization

References

1. Altürk, A., Keinert, F.: Regularity of boundary wavelets. Appl. Comput. Harmon. Anal. **32**(1), 65–85 (2012)
2. Altürk, A., Keinert, F.: Construction of multiwavelets on an interval wavelets. Axioms **2**(2), 122–141 (2013)

3. Andersson, L., Hall, N., Jawerth, B., Peters, G.: Wavelets on closed subsets of the real line. In: Recent advances in wavelet analysis, *Wavelet Anal. Appl.*, vol. 3, pp. 1–61. Academic Press, Boston, MA (1994)
4. Chui, C.K., Quak, E.: Wavelets on a bounded interval. In: Numerical methods in approximation theory, Vol. 9 (Oberwolfach, 1991), *Internat. Ser. Numer. Math.*, vol. 105, pp. 53–75. Birkhäuser, Basel (1992)
5. Cohen, A., Daubechies, I., Vial, P.: Wavelets on the interval and fast wavelet transforms. Appl. Comput. Harmon. Anal. **1**(1), 54–81 (1993)
6. Daubechies, I., Lagarias, J.C.: Two-scale difference equations II: local regularity, infinite products of matrices and fractals. SIAM J. Math. Anal. **23**(4), 1031–1079 (1992)
7. Jiang, Q.: On the regularity of matrix refinable functions. SIAM J. Math. Anal. **29**(5), 1157–1176 (1998)
8. Keinert, F.: Wavelets and multiwavelets. Studies in Advanced Mathematics. Chapman & Hall/CRC, Boca Raton, FL (2004)
9. Yang, S., Cheng, Z., Wang, H.: Construction of biorthogonal multiwavelets. J. Math. Anal. Appl. **276**, 1–12 (2002)

Machine Learning in Adaptive FETI-DP: Reducing the Effort in Sampling

Alexander Heinlein, Axel Klawonn, Martin Lanser, and Janine Weber

Abstract The convergence rate of classic domain decomposition methods in general deteriorates severely for large discontinuities in the coefficient functions of the considered partial differential equation. To retain the robustness for such highly heterogeneous problems, the coarse space can be enriched by additional coarse basis functions. These can be obtained by solving local generalized eigenvalue problems on subdomain edges. In order to reduce the number of eigenvalue problems and thus the computational cost, we use a neural network to predict the geometric location of critical edges, i.e., edges where the eigenvalue problem is indispensable. As input data for the neural network, we use function evaluations of the coefficient function within the two subdomains adjacent to an edge. In the present article, we examine the effect of computing the input data only in a neighborhood of the edge, i.e., on slabs next to the edge. We show numerical results for both the training data as well as for a concrete test problem in form of a microsection subsection for linear elasticity problems. We observe that computing the sampling points only in one half or one quarter of each subdomain still provides robust algorithms.

A. Heinlein · A. Klawonn (✉) · M. Lanser
Department of Mathematics and Computer Science, University of Cologne, Köln, Germany
http://www.numerik.uni-koeln.de

Center for Data and Simulation Science, University of Cologne, Köln, Germany
e-mail: alexander.heinlein@uni-koeln.de; axel.klawonn@uni-koeln.de;
martin.lanser@uni-koeln.de
http://www.cds.uni-koeln.de

J. Weber
Department of Mathematics and Computer Science, University of Cologne, Köln, Germany
e-mail: janine.weber@uni-koeln.de
http://www.numerik.uni-koeln.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_58

1 Introduction

Domain decomposition methods are highly scalable iterative solvers for large linear systems of equations, e.g., arising from the discretization of partial differential equations. While scalability results from a decomposition of the computational domain into local subdomains, i.e., from a divide and conquer principle, robustness is obtained by enforcing certain constraints, e.g., continuity in certain variables or averages over variables on the interface between neighboring subdomains. These constraints build a global coarse problem or second level. Nevertheless, the convergence rate of classic domain decomposition approaches deteriorates or even stagnates for large discontinuities in the coefficients of the partial differential equation considered. As a remedy, to retrieve a robust algorithm, several adaptive approaches to enrich the coarse space with additional constraints obtained from the solution of generalized eigenvalue problems have been developed, e.g., [2, 3, 7, 8, 11, 12]. The eigenvalue problems are in general localized to parts of the interface, e.g., edges or faces. In the present paper, we only consider two-dimensional problems for simplicity and thus only eigenvalue problems on edges. Let us remark that for many realistic coefficient distributions, only a few adaptive constraints on a few edges are necessary to obtain a robust algorithm and thus many expensive solutions of eigenvalue problems can be omitted. Although some heuristic approaches [7, 8] exist to reduce the number of eigenvalue problems, in general, it is difficult to predict a priori which eigenvalue problems are necessary for robustness. In [5], we successfully used a neural network to predict the geometric location of necessary constraints in a preprocessing step, i.e., to automatically make the decision whether or not we have to solve a specific eigenvalue problem. Additionally, we discussed the feasibility of randomly, and thus automatically generated training data in [4]. In the present paper, we extend the results given in [4] by providing also results for linear elasticity problems.

Both in [5] and [4], we use samples of the coefficient function as input data for the neural network. In particular, in case of regular decompositions, the resulting sampling points cover the complete neighboring subdomains for a specific edge. Even though the training of the neural network as well as the generation of the training data can be performed in an offline-phase, we aim to further optimize the complexity of our approach by reducing the size of the input data by using fewer sampling points; see also Fig. 1 for an illustration. In particular, for the first time, we only compute sampling points for slabs of varying width around a specific edge between two subdomains. Since our machine learning problem, in principle, is an image classification task, this corresponds to the idea of using only a fraction of pixels of the original image as input data for the neural network. We show numerical results for linear elasticity problems in two dimensions. As in [5], we focus on a certain adaptive coarse space for the FETI-DP (Finite Element Tearing and Interconnecting—Dual Primal) algorithm [11, 12].

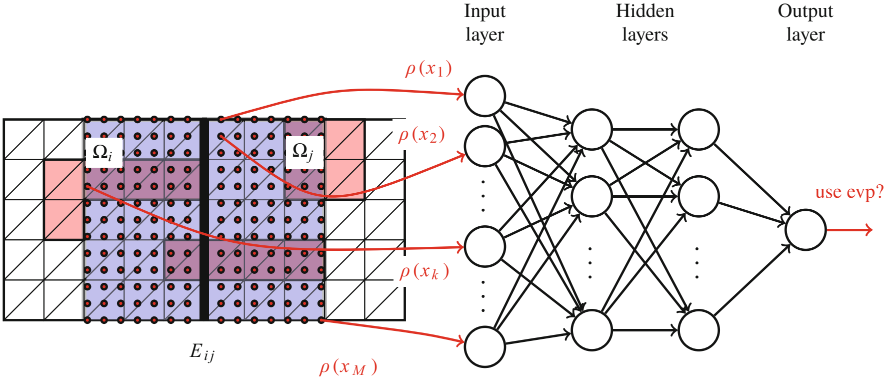


Fig. 1 Sampling of the coefficient function; white color corresponds to a low coefficient and red color to a high coefficient. In this representation, the samples are used as input data for a neural network with two hidden layers. Only sampling points from slabs around the edge are chosen

2 Linear Elasticity and an Adaptive FETI-DP Algorithm

In our numerical experiments, we exclusively consider linear elasticity problems. We denote by $\mathbf{u} : \Omega \to \mathbb{R}^2$ the displacement of an elastic body, which occupies the domain Ω in its undeformed state. We further denote by f a given volume force and by g a given surface force onto the body. The problem of linear elasticity then consists in finding the displacement $\mathbf{u} \in \mathbf{H}_0^1(\Omega, \partial\Omega_D)$, such that

$$\int_\Omega G\, \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v})\, d\mathbf{x} + \int_\Omega G\beta\, \mathrm{div}\mathbf{u}\, \mathrm{div}\mathbf{v}\, d\mathbf{x} = \langle \mathbf{F}, \mathbf{v} \rangle$$

for all $\mathbf{v} \in \mathbf{H}_0^1(\Omega, \partial\Omega_D)$ for given material functions $G : \Omega \to \mathbb{R}$ and $\beta : \Omega \to \mathbb{R}$ and the right-hand side

$$\langle \mathbf{F}, \mathbf{v} \rangle = \int_\Omega \mathbf{f}^T \mathbf{v}\, d\mathbf{x} + \int_{\partial\Omega_N} \mathbf{g}^T \mathbf{v}\, d\sigma.$$

The material parameters G and β depend on the Young modulus $E > 0$ and the Poisson ratio $\nu \in (0, 1/2)$ given by $G = E/(1+\nu)$ and $\beta = \nu/(1-2\nu)$. Here, we restrict ourselves to compressible linear elasticity; hence, the Poisson ratio ν is bounded away from $1/2$.

In the present article, we apply the proposed machine learning based strategy to an adaptive FETI-DP method, which is based on a nonoverlapping domain decomposition of the computational domain Ω. Here, we decompose Ω into N regular subdomains of width H and discretize each subdomain by finite elements of width h. For simplicity, we assume matching nodes on the interface between subdomains. Due to space limitations, we do not explain the standard FETI-DP algorithm in detail. For a detailed description, see, e.g., [10]. Let us just note that we enforce continuity in all vertices of all subdomains to define an initial coarse space.

As already mentioned in Sect. 1, for arbitrary and complex material distributions, e.g., a highly varying Young modulus E, using solely primal vertex constraints is not sufficient to guarantee a robust condition number bound. Thus, additional adaptive constraints, typically obtained from the solution of local generalized eigenvalue problems, are used to enrich the coarse space and retrieve robustness.

The central idea of the adaptive FETI-DP algorithm [11, 12] is to solve local generalized eigenvalue problems for all edges between two neighboring subdomains. For a description of the local edge eigenvalue problems and the resulting enforced coarse constraints, see [11, 12]. In a parallel implementation, the set-up, e.g., the computation of local Schur complements, and the solution of the local eigenvalue problems take up a significant amount of the total time to solution. To reduce the set-up cost without losing robustness, a precise a priori prediction of all edges, where an eigenvalue problem is useful, is necessary; see Sect. 3 for a description of our machine learning based approach. Let us remark, that the additional adaptive constraints are implemented in FETI-DP using a balancing preconditioner. For a detailed description of projector or balancing preconditioning, see [6, 9].

3 Machine Learning for Adaptive FETI-DP

Our approach (ML-FETI-DP) is to train a neural network to automatically make the decision whether an adaptive constraint needs to be enforced or not on a specific edge to retain the robustness of the adaptive FETI-DP algorithm. This corresponds to a supervised machine learning technique.

Sampling Strategy and Neural Network More precisely, we use a dense feedforward neural network, i.e., a multilayer perceptron, to make this decision. For more details on multilayer perceptrons, see, e.g., [1, 13, 14]. As in [5], we use samples, i.e., function evaluations of Young's modulus within the two subdomains adjacent to an edge, as input data for our neural network. Note that our sampling approach is independent of the finite element discretization. In particular, we assume that the sampling grid resolves all geometrical details of the coefficient function or the material distribution. The output of our neural network is the classification whether an adaptive constraint has to be included for a specific edge or not. Our neural network consists of three hidden layers with 30 neurons for each hidden layer. We use the ReLU activation function for all hidden layers and a dropout rate of 20%. See [5] for more details on the machine learning techniques and the preparation of data.

Training Data Sets For the numerical results presented here, we train on two regular subdomains sharing a straight edge. In general, also irregular domain

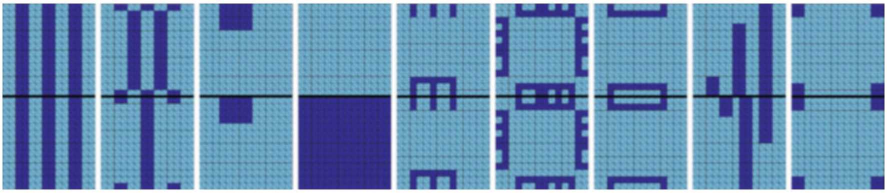


Fig. 2 Nine different types of coefficient functions used for training and validation of the neural network. The inclusions, channels, boxes, and combs with high coefficient are displaced, modified in sized, and mirrored with respect to the edge in order to generate the complete *smart data* set

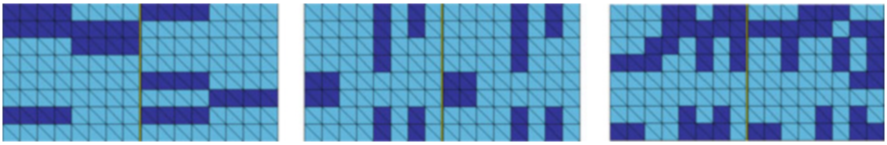


Fig. 3 Examples of three different coefficient functions taken from the *random data* set obtained by using the same randomly generated coefficient for a horizontal (left) or vertical (middle) stripe of a maximum length of four finite element pixels, as well as by pairwise superimposing (right)

decompositions can be considered; see [5]. As in [4], we use different sets of coefficient distributions to generate different sets of training data for the neural network. For the first set of training data, we use a total of 4500 configurations varying the coefficient distributions as presented in Fig. 2. We set the Poisson ratio ν constantly and just vary the Young modulus E. The coefficient distributions in Fig. 2 are varied in size, orientation and location to obtain the full set of training data. We refer to this set of training data as *smart data*; see also [4]. We further consider a randomly generated training data set. In particular, we use the same training sets as in [4] but now additionally provide results for linear elasticity problems. Note that completely randomized coefficient distributions lead to insufficient accuracies and too many false negatives; see [4] for details. Instead, we explicitly control the ratio of high coefficients as well as the distributions of the coefficients to a certain degree by randomly generating either horizontal or vertical stripes of a maximum length of four or eight pixels, respectively; see Fig. 3. We refer to this second set of training data as *random data* and we also consider combinations of both, the *smart* and *random data* sets. For more technical details on the construction of the data sets, we refer to [4].

Sampling Strategy on Slabs We now describe how we reduce the number of sampling points used as input data for the neural network. In [5], the computed sampling grid covers both neighboring subdomains of an edge entirely—at least in case of a regular domain decomposition. Let us remark that in case of irregular domain decompositions, our sampling strategy might miss small areas further away from the edge; see, e.g., [5, Fig. 4]. However, this does not affect the performance of our algorithm. Although the preparation of the training data as well as the training

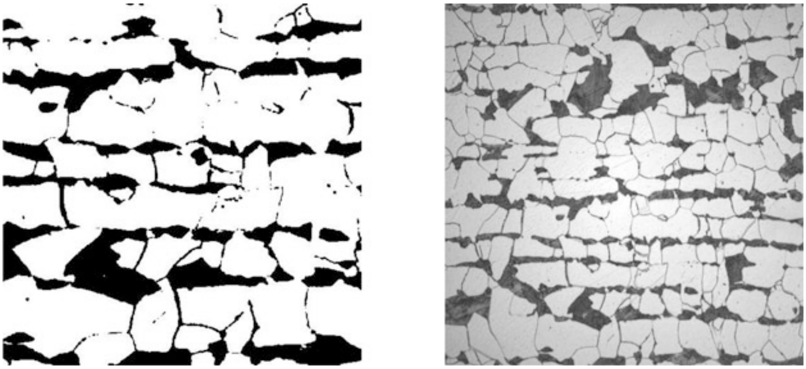


Fig. 4 Left: Subsection of a microsection of a dual-phase steel obtained from the image on the right. We consider $E = 1e3$ in the black part and $E = 1$ elsewhere. Right: Complete microsection of a dual-phase steel. Right image: Courtesy of Jörg Schröder, University of Duisburg-Essen, Germany, orginating from a cooperation with ThyssenKruppSteel

of the neural network can be performed in an offline-phase, we try to generate the training data as efficient and fast as possible. For all sampling points, we need to determine the corresponding finite element as well as to evaluate the coefficient function for the respective finite element. Therefore, there is clearly potential to save resources and compute time in the training as well as in the evaluation phase by reducing the number of sampling points used as input data for the neural network. In general, the coefficient variations close to the edge are the most relevant, i.e., the most critical for the condition number bound of FETI-DP. Therefore, to reduce the total number of sampling points in the sampling grid, reducing the density of the grid points with increasing distance to the edge is a natural approach. More drastically, one could exclusively consider sampling points in a neighborhood of the edge, i.e., on slabs next to the edge. We consider the latter approach here; see also Fig. 1 for an illustration of the sampling points inside slabs.

To generate the output data necessary to train the neural network, we solve the eigenvalue problems as described in [11, 12] for all the aforementioned training and validation configurations. Concerning the classification of the edges, we use both a two-class and a three-class classification approach. For the two-class classification, we only distinguish between edges of class 0 and class 1. By class 0 we denote edges for which no additional constraints are necessary for a robust algorithm, and by class 1 edges where at least one constraint is required. For the three-class classification, we further differentiate between class 1 and class 2. In this case, we assign only edges to class 1, where exactly one additional constraint is necessary, and assign all other edges, for which more than one constraint is necessary, to class 2.

4 Numerical Results

We first provide results for some microsection problems, i.e., linear elasticity problems with a material distribution as shown in Fig. 4 (left). We considered the different training sets *smart data*, *random data*, and a combination of both; see Table 1. Here, we exclusively use the approach of sampling on the complete subdomains. Since training with the *smart data* set seems to be the best choice for this specific example, we exclusively use this data set in the following investigations with slabs of different width. Additionally, using the ML threshold $\tau = 0.45$ for the two-class classification or $\tau = 0.4$ for the three-class classification, respectively, leads to the most robust results when sampling on the complete subdomains. We therefore focus on these thresholds in the following discussion.

We compare the performance of the original sampling approach introduced in [5] to sampling in parts of each subdomain of width H, i.e., in one half and in one quarter (see also Fig. 1). We also consider the extreme case, i.e., sampling only inside minimal slabs of the width of a single finite element. For the training data, both sampling in $H/2$ and $H/4$ leads to accuracy values which are only slightly lower than for the full sampling approach (see Table 2). In particular, we get slightly higher false positive values, especially for the three-class classification. For the extreme case of sampling only in slabs of width h, i.e., using slabs with the minimal possible width in terms of finite elements, the accuracy value drops from 92.8% for the three-class model to only 68.4% for the threshold $\tau = 0.4$. Note that we did not observe a significant improvement for this sampling strategy for more complex network architectures. Thus, it is questionable if the latter sampling approach still provides a reliable machine learning model. For the microsection problem, sampling in slabs of width $H/2$ and $H/4$ results in robust algorithms for both the two-class and the three-class model when using the ML threshold $\tau = 0.45$ or $\tau = 0.4$, respectively; see Tables 3 and 4. For all these approaches, we obtain no false negative edges, which are critical for the convergence of the algorithm. However, the use of fewer sampling points results in more false positive edges and therefore in a larger number of computed eigenvalue problems. When sampling only in slabs of width h, we do not obtain a robust algorithm for the microsection problem for neither the two-class nor the three-class classification. This is caused by the existence of a relatively high number of false negative edges.

Let us summarize that reducing the effort in the training and evaluation of the neural network by reducing the size of the sampling grid still leads to a robust algorithm for our model problems. Nevertheless, the slab width cannot be chosen too small and enough finite elements close to the edge have to be covered by the sampling grid.

Table 1 Comparison of standard FETI-DP, adaptive FETI-DP, and ML-FETI-DP for a **regular domain decomposition** into 8 × 8 subdomains with $H/h = 64$, **linear elasticity**, the **two-class model**, and 10 different subsections of the microsection in Fig. 4 (right). We denote by 'S' the training set of 4500 *smart data*, by 'R1' and 'R2' a set of 4500 and 9000 *random data*, respectively, and by 'SR' the combination of 4500 *smart* and 4500 *random data*. We show the ML-threshold (τ), the condition number (cond), the number of CG iterations (it), the number of solved eigenvalue problems (evp), the number of false positives (fp), the number of false negatives (fn), and the accuracy in the classification (acc). We define the accuracy (acc) as the number of true positives and true negatives divided by the total number of training configurations. We show the average values as well as the maximum values (in brackets)

Alg.	T-data	τ	cond	it	evp	fp	fn	acc
Standard	–	-	–	>300	0	–	–	–
Adaptive	–	-	79.1 (92.8)	87.4 (91)	112.0 (112)	–	–	–
ML	S	0.5	9.3e4 (1.3e5)	92.2 (95)	44.0 (56)	2.2 (3)	2.4 (3)	0.96 (0.95)
	S	0.45	79.1 (92.8)	87.4 (91)	48.2 (61)	4.8 (7)	0 (0)	0.95 (0.93)
	R1	0.45	1.7e3 (2.1e4)	90.4 (91)	53.6 (57)	13.4 (16)	0.8 (1)	0.87 (0.86)
	R2	0.45	79.1 (92.8)	87.4 (91)	52.8 (57)	11.6 (12)	0 (0)	0.90 (0.87)
	SR	0.45	79.1 (92.8)	87.4 (91)	50.6 (61)	8.8 (10)	0 (0)	0.92 (0.90)

Table 2 Results on the complete training data set for **linear elasticity**; the numbers are averages over all training configurations. See Table 1 for the column labelling

	Two-class				Three-class			
Training configuration	τ	fp	fn	acc	τ	fp	fn	acc
Full sampling	0.45	8.9%	2.7%	88.4%	0.4	5.2%	2.0%	92.8%
	0.5	5.5%	5.6%	88.9%	0.5	3.3%	3.3%	93.4%
Sampling in $H/2$	0.45	8.0%	2.6%	89.4%	0.4	9.6%	4.3%	86.1%
	0.5	5.9%	4.0%	90.1%	0.5	7.4%	5.0%	87.6%
Sampling in $H/4$	0.45	8.2%	2.7%	89.1%	0.4	10.4%	3.9%	85.7%
	0.5	5.7%	4.5%	89.8%	0.5	8.1%	4.8%	87.1%
Sampling in h	0.45	20.8%	7.5%	71.7%	0.4	22.4%	9.2%	68.4%
	0.5	15.4%	12.9%	72.3%	0.5	15.0%	15.3%	69.7%

Table 3 Comparison of standard FETI-DP, adaptive FETI-DP, and ML-FETI-DP for a **regular domain decomposition** into 8 × 8 subdomains with $H/h = 64$, **linear elasticity**, the **two-class model**, and the microsection subsection in Fig. 4 (left). See Table 1 for the column labelling

Model problem	Algorithm	τ	cond	it	evp	fp	fn	acc
Microsection problem	Standard	–	–	>300	0	–	–	–
	Adaptive	–	84.72	89	112	–	–	–
	ML, full sampling	0.5	9.46e4	91	41	2	2	0.96
	ML, full sampling	0.45	84.72	89	46	5	0	0.95
	ML, sampling in $H/2$	0.45	84.72	89	47	6	0	0.95
	ML, sampling in $H/4$	0.45	85.31	90	48	7	0	0.94
	ML, sampling in h	0.45	10.9e5	137	50	19	10	0.74

Table 4 Comparison of standard FETI-DP, adaptive FETI-DP, and ML-FETI-DP for a **regular domain decomposition** into 8 × 8 subdomains with $H/h = 64$, **linear elasticity**, the **three-class model**, and the microsection subsection in Fig. 4 (left). See Table 1 for the column labelling. Here, **e-avg** denotes an approximation of the coarse constraints for edges in class 1, see also [5]

Model problem	Algorithm	τ	cond	it	evp	e-avg	fp	fn	acc
Microsection problem	Standard	–	–	>300	0	–	–	–	–
	Adaptive	–	84.72	89	112	–	–	–	–
	ML, full sampling	0.5	274.73	101	15	31	3	2	0.96
	ML, full sampling	0.4	86.17	90	22	24	6	0	0.95
	ML, sampling in $H/2$	0.4	85.29	90	25	26	9	0	0.92
	ML, sampling in $H/4$	0.4	85.37	90	25	27	10	0	0.92
	ML, sampling in h	0.4	2.43e4	111	27	52	29	7	0.68

References

1. I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*, volume 1. MIT press Cambridge, 2016.
2. A. Heinlein, A. Klawonn, J. Knepper, and O. Rheinbach. An adaptive GDSW coarse space for two-level overlapping Schwarz methods in two dimensions. 2019. Proceedings of the International Conference on Domain Decomposition Methods 24, Springer LNCSE, Vol. 125, January 2019, pp. 373–382. https://doi.org/10.1007/978-3-319-93873-8_35.
3. A. Heinlein, A. Klawonn, J. Knepper, and O. Rheinbach. Multiscale coarse spaces for overlapping Schwarz methods based on the ACMS space in 2D. *Electronic Transactions on Numerical Analysis (ETNA)*, 48:156–182, 2018.
4. A. Heinlein, A. Klawonn, M. Lanser, and J. Weber. Machine Learning in Adaptive FETI-DP - A Comparison of Smart and Random Training Data. 2018. TR series, Center for Data and Simulation Science, University of Cologne, Germany, Vol. 2018-5. http://kups.ub.uni-koeln.de/id/eprint/8645. Accepted for publication in the proceedings of the International Conference on Domain Decomposition Methods 25, Springer LNCSE, May 2019.
5. A. Heinlein, A. Klawonn, M. Lanser, and J. Weber. Machine Learning in Adaptive Domain Decomposition Methods - Predicting the Geometric Location of Constraints. *SIAM J. Sci. Comput.*, 41(6):A3887–A3912, 2019.
6. M. Jarošová, A. Klawonn, and O. Rheinbach. Projector preconditioning and transformation of basis in FETI-DP algorithms for contact problems. *Math. Comput. Simulation*, 82(10):1894–1907, 2012.

7. A. Klawonn, M. Kühn, and O. Rheinbach. Adaptive coarse spaces for FETI-DP in three dimensions. *SIAM J. Sci. Comput.*, 38(5):A2880–A2911, 2016.
8. A. Klawonn, M. Kühn, and O. Rheinbach. Adaptive FETI-DP and BDDC methods with a generalized transformation of basis for heterogeneous problems. *Electron. Trans. Numer. Anal.*, 49:1–27, 2018.
9. A. Klawonn and O. Rheinbach. Deflation, projector preconditioning, and balancing in iterative substructuring methods: connections and new results. *SIAM J. Sci. Comput.*, 34(1):A459–A484, 2012.
10. A. Klawonn and O. B. Widlund. Dual-primal FETI methods for linear elasticity. *Comm. Pure Appl. Math.*, 59(11):1523–1572, 2006.
11. J. Mandel and B. Sousedík. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1389–1399, 2007.
12. J. Mandel, B. Sousedík, and J. Sístek. Adaptive BDDC in three dimensions. *Math. Comput. Simulation*, 82(10):1812–1831, 2012.
13. A. Müller and S. Guido. *Introduction to Machine Learning with Python: A Guide for Data Scientists*. O'Reilly Media, 2016.
14. S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning*. Cambridge University Press, 2014.

A New Algebraically Stabilized Method for Convection–Diffusion–Reaction Equations

Petr Knobloch

Abstract This paper is devoted to algebraically stabilized finite element methods for the numerical solution of convection–diffusion–reaction equations. First, the algebraic flux correction scheme with the popular Kuzmin limiter is presented. This limiter has several favourable properties but does not guarantee the validity of the discrete maximum principle for non-Delaunay meshes. Therefore, a generalization of the algebraic flux correction scheme and a modification of the limiter are proposed which lead to the discrete maximum principle for arbitrary meshes. Numerical results demonstrate the advantages of the new method.

1 Introduction

The aim of this paper is the numerical solution of the scalar steady-state convection–diffusion–reaction problem

$$-\,\varepsilon\,\Delta u + \boldsymbol{b}\cdot\nabla u + c\,u = g \quad \text{in}\ \Omega\,, \qquad\qquad u = u_b \quad \text{on}\ \partial\Omega\,, \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, $d \ge 1$, is a bounded domain with a Lipschitz-continuous boundary $\partial\Omega$ that is assumed to be polyhedral (if $d \ge 2$). Furthermore, $\varepsilon > 0$ is a constant and $\boldsymbol{b} \in W^{1,\infty}(\Omega)^d$, $c \in L^\infty(\Omega)$, $g \in L^2(\Omega)$, and $u_b \in H^{\frac{1}{2}}(\partial\Omega) \cap C(\partial\Omega)$ are given functions satisfying $\nabla\cdot\boldsymbol{b} = 0$ and $c \ge 0$ in Ω.

In particular, we are interested in the convection-dominated case $\varepsilon \ll |\boldsymbol{b}|$ whose numerical solution still represents a challenge. This is due to the fact that, in this case, the solution of (1) typically contains layers and the approximate solutions obtained using standard discretization techniques are then usually polluted by spurious oscillations unless the layers are resolved by the mesh. In the finite

P. Knobloch (✉)
Charles University, Faculty of Mathematics and Physics, Department of Numerical Mathematics, Prague, Czech Republic
e-mail: knobloch@karlin.mff.cuni.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_59

element framework, the usual strategy to suppress the spurious oscillations is to add stabilization terms to the discrete variational formulation. The stabilization introduces a certain amount of artificial diffusion which should be not too small to suppress the spurious oscillations sufficiently but also not too large to avoid excessive smearing of the layers. To restrict the addition of artificial diffusion to regions where it is really needed, nonlinear techniques have been developed. An alternative to modifying the discrete variational formulation by adding additional integral terms is to modify the corresponding algebraic problem by purely algebraic means, see, e.g., [7]. The advantage of these techniques is that they satisfy the discrete maximum principle (DMP) by construction (so that spurious oscillations cannot appear) and often provide sharp approximations of layers. Techniques of this type are the subject of the present paper.

First, in Sect. 2, we formulate two finite element discretizations of problem (1) and write down the corresponding linear algebraic problem. Then, in Sect. 3, we formulate the algebraic flux correction (AFC) scheme as considered, e.g., in [6, 7] and describe the Kuzmin limiter [5]. This limiter has various favourable properties, however, it does not guarantee the validity of the DMP for non-Delaunay meshes. Moreover, if the reaction term is not lumped, then the DMP may be not satisfied also on Delaunay meshes. Therefore, in Sect. 4, we generalize the AFC scheme and introduce a modification of the Kuzmin limiter, leading to a new algebraically stabilized method satisfying the DMP on arbitrary meshes and without a lumping of the reaction term. Finally, in Sect. 5, we present numerical results illustrating the properties of the new method.

2 Finite Element Discretization

To define a finite element discretization of problem (1), we first introduce its weak formulation: The weak solution of (1) is a function $u \in H^1(\Omega)$ satisfying the boundary condition $u = u_b$ on $\partial\Omega$ and the variational equation

$$a(u, v) = (g, v) \qquad \forall\, v \in H_0^1(\Omega)\,,$$

where

$$a(u, v) = \varepsilon\,(\nabla u, \nabla v) + (\boldsymbol{b} \cdot \nabla u, v) + (c\, u, v)\,. \tag{2}$$

As usual, $(\cdot, \cdot)$ denotes the inner product in $L^2(\Omega)$ or $L^2(\Omega)^d$. It is well known that the weak solution of (1) exists and is unique. An important property of problem (1) is that, for $c \geq 0$ in Ω, its solutions satisfy the maximum principle.

Now let $\mathscr{T}_h$ be a simplicial triangulation of $\overline{\Omega}$ that belongs to a regular family of triangulations. Furthermore, let us introduce finite element spaces

$$W_h = \{v_h \in C(\overline{\Omega})\,;\ v_h|_T \in P_1(T)\ \forall\, T \in \mathscr{T}_h\}\,, \qquad V_h = W_h \cap H_0^1(\Omega)\,,$$

consisting of continuous piecewise linear functions. The vertices of the triangulation $\mathscr{T}_h$ will be denoted by $x_1, \dots, x_N$ and we assume that $x_1, \dots, x_M \in \Omega$ and $x_{M+1}, \dots, x_N \in \partial\Omega$. Then the usual basis functions $\varphi_1, \dots, \varphi_N$ of W_h are defined by the conditions $\varphi_i(x_j) = \delta_{ij}$, $i, j = 1, \dots, N$, where δ_{ij} is the Kronecker symbol. Obviously, the functions $\varphi_1, \dots, \varphi_M$ form a basis of V_h.

Now an approximate solution of problem (1) can be introduced as the solution of the following finite-dimensional problem: Find $u_h \in W_h$ such that $u_h(x_i) = u_b(x_i)$, $i = M+1, \dots, N$, and

$$a_h(u_h, v_h) = (g, v_h) \qquad \forall\ v_h \in V_h\,, \tag{3}$$

where a_h is a bilinear form approximating the bilinear form a. In what follows, we shall consider two choices of a_h. The first one is simply $a_h = a$, the second one is to set

$$a_h(u_h, v_h) = \varepsilon\,(\nabla u_h, \nabla v_h) + (\boldsymbol{b}\cdot\nabla u_h, v_h) + \sum_{i=1}^{M} (c, \varphi_i)\, u_i\, v_i \tag{4}$$

for any $u_h \in W_h$ and $v_h \in V_h$, i.e., to consider a lumping of the reaction term $(c\,u_h, v_h)$ in $a(u_h, v_h)$. This may help to satisfy the discrete maximum principle for problem (3), cf. Sect. 3.

We denote $a_{ij} = a_h(\varphi_j, \varphi_i)$, $i, j = 1, \dots, N$, $g_i = (g, \varphi_i)$, $i = 1, \dots, M$, and $u_i^b = u_b(x_i)$, $i = M+1, \dots, N$. Then $u_h \equiv \sum_{i=1}^N u_i\,\varphi_i$ is a solution of the finite-dimensional problem (3) if and only if the coefficient vector $\mathrm{U} = (u_1, \dots, u_N)$ satisfies the algebraic problem

$$\sum_{j=1}^{N} a_{ij}\,u_j = g_i\,, \qquad i = 1, \dots, M\,, \tag{5}$$

$$u_i = u_i^b\,, \qquad i = M+1, \dots, N\,. \tag{6}$$

In the convection-dominated regime, the above discretizations do not satisfy the DMP and the approximate solutions are usually polluted by spurious oscillations. To enforce the DMP, one can add a sufficient amount of artificial diffusion to (5). A possible way will be described in the following section.

3 Algebraic Flux Correction

In this section we present an example of algebraic stabilization based on algebraic flux correction (AFC), as presented, e.g., in [6, 7]. A detailed derivation of an AFC scheme for problem (5)–(6) can be found, e.g., in [2]. First, one defines a symmetric artificial diffusion matrix $\mathbb{D} = (d_{ij})_{i,j=1}^N$ possessing the entries

$$d_{ij} = d_{ji} = -\max\{a_{ij}, 0, a_{ji}\} \qquad \forall\, i \neq j\,, \qquad\qquad d_{ii} = -\sum_{j\neq i} d_{ij}\,.$$

Then, the idea is to add the term $(\mathbb{D}\,\mathrm{U})_i$ to both sides of (5) and, on the right-hand side, to use the identity

$$(\mathbb{D}\,\mathrm{U})_i = \sum_{j=1}^N f_{ij} \qquad \text{with} \qquad f_{ij} = d_{ij}\,(u_j - u_i)$$

and to limit those anti-diffusive fluxes f_{ij} that would otherwise cause spurious oscillations. The limiting is achieved by multiplying the fluxes by solution dependent limiters $\alpha_{ij} \in [0, 1]$. This leads to the nonlinear algebraic problem

$$\sum_{j=1}^N a_{ij}\,u_j + \sum_{j=1}^N (1 - \alpha_{ij}(\mathrm{U}))\,d_{ij}\,(u_j - u_i) = g_i\,, \qquad i = 1, \dots, M\,, \tag{7}$$

$$u_i = u_i^b\,, \qquad i = M + 1, \dots, N\,. \tag{8}$$

It is assumed that

$$\alpha_{ij} = \alpha_{ji}\,, \qquad i, j = 1, \dots, N\,, \tag{9}$$

and that, for any $i, j \in \{1, \dots, N\}$, the function $\alpha_{ij}(\mathrm{U})(u_j - u_i)$ is a continuous function of $\mathrm{U} \in \mathbb{R}^N$. A theoretical analysis of the AFC scheme (7)–(8) concerning the solvability, discrete maximum principle and error estimation can be found in [2].

The symmetry condition (9) is particularly important since it guarantees conservativity and implies that the matrix corresponding to the term arising from the AFC is positive semidefinite. Moreover, it was demonstrated in [1] that, without the symmetry condition (9), the nonlinear algebraic problem (7)–(8) is not solvable in general.

Of course, the properties of the AFC scheme (7)–(8) significantly depend on the choice of the limiters α_{ij}. Here we present the popular Kuzmin limiter [5]. To define it, one first computes, for $i = 1, \dots, M$,

$$P_i^+ = \sum_{\substack{j=1\\ a_{ji}\,\le\, a_{ij}}}^N f_{ij}^+\,, \qquad P_i^- = \sum_{\substack{j=1\\ a_{ji}\,\le\, a_{ij}}}^N f_{ij}^-\,, \qquad Q_i^+ = -\sum_{j=1}^N f_{ij}^-\,, \qquad Q_i^- = -\sum_{j=1}^N f_{ij}^+\,, \tag{10}$$

where $f_{ij} = d_{ij}\,(u_j - u_i)$, $f_{ij}^+ = \max\{0, f_{ij}\}$, and $f_{ij}^- = \min\{0, f_{ij}\}$. Then, one defines

$$R_i^+ = \min\left\{1, \frac{Q_i^+}{P_i^+}\right\}, \quad R_i^- = \min\left\{1, \frac{Q_i^-}{P_i^-}\right\}, \qquad i = 1, \dots, M. \tag{11}$$

If P_i^+ or P_i^- vanishes, one sets $R_i^+ = 1$ or $R_i^- = 1$, respectively. For $i = M + 1, \dots, N$, one defines $R_i^+ = R_i^- = 1$. Furthermore, one sets

$$\widetilde{\alpha}_{ij} = \begin{cases} R_i^+ \text{ if } f_{ij} > 0, \\ \;1 \text{ if } f_{ij} = 0, \\ R_i^- \text{ if } f_{ij} < 0, \end{cases} \qquad i, j = 1, \dots, N. \tag{12}$$

Finally, one defines

$$\alpha_{ij} = \alpha_{ji} = \widetilde{\alpha}_{ij} \qquad \text{if} \quad a_{ji} \le a_{ij}, \qquad i, j = 1, \dots, N. \tag{13}$$

It was proved in [4] that the AFC scheme (7)–(8) with the above limiter satisfies a local DMP provided that

$$\min\{a_{ij}, a_{ji}\} \le 0 \qquad \forall\, i = 1, \dots, M, \;\; j = 1, \dots, N, \;\; i \neq j. \tag{14}$$

If a_h is given by (4), then, as discussed in [2], the validity of (14) is guaranteed if the triangulation $\mathscr{T}_h$ is weakly acute. In the two-dimensional case, it is sufficient if $\mathscr{T}_h$ is a Delaunay triangulation. However, for non-Delaunay triangulations, the validity of (14) cannot be guaranteed. Moreover, if the lumped bilinear form (4) is replaced by the original bilinear form (2), then the validity of the condition (14) may be lost also for Delaunay triangulations since some off-diagonal entries of the matrix corresponding to the reaction term from (2) are positive.

It was shown in [4] that the DMP generally does not hold if the condition (14) is not satisfied. This is due to the condition $a_{ji} \le a_{ij}$ used in (13) to symmetrize the factors $\widetilde{\alpha}_{ij}$. As discussed in [4], in many cases (depending on $\boldsymbol{b}$ and the geometry of the triangulation), this condition causes that $\alpha_{ij} = \alpha_{ji}$ is defined using quantities computed at the upwind vertex of the edge with end points x_i, x_j. It turns out that this feature has a positive influence on the quality of the approximate solutions and on the convergence of the iterative process for solving the nonlinear problem (7)–(8).

4 A New Algebraically Stabilized Method

As we discussed in the preceding section, the symmetrization (13) of the limiter causes that the DMP does not hold for the AFC scheme (7)–(8) in general. In this section we modify the AFC scheme in such a way that the symmetry of the limiter will not be needed and the DMP will be always satisfied.

Obviously, the AFC scheme (7)–(8) can be written in the form

$$\sum_{j=1}^{N} a_{ij}\, u_j + \sum_{j=1}^{N} b_{ij}(\mathrm{U})\,(u_j - u_i) = g_i\,, \qquad i = 1,\dots,M\,, \tag{15}$$

$$u_i = u_i^b\,, \qquad i = M+1,\dots,N\,, \tag{16}$$

with

$$b_{ij}(\mathrm{U}) = -\max\{(1-\alpha_{ij}(\mathrm{U}))\,a_{ij}, 0, (1-\alpha_{ji}(\mathrm{U}))\,a_{ji}\}\,, \tag{17}$$

where we employed the symmetry condition (9). In the preceding section, we discussed the important consequences of this symmetry condition. However, these consequences remain valid for the scheme (15)–(16) with any symmetric matrix $(b_{ij}(\mathrm{U}))_{i,j=1}^{N}$ with nonpositive off-diagonal entries. Now, it is easy to see that the entries $b_{ij}(\mathrm{U})$ defined in (17) are symmetric also if the limiters α_{ij} are not symmetric. This enables us to get rid of the symmetry condition (9) and to consider the problem (15)–(17) with any functions α_{ij} satisfying, for any $i, j \in \{1,\dots,N\}$,

$$\alpha_{ij} : \mathbb{R}^N \to [0,1]\,, \tag{18}$$

$$\text{if } a_{ij} > 0, \text{ then } \alpha_{ij}(\mathrm{U})(u_j - u_i) \text{ is a continuous function of } \mathrm{U} \in \mathbb{R}^N\,. \tag{19}$$

Then it follows like in [2] that the nonlinear algebraic problem (15)–(17) has a solution. Of course, if the functions α_{ij} form a symmetric matrix, then the AFC scheme (7)–(8) is recovered.

If the condition (14) is satisfied, then

$$b_{ij}(\mathrm{U}) = \begin{cases} (1-\alpha_{ij}(\mathrm{U}))\,d_{ij} & \text{if } a_{ji} \le a_{ij}\,, \\ (1-\alpha_{ji}(\mathrm{U}))\,d_{ij} & \text{otherwise}\,. \end{cases}$$

Thus, if (14) holds, then the definition (17) implicitly comprises the favourable upwind feature mentioned in the preceding section and the method (15)–(16) can be written in the form of the AFC scheme (7)–(8). Moreover, if one sets

$$b_{ij}(\mathrm{U}) = -\max\{(1-\widetilde{\alpha}_{ij}(\mathrm{U}))\,a_{ij}, 0, (1-\widetilde{\alpha}_{ji}(\mathrm{U}))\,a_{ji}\}\,, \tag{20}$$

then one obtains the AFC scheme (7)–(8) with limiters α_{ij} defined by (13).

Now, we would like to use the scheme (15)–(16) with $b_{ij}(\mathrm{U})$ defined by (20) also if (14) is not satisfied. However, then $P_i^{\pm}$ may vanish and one can show that, independently of how $R_i^{\pm}$ are defined in these cases, the continuity assumption (19) is not satisfied in general. Therefore, we replace the definition of $P_i^{\pm}$ by

$$P_i^+ = \sum_{\substack{j=1\\ a_{ij}>0}}^{N} a_{ij}\,(u_i - u_j)^+\,, \qquad P_i^- = \sum_{\substack{j=1\\ a_{ij}>0}}^{N} a_{ij}\,(u_i - u_j)^-\,. \tag{21}$$

If (14) holds, then these formulas give the same values as (10).

Thus, the algebraically stabilized method (ASM) introduced in this paper is given by (15)–(16) with $b_{ij}(\mathrm{U})$ defined by (20), where $\widetilde{\alpha}_{ij}$ are given as in Sect. 3, but with $P_i^{\pm}$ defined by (21). It follows from [2], that the continuity assumption (19) is satisfied and hence the ASM is solvable.

It is clear that, if the condition (14) holds, then the ASM is equivalent to the AFC scheme (7)–(8) with limiters α_{ij} defined by (10)–(13). Therefore, the new method preserves the advantages of the AFC scheme from the preceding section which are available under the condition (14). However, in contrast to the method from the preceding section, the new method satisfies the DMP also if the condition (14) is not satisfied, which can be verified using the techniques of [4] and [3]. Consequently, the ASM satisfies the DMP for both definitions (2) and (4) of the bilinear form and for any triangulation $\mathscr{T}_h$.

5 Numerical Results

Since the AFC scheme from Sect. 3 and the ASM from Sect. 4 are equivalent under the condition (14), we shall present numerical results only for cases when (14) is not satisfied. To this end, we shall consider the following two examples.

Example 1 We consider problem (1) defined in $\Omega = (0,1)^2$ with the data $\varepsilon = 10^{-8}$, $\boldsymbol{b} = (\cos(-\pi/3), \sin(-\pi/3))^T$, $c = 100$, $g = 1$, and the boundary condition

$$u_b(x,y) = \begin{cases} 0 & \text{for } x = 1 \text{ or } y = 0,\\ 1 & \text{else.} \end{cases}$$

Example 2 We consider problem (1) defined in $\Omega = (0,1)^2$ with the data $\varepsilon = 10^{-2}$, $\boldsymbol{b} = (\cos(-\pi/3), \sin(-\pi/3))^T$, $c = 0$, $g = 0$, and the boundary condition

$$u_b(x,y) = \begin{cases} 0 & \text{for } x = 1 \text{ or } y \le 0.7,\\ 1 & \text{else.} \end{cases}$$

Example 1 was computed on a triangulation of the type from Fig. 1 (left) and for $a_h = a$. One observes in Fig. 2 that the solution of the AFC scheme contains undershoots violating the DMP whereas the ASM provides the nodally exact solution. For a_h defined by (4), the condition (14) is satisfied so that also the AFC scheme satisfies the DMP, however, the layers are then smeared.

To violate the condition (14) for Example 2, we used a non-Delaunay triangulation of the type depicted in Fig. 1 (middle) which was constructed starting from the triangulation shown in Fig. 1 (right) by shifting interior nodes to the right by half of the horizontal mesh width on each even horizontal mesh line. One observes in Fig. 3 that the solution of the AFC scheme again violates the DMP while the DMP is satisfied for the ASM.

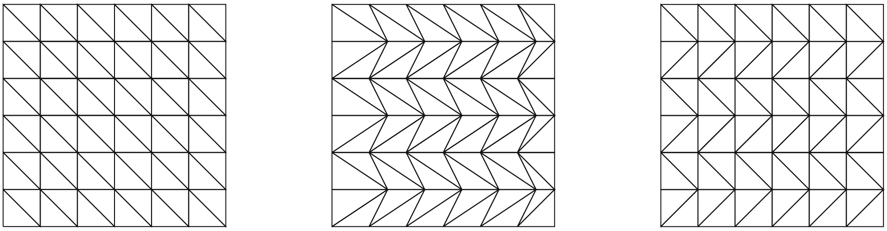


Fig. 1 Types of triangulations used in the computations

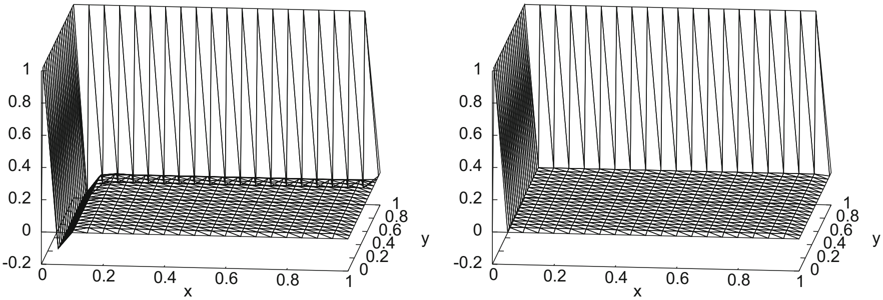


Fig. 2 Example 1: approximate solutions obtained using the AFC scheme (left) and the ASM (right)

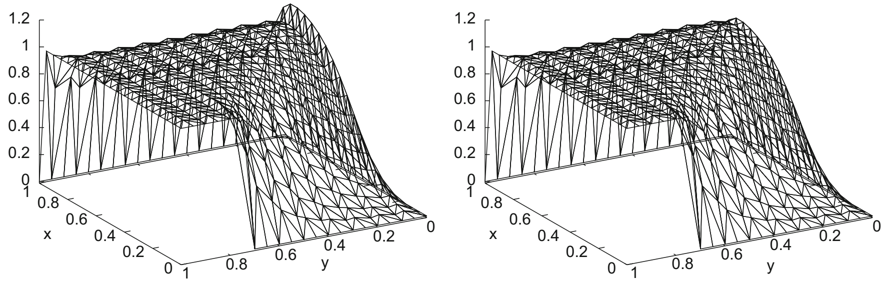


Fig. 3 Example 2: approximate solutions obtained using the AFC scheme (left) and the ASM (right)

Acknowledgement This work has been supported through the grant No. 19-04243S of the Czech Science Foundation.

References

1. Gabriel R. Barrenechea, Volker John, and Petr Knobloch. Some analytical results for an algebraic flux correction scheme for a steady convection-diffusion equation in one dimension. *IMA J. Numer. Anal.*, 35(4):1729–1756, 2015.
2. Gabriel R. Barrenechea, Volker John, and Petr Knobloch. Analysis of algebraic flux correction schemes. *SIAM J. Numer. Anal.*, 54(4):2427–2451, 2016.
3. Gabriel R. Barrenechea, Volker John, Petr Knobloch, and Richard Rankin. A unified analysis of algebraic flux correction schemes for convection-diffusion equations. *SeMA J.*, 75(4):655–685, 2018.
4. Petr Knobloch. On the discrete maximum principle for algebraic flux correction schemes with limiters of upwind type. In Z. Huang, M. Stynes, and Z. Zhang, editors, *Boundary and Interior Layers, Computational and Asymptotic Methods BAIL 2016*, volume 120 of *Lecture Notes in Computational Science and Engineering*, pages 129–139. Springer, 2017.
5. Dmitri Kuzmin. Algebraic flux correction for finite element discretizations of coupled systems. In M. Papadrakakis, E. Oñate, and B. Schrefler, editors, *Proceedings of the Int. Conf. on Computational Methods for Coupled Problems in Science and Engineering*, pages 1–5. CIMNE, Barcelona, 2007.
6. Dmitri Kuzmin. Linearity-preserving flux correction and convergence acceleration for constrained Galerkin schemes. *J. Comput. Appl. Math.*, 236:2317–2337, 2012.
7. Dmitri Kuzmin and Matthias Möller. Algebraic flux correction I. Scalar conservation laws. In Dmitri Kuzmin, Rainald Löhner, and Stefan Turek, editors, *Flux-Corrected Transport. Principles, Algorithms, and Applications*, pages 155–206. Springer-Verlag, Berlin, 2005.

Analysis of Kuramoto-Sivashinsky Model of Flame/Smoldering Front by Means of Curvature Driven Flow

Miroslav Kolář, Shunsuke Kobayashi, Yasuhide Uegata, Shigetoshi Yazaki, and Michal Beneš

Abstract In this paper we summarize our results on the investigation of the Kuramoto-Sivashinsky model, which describes the motion of flame/smoldering interface. We propose the generalization of the model formulated in terms of mathematical theory of moving parametrized curves, and investigate it from numerical and analytical point of view. In the part dedicated to computational studies, we present the verification of our scheme by measurement of experimental order of convergence. In the analytical part of the paper we summarize biffurcation analysis of the model and study of rotational wave solutions.

1 Introduction

Interfacial dynamics of flame fronts is a topic of increased interest in fire research and combustion phenomena (c.f. the introduction in [1]). Also, this topic has attracted an interest in the field of applied mathematics. The first pioneering works on this topic are dated to 1970s in, e.g., [2, 3]. In this paper, we investigate the Kuramoto-Sivashinsky (KS model in short), which describes the propagation of a curved flame-front represented by a graph of a smooth function $y = f(x, t)$

M. Kolář (✉) · S. Yazaki
Graduate School of Science and Technology, Meiji University, Kanagawa, Japan
e-mail: Miroslav.Kolar@fjfi.cvut.cz; kolarmir@fjfi.cvut.cz; syazaki@meiji.ac.jp

S. Kobayashi
Graduate School of Science, Kyoto University, Kyoto, Japan
e-mail: s.kobayashi@math.kyoto-u.ac.jp

Y. Uegata
Setagaya Gakuen School, Tokyo, Japan

M. Beneš
Department of Mathematics, Faculty of Nuclear Sciences and Physical Engineering,
Czech Technical University in Prague, Prague, Czech Republic
e-mail: Michal.Benes@fjfi.cvut.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_60

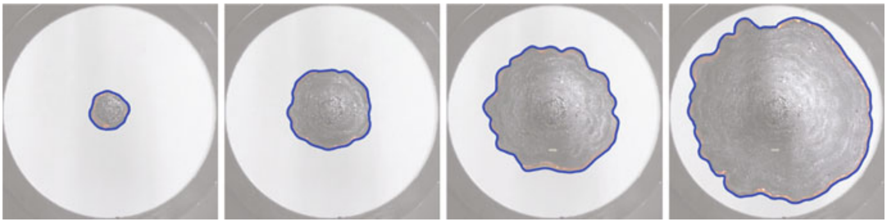


Fig. 1 Experimental images of flame/smoldering front on a paper sheet. Blue curve represents flame/smoldering interface segmented to determine the parameters of the KS model [5]

satisfying the following dynamical system

$$f_t + f_x^2/2 + (\alpha - 1)f_{xx} + 4f_{xxxx} = 0. \tag{1}$$

Here, parameter $\alpha > 0$ depends on the scaled Lewis number [1], $f_t = \partial f/\partial t$, $f_x = \partial f/\partial x$, $f_{xx} = \partial^2 f/\partial x^2$ and $f_{xxxx} = \partial^4 f/\partial x^4$. The typical profile of flame/smoldering interface is in Fig. 1. Recently, this problem was investigated in, e.g. [1, 4] from numerical and experimental point of view. In [5], the framework for validation of experimental data was established.

When certain conditions are satisfied, the KS model can be generalized as a flow of a family of smooth Jordan curves Γ_t in the plane, representing the flame/smoldering front. For details, we refer the reader to, e.g. [4, 6]. The curve Γ_t is parametrized by a smooth mapping $\mathbf{x} : [0, 1] \times [0, T_{max}] \to \mathbb{R}^2$ as

$$\Gamma_t = \{\mathbf{x}(u, t) = (x_1(u, t), x_2(u, t)) : u \in [0, 1]\}. \tag{2}$$

Then parametrization $\mathbf{x}$ for the KS model (1) satisfies the following geometric evolution equation

$$\mathbf{x}_t = V\mathbf{N} + W\mathbf{T}, \tag{3}$$

where $\mathbf{T} = \partial_u\mathbf{x}/|\partial_u\mathbf{x}|$ and $\mathbf{N} = \partial_u\mathbf{x}^\perp/|\partial_u\mathbf{x}|$ are unit tangent and normal vectors to Γ_t, respectively. Here, $(a, b)^\perp = (-b, a)$ and the orientation of the parametrization $\mathbf{x}$ is chosen to be counterclockwise. The normal component of the velocity is given as

$$V = V_0 + (\alpha - 1)\kappa + \delta\kappa_{ss}. \tag{4}$$

Flow (3) with the normal velocity (4) corresponds to the KS model (1) by the choice of parameter δ as $\delta = 4$ (see [4]). Next, V_0 is a prescribed constant speed and κ denotes the curvature of Γ_t. Parameters V_0 and α can be determined from experimental data, e.g., by means of the image segmentation of the flame/smoldering front—see [5].

Here and hereafter, we use the subscript s to denote the derivative with respect to the arc-length, i.e., $F_s = g^{-1}\partial_u F$ for a quantity F, where $g = |\partial_u \mathbf{x}|$ is the relative local length. It is well known that the curve Γ_t is determined by the normal velocity V and the tangential velocity W does not affect its shape. However, the particular choices of W can provide some desirable properties. For example, the tangential velocity proposed by Ševčovič and Yazaki [7] or Beneš et al. [8] helps to control the position of discretization points along Γ_t, which is especially helpful in long-term numerical computations. In this paper, we propose the following form of tangential velocity

$$W = 3\delta(\kappa^2)_s \big/ 2, \tag{5}$$

which is particularly helpful in a bifurcation analysis of the KS model in Sect. 3.

The organization of the rest of the paper is the following. In Sect. 2, we demonstrate the qualitative and quantitative behavior of the numerical solution of the KS model. The numerical approximation scheme for flow (3) with the normal velocity (4) was proposed in [4]. We verify the scheme by means of the measurement of experimental orders of convergence and apply it to the case of rotational solution. In Sect. 3, we present our latest results on theoretical analysis of rotating wave solutions of KS model for the case of the flame/smoldering front in the shape of the initial expanding circle.

2 Computational Studies of the KS Model

In this section, we demonstrate the qualitative and quantitative behavior of a closed curve expanding according to the geometric evolution equation (3) with the normal velocity given by the KS model (4). The details of spatial discretization of the evolution equation are elaborated in [4]. For the time integration, the fourth-order Runge-Kutta method with the time step $\Delta t = 2^{-18} \approx 10^{-6}$ was employed.

As the initial condition for the KS model, we consider the following Cassini curve

$$\begin{aligned} R &= R_C\big[C_C\cos(4\pi u) + [(C_C\cos(4\pi u))^2 + D_C]^{\frac{1}{2}}\big]^{\frac{1}{2}}, \\ \mathbf{x} &= R(\cos(2\pi u), \sin(2\pi u)), \end{aligned} \tag{6}$$

where $u \in [0, 1]$ and $R_C = 10$, $D_C = 4$ and $C_C = 10$. We set the final time for our simulation as $T = 250$. In Fig. 2 the numerical solution of (3) with the normal velocity (4) is shown. The quantitative analysis of numerical solution of the flow (3) is given by the measurement of experimental order of convergence (EOC). The EOCs were measured by means of the Hausdorff distances H_k between the numerical solution and the solution calculated on extra-fine mesh at the $k-$th time level ($k = 1, 2, \ldots, N$) with the time step Δt. For a mesh with M discretization

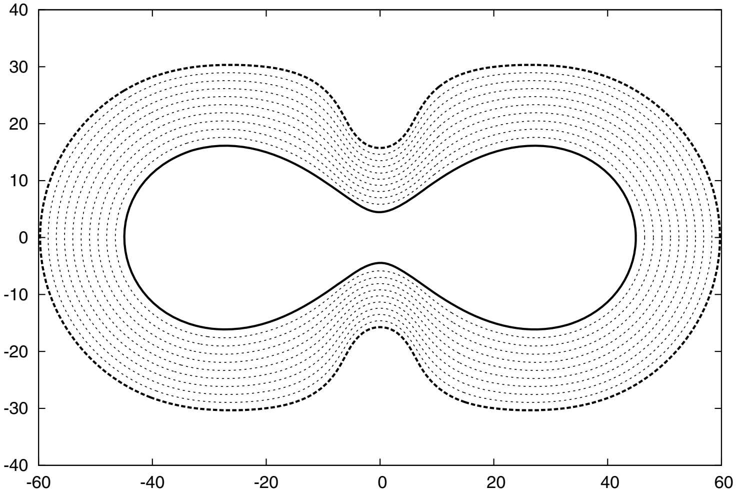


Fig. 2 Time evolution of initial Cassini curve (solid line) driven by (3) with normal velocity (4) and tangential velocity calculated according to [7, 8] in time $t \in (0, 250)$. The corresponding EOCs are shown in Table 1

points, the maximum norm $err_{\max}(M) = \max_{k=1,2,\ldots,N} H_k$ and the discrete L_1 norm $err_{L_1}(M) = \Delta t \sum_{k=1}^{N} H_k/T$ were used. Then, the experimental order of convergence between two meshes with M_1 and M_2 discretization points was estimated as

$$\mathrm{EOC} = [\log err_I(M_1) - \log err_I(M_2)] \big/ [\log M_2 - \log M_1], \quad I = \max, L_1.$$

In Table 1 we summarize the values of EOC for the computational experiment with the Cassini curve (6) as the initial condition. The values of calculated EOCs suggest that the numerical approximation scheme proposed in [4] has about the first order of convergence, which is caused by the discretization procedure which assumes the uniform redistribution of discretization points along the curve. In the case of nontrivial shape, such as the Cassini curve, this assumption can be fulfilled only approximately by evolving the initial curve with zero normal velocity and asymptotically uniform tangential velocity chosen according to [7, 8] for a short time.

Table 1 EOCs for errors measured in discrete L_1 and maximum norms for the KS model depicted in Fig. 2 with Cassini curve as the initial condition

M	EOC (L_1)	EOC (max)
64	–	–
128	1.0045	1.0251
256	1.0017	1.0013
512	1.0005	1.0025
1024	1.0001	1.0002

3 Rotating Wave Solution

In this section we present our latest result on theoretical analysis of the KS model. We present the study of the KS model by means of the bifurcation theory and show the existence of a rotating wave solution bifurcating from an expanding circle.

Rewriting the curvature κ in terms of parametrization $\mathbf{x}$ by means of the Frenet formulae and substituting (4) to (3), we have

$$\mathbf{x}_t = -\delta \mathbf{x}_{ssss} - ((\alpha - 1) + \delta\kappa^2)\mathbf{x}_{ss} - (3\delta(\kappa^2)_s/2 - W)\mathbf{x}_s + V_0\mathbf{N}. \tag{7}$$

We chose the tangential velocity as (5) and consider a perturbation $\varepsilon(u, t)$ from an expanding circle solution, such that

$$\mathbf{x}(u, t) = \boldsymbol{C}(u, t) + \varepsilon(u, t)\mathbf{y}(u), \tag{8}$$

where $\mathbf{y}(u) = (\cos(2\pi u), \sin(2\pi u))$, $\varepsilon \in \mathbb{R}$ is a periodic function w.r. to $u \in [0, 1]$, $\boldsymbol{C}(u, t) = R(t)\mathbf{y}(u)$ is the expanding circle solution, and $R(t)$ is the solution of $\dot{R} = V_0 + (\alpha - 1)R^{-1}$. Here and hereafter, we denote $\dot{\mathsf{F}} = d\mathsf{F}/dt$. By substituting (8) to (7), we derive an evolution equation for ε as the following

$$\begin{aligned}\varepsilon_t = &-\frac{\delta}{16\pi^4 R^4}\varepsilon_{uuuu} - \frac{\delta + R^2(\alpha - 1)}{4\pi^2 R^4}\varepsilon_{uu} - \frac{\alpha - 1}{R^2}\varepsilon + \frac{\alpha - 1}{R^3}\varepsilon^2 - \frac{V_0}{8\pi^2 R^2}\varepsilon_u^2 \\ &+ \frac{\varepsilon\varepsilon_{uu}}{\pi^2 R^3}\left(\frac{\alpha - 1}{2} + \frac{\delta}{R^2}\right) + \frac{3\delta}{16\pi^4 R^5}\varepsilon_{uu}^2 + \frac{\delta}{4\pi^4 R^5}\varepsilon\varepsilon_{uuuu} + \frac{3\delta}{16\pi^4 R^5}\varepsilon_u\varepsilon_{uuu} + \mathcal{O}_3,\end{aligned} \tag{9}$$

where $\mathcal{O}_3 = \mathcal{O}(|(\varepsilon, \varepsilon_u, \varepsilon_{uu}, \varepsilon_{uuu}, \varepsilon_{uuuu})|^3)$.

Since $R(t)$ is the known function of t, the dynamical system (9) is non-autonomous. In order to apply the standard bifurcation theory, $R(t)$ is regarded as the bifurcation parameter R, and by omitting $\mathcal{O}_3$ in (9), the truncated dynamical system is defined $\mathcal{H}^4_{\mathrm{per}} = \left\{\varepsilon \in \mathcal{H}^4_{\mathrm{loc}};\ \varepsilon(u) = \varepsilon(u + 1)\right\}$. Substituting the Fourier expansion $\varepsilon(u, t) = \sum_{m\in\mathbb{Z}} \varepsilon_m(t) \mathrm{e}^{2im\pi u}$ into (9) and omitting $\mathcal{O}_3$, we arrive at the following infinite dimensional dynamical system:

$$\dot{\varepsilon}_m(t) = \lambda_m \varepsilon_m(t) + \sum_{m_1 + m_2 = m} C_{m_1, m_2}\varepsilon_{m_1}(t)\varepsilon_{m_2}(t), \tag{10}$$

where $\lambda_m = (m^2 - 1)(R^2(\alpha - 1) - \delta m^2)/R^4$ and

$$C_{m_1, m_2} = \frac{\alpha - 1}{R^3} + \frac{V_0 m_1 m_2}{2R^2} - \frac{4m_2^2}{R^3}\left(\frac{\alpha - 1}{2} + \frac{\delta}{R^2}\right) + \frac{\delta m_2^2(3m_1^2 + 3m_1 m_2 + 4m_2^2)}{R^5}$$

in the phase space $\mathcal{F} = \left\{\{\varepsilon_m\}_{m\in\mathbb{Z}};\ \varepsilon_{-m} = \bar{\varepsilon}_m,\ \|\{\varepsilon_m\}_{m\in\mathbb{Z}}\|_{\mathcal{F}}^2 < \infty\right\}$ with the norm $\|\{\varepsilon_m\}_{m\in\mathbb{Z}}\|_{\mathcal{F}}^2 = \sum_{m\in\mathbb{Z}}(1+m^2)^4|\varepsilon_m|^2$. We note that $\varepsilon_{-m} = \bar{\varepsilon}_m$ always holds. Although ε_m is a complex number, one can verify $\varepsilon_{-m} = \bar{\varepsilon}_m$ since ε is a real number. Also it is well known that the linearized operator becomes a generator for the analytical semi-group (see Sec. 2 in [9]).

To find the primary bifurcation point of the circle solution, it is convenient to introduce the neutral stability curves defined from $\lambda_m = 0$.

Definition 1 The neutral stability curves are defined as a set of parameters $\{(\delta, R);\ \delta = (\alpha-1)R^2/m^2,\ m \in \mathbb{Z}\}$ on which the linearized operator of (10) has a simple zero eigenvalue.

Note that the eigenvalues $\lambda_{\pm 1} \equiv 0$ hold for any $R > 0$. On the other hand, $\lambda_m < 0$ holds for $|m| = 2, 3, \cdots$ and $R < R^* = 2\sqrt{\delta/(\alpha-1)}$. Therefore, the circle solution is neutral stable as in the gray region in Fig. 3. In particular, when $|m| = 2$, for any fixed $\delta > 0$ the value R^* is the minimum value on which the stability of the circle solution changes from neutral stable to unstable.

Around the bifurcation parameter value $R = R^*$, we apply the central manifold theory to the dynamical system (10). Let $\mu = R - R^*$. Then the ±2-mode eigenvalues are expanded as $\lambda_{\pm 2} = 12\delta(2 - R^*)/(R^*)^5\mu + \mathcal{O}(\mu^2)$. For a small $\rho > 0$, we define a neighborhood $\mathcal{N}_\rho$ in $\mathcal{F} \times \mathbb{R}$ such that $\mathcal{N}_\rho = \{(\varepsilon_m, \mu) \in \mathcal{F} \times \mathbb{R};\ \|\{\varepsilon_m\}_{m\in\mathbb{Z}}\|_{\mathcal{F}} + |\mu| < \rho\}$. By the center manifold reduction, we obtain the following reduced system.

Proposition 1 *For $\alpha > 1$ and $\delta > 0$ there exists a positive constant ρ such that the local center manifold $\mathcal{M}^{\mathrm{c}}_{\mathrm{loc}}$ of (10) is contained in $\mathcal{N}_\rho$. Furthermore, the dynamics of (10) on $\mathcal{M}^{\mathrm{c}}_{\mathrm{loc}}$ is given by the following system:*

$$\begin{cases} \dot{\varepsilon}_1 = a_1\bar{\varepsilon}_1\varepsilon_2 + a_2\varepsilon_1|\varepsilon_2|^2 + \mathcal{O}_4, \\ \dot{\varepsilon}_2 = \lambda_2\varepsilon_2 + b_1\varepsilon_1^2 + (b_2|\varepsilon_1|^2 + b_3|\varepsilon_2|^2)\varepsilon_2 + \mathcal{O}_4. \end{cases} \tag{11}$$

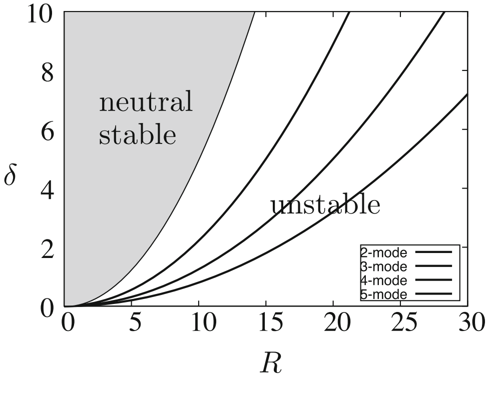


Fig. 3 The $|m|$-mode neutral stability curves when $\alpha = 1.2$ with $|m| = 2, 3, 4, 5$

Here we put $c_{i,j} = C_{i,j} + C_{j,i}$, *and* $a_1 = c_{2,-1}$, $a_2 = -c_{1,2}c_{3,-2}/\lambda_3$, $b_1 = C_{1,1}$, $b_2 = -c_{2,0}c_{1,-1}/\lambda_0 - c_{1,2}c_{3,-1}/\lambda_3$, $b_3 = -c_{2,0}c_{2,-2}/\lambda_0 - C_{2,2}c_{4,-2}/\lambda_4$ *and* $O_4 = O(|(\varepsilon_{\pm 1}, \varepsilon_{\pm 2})|^4)$.

Remark 1 To compute the third-order terms of (11) rigorously, we must calculate (9) up to the third-order terms. However, the form of the reduced system are the same as (11) due to the normal form theory. Moreover, we will see that the necessary condition for the existence of rotating wave solution is independent of the third-order terms.

Set $\varepsilon_j = r_j(t)\mathrm{e}^{i\theta_j(t)}$ for $j = 1, 2$ and $\phi = 2\theta_1 - \theta_2$. Then the third-order truncated system of (11) is

$$\begin{cases} \dot{r}_1 = r_1 r_2(a_1 \cos\phi + a_2 r_2), \\ \dot{r}_2 = \lambda_2 r_2 + b_1 r_1^2 \cos\phi + r_2(b_2 r_1^2 + b_3 r_2^2), \\ \dot{\phi} = -(2a_1 r_2 + b_1 r_1^2/r_2)\sin\phi. \end{cases} \tag{12}$$

In this paper, we call the *rotating wave* solution a non-trivial stationary solution of (12) under $\sin\phi \neq 0$. Hence the equilibrium which corresponds to the rotating wave solution is as follows

$$r_1 = \sqrt{-\frac{2a_1}{b_1}} r_2, \quad r_2 = \sqrt{\frac{\lambda_2 b_1}{2a_1 b_2 - b_1(2a_2 + b_3)}}, \quad \cos\phi = -\frac{a_2}{a_1} r_2 \tag{13}$$

in the phase space $\{(r_1, r_2, \phi)\}$. Therefore, the necessary condition for the existence of rotating wave is $a_1 b_1 < 0$, which holds for arbitrary $V_0 > 0$, $\alpha > 1$ and $\delta > 0$. The rotating wave of (11) means that the phase difference ϕ remains constant, but θ_1 and θ_2 both increase or both decrease linearly w.r. to the time, and this is the reason of the rotating. The rotating wave solution is approximately inherent in the solution structure of the dynamical system (9) without O_3, and the leading terms of it are

$$\varepsilon(u, t) \sim r_1 \mathrm{e}^{i(\theta_1(t) + 2\pi u)} + r_2 \mathrm{e}^{i(\theta_2(t) + 4\pi u)} + \text{c.c.},$$

where "c.c." denotes complex conjugate for the first two terms. Then, the linearized matrix at the equilibrium point (13) is given by

$$\begin{pmatrix} 0 & p a_2 r_2^2 & -p a_1 r_2^2 \sin\phi \\ 2 p r_2^2 \left(b_2 - \dfrac{a_2 b_1}{a_1} \right) & 2(b_3 - a_2) r_2^2 & 2 a_1 r_2^2 \sin\phi \\ -2 b_1 p \sin\phi & -4 a_1 \sin\phi & 0 \end{pmatrix}, \tag{14}$$

where $p = \sqrt{-2a_1/b_1}$. Thus, we conclude that the rotating wave equilibrium is locally asymptotically stable provided that all eigenvalues of this matrix have negative real part.

In the case where $V_0 = 1.8$, $\alpha = 1.2$, $\delta = 4.0$, we have the bifurcation point $R^* = 4\sqrt{5}$, and the coefficients of (11) are $a_1 = (\sqrt{5}-144)/3200$, $a_2 = 3(288\sqrt{5}-102{,}385)/1{,}280{,}000$, $b_1 = (\sqrt{5}+180)/16{,}000$, $b_2 = -3(108\sqrt{5}+5095)/128{,}000$ and $b_3 = -(16{,}776\sqrt{5} + 47{,}735)/720{,}000$. In Fig. 4 we show the numerical solution of (12) with the above parameters and $\lambda_2 = 10^{-5}$. Since the rotating wave solution is unstable in this case, the orbit escapes from the rotating wave equilibrium point and the numerical experiment suggests its trajectory converges to a heteroclinic cycle which connects two equilibria $(0, r_2^*, 0)$ and $(0, r_2^*, \pi)$, where $r_2^* = \sqrt{-\lambda_2/b_3}$.

Figure 5 shows the time evolution of the perturbed curves $\tilde{\mathbf{x}}(u,t) = (R^*/500 + \varepsilon(u,t))\mathbf{y}(u)$ by using the numerical data given by (12) (see also Fig. 4). To visualize the form of rotating solution we formally set $R^*/500$. Here $\tilde{\mathbf{x}}$ is an approximation of $\mathbf{x}$ in (8) in the sense that $R(t)$ is replaced by $R^* := 4\sqrt{5}$. In the snapshots, the circles are of the radius $0.018 \approx R^*/500$. The color-change of the perturbed curves visualize the perturbation ε (the origin is represented by "+" and the parts far from the origin is colored brightly, whereas the parts close to the origin are

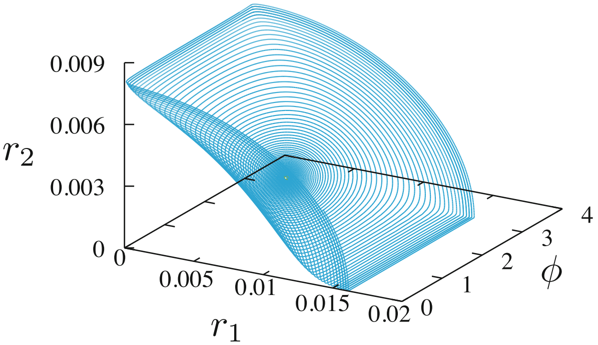


Fig. 4 Orbit of the numerical solution of (12). The initial value is set near the rotating wave equilibrium (13)

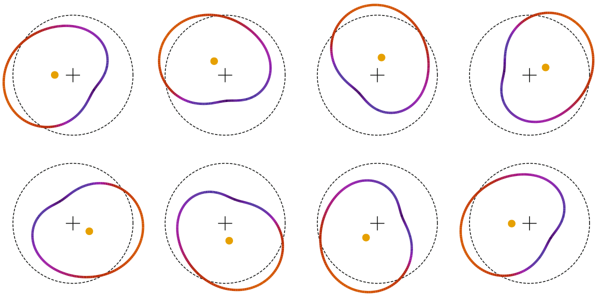


Fig. 5 The numerical approximation of rotating wave solution of (8) obtained by the orbit in Fig. 4. Time snapshots are depicted for $t \in [20{,}500, 76{,}500]$, which corresponds to the yellow region in Fig. 6

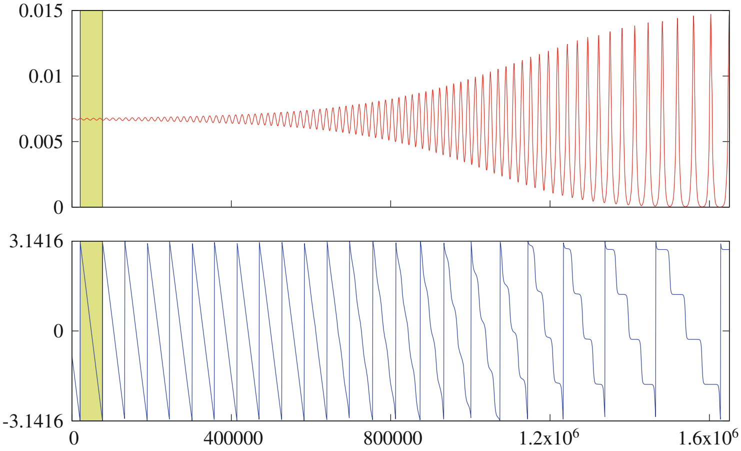


Fig. 6 The time sequence of gravity point (yellow points in Fig. 5) of the solution curves $(R^*/500 + \varepsilon(u,t))\mathbf{y}(u)$. The upper figure represents its amplitude, and the lower one represents its argument (angle)

colored darkly). The yellow dots represent the gravity points of perturbed curves and the time evolution of the gravity points is shown in Fig. 6. This phenomena suggests that not only normal component but also rotating component are inherent in a curved propagating front.

4 Conclusions

In this paper we summarized our recent results on investigation of the Kuramoto-Sivashinsky model for evolution of flame/smoldering front. We introduced the formulation of the problem by parametrically described moving curves with nontrivial tangential velocity, and verified our numerical approximation scheme originally proposed in [4] by measurement of experimental orders of convergence. Our results suggest that even for nontrivial initial conditions, our scheme is of about first order. The last section gives mathematical results on quantitative properties of solutions. There, we derived the perturbation equation (9) from a circle solution, and obtained the reduced system (11). Furthermore, analysis of (11) revealed that rotating wave solution bifurcates from the circle solution. In fact, modulated traveling waves and heteroclinic cycles also had been found in (1) by Armbruster et al. [10, 11]. However, we cannot obtain them since (11) has codimension one bifurcation.

Acknowledgments M. Kolář was supported by the International Research Fellowship of Japan Society for Promotion of Science. S. Yazaki was supported by KAKENHI No.19H01807. M. Beneš was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the OPRDE grant number CZ.02.1.01/0.0 /0.0/16_019/0000753 "Research centre for low-carbon energy technologies".

References

1. Goto M., Kuwana K., Kushida G., Yazaki S.: Experimental and theoretical study on near-floor flame spread along a thin solid. P. Combust. Inst. **37**, 3783–3791 (2019)
2. Kuramoto Y., Tsuzuki T.: Persistent propagation of concentration waves in dissipative media far from thermal equilibrium, Prog. Theor. Phys. **55**, 356–369 (1976)
3. Sivashinsky G. I.: Nonlinear analysis of hydrodynamic instability in laminar flames–I. Derivation of basic equations, Acta Astronaut. **4**, 1177–1206 (1977)
4. Goto M., Kuwana K., Yazaki S.: A simple and fast numerical method for solving flame/smoldering evolution equations. JSIAM Lett. **10**, 49–52 (2018)
5. Goto M., Kuwana K., Uegata Y., Yazaki S.: A method how to determina parameters arising in a smoldering evolution equation by image segmentation for experiments's movies. To appear in DCDS-S (2019)
6. Frankel M. L., Sivashinsky G. I.: On the nonlinear thermaldiffusive theory of curved flames, J. Phys. France **48** 25–28 (1987)
7. Ševčovič D., Yazaki S.: Computational and qualitative aspects of motion of plane curves with a curvature adjusted tangential velocity, Math. Method. in Appl. Sci. **35** 1784–1798 (2012)
8. Beneš M., Kratochvíl J., Křišťan J., Minárik V., Pauš P.: A parametric simulation method for discrete dislocation dynamics, Eur. Phys. J-Spec. Top. **177**, 177–192 (2009)
9. Brauner C. M., Frankel M., Hulshof J., Sivashinsky G. I.: Weakly nonlinear asymptotics of the κ-θ model of cellular flames: the Q-S equation, Interfaces Free Bound., **7**, 131–146, (2005)
10. Armbruster D., Guckenheimer J., Holmes P.: Heteroclinic cycles and modulated travelling waves in systems with O(2) symmetry Physica D **29** 257–282 (1988)
11. Armbruster D., Guckenheimer J., Holmes P.: Kuramoto-Sivashinsky dynamics on the center-unstable manifold, SIAM J. Appl. Math. **49** 676–691 (1988)

The Master-Slave Splitting Extended to Power Flow Problems on Integrated Networks with an Unbalanced Distribution Network

M. E. Kootte and C. Vuik

Abstract An integrated network consists of a transmission network and at least one distribution network which are connected to each other via a substation. One way to do power flow simulations on these integrated networks is the Master-Slave splitting method. This method splits the integrated network and iterates between the separate transmission (the master) and distribution (the slave) network. In this paper, we extend the method to hybrid networks: a network consisting of a balanced transmission and an unbalanced distribution network. An extra handling is necessary to get the Master-slave splitting to work on hybrid networks. We explain two approaches to use the Master-Slave splitting on a hybrid network and compare these approaches on accuracy, computational time, and convergence, by doing test-simulations. The Master-Slave splitting is interesting when distribution and transmission systems have different characteristics, are in geographically distinct locations, or when system operators are not able or allowed to share data of their network with each other. The extension to hybrid networks makes this method generally applicable and an interesting choice to do power flow simulations on integrated networks.

1 Introduction

System operators (SO's) use power flow simulations for safe operation and planning of the electricity grid. In general, a country has one high-voltage transmission network and several medium/low-voltage distribution networks and each SO studies its network separately. The electricity system is changing because of the increasing demand of electricity, the supply of renewable resources, and distributed generation. These changes lead to network interactions that need to be studied with power flow simulations that run on integrated transmission-distribution networks. The different

M. E. Kootte (✉) · C. Vuik
Delft Institute of Applied Mathematics, Delft, TU, Netherlands
e-mail: m.e.kootte@tudelft.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_61

characteristics of the transmission and distribution network require different power flow models and makes integration difficult.

Researchers are paying more attention to integrating power flow models. One of the presented ideas is to unify the power flow models and solve them as a whole [1]. This method has several disadvantages: system operators are not always allowed to share data of their complete network with each other, transmission and distribution systems have different characteristics and require own power flow solvers [2], and the systems are modeled in different units.

Another way to do integrated simulations that overcomes these disadvantages is the Master-Slave splitting (MSS) method [3]. The MSS-method is an iterative method, in which the solution is based on convergence of the voltage mismatch on the boundary between the master (the transmission network) and the slave (the distribution network). At every iteration, the two systems are solved on its own and share only information of the boundary bus with each other. This allows for using power flow algorithms that are appropriate for the specific network conditions and minimizes the data communication between systems. Previous research has shown that the MSS-method has good convergence characteristics when they are applied on a balanced integrated network. However, the distribution network is in general not balanced. Although the authors of the MSS-method describe how the MSS-method is applied on unbalanced distribution networks, they do not test if their method indeed still works.

In this paper, we extend the MSS-method to work on a balanced-unbalanced integrated network and evaluate its behavior on several test-cases. We compare its solution on accuracy and convergence with non-integrated network simulations, i.e. simulations that SO's currently use to study their network separately. In the rest of the paper we describe how we model the power flow problem for balanced and unbalanced networks (Sect. 2), we explain and extend the MSS-method (Sect. 3), we run several numerical simulations and analyze the results (Sect. 4), and draw conclusions from these results (Sect. 5).

2 Characterization of the Power Flow Problem

The steady-state power flow problem is the problem of determining the voltages V in a network, given the specified power $S = P + \iota Q$ and current I [4], ι being the imaginary number. V and I are related by Ohm's Law, $I = YV$, where Y is the admittance. S and V are related by $S = VI^*$. Because the currents are never given in an electricity system, we substitute Ohm's Law into $S = VI^*$ and get a nonlinear equation for S. Power is generated in three phases leading to three sinusoidal functions that describe phase a, b, and c of the voltage, represented by $V = |V|e^{\delta\iota}$ (a magnitude and phase-angle), and of the current. In balanced systems, the phase magnitudes ($|\cdot|$) and angles (ϕ) between two phases are equal: For a voltage V this means that $|V|_a = |V|_b = |V|_c$ and $\phi_{ab} = \phi_{bc} = \phi_{ca} = \frac{2}{3}\pi$. To simplify and speed-up the computations, we only have to model phase a and deduct the other two phases from here. In unbalanced systems, the magnitudes and angles are not equal: All the three-phases are included in the model. The nonlinear power flow equation can be described as follows:

$$S_p = V_p (YV)_p^*, \qquad \begin{cases} p \in \{a\}, & \text{balanced systems,} \\ p \in \{a, b, c\} & \text{unbalanced systems.} \end{cases} \tag{1}$$

We represent an electricity network as a graph consisting of buses $i = 1, .., N$ and branches (named after the two buses connecting them). These buses are either a PQ-bus, a PV-bus, or a slack bus, depending on the information we know at that point [5]. We solve equation (1) in an iterative manner for V. All loads in a network are modeled as PQ-buses: Power is consumed at these buses. Generators, buses where power is supplied, are modeled as PV-buses, except for the first generator bus: This is the slack bus. Each network has exactly one slack bus and can have one or multiple PQ and PV-buses. Table 1 describes the known and unknown variables of these buses. We explain in Sect. 3 how we treat the slack buses.

We do power flow simulations in per-unit (pu) quantities and not in engineering quantities. This means that the quantities are scaled by base values such that the voltage is close to unity. This has the advantage that it eliminates erroneous values by scaling them in a narrow range [4].

2.1 Integrated Networks

An integrated network consists of a transmission network and at least one distribution network. The separate transmission and distribution networks have distinct characteristics and therefore require own appropriate algorithms. Transmission networks are balanced networks and modeled in single-phase. As the MSS-method solves the two systems on its own, it allows for using a preferred algorithm for each network. We use the Newton-Raphson power mismatch (NR-power) [4] method to solve single-phase transmission networks. Distribution networks are in general unbalanced networks and must be modeled in three-phase. We use the Newton-Raphson three-phase current injection method (NR-TCIM) [6] to solve distribution networks.

Table 1 Bustypes in a network and the information we know and not know at each bus i

Bus type	Known	Unknown
PQ-bus	$P_i,\ Q_i$	$\delta_i,\ \lvert V_i\rvert$
PV-bus	$P_i,\ \lvert V_i\rvert$	$Q_i,\ \delta_i$
Slack bus	$\delta_i,\ \lvert V_i\rvert$	$P_i,\ Q_i$

3 Solving the Power Flow Problem with the Master-Slave Splitting Method

The Master-Slave splitting method [3] is an iterative method that splits the integrated network in a master, the transmission network, and a slave, the distribution network, and solves them on their own. Because the master and slave are solved on its own, they both require a slack bus. One of the load buses of the master is taken as the slack bus of the slave and this bus becomes a direct voltage source for the slave. This load bus is called the boundary B. When multiple slaves are connected to the master, the connecting load buses form the boundary-set $\mathbf{B}$. The voltage source must be equal to the loads in the slave system.

The MSS-method starts by solving the slave: The voltage source from the master is taken as the slack bus for the slave. The slack bus needs the voltage magnitude and angle as known parameter, but as the master is not yet solved, we start with an initial guess of the voltage source, i.e. V_B^0. With this information, we solve the slave with NR-TCIM. We then continue to the master: The boundary bus B is a load-bus for the master, hence we must know active and reactive power $S_B = P_B + \iota Q_B$ at this bus. From the slave, we know S_B and we inject this output into the master. The slack bus, as present in the original transmission network when it is modeled as a separated network, remains the slack bus for the master. This gives us enough information to solve the master, which we do with the NR-power method. Solving the master gives us the voltage V_B. We compare this voltage with the voltage that was previous injected into the slave. When the difference is smaller than a certain tolerance value ϵ, the system has converged. Otherwise, we repeat these steps until we reach convergence. We summarize these steps in Algorithm 1.

Algorithm 1 General algorithmic approach of the Master-Slave splitting method

```
1: Set iteration counter $\nu = 0$. Initialize the voltage $V_B^0$ of the Slave.
2: Solve the slave system. Output: $S_B^{\nu+1}$.
3: Inject $S_B^{\nu+1}$ into the Master.
4: Solve the Master. Output: $V_B^{\nu+1}$.
5: Is $|V_B^{\nu+1} - V_B^{\nu}|_1 > \epsilon$ ? Repeat step 2 till 5.
```

3.1 *The Master-Slave Splitting Extended to Balanced/Unbalanced Networks*

We are working with a combined balanced-unbalanced system. In order to use the boundary output from the slave and use it as input for the master (and vice-versa), we need to make some modifications. We can do this in two ways: (1) modeling the transmission network in three-phase or (2) transform only the boundary state output such that it matches the input format. If we model the transmission network in three-phase, we integrate two three-phase networks. We call this network an homogeneous network. We keep the assumption that the entire transmission network is balanced. For method (2), we need to transform the three-phase power output S_B^{abc} to a single-phase quantity and the single-phase voltage output V_B^a to a three-phase quantity. We make the assumption that this boundary bus B is balanced. This means that the power injected into the single-phase system is equally influenced by all three-phases:

$$S_B^a = \frac{1}{3}[1\ 1\ 1][S_B^a\ S_B^b\ S_B^c]^T. \tag{2}$$

For the voltage, this means that we can deduct phase b and c, as explained in the beginning of Sect. 2, from phase a:

$$[V_B^a\ V_B^b\ V_B^c] = V_B^a[1\ a^2\ a]^T, \quad a = e^{2/3\pi\iota}. \tag{3}$$

After we received the output in line 2 and 3 of Algorithm 1, we apply transformations (2) and (3) respectively, before we continue to the next line of the algorithm.

The Master-Slave Iterative Schemes

Two iterative schemes to solve the Master-Slave splitting are the Convergence-Alternating-Iterative (CAI)-scheme and the Multistep-Alternating-Iterative (MAI)-scheme [7]. In the CAI-scheme, explicit convergence tolerance is defined for the transmission and distribution system. At each MSS iteration step, the system is solved once its convergence condition is met. Then its output is injected into the other system. In the MAI-scheme, a maximum number of iterations per transmission and distribution system, $I_{T_{max}}$ and $I_{D_{max}}$ respectively, is defined. At each MSS iteration step, the system is solved within this number of subiterations.

4 Numerical Assessment of Integration Methods

We work in the Matpower[1] library where we created 5 test-cases: T9-D13, T118-D37, T3120-D37, T9-2D13, and T9-3D13. The distribution networks are connected via their original slack bus. The connection node of the transmission network is given. Table 2 explains the networks.

[1] MATPOWER is a package of free, open-source Matlab-language M-files for solving steady-state power system simulation and optimization problems [8].

Table 2 Comparison on number of iterations (for the MSS-method (I_{MSS}) and the necessary iterations per system (I_T and I_D)), and CPU-time of the four different MSS-methods, applied on five test-cases. The star-marked numbers converged to wrong results. In this case, the MAI method required more than 2 subiterations to converge correctly. The lowest CPU times are printed in bold

	MS-homo-CAI				MS-hybrid-CAI				MS-homo-MAI				MS-hybrid-MAI			
	I_{MSS}	I_T	I_D	CPU	I_{MSS}	I_T	I_D	CPU	I_{MSS}	I_T	I_D	CPU	I_{MSS}	I_T	I_D	CPU
	#	#	#	s	#	#	#	s	#	#	#	s	#	#	#	s
T9-D13	4	4	4	**0.247**	4	5	4	0.296	6	2	2	0.254	6	2	2	0.315
T118-D37	4	6	4	0.352	4	5	4	0.368	8*	2*	2*	0.376*	6	2	2	**0.332**
T3120-D13	4	5	3	2.27	4	8	4	0.635	4*	2*	2*	1.56*	4	2	2	**0.458**
T9-2D13	4	4	4	**0.247**	4	4	4	0.306	6	2	2	0.285	6	2	2	0.334
T9-3D13	5	4	4	**0.288**	4	4	4	0.346	7	2	2	0.313	7	2	2	0.415

- The Matpower Transmission 9-bus network connected at node 7 with a IEEE Distribution 13-bus network (T9-D13).
- The Matpower Transmission 118-bus network connected at node 117 with a IEEE Distribution 37-bus network (T118-D37).
- The Matpower Transmission 3120-bus network connected at node 1000 with the 13-bus Distribution network (T3120-D13).

We changed the 13-bus Distribution network to a 10-bus network by deleting the buses that are connected to a regulator. The 37-bus network is originally a balanced distribution network. We changed it to an unbalanced network by shifting 20% of the loads of phase b equally to phase a and c, as explained by Taranto and Marinho [1]. We created two test-cases with multiple Distribution networks: T9-2D13 and T9-3D13, respectively 2 and 3 D13-networks connected to the T9-network.

We run all simulations in Matlab. We set the tolerance value of the MSS-method to $\epsilon_{MSS} = 1e^{-7}$ and the tolerance of the NR-power method and NR-TCIM both as $\epsilon_{NR-P} = \epsilon_{NR-TCIM} = 1e^{-8}$.

Table 2 shows the number of iterations and CPU time to solve integrated networks with four different MSS-methods. At first glance, the results show that all the methods have good convergence characteristics: they converge within a small number of iterations and amount of time. These numbers are comparable for most of the test-cases. If we take a closer look at the bigger test-case, T3120-D13, we see that the MS-homogeneous methods are slower than the MS-hybrid methods. In this bigger test-case, the difference in size of a single-phase and three-phase model becomes more significant and it is thus expected that hybrid methods would perform better. A last remark is that the MS-homo-MAI method did not always converge to the correct results. Therefore, one should be careful here when using this method. Increasing the number of sub iterations of the transmission system leads to better results, but this brings MAI-method closer to the CAI-method.

In Table 3, we compare the outcome of the first test-case, T9-D13, with the output from separated networks T9 and D13. To make generation and load output of the distribution system match, we changed the load at bus 7 in the transmission network, to the total contribution of the loads in the distribution network. To make a fair

Table 3 Comparison on voltage magnitude $|V|$ and angle δ (in radians) of phase a only of the four Master-Slave splitting methods. The first four rows compare the exact values of the boundary bus B in the MSS network with connection bus (bus 7) of the separated transmission network. The last four rows compare the relative differences of the voltage magnitudes and angles with the separated networks. The two networks are compared individually. E.g.: $||\frac{||V|_{MSS}-|V|_T|}{|V|_T}||_\infty$ is the infinity norm of relative difference between the voltage magnitude of the transmission part of the MSS-method and the separated transmission model

	MS-homo-CAI	MS-hybrid-CAI	MS-homo-MAI	MS-hybrid-MAI
$\|V\|^B_{MSS}$	1.0440	1.0440	1.0443	1.0443
$\|V\|^B_{Sep}$	1.0446	1.0446	1.0446	1.0446
δ^B_{MSS}	0.2261	0.2260	0.2271	0.2272
δ^B_{Sep}	0.2308	0.2308	0.2308	0.2308
$\|\|\frac{\|\|V\|_{MSS}-\|V\|_T\|}{\|V\|_T}\|\|_\infty$	5.9E–4	5.6E–4	9.2E–4	9.1E–4
$\|\|\frac{\|\|V\|_{MSS}-\|V\|_D\|}{\|V\|_D}\|\|_\infty$	5.9E–4	5.6E–4	3.4E–4	3.2E–4
$\|\|\frac{\|\delta_{MSS}-\delta_T\|}{\delta_T}\|\|_\infty$	8.2E–2	8.3E–2	6.5E–2	6.3E–2
$\|\|\frac{\|\delta_{MSS}-\delta_D\|}{\delta_D}\|\|_\infty$	2.1E–2	2.2E–2	1.8E–2	1.7E–2

comparison, we multiplied the pu values of the voltage of the separated distribution network by the pu value of the voltage of the connection bus of the transmission network. Because the slack bus of the distribution network is a reference value for the rest of the network and thus it always holds that $V = 1$ pu. If we multiply this value by the value of V_7^T, we receive this as a new reference for the rest of the network.

Table 3 shows that all methods have similar and accurate output compared with the separated systems. It is also clear that the voltage magnitude is more accurate than the voltage angle. To explain this is an interesting follow-up study.

The MSS-method is an excellent choice if one wants to run parallel computations. The amount of communication between two networks is limited, on average 4 iterations, which makes it a suitable option for parallel computing where one master is connected to several slaves, which are all solved in parallel. Real electricity grids are designed like this, with the distribution networks having a size up to millions of buses, which makes parallel high performance computing a necessary choice. In future research, we want to test the four different MSS-methods on realistic size test-cases.

5 Conclusion

We studied the MSS-method applied on hybrid integrated networks. We showed four possible MSS-methods that deal with unbalanced distribution networks. The four different methods were named after the two possible ways we modeled the balanced networks, as a three-phase or as a single-phase network leading to MS-homogeneous and MS-hybrid methods respectively, and after how we put up the iterative schemes, the CAI-scheme and the MAI-scheme. Three out of four methods have shown to be accurate and efficient to run power flow simulations on integrated networks: The MS-homo-CAI, MS-hybrid-CAI, and MS-hybrid-MAI method. They all converged within reasonable amount of time and number of iterations, while obtaining accurate solutions. The MS-hybrid methods showed their speed-up potential when they are applied on bigger test-cases. The MS-homo-MAI method performed not so well over-all: Although the method has shown to be efficient, it does not always converge to the accurate solution. Therefore, we would not recommend to use this one.

With this extension, we showed that the MSS-method can solve integrated balanced/unbalanced networks with different characteristics. The splitting allows for solving the subsystems with their required algorithm and for sharing of information of only one overlapping boundary bus. Furthermore, it has good potential for parallel high-performance computing, which is necessary to do power flow simulations on real integrated networks.

References

1. G. N. Taranto and J. M. Marinho, "A Hybrid Three-Phase Single-Phase Power Flow Formulation," *IEEE Transactions on Power Systems*, vol. 23, no. 3, pp. 1063–1070, 2008.
2. U. Eminoglu and M. H. Hocaoglu, "The MeridiDistribution Systems Forward/Backward Sweepbased Power Flow Algorithms: A Review and Comparison Study," *Electric Power Components and Systems*, 2008.
3. H. Sun, Q. Guo, B. Zhang, and Y. Guo, "Master – Slave-Splitting Based Distributed Global Power Flow Method for Integrated Transmission and Distribution Analysis," *IEEE Transactions on Smart Grid*, vol. 6, no. 3, pp. 1484–1492, 2015.
4. P. Schavemaker and L. van der Sluis, "Energy Management Systems," in *Electrical Power System Essentials*, ch. 6, Sussex, United Kingdom: John Wiley & Sons, Inc., 2008.
5. B. Sereeter, K. Vuik, and C. Witteveen, "Newton power flow methods for unbalanced three-phase distribution networks," *Energies*, vol. 10, no. 10, p. 1658, 2017.
6. P. A. N. Garcia, J. L. R. Pereira, S. Carneiro, and V. M. Da Costa, "Three-phase power flow calculations using the current injection method," *IEEE Transactions on Power Systems*, vol. 15, no. 2, pp. 508–514, 2000.
7. H. B. Sun and B. M. Zhang, "Global state estimation for whole transmission and distribution networks," *Electric Power Systems Research*, vol. 74, pp. 187–195, 2005.
8. R. D. Zimmerman, C. E. Murillo-Sánchez, and R. J. Thomas, "MATPOWER: Steady-state operations, planning, and analysis tools for power systems research and education," *IEEE Transactions on Power Systems*, vol. 26, no. 1, pp. 12–19, 2011.

On Mesh Regularity Conditions for Simplicial Finite Elements

Ali Khademi, Sergey Korotov, and Jon Eivind Vatne

Abstract We review here various results (including own very recent ones) on mesh regularity conditions commonly imposed on simplicial finite element meshes in the interpolation theory and finite element analysis. Several open problems are listed as well.

1 Introduction

In 1968, M. Zlámal [32] introduced the so-called *minimum angle condition* that ensures the convergence of the finite element approximations for solving linear elliptic boundary value problems of the second and fourth order on planar triangulations: there exists a positive constant α_0 such that the minimal angle α_T of each triangle T in all triangulations used satisfies

$$\alpha_T \geq \alpha_0 > 0. \tag{1}$$

(To be more precise, Zlámal used the equivalent condition $\sin \alpha_T \geq \sin \alpha_0$.) The same condition was also introduced by A. Ženíšek [31] for the finite element method applied to a system of linear elasticity equations of second order.

Later, the so-called inscribed ball condition was introduced, see, e.g. [11, p. 124], which uses a ball (or its radius) contained in a given element (cf. (4)). Thus, it can also be used for nonsimplicial elements in any dimension. This condition reads as follows: the ratio of the radius of the inscribed ball of any element and the diameter of this element must be bounded from below by a positive constant over all partitions from a given family, thus preventing elements from shrinking. For a family of planar triangulations it is, obviously, equivalent to Zlámal's condition (1).

A. Khademi · S. Korotov (✉) · J. E. Vatne
Western Norway University of Applied Sciences, Bergen, Norway
e-mail: ali.khademi@hvl.no; sergey.korotov@hvl.no; jon.eivind.vatne@hvl.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_62

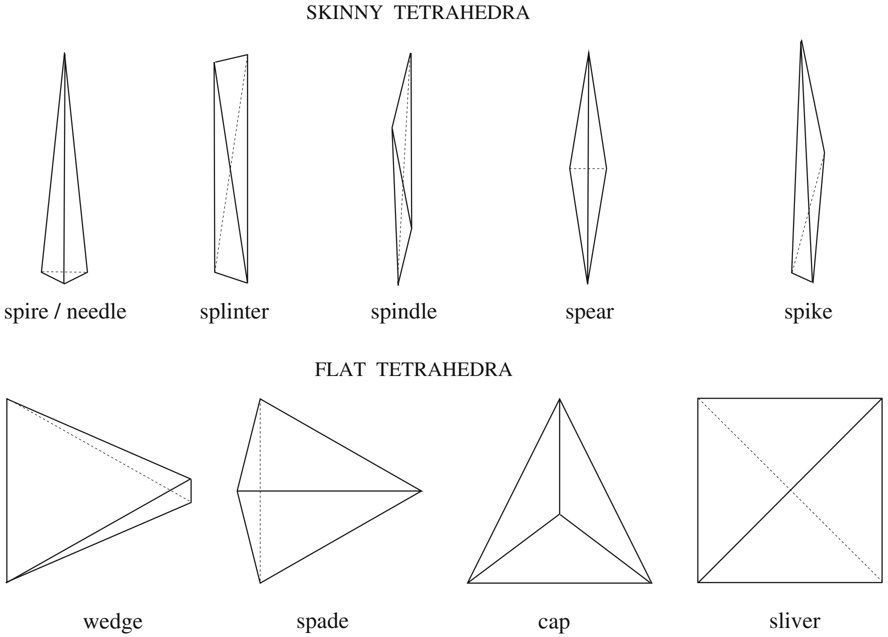


Fig. 1 Classification of degenerated tetrahedra according to [12]

In [24] the inscribed ball condition was replaced by a simpler equivalent condition on the volume of every element (cf. (3)). Another equivalent circumscribed ball condition for simplices (cf. (5)) was first introduced in [10].

In practical calculations we sometimes produce simplicial elements degenerating in some way, see Fig. 1 for the tetrahedral case. Shrinking (for example, flat and narrow) elements are also commonly used in covering thin slots, gaps or strips of different materials or to approximate functions that change more rapidly in one direction than in another direction [3]. Therefore, much effort has been devoted to finding suitable (and practical) concepts, which are weaker than the Zlámal-type conditions. The first result in this direction is usually attributed to J. L. Synge [30], who however did not consider any convergence of the finite element method, but got it in the context of interpolation theory. Already in 1957, he proved that linear triangular elements yield the optimal interpolation order in the maximum norm under the so-called maximum angle condition: there exists a constant $\gamma_0 < \pi$ such that for any triangulation $\mathcal{T}_h \in \mathcal{F}$ and any triangle $T \in \mathcal{T}_h$ we have

$$\gamma_T \le \gamma_0 < \pi, \tag{2}$$

where γ_T is the maximum angle of the triangle T. This condition is obviously weaker than the minimum angle condition (1) and was widely used later on in the finite element community for various convergence results, see e.g. [3–5, 26, 28] and many other works.

Whereas the connection between the notions introduced was clear in dimension two, much effort was later focused on extending the notions to higher dimensions and clarifying the interplay between them. This article surveys various results in this direction including our very recent contributions.

2 Main Definitions and Concepts

Recall that a *simplex* S in $\mathbf{R}^d$, $d \in \{1, 2, 3, \dots\}$, is the convex hull of $d+1$ vertices $A_0, A_1, \dots, A_d$ that do not belong to the same $(d-1)$-dimensional hyperplane, i.e., $S = \operatorname{conv}\{A_0, A_1, \dots, A_d\}$. We denote by h_S the length of the longest edge of S.

The dihedral angle α between two facets of S is defined by means of the inner product of their outward unit normals n_1 and n_2,

$$\cos\alpha = -n_1 \cdot n_2.$$

Let $\Omega \subset \mathbf{R}^d$ be a bounded domain. Assume that $\overline{\Omega}$ is *polytopic*. By this we mean that $\overline{\Omega}$ is the closure of a domain whose boundary $\partial\overline{\Omega}$ is contained in a finite number of $(d-1)$-dimensional hyperplanes.

Next we define a simplicial partition of a bounded closed polytopic domain $\overline{\Omega} \subset \mathbf{R}^d$ as follows. We subdivide $\overline{\Omega}$ into a finite number of simplices (called *elements* and denoted by S), so that their union is $\overline{\Omega}$, any two distinct simplices have disjoint interiors, and any facet of any simplex is either a facet of another simplex from the partition or belongs to the boundary $\partial\overline{\Omega}$. The set of such simplices will be called *simplicial partition* and denoted by $\mathcal{T}_h$, where $h = \max\limits_{S\in\mathcal{T}_h} h_S$.

Definition 1 The sequence of simplicial partitions $\mathcal{F} = \{\mathcal{T}_h\}_{h\to 0}$ of $\overline{\Omega}$ is called a *family of simplicial partitions* if for every $\varepsilon > 0$ there exists $\mathcal{T}_h \in \mathcal{F}$ with $h < \varepsilon$.

In this paper, all constants C_i are independent of S and h, but can depend on the dimension d. The p-dimensional volume for $p \le d$ is denoted by vol_p.

2.1 Volumic Regularity Conditions

First we present three volumic regularity conditions usually imposed on simplicial partitions in $\mathbf{R}^d$. They guarantee the optimal order of the interpolation error of simplicial finite elements, which is employed in various convergence proofs of the finite element method by means of Céa's lemma [11].

Condition 1 There exists $C_1 > 0$ such that for any $\mathcal{T}_h \in \mathcal{F}$ and any $S \in \mathcal{T}_h$ we have

$$\operatorname{vol}_d S \geq C_1 h_S^d. \tag{3}$$

Condition 2 There exists $C_2 > 0$ such that for any $\mathcal{T}_h \in \mathcal{F}$ and any $S \in \mathcal{T}_h$ we have

$$\operatorname{vol}_d b \geq C_2 h_S^d, \tag{4}$$

where $b \subset S$ is the inscribed ball of S.

Condition 3 There exists $C_3 > 0$ such that for any $\mathcal{T}_h \in \mathcal{F}$ and any $S \in \mathcal{T}_h$ we have

$$\operatorname{vol}_d S \geq C_3 \operatorname{vol}_d B, \tag{5}$$

where $B \supset S$ is the circumscribed ball about S.

Theorem 1 *Conditions 1–3 are equivalent.*

For the proof see [8].

2.2 Minimum Angle Conditions in Higher Dimensions

In this section we present two equivalent generalizations of Zlámal's condition to higher dimensions.

In 1978, F. Eriksson proposed an appropriate definition for the d-dimensional sine of angles in $\mathbf{R}^d$. In terms of the simplex S, for any of its vertices A_i, the d-dimensional sine of the angle of S at A_i, denoted by $\hat{A}_i$, is defined as follows (see (3) in [13, p. 72]):

$$\sin_d(\hat{A}_i | A_0 A_1 \ldots A_d) = \frac{d^{d-1} (\operatorname{vol}_d S)^{d-1}}{(d-1)! \prod_{j=0, j\neq i}^{d} \operatorname{vol}_{d-1} F_j}. \tag{6}$$

In [9] the following new angle-type condition was proposed.

Condition 4 There exists $C_4 > 0$ such that for any $\mathcal{T}_h \in \mathcal{F}$ and any $S = \operatorname{conv}\{A_0, \ldots, A_d\} \in \mathcal{T}_h$ we have

$$\sin_d(\hat{A}_i | A_0 A_1 \ldots A_d) \geq C_4 > 0 \qquad \forall\, i \in \{0, 1, \ldots, d\}, \tag{7}$$

where $\sin_d$ is defined in (6).

Remark 1 For $d = 2$, $\sin_2(\hat{A}_i|A_0A_1A_2)$ is the standard sine of the angle $\hat{A}_i$ in the triangle $A_0A_1A_2$, due to the following well-known formula, e.g. for $i = 0$,

$$\mathrm{vol}_2(A_0A_1A_2) = \frac{1}{2}|A_0A_1||A_0A_2|\sin\hat{A}_0, \tag{8}$$

therefore we observe that (7) presents a generalization of Zlámal's condition to higher dimensions, i.e. for $d \geq 3$.

Recently, in [20] another angle-type condition was introduced, which generalizes the three-dimensional version of the condition proposed in [7].

Condition 5 There exists a constant $C_5 > 0$ such that for any partition $\mathcal{T}_h \in \mathcal{F}$, any simplex $S \in \mathcal{T}_h$ and any subsimplex $S' \subset S$ with vertex set contained in the vertex set of S, the minimum dihedral angle $\alpha_{S'}$ in S' satisfies

$$\alpha_{S'} \geq C_5. \tag{9}$$

Theorem 2 *Conditions 1–5 are equivalent in* $\mathbf{R}^d$ *for any* $d \geq 2$.

For the proof see [20].

Remark 2 Each of Conditions 4 and 5 can be called the *minimum angle condition* as they really present a limitation of angles (or its sines) from below.

Definition 2 A family of simplicial partitions satisfying one of Conditions 1–5 is called *regular*.

3 The Maximum Angle Conditions

Now we present two conditions which weaken Conditions 4 and 5. They turn to be equivalent and might be called the *maximum angle conditions*.

As mentioned in the introduction, the Synge-condition (2) was essentially the first step in this direction. It was later on generalized by M. Křížek for tetrahedra as follows: there exists a constant $\gamma_0 < \pi$ such that for any tetrahedral partition $\mathcal{T}_h \in \mathcal{F}$ and any tetrahedron $T \in \mathcal{T}_h$ one has

$$\gamma_{\mathrm{D}}^T \leq \gamma_0 \quad \& \quad \gamma_{\mathrm{F}}^T \leq \gamma_0, \tag{10}$$

where γ_{D}^T is the maximum dihedral angles between faces of T and γ_{F}^T is the maximum angle in all four triangular faces of T, see [22].

Probably a large number of various angles in simplices for large values of d has been the reason for not elaborating the natural higher-dimensional analog of the Synge and Křížek condition until a very recent work [18], see Definition 4.

Therefore following the chronology we first present the weakened version of Condition 4 as follows, see [15].

Definition 3 A family $\mathcal{F}$ is called a *semiregular family of partitions of a polytope into simplices* if there exists $C_6 > 0$ such that for any $\mathcal{T}_h \in \mathcal{F}$ and any $S = \text{conv}\{A_0, \ldots, A_d\} \in \mathcal{T}_h$ we can always find d edges of S, which when considered as vectors, constitute a (higher-dimensional) angle whose d-sine is bounded from below by the constant C_6.

Remark 3 We observe that for the case $d = 2$, Definition 3 is equivalent to the maximum angle condition of Synge.

Remark 4 The d edges mentioned in the above definition do not necessarily emanate from the same vertex. An example is a path-simplex with its d orthogonal edges forming a path (in the sense of graph theory). The path element can degenerate (e.g. the d-dimesional sine of some of its angles can be close to zero) but the d-sine made by the orthogonal d edges stays the same. Thus, families of needle, splinter, and wedge elements from Fig. 1 satisfy Definition 3. They yield the optimal interpolation order of linear elements provided the lengths of their edges are as indicated, for example, in Fig. 2.

Recently, in [18] another condition was introduced, which presents a natural generalization of maximum angle conditions by Synge and Křížek to any dimension.

Definition 4 A family $\mathcal{F}$ of partitions of a polytope into simplices satisfies the *d-dimensional maximum angle condition* if there exists a constant $C_7 > 0$ such that for any partition $\mathcal{T}_h \in \mathcal{F}$, any simplex $S \in \mathcal{T}_h$ and any subsimplex $S' \subset S$ with vertex set contained in the vertex set of S, the maximum dihedral angle $\gamma_{S'}$ in S' satisfies

$$\gamma_{S'} \leq \pi - C_7. \tag{11}$$

The following result was proved in [18].

Theorem 3 *Definitions 3 and 4 are equivalent in $\mathbf{R}^d$ for any $d \geq 2$.*

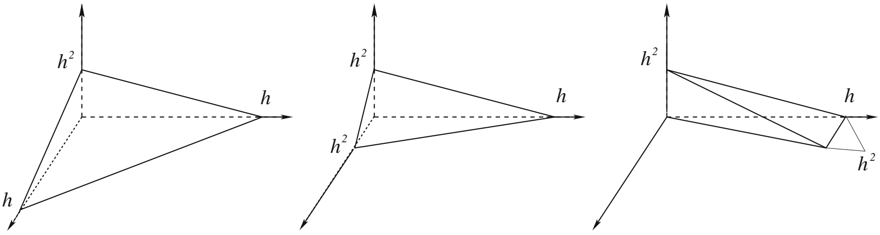


Fig. 2 Three types of degenerating tetrahedra which do not deteriorate the optimal interpolation order [22]. The length h^2 can be replaced by $h^{1+\varepsilon}$ for any $\varepsilon > 0$

4 Final Remarks and Some Open Problems

- Having several equivalent forms of some condition may sometimes considerably simplify proofs of regularity of produced meshes, see e.g. [21].
- A suitable analogue of the maximum angle condition for prismatic finite elements has been recently proposed in [17].
- We note that the maximum angle condition is, in fact, not necessary for convergence of finite element approximations as shown e.g. in [14, 19, 23]. Is there a way to prove some interpolation/convergence properties of finite element approximations constructed on simplicial meshes with other types of degeneracies than those covered by the semiregularity property?
- Is there a natural generalization of the volumic conditions that characterize semiregularity, or some weaker property (addressing to the open problem formulated above)?
- In [18] it is proved that the condition of Definition 4 is equivalent to the condition of P. Jamet from [16]. In [6], N. Baidakova proposed another angle condition which is also equivalent to the condition of Jamet.
- Some other angle-type regularity conditions used in the literature (mostly for tetrahedra) can be found in [1, 2, 25, 29].
- Closely related interesting issue of divergence of finite element approximations is discussed in [27].

References

1. G. Acosta, T. Apel, R. G. Durán, and A. L. Lombardi, *Error estimates for Raviart–Thomas interpolation of any order on anisotropic tetrahedra*, Math. Comput. **80** (2011), 141–163.
2. G. Acosta and R. G. Durán, *The maximum angle condition for mixed and nonconforming elements: Application to the Stokes equations*, SIAM J. Numer. Anal. **37** (1999), 18–36.
3. T. Apel, *Anisotropic Finite Elements: Local Estimates and Applications*, Adv. in Numer. Math., B. G. Teubner, Stuttgart, 1999.
4. I. Babuška and A. K. Aziz, *On the angle condition in the finite element method*, SIAM J. Numer. Anal. **13** (1976), 214–226.
5. R. E. Barnhill and J. A. Gregory, *Sard kernel theorems on triangular domains with applications to finite element error bounds*, Numer. Math. **25** (1976), 215–229.
6. N. V. Baidakova, *On Jamet's esimates for the finite element method with interpolation at uniform nodes of a simplex*, Sib. Adv. Math. **28** (2018), 1–22.
7. J. Brandts, S. Korotov, and M. Křížek, *On the equivalence of regularity criteria for triangular and tetrahedral finite element partitions*, Comput. Math. Appl. **55** (2008), 2227–2233.
8. J. Brandts, S. Korotov, and M. Křížek, *On the equivalence of ball conditions for simplicial finite elements in* $\mathbf{R}^d$, Appl. Math. Lett. **22** (2009), 1210–1212.
9. J. Brandts, S. Korotov, and M. Křížek, *Generalization of the Zlámal condition for simplicial finite elements in* $\mathbf{R}^d$, Appl. Math. **56** (2011), 417–424.
10. J. Brandts, M. Křížek, Gradient superconvergence on uniform simplicial partitions of polytopes. *IMA J. Numer. Anal.* **23** (2003), 489–505.

11. P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.
12. H. Edelsbrunner, *Triangulations and meshes in computational geometry*, Acta Numer. **9** (2000), 133–213.
13. F. Eriksson, *The law of sines for tetrahedra and n-simplices*, Geom. Dedicata **7** (1978), 71–80.
14. A. Hannukainen, S. Korotov, and M. Křížek, *The maximum angle condition is not necessary for convergence of the finite element method*, Numer. Math. **120** (2012), 79–88.
15. A. Hannukainen, S. Korotov, and M. Křížek, *Generalizations of the Synge-type condition in the finite element method*, Appl. Math. **62** (2017), 1–13.
16. P. Jamet, *Estimation d'erreur pour des éléments finis droits presque dégénérés*, RAIRO Anal. Numér. **10** (1976), 43–61.
17. A. Khademi, S. Korotov, and J. E. Vatne, *On interpolation error on degenerating prismatic elements*, Appl. Math. **63** (2018), 237–258.
18. A. Khademi, S. Korotov, J. E. Vatne, *On the generalization of the Synge-Křížek maximum angle condition for d-simplices*, J. Comput. Appl. Math. **358** (2019), 29–33.
19. K. Kobayashi and T. Tsuchiya, *On the circumradius condition for piecewise linear triangular elements*, Japan J. Indust. Appl. Math. **32** (2015), 65–76.
20. S. Korotov and J. E. Vatne, *The minimum angle condition for d-simplices*, *Comput. Math. Appl.* **80** (2020), 367–370.
21. S. Korotov and J. E. Vatne, *On regularity of tetrahedral meshes produced by some red-type refinements*, In: Proc. of Inter. Conf. ICDDEA-2019, Lisbon, Portugal (ed. by S. Pinelas et al.) (in press).
22. M. Křížek, *On the maximum angle condition for linear tetrahedral elements*, SIAM J. Numer. Anal. **29** (1992), 513–520.
23. V. Kučera, *Several notes on the circumradius condition*, Appl. Math. **61** (2016), 287–298.
24. J. Lin and Q. Lin, *Global superconvergence of the mixed finite element methods for 2-d Maxwell equations*, *J. Comput. Math.* **21** (2003), 637–646.
25. A. Liu and B. Joe, *Relationship between tetrahedron shape measures*, BIT **34** (1994), 268–287.
26. S. Mao and Z. Shi, *Error estimates of triangular finite elements under a weak angle condition*, J. Comput. Appl. Math. **230** (2009), 329–331.
27. P. Oswald, *Divergence of FEM: Babuška-Aziz triangulations revisited*, Appl. Math. **60** (2015), 473–484.
28. A. Rand, *Average interpolation under the maximum angle condition*, SIAM J. Numer. Anal. **50** (2012), 2538–2559.
29. Yu. N. Subbotin, *Dependence of estimates of a multidimensional piecewise polynomial approximation on the geometric characteristics of the triangulation*, Tr. Mat. Inst. Steklova **189**, 117 (1989).
30. J. L. Synge, *The Hypercircle in Mathematical Physics*, Cambridge Univ. Press, Cambridge, 1957.
31. A. Ženíšek, *The convergence of the finite element method for boundary value problems of a system of elliptic equations (in Czech)*, Apl. Mat. **14** (1969), 355–377.
32. M. Zlámal, *On the finite element method*, Numer. Math. **12** (1968), 394–409.

Assembly of Multiscale Linear PDE Operators

Miroslav Kuchta

Abstract In numerous applications the mathematical model consists of different processes coupled across a lower dimensional manifold. Due to the multiscale coupling, finite element discretization of such models presents a challenge. Assuming that only *singlescale* finite element forms can be assembled we present here a simple algorithm for representing multiscale models as linear operators suitable for Krylov methods. Flexibility of the approach is demonstrated by numerical examples with coupling across dimensionality gap 1 and 2. Preconditioners for several of the problems are discussed.

1 Introduction

This paper is concerned with implementation of the finite element method (FEM) for multiscale models, that is, systems where the unknowns are defined over domains of (in general) different topological dimension and are coupled on a manifold, which is possibly a different domain. The systems arise naturally in applications where Lagrange multipliers are used to enforce boundary conditions, e.g. [4, 7], or interface coupling conditions e.g. [3, 23]. In modeling reservoir flows [10], tissue perfusion [9, 11, 20] or soil-root interaction [19] resolving the interface as a manifold of co-dimension 1 can be prohibitively expensive. In this case it is convenient to represent the three-dimensional structures as curves and the model reduction gives rise to multiscale systems with a dimesionality gap 2.

Crucial for the FEM discretization of the multiscale models is the assembly of coupling terms, in particular, integration over the coupling manifold. There exists a number of open source FEM libraries, e.g. [1, 6, 14, 17], which expose this (low-level) functionality and as such can be used for implementation. However, for rapid

M. Kuchta (✉)
Simula Research Laboratory, Lysaker, Norway
e-mail: miroslav@simula.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_63

prototyping, it is advantageous if the new models are described in a more abstract way which is closer to the mathematical definition of the problem.

FEniCS is a popular open source FEM framework which employs a compiler to generate low level (C++) assembly code from the high-level symbolic representation of the variational forms in the UFL language embedded in Python, see [24]. Here the code generation pipeline provides convenience for the user. At the same time, implementing new features is complicated by the fact that interaction with all the components of the pipeline is required. As a result, support for multiscale models has only recently been added to the core of the library [12] and is currently limited to problems with dimensionality gap 0 and 1. Moreover, in case of the trace constrained systems the coupling manifold needs to be triangulated in terms of facets of the bulk discretization. We remark that similar functionality for multiscale systems is offered by the FEniCS based library [2].

Here we present a simple algorithm[1] which extends FEniCS to support a more general class of multiscale systems by transforming symbolic variational forms in UFL language into a domain specific language [25] which represents (actions of) discrete linear operators. As this representation targets solutions by iterative methods, preconditioning strategies shall also be discussed. Our work is structured as follows. Section 2 details the algorithm. Numerical examples with dimensionality gap 1 and 2 are presented in Sects. 3 and 4 respectively.

2 Multiscale Assembler

In the following $(\cdot, \cdot)_\Omega$ denotes the L^2 inner product over a bounded domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. The duality pairing between the Hilbert space V and its dual space V' is denoted by $\langle \cdot, \cdot \rangle$. Given basis of a discrete finite element space V_h, the matrix representation of operator A is A_h. Adjoints of A and A_h are denoted as A' and A'_h respectively.

Our representation of multiscale systems builds on two observations, which shall be presented using the Babuška problem [4]. Let $\Gamma = \partial\Omega$ and $V = H^1(\Omega)$, $Q = H^{-1/2}(\Gamma)$, $W = V \times Q$. Then for every $L \in W'$ there exists a unique solution $w = (u, p) \in W$ satisfying $\mathcal{A}w = L$ where

$$\mathcal{A} = \begin{pmatrix} A & B' \\ B & 0 \end{pmatrix} \quad \text{and} \quad \begin{aligned} \langle Au, v \rangle &= (\nabla u, \nabla v)_\Omega + (u, v)_\Omega \quad v \in V, \\ \langle Bu, q \rangle &= (Tu, q)_\Gamma \quad q \in Q. \end{aligned} \tag{1}$$

[1] Implementation can be found in the Python module `FEniCS_ii` https://github.com/MiroK/fenics_ii.

Here $T : H^1(\Omega) \rightarrow H^{1/2}(\Gamma)$ is the trace operator such that $Tu = u|_\Gamma$, $u \in C(\overline{\Omega})$. We remark that (1) is the weak form of $-\Delta u + Iu = f$ in Ω with $u = g$ on $\partial\Omega$ enforced by the Lagrange multiplier p.

Given the structure of $\mathcal{A}$ in (1) it is natural to represent the operator on a finite element space W_h as a block structured matrix (rather then a monolithic one). Moreover, observe that the *multiscale* operator $B : V \rightarrow Q$ (operator $A : V \rightarrow V'$ is *singlescale*) is a composition $B = I \circ T$ where $I : H^{1/2}(\Gamma) \rightarrow Q$ is a singlescale operator. Therefore, matrix representation of B is a matrix product $B_h = I_h T_h$. Assuming that the FEM library at hand can only assemble singlescale operators, e.g. I and A, the multiscale operators B_h and $\mathcal{A}_h$ can be formed, if representation of the trace operator is available. We remark that the block representation is advantageous for construction of preconditioners; for example the blocks can be easily shared between the system and the preconditioner, cf. [18, 25].

Based on the above observations the multiscale systems can be represented as block structured operators where the blocks are not necessarily matrices. Cbc.block [25] defines a language for matrix expressions using the lazy evaluation pattern. In particular, block matrix(`block_mat`) and matrix product($*$) are built-in operators. We remark that the operators are not formed explicitly, however, they can be evaluated if e.g. action in a matrix-vector product in a Krylov solver is needed. Using B from (1) as an example we thus aim to build an interpreter which translates UFL representation of $(Tu, q)_\Gamma$ into a cbc.block representation $I_h{*}T_h$. We remark that T_h is here assumed to be a mapping between primal representations, cf. [26].

The core of the multiscale interpreter is the algorithm (Fig. 1) for translating from one symbolic representation to another. Observe that in `multi_assemble` different *reduced* assemblers are recursively called on the transformed UFL form with the singlescale form being the base case. An example of a reduced assembler is the `trace_assemble` function which, having found *trace*-reduced argument (ln. 12) in form a, e.g. $a(u, q) = (Bu, q) = (Tu, q)_\Gamma$, $u \in V_h$, $q \in Q_h$ builds a finite element *trace space* $\bar{V}_h = \bar{V}_h(\Gamma)$ (ln. 14), an algebraic representation of the operator $T : V_h \rightarrow \bar{V}_h$ (ln. 15) and delegates assembly of the transformed form $I(\bar{u}, q) = (\bar{u}, q)_\Gamma$, $\bar{u} \in \bar{V}_h$, $q \in Q_h$ (ln. 19) to `multi_assemble` (ln. 20). As I is singlescale the native FEniCS assemble function can be used to form the matrix I_h and the symbolic matrix-matrix product representation can be formed (ln. 20). The translation can thus be summarized as $(Tu, q)_\Gamma \rightarrow (\bar{u}, q)_\Gamma{*}T_h \rightarrow I_h{*}T_h$.

Algorithm 1 can be easily extended to different multiscale couplings by adding a dedicated assembler. In particular, given $\Omega \subset \mathbb{R}^3$ and γ a curve contained in Ω, the 3d-1d coupled problems [10, 11] require operators T, Π such that for $u = C(\Omega)$, $Tu = u|_\gamma$ and

$$(\Pi u)(x) = |C_R(x)|^{-1} \int_{C_R(x)} u(y)\, \mathrm{d}y. \tag{2}$$

Here $C_R(x)$ is a circle of radius R in a plane $\{y \in \mathbb{R}^3, (y - x) \cdot \frac{\mathrm{d}\gamma}{\mathrm{d}s}(x) = 0\}$ defined by the tangent vector of γ at x. Observe that assembling 3d-1d constrained operators

Algorithm 1: multi_assemble

```
Data: a::UFL.Form or list of UFL.Form
Result: cbc.block matrix expression
begin
    // Single form
    if a is UFL.Form then
        // Attempt to reduce
        for assemble ∈ assemblers do
            tensor = assemble(a)
            if tensor is not None then
                return tensor
        // Singlescale operator
        return FEniCS.assemble(a)
    // Functional
    if is_number(form) then
        return form
    shape ← sizes(form)
    // Assemble blocks
    blocks ← map(multi_assemble, form)
    // List/List of operators
    tensor ← reshape(blocks, shape)
    // Reshape for cbc.block
    // Form had test functions only
    if is_vector(tensor) then
        return block.block_vec(tensor)
    // Bilinear form
    return block.block_mat(tensor)
```

Algorithm 2: trace_assemble

```
Data: a::UFL.Form
Result: cbc.block matrix expression
begin
    trace_integrals ← get_trace_integrals(a)
    all_integrals ← integrals(form)
    if not trace_integrals then
        return None
    // Form is sum of integrals...
    cs ← []
    for $i$ ∈ all_integrals do
        if $i \notin$ trace_integrals then
            cs += [multi_assemble(Form([i]))]
            continue
        intgrnd ← integrand(i)
        $u$, ← trace_terminals(intgrnd)
        $V_h$ ← function_space($u$)
        $\bar{V}_h$ ← trace_space($V_h$, $u$)
        $T_h$ ← trace_matrix($V_h$, $\bar{V}_h$)
        if is_trial_function(u) then
            $\bar{u}$ ← TrialFunction($\bar{V}_h$)
            ii ← replace(intgrnd, u, $\bar{u}$)
            $I$ = Form([reconstruct(i, ii)])
            $B_h$ ← multi_assemble($I$)*T
            cs += [$B_h$]
        // Handle test/function
    // ...cbc.block sum of operators
    return reduce(+, cs)
```

Fig. 1 Translation of UFL representation of multiscale variational form into cbc.block matrix expression. Several passes by different scale assemblers might be needed to reduce the form into singlescale base case which can be assembled as matrix or vector by FEniCS. Handling of test function and function type terminals is omitted for brevity

follows closely Algorithm 2, with the non-trivial difference being the representation of Π. We remark that in assembly of Π or T we do not require that γ is discretized in terms of edges of the mesh of Ω. In fact, the two meshes can be independent. This is also the case for d–$(d-1)$ trace. Let us also note that the restriction operator $Ru = u|_\omega$, where $\omega \subseteq \Omega \subset \mathbb{R}^d$ can be implemented similar to the trace operator. Finally, observe that the Algorithm 1 is not limited to forms where the arguments are *reduced* to the coupling manifold. Indeed, [10, 16, 21] utilize *extension* from γ to Ω by a constant or as Green function of a line source respectively. Such couplings can be readily handled if realization of the discrete *extension* operator is available.

We conclude the discussion by listing the limitations of our current implementation. Unlike in [2, 12] the MPI-parallelism is missing[2] as is the support for nonlinear

[2] The serial performance of our pure Python implementation is cca. 2× slower than the native FEniCS implementation [12]. More precisely, assembling (1) on $\Omega = [0, 1]^2$ discretized by $2 \cdot 1024^2$ triangles and continuous linear Lagrange elements (the system matrix size is cca. 10^6, however, it is *not* explicitly formed here) takes 3.86 s (to be compared with 1.79 s). Most of the time is spent building T_h. The trace matrix is reused by the interpreter to evaluate both B_h and B_h'.

forms. Moreover, the reduction operators cannot be nested and can only be applied to terminal expressions in UFL, e.g. $T(u+v)$ cannot be interpreted. In addition, point constraints are not supported. With the exception of parallelism the limitations will be addressed by future versions.

In the following we showcase the multiscale interpreter by considering coupled problems with dimensionality gap 1 and 2. We begin by a trace constrained $2d$-$1d$ Darcy-Stokes system.

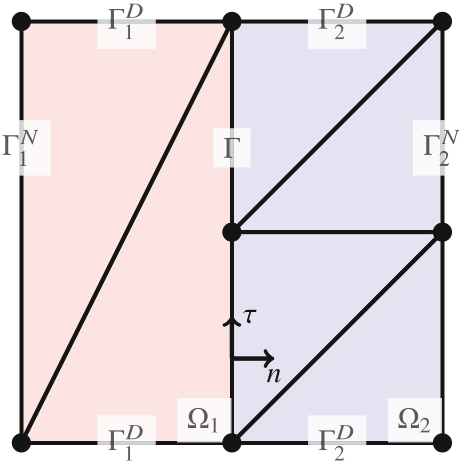
3 Trace Constrained Systems

Let $\Omega_1, \Omega_2 \subset \mathbb{R}^2$ be such that $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ and $|\Gamma| \neq 0$. Further let $\partial\Omega_i = \Gamma \cup \Gamma_i^D \cup \Gamma_i^N$ where $|\Gamma_i^k| \neq 0$, $i = 1, 2$, $k = N, D$ and $\Gamma \cap \Gamma_i^N = \emptyset$, cf. Fig. 2. We then wish to solve the coupled Darcy-Stokes problem (with unit parameters)

$$
\begin{aligned}
-\nabla\cdot\sigma = f_1, \nabla\cdot u_1 = 0 &\quad \text{in } \Omega_1,\\
u_2 + \nabla p_2 = 0, \nabla\cdot u_2 = f_2 &\quad \text{in } \Omega_2,\\
u_1\cdot n - u_2\cdot n = 0 &\quad \text{on } \Gamma,\\
n\cdot\sigma\cdot n + p_2 = 0 &\quad \text{on } \Gamma,\\
-n\cdot\sigma\cdot\tau - u_1\cdot\tau = 0 &\quad \text{on } \Gamma.
\end{aligned}
\tag{3}
$$

Here $\sigma(u_1, p_1) = D(u_1) - p_1 I$ with $D(u) = \frac{1}{2}((\nabla u) + (\nabla u)')$. The unknowns u_1, p_1 and u_2, p_2 are respectively the Stokes and Darcy velocity and pressure. The

Fig. 2 Domain for (3)



system is closed by prescribing Dirichlet conditions on Γ_i^D and Neumann conditions on Γ_i^N.

Let T_n, T_t be the normal and tangential trace operators on Γ. We shall consider variational formulations of (3) induced by a pair of operators

$$\mathcal{A}_p = \left(\begin{array}{cc|c} -\nabla\cdot D + T_t'T_t & -\nabla & T_n' \\ \mathrm{div} & & \\ \hline -T_n & & -\Delta \end{array}\right), \mathcal{A}_m = \left(\begin{array}{cc|cc|c} -\nabla\cdot D + T_t'T_t & -\nabla & & & T_n' \\ \mathrm{div} & & & & \\ \hline & & I & -\nabla & -T_n' \\ & & \mathrm{div} & & \\ \hline T_n & & -T_n & & \end{array}\right). \tag{4}$$

Using the (mixed) operator $\mathcal{A}_m$ problem (3) is solved for both u_2, p_2 and an additional unknown, the Lagrange multiplier, which enforces mass conservation $u_1\cdot n - u_2\cdot n = 0$ on Γ. In the (primal) operator $\mathcal{A}_p$ the condition appears naturally. Observe that the primal operator is non-symmetric.

Well-posedness of the primal and mixed formulations as well the corresponding solution strategies have been studied in a number of works, e.g. [13] and [15, 23]. Here we compare the formulations and discuss monolithic solvers which utilize block diagonal preconditioners

$$\begin{aligned} \mathcal{B}_p &= \mathrm{diag}\left(-\nabla\cdot D + T_t'T_t, I, -\Delta\right)^{-1}, \\ \mathcal{B}_m &= \mathrm{diag}\left(-\nabla\cdot D + T_t'T_t, I, I - \nabla\mathrm{div}, I, (-\Delta + I)^{1/2}\right)^{-1}. \end{aligned} \tag{5}$$

The preconditioner $\mathcal{B}_p$ has been proposed by Cai et al. [8], while $\mathcal{B}_m$ follows from the analysis [15] by operator preconditioning technique [26]. More precisely, $\mathcal{B}_m$ is a Riesz map with respect to the inner product of the space in which [15] prove well-posedness of $\mathcal{A}_m$, i.e. $H^1_{0,\Gamma_1^D}(\Omega_1)\times L^2(\Omega_1)\times H_{0,\Gamma_2^D}(\mathrm{div},\Omega_2)\times L^2(\Omega_2)\times H^{1/2}(\Gamma)$. We remark that all the blocks of the preconditioners can be realized by efficient and order optimal multilevel methods. In particular, we shall use further the multigrid realization of the fractional Laplace preconditioner [5].

In order to check mesh independence of the preconditioners let us consider the geometry from Fig. 2 and let $\Omega_1 = [0, 0.5]\times[0, 1]$, $\Omega_2 = [0.5, 1]\times[0, 1]$. In both $\mathcal{A}_m$, $\mathcal{A}_p$ the triangulations of the domains shall be *independent*,[3] cf. Fig. 2, with the mesh of Γ defined in terms of facets of Ω_2. Finally, the finite element approximation

[3] Details of experimental setup. We discretize Ω_i uniformly by first dividing the domains into $n\times m$ rectangles and afterwords splitting each rectangle into two triangles. For Ω_1 we have $m = n$, $m = 2n$ for Ω_2 so that the trace meshes of the domains are different. Krylov solvers are started from random initial guess. Convergence tolerance for relative preconditioned residual norm of 10^{-10} is used. Unless specified otherwise the preconditioner blocks use LU factorization.

Table 1 Number of iterations required for convergence of GMRes($\mathcal{A}_p$) and MinRes($\mathcal{A}_m$) using preconditioners (5). Multigrid preconditioner for $H^{1/2}$ leads to slightly increased number of iterations compared to eigenvalue realization [22]

h	$\mathcal{B}_p\mathcal{A}_p$	$\mathcal{B}_p^{\text{EIG}}\mathcal{A}_p$	$\mathcal{B}_p^{\text{MG}}\mathcal{A}_p$
2^{-3}	48	53	59
2^{-4}	48	51	59
2^{-5}	47	50	63
2^{-6}	47	49	65
2^{-7}	46	49	65

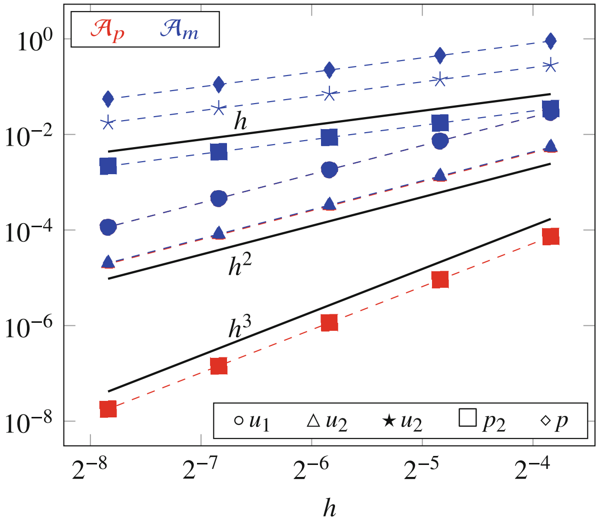


Fig. 3 Convergence of the primal(red) and mixed formulation of (3). The approximation error is computed in the norms of $\mathcal{B}_p^{-1}$ and $\mathcal{B}_m^{-1}$

of $\mathcal{A}_p$ shall be constructed using P_2-P_1-P_2 elements[4] while P_2-P_1-RT_0-P_0-P_0 is used for the mixed formulation $\mathcal{A}_m$.

Results of the numerical experiment are summarized in Table 1. It can be seen that the preconditioners (5) are robust with respect to the discretization. Further, Fig. 3 shows that both formulations lead to expected order of convergence in all the unknowns. The approximation of Stokes variables is practically identical. We remark that p_2 convergence in $\mathcal{A}_p$ is reported in the L^2 norm for the sake of comparison with the mixed formulation,[5]

[4] Finite element space of continuous Lagrange elements of order k is denoted by P_k while RT_0 denotes the space of lowest order Raviart-Thomas elements.

[5] The implementation of the Darcy-Stokes problems with *conforming* meshes can be found at https://github.com/MiroK/fenics_ii/blob/master/demo/ as dq_darcy_stokes_2d.py (primal formulation) and darcy_stokes_2d.py (mixed formulation).

4 More General Multiscale Systems

To show flexibility of the interpreter we finally consider a simple prototypical $3d$-$1d$ coupled problem. Let $\Omega \subset \mathbb{R}^3$ be a bounded domain and let γ be a curve embedded in Ω. Assuming γ is a representation of the vasculature (e.g. as center lines) parameterized by arc length coordinate s a model of tissue *perfusion* by D'Angelo and Quarteroni [11] can be represented as an operator equation

$$\mathcal{A}_p \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} -k\Delta + T'\Pi & \beta T' \\ -\beta\Pi & -\hat{k}\Delta + \beta I \end{pmatrix} \begin{pmatrix} u \\ p \end{pmatrix} = \begin{pmatrix} f_u \\ f_p \end{pmatrix}. \tag{6}$$

Here k, $\hat{k}$ are the conductivities of the tissue and the vasculature, while β is the permeability. We denote f_u and f_p potential source terms in Ω and Γ respectively.

Compared to Babuška problem (1) or Darcy-Stokes problem (4) the system (6) introduces new multiscale coupling as in the perfusion operator $\mathcal{A}_p$ the test functions in the bulk are reduced to γ by a $3d$-$1d$ trace operator (T) while Π in (2) is used for the trial functions.

We test the abilities of the assembler by considering FEM discretization of (6) in terms of P_1-P_1 elements with the problem setup on a uniform discretization of $[0, 1]^3$ and γ a straight line which, in general, is not aligned with the edges of the mesh of Ω. Figure 4 shows that the relative norm of the refined solution decreases linearly.

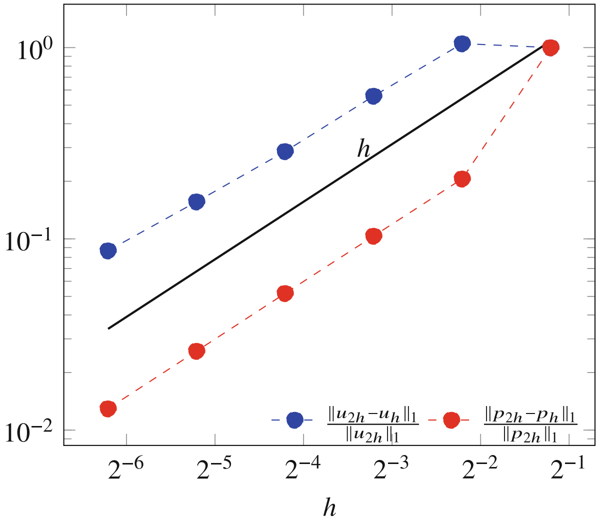


Fig. 4 Error convergence of the FEM approximation of $3d$-$1d$ coupled problem (6)

References

1. MFEM: Modular finite element methods library. mfem.org. https://doi.org/10.11578/dc.20171025.1248
2. multiphenics - easy prototyping of multiphysics problems in FEniCS. https://mathlab.sissa.it/multiphenics. Accessed: 2019-12-16
3. Ambartsumyan, I., Khattatov, E., Yotov, I., Zunino, P.: A Lagrange multiplier method for a Stokes–Biot fluid–poroelastic structure interaction model. Numerische Mathematik **140**(2), 513–553 (2018)
4. Babuška, I.: The finite element method with Lagrangian multipliers. Numerische Mathematik **20**(3), 179–192 (1973)
5. Bærland, T., Kuchta, M., Mardal, K.A.: Multigrid methods for discrete fractional Sobolev spaces. SIAM Journal on Scientific Computing **41**(2), A948–A972 (2019)
6. Bangerth, W., Hartmann, R., Kanschat, G.: deal.II – a general purpose object oriented finite element library. ACM Trans. Math. Softw. **33**(4), 24/1–24/27 (2007)
7. Bertoluzza, S., Chabannes, V., Prud'Homme, C., Szopos, M.: Boundary conditions involving pressure for the Stokes problem and applications in computational hemodynamics. Computer Methods in Applied Mechanics and Engineering **322**, 58–80 (2017)
8. Cai, M., Mu, M., Xu, J.: Preconditioning techniques for a mixed Stokes/Darcy model in porous media applications. Journal of computational and applied mathematics **233**(2), 346–355 (2009)
9. Cattaneo, L., Zunino, P.: A computational model of drug delivery through microcirculation to compare different tumor treatments. International journal for numerical methods in biomedical engineering **30**(11), 1347–1371 (2014)
10. Cerroni, D., Laurino, F., Zunino, P.: Mathematical analysis, finite element approximation and numerical solvers for the interaction of 3d reservoirs with 1d wells. GEM-International Journal on Geomathematics **10**(1), 4 (2019)
11. D'Angelo, C., Quarteroni, A.: On the coupling of 1d and 3d diffusion-reaction equations: application to tissue perfusion problems. Mathematical Models and Methods in Applied Sciences **18**(08), 1481–1504 (2008)
12. Daversin-Catty, C., Richardson, C.N., Ellingsrud, A.J., Rognes, M.E.: Abstractions and automated algorithms for mixed domain finite element methods. arXiv preprint arXiv:1911.01166 (2019)
13. Discacciati, M., Miglio, E., Quarteroni, A.: Mathematical and numerical models for coupling surface and groundwater flows. Applied Numerical Mathematics **43**(1–2), 57–74 (2002)
14. Fournié, M., Renon, N., Renard, Y., Ruiz, D.: CFD parallel simulation using GetFem++ and MUMPS. In: P. D'Ambra, M. Guarracino, D. Talia (eds.) Euro-Par 2010 - Parallel Processing, pp. 77–88. Springer Berlin Heidelberg, Berlin, Heidelberg (2010)
15. Galvis, J., Sarkis, M.: Non-matching mortar discretization analysis for the coupling Stokes-Darcy equations. Electron. Trans. Numer. Anal **26**(20), 07 (2007)
16. Gjerde, I.G., Kumar, K., Nordbotten, J.M.: A singularity removal method for coupled 1d-3d flow models. arXiv preprint arXiv:1812.03055 (2018)
17. Hecht, F.: New development in FreeFem++. J. Numer. Math. **20**(3–4), 251–265 (2012). URL https://freefem.org/
18. Kirby, R.C., Mitchell, L.: Solver composition across the PDE/linear algebra barrier. SIAM Journal on Scientific Computing **40**(1), C76–C98 (2018)
19. Koch, T., Heck, K., Schröder, N., Class, H., Helmig, R.: A new simulation framework for soil–root interaction, evaporation, root growth, and solute transport. Vadose Zone Journal **17**(1) (2018)
20. Koch, T., Schneider, M., Helmig, R., Jenny, P.: Modeling tissue perfusion in terms of 1d-3d embedded mixed-dimension coupled problems with distributed sources. arXiv preprint arXiv:1905.03346 (2019)

21. Kuchta, M., Laurino, F., Mardal, K.A., Zunino, P.: Analysis and approximation of mixed-dimensional pdes on 3d-1d domains coupled with Lagrange multipliers. arXiv preprint arXiv:2004.02722 (2020)
22. Kuchta, M., Nordaas, M., Verschaeve, J.C., Mortensen, M., Mardal, K.A.: Preconditioners for saddle point systems with trace constraints coupling 2d and 1d domains. SIAM Journal on Scientific Computing **38**(6), B962–B987 (2016)
23. Layton, W.J., Schieweck, F., Yotov, I.: Coupling fluid flow with porous media flow. SIAM Journal on Numerical Analysis **40**(6), 2195–2218 (2002)
24. Logg, A., Mardal, K.A., Wells, G.: Automated solution of differential equations by the finite element method: The FEniCS book, vol. 84. Springer Science & Business Media (2012)
25. Mardal, K.A., Haga, J.B.: Block preconditioning of systems of PDEs, pp. 643–655. Springer Berlin Heidelberg, Berlin, Heidelberg (2012)
26. Mardal, K.A., Winther, R.: Preconditioning discretizations of systems of partial differential equations. Numerical Linear Algebra with Applications **18**(1), 1–40 (2011)

A Least-Squares Galerkin Gradient Recovery Method for Fully Nonlinear Elliptic Equations

Omar Lakkis and Amireh Mousavi

Abstract We propose a least squares Galerkin based gradient recovery to approximate Dirichlet problems for strong solutions of linear elliptic problems in non-divergence form and corresponding a priori and a posteriori error bounds. This approach is used to tackle fully nonlinear elliptic problems, e.g., Monge–Ampère, Hamilton–Jacobi–Bellman, using the smooth (vanilla) and the semismooth Newton linearization. We discuss numerical results, including adaptive methods based on the a posteriori error indicators.

1 Introduction

Let Ω denote a bounded convex domain in $\mathbb{R}^d$, $d \in \mathbb{N}$ (typically $d = 2, 3$). Consider the Dirichlet problem of finding a function $u : \Omega \to \mathbb{R}$ such that

$$\mathscr{F}[x, u, \nabla u, \mathrm{D}^2 u] = 0 \text{ and } u|_{\partial\Omega} = r. \tag{1}$$

Here, ∇u, $\mathrm{D}^2 u$ denote the gradient and the Hessian of u and $\mathscr{F} : \Omega \times \mathbb{R} \times \mathbb{R}^d \times \mathbb{R}^{d\times d} \to \mathbb{R}$ is assumed to be elliptic and Newton differentiable which is defined by Definition 2.

While viscosity solutions are possible, in a natural way, for this type of equations, we here focus on smoother solutions. Namely, we look at the numerical approximations of u in $\mathrm{H}^2(\Omega)$ satisfying (1), termed *strong solution*. We follow a series of papers on the matter [3, 4, 11], but with a focus on the different and somewhat more flexible numerical methodology of least squares gradient recovery

O. Lakkis (✉)
University of Sussex, Brighton, England
e-mail: o.lakkis@sussex.ac.uk

A. Mousavi
Isfahan University of Technology, Isfahan, Iran
e-mail: amireh.mousavi@math.iut.ac.ir

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_64

Galerkin finite element method to discretize the linear equations in nondivergence form that ensue from linearizing (1) using semismooth Newton method. We only state the results here, respectively referring for the details of Sects. 2 and 3 to Lakkis and Mousavi [7] and Lakkis and Mousavi [8]. We look at some numerical examples, outlining an adaptive algorithm based on a posteriori error estimates for the linear elliptic equations in nondivergence form with Cordes coefficients.

2 A Least-Squares Galerkin Approach to Gradient Recovery for Linear Equations in Nondivergence Form

We outline the proposed numerical method of the strong solution of the linear second order equation in nondivergence form; the details, including the proofs of all stated results can be found in [7]. To prevent difficulties arising from numerically working in $\mathrm{H}^2(\Omega)$ space, we consider an equivalent problem with solution in a H^1-regularity space. For this we minimize a *cost* (least-squares) functional associated to the main problem. We prove that the equivalent problem is well posed using a coercivity argument, deducing thus the same result for the discrete counterpart. By setting Galerkin finite element spaces within $\mathrm{H}^1(\Omega)$, we provide a priori and a posteriori error bounds.

In this section we consider the following linear second order elliptic equations in nondivergence form of finding $u \in \mathrm{H}^2(\Omega)$ such that

$$\mathscr{L}u := A : \mathrm{D}^2 u + b^{\mathsf{T}}\nabla u - cu = f \text{ and } u|_{\partial\Omega} = 0 \tag{2}$$

where the coefficients $A \in \mathrm{L}_\infty(\Omega; \mathrm{Sym}(\mathbb{R}^d))$, with $\mathrm{Sym}(X) =:$ symmetric operators on X, is uniformly elliptic, $b \in \mathrm{L}_\infty(\Omega; \mathbb{R}^d)$ and $c \in \mathrm{L}_\infty(\Omega)$, $c \geqslant 0$ satisfy exactly one of the following two *Cordes conditions* for some $\varepsilon \in (0, 1)$

$$b \neq 0 \text{ or } c \neq 0 \;\Rightarrow\; \frac{|A|^2 + {|b|^2}/{2\lambda} + ({c}/{\lambda})^2}{(\operatorname{tra} A + {c}/{\lambda})^2} \leqslant \frac{1}{d+\varepsilon} \text{a.e. in } \Omega \text{ for some } \lambda > 0, , \tag{3}$$

$$b \equiv 0 \text{ and } c \equiv 0 \;\Rightarrow\; \frac{|A|^2}{(\operatorname{tra} A)^2} \leqslant \frac{1}{d-1+\varepsilon} \text{a.e. in } \Omega, \tag{4}$$

where $|X| = \left(\operatorname{tra} X^{\mathsf{T}} X\right)^{1/2}$. The right-hand side f is a generic element of $\mathrm{L}_2(\Omega)$. We consider the right-hand side r in (1) to be 0 for simplicity (although the developments can be extended to $r|_{\partial\Omega}$ being the trace of a function $r \in \mathrm{H}^2(\Omega)$.

Problem (2) is well posed under these assumptions as shown by Smears and Süli [10]. In numerical approximating solutions, dealing with more regular than $\mathrm{H}^1(\Omega)$ spaces leads to complicated computations. To avoid this difficulty, we intend to consider an alternative equivalent problem with $\mathrm{H}^1(\Omega)$ solution.

We denote the outer normal to Ω at $x \in \partial\Omega$ by $n_\Omega(x)$, which we assume defined for $\mathscr{S}$-almost every $x \in \partial\Omega$ ($\mathscr{S}$ being the $(d-1)$-dimensional "surface" measure) and recall the tangential trace of $\psi \in \mathrm{H}^1(\Omega; \mathbb{R}^d)$ is expressed (or defined) by

$$\left(I - n_\Omega {n_\Omega}^\mathsf{T}\right) \psi|_{\partial\Omega} \,. \tag{5}$$

Define the following function spaces

$$\mathscr{W} := \left\{\psi \in \mathrm{H}^1(\Omega; \mathbb{R}^d) : \left(I - n_\Omega {n_\Omega}^\mathsf{T}\right) \psi|_{\partial\Omega} = 0\right\}, \tag{6}$$

$$\mathscr{Y} := \mathrm{H}^1(\Omega) \times \mathrm{H}^1\left(\Omega; \mathbb{R}^d\right) \tag{7}$$

$$\mathscr{V} := \mathrm{H}^1_0(\Omega) \times \mathscr{W} \subseteq \mathscr{Y}, \tag{8}$$

endowed with the H^1-norm for $\mathscr{W}$ and the following norm for $\mathscr{Y}$ and $\mathscr{V}$,

$$\|(\varphi, \psi)\|^2_{\mathscr{Y}} := \|\varphi\|^2_{\mathrm{H}^1(\Omega)} + \|\psi\|^2_{\mathrm{H}^1(\Omega)} \text{ for each } (\varphi, \psi) \in \mathscr{Y} \supseteq \mathscr{V}. \tag{9}$$

We denote by $\langle\varphi, \psi\rangle$ the $\mathrm{L}_2(D; V)$ inner product with respect to the Lebesgue or surface measure on D. For a fixed $\theta \in [0, 1]$ we introduce the linear operator $\mathscr{M}_\theta : \mathscr{Y} \to \mathrm{L}_2(\Omega)$

$$(\varphi, \psi) \mapsto A : \mathrm{D}\,\psi + b^\mathsf{T}(\theta\psi + (1-\theta)\nabla\varphi) - c\varphi =: \mathscr{M}_\theta(\varphi, \psi). \tag{10}$$

The parameter θ is at the user's disposal, but the most useful values are 0, 1/2 and 1. We introduce the following quadratic functional of $(\varphi, \psi) \in \mathscr{V}$

$$E_\theta(\varphi, \psi) := \|\nabla\varphi - \psi\|^2_{\mathrm{L}_2(\Omega)} + \|\nabla\times\psi\|^2_{\mathrm{L}_2(\Omega)} + \|\mathscr{M}_\theta(\varphi, \psi) - f\|^2_{\mathrm{L}_2(\Omega)} \tag{11}$$

where $\nabla\times\psi$ denotes curl (rotation) of ψ, and then consider the convex minimization problem of finding

$$(u, g) = \underset{(\varphi,\psi)\in\mathscr{V}}{\arg\min} E_\theta(\varphi, \psi). \tag{12}$$

Remark 1 (Equivalent Problems) The problem of finding strong solution to (2) and convex minimization problem (12) are equivalent and in (12), $g = \nabla u$ holds. Thus, in the rest of the paper, g is equal to ∇u.

The Euler–Lagrange equation of the minimization problem (12) consists in finding $(u, g) \in \mathscr{V}$ such that

$$\langle\nabla u - g, \nabla\varphi - \psi\rangle + \langle\nabla\times g, \nabla\times\psi\rangle + \langle\mathscr{M}_\theta(u, g), \mathscr{M}_\theta(\varphi, \psi)\rangle$$
$$= \langle f, \mathscr{M}_\theta(\varphi, \psi)\rangle \quad \text{for each } (\varphi, \psi) \in \mathscr{V}. \tag{13}$$

We introduce the symmetric bilinear form $a_\theta : \mathscr{Y}^2 \to \mathbb{R}$ via

$$a_\theta(\varphi, \psi; \varphi', \psi') := \left\langle \nabla\varphi - \psi, \nabla\varphi' - \psi' \right\rangle + \left\langle \nabla\times\psi, \nabla\times\psi' \right\rangle + \left\langle \mathscr{M}_\theta(\varphi, \psi), \mathscr{M}_\theta(\varphi', \psi') \right\rangle. \tag{14}$$

Theorem 1 (Coercivity and Continuity) *Let Ω be a bounded convex open subset of $\mathbb{R}^d$ and the uniformly bounded coefficients A, b, c satisfy either (3) with $\lambda > 0$ or (4) with $b \equiv 0$ and $c \equiv 0$. Then a_θ on $\mathscr{V}$ is coercive and continuous, there exist $C_{15}, C_{16} > 0$ such that*

$$a_\theta(\varphi, \psi; \varphi, \psi) \geqslant C_{15} \|(\varphi, \psi)\|^2_{\mathscr{Y}} \text{ for each } (\varphi, \psi) \in \mathscr{V}, \tag{15}$$

$$a_\theta\left(\varphi, \psi; \varphi', \psi'\right) \leqslant C_{16} \|(\varphi, \psi)\|_{\mathscr{Y}} \left\|(\varphi', \psi')\right\|_{\mathscr{Y}} \text{ for each } (\varphi, \psi), (\varphi', \psi') \in \mathscr{V}. \tag{16}$$

Theorem 1 ensures the well-posedness of the problem (13) through the Lax-Milgram setting.

Definition 1 (A Least Squares Finite Element Method) Let $\mathfrak{T}$ be a collection of conforming shape-regular triangulations on Ω which also known as meshes. If the domain, Ω, is a polyhedral then it coincides with the interior area of the mesh. Otherwise, if the domain includes curved boundary, the coincidence is lost. Hence this leads to have simplices with curved sides and isoparametric elements. For each element $K \in \mathscr{T} \in \mathfrak{T}$, denote $h_K := \operatorname{diam} K$, and $h := h_{\mathscr{T}} := \max_{K\in\mathscr{T}} h_K$. Now, consider the following Galerkin finite element spaces

$$\mathbb{U} := \mathbb{P}^k(\mathscr{T}) \cap \mathrm{H}^1_0(\Omega), \quad \mathbb{G} := \mathbb{P}^k(\mathscr{T}; \mathbb{R}^d) \cap \mathscr{W} \subseteq \mathrm{H}^1(\Omega; \mathbb{R}^d). \tag{17}$$

Corresponding to these spaces, the discrete problem corresponding to (13) turns to finding $(\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}}) \in \mathbb{U} \times \mathbb{G}$ such that

$$a_\theta(\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}}; \varphi, \psi) = \langle f, \mathscr{M}_\theta(\varphi, \psi)\rangle \text{ for each } (\varphi, \psi) \in \mathbb{U} \times \mathbb{G}. \tag{18}$$

The coercivity is inherited to subspaces, therefore the solution of discrete problem (18) is also well-posed. The discrete problem (18) leads to an approximate solution satisfying the following error estimate theorems.

Remark 2 (Implementing the Boundary Conditions) Since imposing zero-tangential trace condition to the finite element spaces is not trivial. In the implementation we used in Sect. 4 we replace in (18) the space $\mathbb{G} := \mathbb{P}^k\left(\mathscr{T}; \mathbb{R}^d\right) \cap \mathscr{W} \subseteq \mathrm{H}^1(\Omega; \mathbb{R}^d)$ with the larger space $\tilde{\mathbb{G}} := \mathbb{P}^k\left(\mathscr{T}; \mathbb{R}^d\right) \cap \mathrm{H}^1(\Omega; \mathbb{R}^d)$.

Theorem 2 (A Priori Error Estimate) *Let $\mathscr{T} \in \mathfrak{T}$ be a mesh on the polyhedral domain $\Omega \subseteq \mathbb{R}^d$. Moreover assume that the strong solution u of (2) satisfies $u \in \mathrm{H}^{\beta+2}(\Omega)$, for some real $\beta > 0$. Let $(\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}}) \in \mathbb{U} \times \mathbb{G}$ be the finite element solution of (18) on the mesh $\mathscr{T}$. Then for some $C_{19} > 0$, independent of u and h we have*

$$\|(u, \nabla u) - (\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}})\|_{\mathscr{Y}} \leqslant C_{19} h^{\min\{k,\beta\}} \|u\|_{\mathrm{H}^{k+2}(\Omega)}. \tag{19}$$

Remark 3 (Curved Domain) In the case that Ω has a curved boundary we use isoparametric finite element. A piecewise smooth domain guarantees an optimal rate error bound using isoparametric finite element similarly to Theorem 2 [2].

Theorem 3 (Error-Residual a Posteriori Estimates) *Let* $(\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}})$ *is the unique solution of the discrete problem (18).*

(i) The following a posteriori residual upper bound holds

$$\|(u, \nabla u) - (\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}})\|_{\mathscr{Y}}^2 \leqslant C_{15}^{-1} \left(\|\nabla \mathsf{u}_{\mathbb{U}} - \mathsf{g}_{\mathbb{G}}\|_{\mathrm{L}_2(\Omega)}^2 + \|\nabla \times \mathsf{g}_{\mathbb{G}}\|_{\mathrm{L}_2(\Omega)}^2 + \|\mathscr{M}_\theta(\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}}) - f\|_{\mathrm{L}_2(\Omega)}^2 \right).$$

(ii) For any open subdomain $\omega \subseteq \Omega$ *we have*

$$\|\nabla \mathsf{u}_{\mathbb{U}} - \mathsf{g}_{\mathbb{G}}\|_{\mathrm{L}_2(\omega)}^2 + \|\nabla \times \mathsf{g}_{\mathbb{G}}\|_{\mathrm{L}_2(\omega)}^2 + \|\mathscr{M}_\theta(\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}}) - f\|_{\mathrm{L}_2(\omega)}^2 \leqslant C_{16,\omega} \left(\|u - \mathsf{u}_{\mathbb{U}}\|_{\mathrm{H}^1(\omega)}^2 + \|\nabla u - \mathsf{g}_{\mathbb{G}}\|_{\mathrm{H}^1(\omega)}^2 \right), \tag{20}$$

where $C_{16,\omega}$ *is the continuity constant of* a_θ *restricted to* $\omega \subseteq \Omega$*.*

3 Linearization of Fully Nonlinear Problems

In this section, we present the Newton differentiability concept to operators, which can even include non-smooth operators. This concept is useful to extend the standard Newton linearization to the problems with non-smooth operator. We state the convergence analysis of a linearization method which is based on this concept. We then discuss linearization of two specific fully nonlinear PDEs, namely Monge–Ampère and Hamilton–Jacobi–Bellman equations that lead to a sequence of linear equations in nondivergence form. We refer the reader to [1] or [6] for details on such equations.

Definition 2 (Newton Differentiable Operator, Ito and Kunisch [5]) Let $\mathscr{X}$ and $\mathscr{Z}$ be Banach spaces and let $\mathscr{U}$ be a non-empty open subset of $\mathscr{X}$. An operator $\mathscr{F} : \mathscr{U} \subset \mathscr{X} \to \mathscr{Z}$ is called *Newton differentiable* at $x \in \mathscr{U}$ if there exists a set-valued map with non-empty images $\mathfrak{D}\mathscr{F} : \mathscr{U} \rightrightarrows \mathrm{Lin}(\mathscr{X} \to \mathscr{Z})$ (where the double arrow signifies values in the power set of the right-hand side) such that

$$\lim_{\|e\|_{\mathscr{X}} \to 0} \frac{1}{\|e\|_{\mathscr{X}}} \sup_{\mathscr{D} \in \mathfrak{D}\mathscr{F}[x]} \|\mathscr{F}[x+e] - \mathscr{F}[x] - \mathscr{D}e\|_{\mathscr{Z}} = 0 \text{ for each } x \in \mathscr{U}. \tag{21}$$

The nonlinear operator $\mathscr{F}$ is called *Newton differentiable on* $\mathscr{U}$ with Newton derivative $\mathfrak{D}\mathscr{F}$ if $\mathscr{F}$ is Newton differentiable at x, for every $x \in \mathscr{U}$.

The set-valued map $\mathfrak{D}\,\mathscr{F}[x]$ is single-valued at x if and only if $\mathscr{F}$ is Fréchet differentiable and $\mathfrak{D}\,\mathscr{F}[x] = \{\mathrm{D}\,\mathscr{F}[x]\}$.

Theorem 4 (Superlinear Convergence) *Suppose that a nonlinear operator $\mathscr{F}$ is Newton differentiable in an open neighborhood $\mathscr{U}$ of x^*, solution of $\mathscr{F}[x] = 0$. If for any $x \in U$, the all $D \in \mathfrak{D}\,\mathscr{F}[x]$ are non-singular and $\left\| D^{-1} \right\|$ are bounded, then the Newton iteration*

$$x_{n+1} = x_n - D_n^{-1}\mathscr{F}[x_n], \quad D_n \in \mathfrak{D}\,\mathscr{F}[x_n] \tag{22}$$

converges superlinearly to x^* provided that x_0 is sufficiently close to x^*.

Definition 3 (The Monge–Ampère Equation) Let $\Omega \subseteq \mathbb{R}^2$ be a bounded convex domain. Consider the Monge–Ampère (MA) equation with Dirichlet boundary condition

$$\det \mathrm{D}^2 u = f \text{ in } \Omega,\ u|_{\partial\Omega} = 0 \text{ and } u \text{ is strictly convex in } \Omega, \tag{23}$$

where $f \in \mathrm{L}_2(\Omega)$, $f > 0$. Let $\mathscr{K} := \left\{ v \in \mathrm{H}^2(\Omega) \cap \mathrm{H}^1_0(\Omega) : \ v \text{ is strictly convex} \right\}$ and define the operators $\mathscr{M} : \mathscr{K} \to \mathrm{L}_2(\Omega)$ by

$$\mathscr{M}[v] := \det \mathrm{D}^2 v - f \tag{24}$$

and $\mathfrak{D}\,\mathscr{M} : \mathscr{K} \to \operatorname{Lin}\left(\mathrm{H}^2(\Omega) \cap \mathrm{H}^1_0(\Omega) \to \mathrm{L}_2(\Omega)\right)$ by

$$\mathfrak{D}\,\mathscr{M}[v] := \operatorname{Cof} \mathrm{D}^2 v : \mathrm{D}^2 . \tag{25}$$

Theorem 5 (Superlinear Convergence of Iterative Method to MA Equation) *The operator $\mathscr{M}$ is Fréchet differentiable and thus Newton differentiable. Moreover, if the initial guess $u_0 \in \mathscr{K}$ is close to the exact solution $u \in \mathrm{H}^2(\Omega) \cap \mathrm{H}^1_0(\Omega)$ of (23), then the recursive problem*

$$\operatorname{Cof} \mathrm{D}^2 u_n : \mathrm{D}^2 u_{n+1} = f - \det \mathrm{D}^2 u_n + \operatorname{Cof} \mathrm{D}^2 u_n : \mathrm{D}^2 u_n \ \text{ in } \Omega, \text{ and } \ u_{n+1}|_{\partial\Omega} = 0 \tag{26}$$

converges with superlinear rate to u.

Definition 4 (Hamilton–Jacobi–Bellman Equation) Let Ω be a bounded convex domain in $\mathbb{R}^d$, $d \in \mathbb{N}$ (typically $d = 2, 3$). Consider the Hamilton–Jacobi–Bellman (HJB) equation with Dirichlet boundary condition

$$\sup_{\alpha \in \mathscr{A}} \left(A^\alpha : \mathrm{D}^2 u + b^{\alpha\intercal} \nabla u - c^\alpha u - f^\alpha \right) = 0 \ \text{ in } \Omega \text{ and } \ u|_{\partial\Omega} = 0 \tag{27}$$

where $\mathscr{A}$ is a compact metric space, $A \in \mathrm{L}_\infty(\Omega; \mathrm{C}^0(\mathscr{A}; \operatorname{Sym}(\mathbb{R}^d)))$, $b \in \mathrm{L}_\infty(\Omega; \mathrm{C}^0(\mathscr{A}; \mathbb{R}^d))$, $c \in \mathrm{L}_\infty(\Omega; \mathrm{C}^0(\mathscr{A}))$ and $f \in \mathrm{L}_2(\Omega; \mathrm{C}^0(\mathscr{A}))$. We suppose $A^\alpha(x)$ is uniformly elliptic in both x and α and together with b^α, c^α meets, for some $\epsilon \in (0, 1)$ the Cordes condition (3), or (4) if $b^\alpha \equiv 0$, $c^\alpha \equiv 0$, independent of $\alpha \in \mathscr{A}$. For each $\alpha \in \mathscr{A}$, define the linear operator

$$\mathscr{L}^\alpha v := A^\alpha : \mathrm{D}^2 v + b^{\alpha\intercal} \nabla v - c^\alpha v, \tag{28}$$

the following set of $\mathscr{A}$-index-valued maps:

$$\mathscr{Q} := \{q : \Omega \to \mathscr{A} \mid q \text{ is measurable}\}\ , \tag{29}$$

and the set-valued map $\mathscr{N}$, for $v \in \mathrm{H}^2(\Omega) \cap \mathrm{H}^1_0(\Omega)$, such that

$$\mathscr{N}[v] := \left\{q \in \mathscr{Q} :\ q(x) \in \mathrm{Argmax}_{\alpha\in\mathscr{A}} \left(\left[\mathscr{L}^\alpha v - f^\alpha\right] x\right) \text{ for almost all } x \text{ in } \Omega\right\}. \tag{30}$$

Now, we define the HJB operator by

$$\mathscr{B}[v] := \sup_{q\in\mathscr{Q}} \mathscr{L}^q v - f^q, \tag{31}$$

and the set-valued map $\mathfrak{D}\,\mathscr{B} : \mathrm{H}^2(\Omega) \cap \mathrm{H}^1_0(\Omega) \rightrightarrows \mathrm{Lin}\left(\mathrm{H}^2(\Omega) \cap \mathrm{H}^1_0(\Omega) \to \mathrm{L}_2(\Omega)\right)$ by

$$\mathfrak{D}\,\mathscr{B}[v] := \Big\{\mathscr{L}^q := (A^q : \mathrm{D}^2 + b^{q\intercal}\nabla - c^q) \mid q \in \mathscr{N}[v]\}\ . \tag{32}$$

Theorem 6 (Superlinear Convergence of Iterative Method to HJB Equation) *The operator $\mathscr{B}$ is Newton differentiable with Newton derivative $\mathfrak{D}\,\mathscr{B}$. Moreover, if the initial guess u_0 is close to the exact solution $u \in \mathrm{H}^2(\Omega) \cap \mathrm{H}^1_0(\Omega)$ of (27), the recursive problem*

$$\mathscr{L}^{q_n} u_{n+1} = f^{q_n} \ \textit{in}\ \Omega,\ \textit{and}\ \ u_{n+1}|_{\partial\Omega} = 0 \tag{33}$$

where $q_n \in \mathscr{N}[u_n]$, converges with superlinear rate to u.

To follow (26) and (33), we need to approximate a linear problem in nondivergence form in each iteration, which we apply the method discussed in Sect. 2. The convergence of the iterative methods (26) and (33) implies that the finite element approximation $(\mathsf{u}_\mathbb{U}, \mathsf{g}_\mathbb{G}) \in \mathbb{U} \times \mathbb{G}$ achieved via the recursive problems also satisfies the error bound of Theorems 2 and 3.

Remark 4 The a posteriori residual bound of Theorem 3 can be used as an explicit error indicator to determine a locally refined mesh in the adaptive scheme.

4 Numerical Experiments

We discuss two numerical tests one for each of Monge–Ampère via Newton and Hamilton–Jacobi–Bellman problems that demonstrate the robustness of our method to the fully nonlinear problems. For both test problems, the domain, Ω, is taken to be the unit disk in $\mathbb{R}^2$ with center at the origin. The criterion to stop the iteration is either $\|(\mathsf{u}_{n+1}, \mathsf{g}_{n+1}) - (\mathsf{u}_n, \mathsf{g}_n)\|_{\mathscr{Y}} < 10^{-8}$ or maximum 8 iterations. In implementation, we take the parameter θ of (18) equal to 0.5. Both implementations were done by using FEniCS package.

In the first test problem, the known solution is considered smooth and we see that the numerical results which obtained on the uniform mesh confirm the convergence analysis of Theorem 2. In the second test problem, we choose the known solution near singular and test the performance of the adaptive scheme as mentioned in Remark 4. Through comparing the convergence rate by the adaptive with uniform refinement, we observe the efficiency of the adaptive scheme.

Problem 1 (Monge–Ampère Test) Consider problem (23) and choose f corresponding to the exact solution

$$u(x) = -\sqrt{R^2 - x_1^2 - x_2^2} + \sqrt{R^2 - 1}, \text{ for a fixed } R > 1. \tag{34}$$

As suggested by Lakkis and Pryer Lakkis and Pryer [9] the first iterate u_0 is the discretization of u_0 satisfying

$$\Delta u_0 = 2\sqrt{f} \text{ in } \Omega, \text{ and } u_0|_{\partial\Omega} = 0 \tag{35}$$

and then we track the recursive problem (26). We show various error norms of linear ($\mathbb{P}^1$) and quadratic ($\mathbb{P}^2$) finite element approximation for two values R in Figs. 1 and 2.

Problem 2 (Hamilton–Jacobi–Bellman Test) Consider problem (27) and let $\mathscr{A} = [0, 2\pi]$,

$$A^\alpha(x) = \begin{bmatrix} \cos(\alpha) & \sin(\alpha) \\ -\sin(\alpha) & \cos(\alpha) \end{bmatrix} \begin{bmatrix} 1 + (x_1^2 + x_2^2) & 0.005 \\ 0.005 & 1.01 - (x_1^2 + x_2^2) \end{bmatrix} \begin{bmatrix} \cos(\alpha) & -\sin(\alpha) \\ \sin(\alpha) & \cos(\alpha) \end{bmatrix}, \tag{36}$$

$$b^\alpha = 0, \quad c^\alpha = 2 - 0.5(\cos(2\alpha) + \sin(2\alpha)), \quad f^\alpha = \mathscr{L}^\alpha u - (1 - \cos(2\alpha - \pi(x_1 + x_2))), \tag{37}$$

with the exact solution

$$u(x) := \begin{cases} r(x)^{5/3}(1 - r(x))^{5/2} \sin(\varphi(x))^{5/2} & \text{if } 0 < r(x) \leqslant 1 \text{ and }, 0 < \varphi(x) < 3\pi/2, \\ 0 & \text{otherwise,} \end{cases} \tag{38}$$

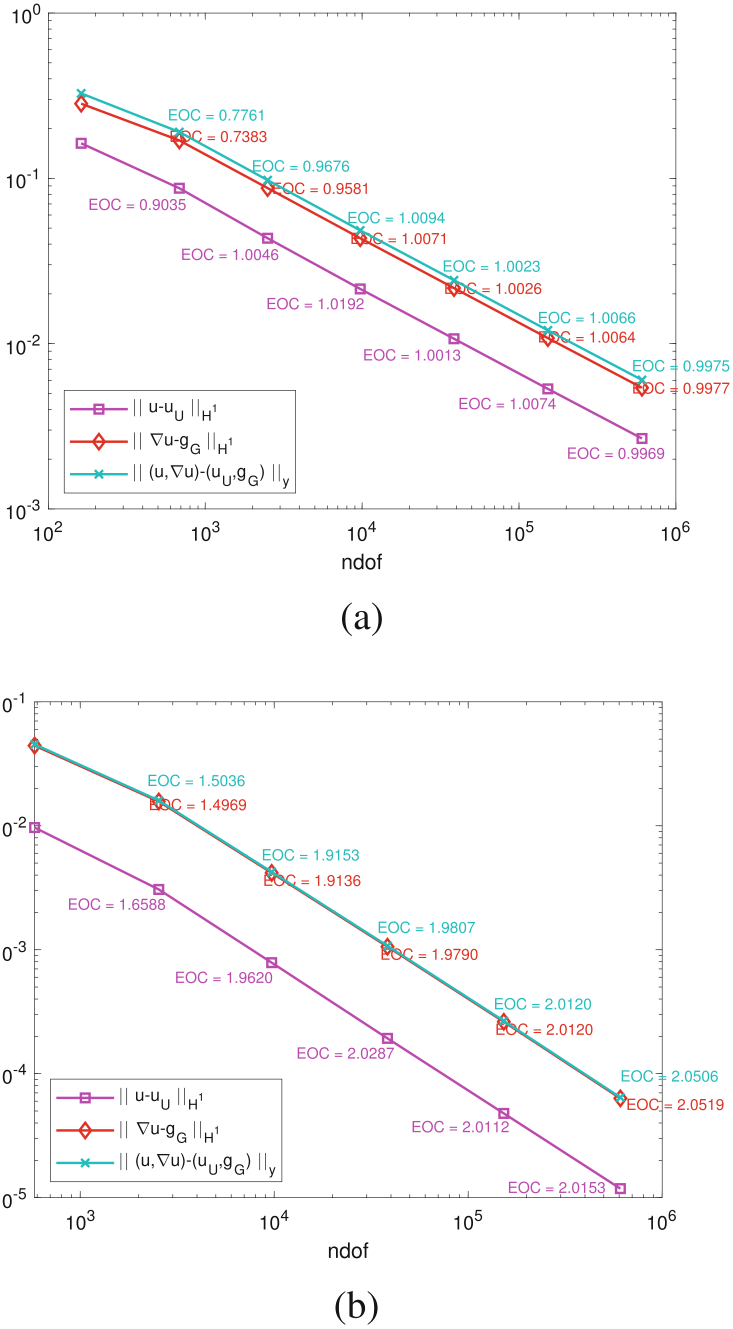


Fig. 1 Experimental order of convergence (EOC) for the Monge–Ampère test problem with $R = \sqrt{2}$. (**a**) $\mathbb{P}^1$ elements. (**b**) $\mathbb{P}^2$ elements

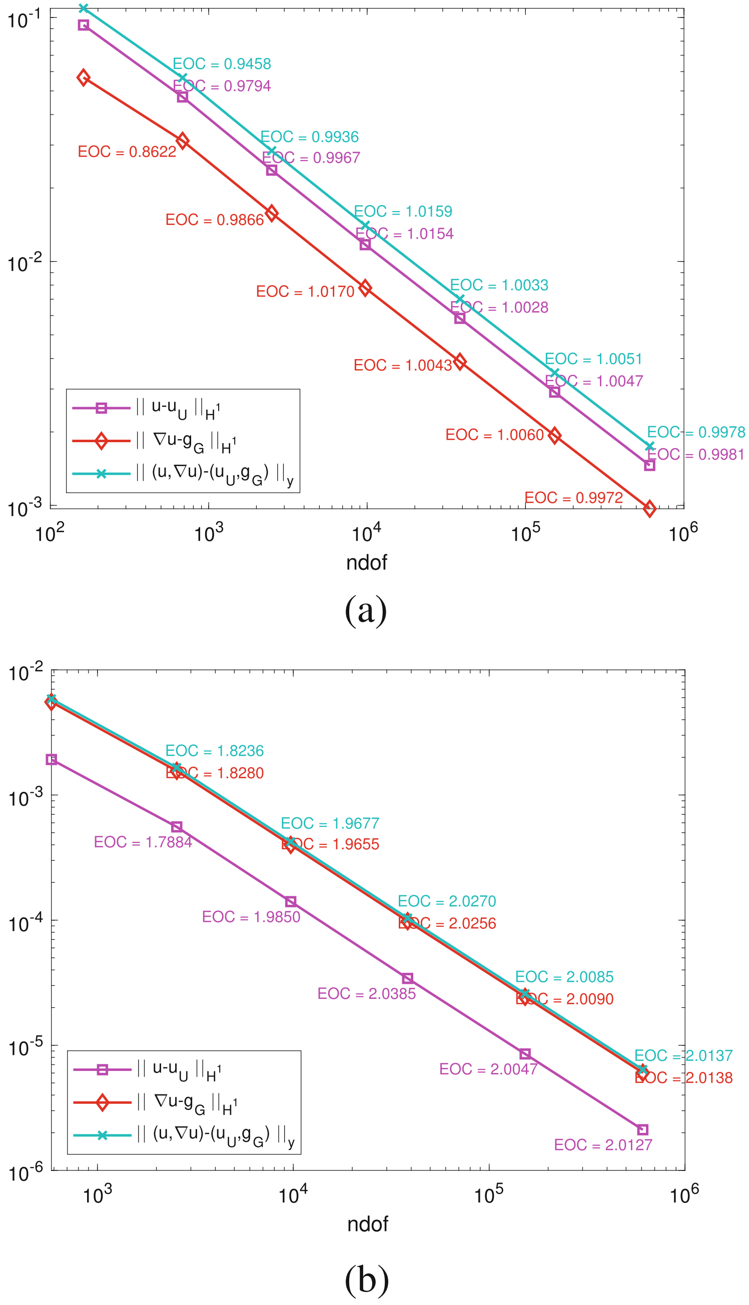
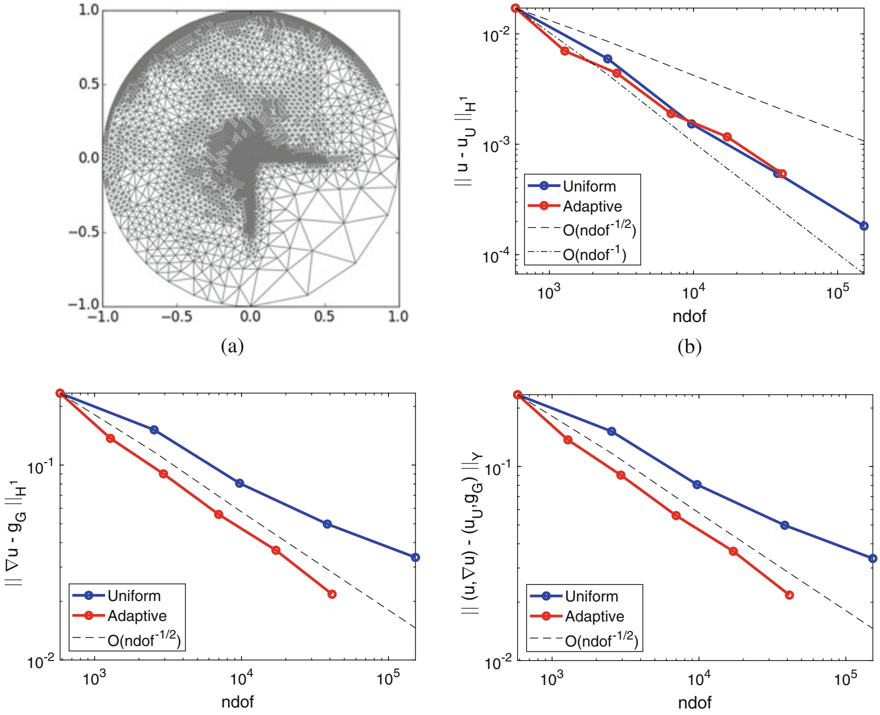


Fig. 2 Experimental order of convergence (EOC) for the Monge–Ampère test problem with $R = 2$. (**a**) $\mathbb{P}^1$ elements. (**b**) $\mathbb{P}^2$ elements



(a) (b)

(c) (d)

Fig. 3 Mesh in a and b–d show the convergence rate in both the uniform and adaptive refinement for the HJB test problem Sect. 2 with $\mathbb{P}^2$ elements. While the adaptive scheme does not yield any noticeable gain for the function value approximation ($\|u - \mathsf{u}_{\mathbb{U}}\|_{\mathrm{H}^1(\Omega)}$), it does so in the reconstructed gradient ($\|\nabla u - \mathsf{g}_{\mathbb{G}}\|_{\mathscr{Y}}$)

$(r(x), \varphi(x))$ are polar coordinates centered in the origin. One can check that the near degenerate diffusion A^α together with b^α and c^α satisfy the Cordes condition (3) with $\lambda = 1$ and $\varepsilon = 0.0032$. Note that $u \in \mathrm{H}^s$ for any $s < 8/3$. As $u \in \mathrm{H}^2(\Omega)$, we do not expect the advantage of the adaptive scheme over than the uniform refinement for $\mathrm{H}^1(\Omega)$-norm of the error of $\mathsf{u}_{\mathbb{U}}$; it is shown in Fig. 3b. But since ∇u does not have such smoothness, we observe the superiority of the adaptive scheme for $\mathrm{H}^1(\Omega)$-norm of the error of $\mathsf{g}_{\mathbb{G}}$ (and $\mathscr{Y}$-norm of the error of $(\mathsf{u}_{\mathbb{U}}, \mathsf{g}_{\mathbb{G}})$) in Fig. 3c and d.

References

1. Luis A. Caffarelli and Xavier Cabré. *Fully nonlinear elliptic equations*, volume 43 of *American Mathematical Society Colloquium Publications*. American Mathematical Society, Providence, RI, 1995. ISBN 0-8218-0437-5. http://www.worldcat.org/oclc/246542992.

2. Philippe G. Ciarlet. *Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 2002. ISBN 0898715148.
3. Xiaobing Feng and Max Jensen. Convergent semi-Lagrangian methods for the Monge-Ampère equation on unstructured grids. *SIAM Journal on Numerical Analysis*, 55(2):691–712, 2017. ISSN 0036-1429. https://doi.org/10.1137/16M1061709. https://epubs.siam.org/doi/10.1137/16M1061709.
4. Dietmar Gallistl and Endre Süli. Mixed Finite Element Approximation of the Hamilton–Jacobi–Bellman Equation with Cordes Coefficients. *SIAM Journal on Numerical Analysis*, 57(2):592–614, 01 2019. ISSN 0036-1429. https://doi.org/10.1137/18M1192299. https://epubs.siam.org/doi/abs/10.1137/18M1192299.
5. Kazufumi Ito and Karl Kunisch. *Lagrange multiplier approach to variational problems and applications*. SIAM, Philadelphia, 2008. ISBN 978-0-89871-649-8. http://www.worldcat.org/oclc/884103565. OCLC: 884103565.
6. N. V. Krylov. *Sobolev and viscosity solutions for fully nonlinear elliptic and parabolic equations*, volume 233 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2018. ISBN 978-1-4704-4740-3. http://www.worldcat.org/oclc/1039672482.
7. Omar Lakkis and Amireh Mousavi. A least-squares Galerkin approach to gradient and Hessian recovery for nondivergence-form elliptic equations. online preprint (under peer-review) 1909.00491, arXiv, 09 2019. https://arxiv.org/abs/1909.00491v1.
8. Omar Lakkis and Amireh Mousavi. A least-squares galerkin approach to gradient recovery for Hamilton-Jacobi-Bellman equations with cordes coefficients. in preparation, 2020.
9. Omar Lakkis and Tristan Pryer. A finite element method for nonlinear elliptic problems. *SIAM Journal on Scientific Computing*, 35(4):A2025–A2045, 2013. https://doi.org/10.1137/120887655. http://arxiv.org/abs/1103.2970.
10. Iain Smears and Endre Süli. Discontinuous Galerkin finite element approximation of Hamilton-Jacobi-Bellman equations with Cordes coefficients. *SIAM J. Numer. Anal.*, 52(2):993–1016, 2014. ISSN 0036-1429. https://doi.org/10.1137/130909536. https://epubs.siam.org/doi/10.1137/130909536.
11. Iain Smears and Endre Süli. Discontinuous Galerkin finite element methods for time-dependent Hamilton–Jacobi–Bellman equations with Cordes coefficients. *Numerische Mathematik*, 133(1):141–176, May 2016. ISSN 0029-599X, 0945-3245. https://doi.org/10.1007/s00211-015-0741-6. http://arxiv.org/abs/1406.4839. arXiv: 1406.4839.

A Posteriori Model Error Analysis of 3D-1D Coupled PDEs

Federica Laurino, Stefano Brambilla, and Paolo Zunino

Abstract The objective of this work is to extend the model reduction technique for coupled 3D-1D elliptic PDEs, previously proposed by the authors, with an a posteriori analysis of the model error, defined as the difference between the solutions of the reference and reduced problem. More precisely, we introduce an estimator for a user-defined functional of the error, computed using a duality approach. This result is particularly useful since it allows to localize the model error on the computational mesh and to investigate the reliability of the model reduction approach.

1 Introduction

Model reduction techniques have been extensively analyzed and applied in many fields. For instance, they result very useful in case of PDEs defined in domains with small inclusions because the scale separation between the domains can be exploited in order to reduce the computational cost. When the inclusions are shaped as slender bodies the model describing the action of the inclusions can be transformed in a simpler 1D problem. In particular, we consider the 3D-1D model reduction approach addressed recently in [2–7] and in particular the formulation proposed in [7] and we perform an a posteriori analysis of the model error. The insight provided by the analysis is twofold: we understand how well the reduced problem represents the original one and we localize the error on the computational mesh.

F. Laurino (✉) · P. Zunino
Politecnico di Milano, Milano, Italy
e-mail: federica.laurino@polimi.it; paolo.zunino@polimi.it

S. Brambilla
MOXOFF, Milano, Italy
e-mail: stefano.brambilla@moxoff.com

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_65

2 Problem Setting

We consider the prototypal problem based on Robin-Neumann coupling conditions addressed in [7]. The domain is denoted by Ω and composed by two parts, Σ and $\Omega_\oplus := \Omega \setminus \overline{\Sigma}$. We assume Ω is convex and Σ (the interior domain) is *completely embedded* into Ω, such that the distance between $\partial\Omega$ and $\partial\Sigma$ is strictly positive. Let Σ be a *generalized cylinder*, that is the swept volume of a two dimensional set moved along a curve in the three-dimensional space. We denote by Γ the lateral boundary of the cylinder and $\Lambda = \{\Lambda(s),\ s \in (0, S)\}$ the centerline. Moreover, $\mathcal{D}(s)$ is the cross section and $\partial\mathcal{D}(s)$ its boundary. Let $|\cdot|$ denote the Lebesgue measure of a set. We assume that Σ has *top* and *bottom* boundaries, namely $|\mathcal{D}(0)|,\ |\mathcal{D}(S)| > 0$, and they are denoted by Γ_0 and Γ_S respectively. We refer to [7] for more details. The problem consists to find $u_\oplus, u_\ominus$ (where $\oplus, \ominus$ denote the exterior and the interior of Σ, respectively) such as:

$$-\Delta u_\oplus = f \qquad \text{in } \Omega_\oplus, \tag{1a}$$

$$-\Delta u_\ominus = g \qquad \text{in } \Sigma, \tag{1b}$$

$$-\nabla u_\oplus \cdot \boldsymbol{n}_\oplus = \kappa\,(u_\oplus - u_\ominus) \qquad \text{on } \Gamma, \tag{1c}$$

$$-\nabla u_\ominus \cdot \boldsymbol{n}_\ominus = \kappa\,(u_\ominus - u_\oplus) \qquad \text{on } \Gamma, \tag{1d}$$

$$-\nabla u_\oplus \cdot \boldsymbol{n}_\oplus = 0 \qquad \text{on } \Gamma_0 \cup \Gamma_S, \tag{1e}$$

$$-\nabla u_\ominus \cdot \boldsymbol{n}_\ominus = 0 \qquad \text{on } \Gamma_0 \cup \Gamma_S, \tag{1f}$$

$$u_\oplus = 0 \qquad \text{on } \partial\Omega\,. \tag{1g}$$

It is assumed that the interface of Σ is permeable, namely it is crossed by a normal flux proportional to $\kappa\,(u_\oplus - u_\ominus)$. The coefficient κ plays the role of *permeability* or *transfer coefficient* and it is uniform on each cross section $\partial\mathcal{D}(s)$. As a result of that, κ is only a (regular) function of the arc-length s. In [7], a topological model reduction technique, based on averaging, is applied to (1) in order to transform the problem on Σ into a simpler one. In particular, the domain Σ is shrunk to its centerline Λ and the corresponding partial differential equation is averaged on the cylinder cross section, namely $\mathcal{D}$. This new problem setting is called the *reduced* problem.

2.1 Reduced 3D-1D Coupled Problem

The averaging technique presented in [7] is based on four fundamental assumptions, described below.

A0 The transversal diameter ε of Σ is small compared to the diameter of Ω.

A1 The function $u_\ominus$ has a *uniform profile* on each cross section $\mathcal{D}(s)$, namely $u_\ominus(r, s, t) = U(s)$.

A2 The domain $\Omega_\oplus$ can be identified with the entire Ω, and we correspondingly omit the subscript $\oplus$ to the functions defined on $\Omega_\oplus$.

We denote by $\overline{\overline{(\cdot)}}$ and $\overline{(\cdot)}$ the average operators computed on $\mathcal{D}$ and $\partial\mathcal{D}$ respectively. Namely, for any sufficiently regular function w, $\overline{\overline{w}}(s) = |\mathcal{D}(s)|^{-1}\int_{\mathcal{D}(s)} w d\sigma$ and $\overline{w}(s) = |\partial\mathcal{D}(s)|^{-1}\int_{\partial\mathcal{D}(s)} w d\gamma$. We decompose the solution and the test functions on every cross section $\partial\mathcal{D}(s)$ as their average plus some fluctuations, namely $u_\oplus = \overline{u}_\oplus + \tilde{u}_\oplus$, $u_\ominus = \overline{u}_\ominus + \tilde{u}_\ominus$, $v = \overline{v} + \tilde{v}$, where $\overline{\tilde{u}}_\oplus = \overline{\tilde{u}}_\ominus = \overline{\tilde{v}} = 0$, and

A3 the energy of fluctuations is small, i.e. $\int_{\partial\mathcal{D}(s)} \tilde{u}_* \tilde{v} d\gamma \simeq 0$.

Applying the assumptions **A0**–**A3** to (1), we obtain the 3D-1D coupled problem that is to find $u \in H_0^1(\Omega)$ and $U \in H^1(\Lambda)$ such that for any $v \in H_0^1(\Omega)$, $V \in H^1(\Lambda)$

$$(\nabla u, \nabla v)_\Omega + (\kappa\overline{u}, \overline{v})_{\Lambda,|\partial\mathcal{D}|} = (\kappa U, \overline{v})_{\Lambda,|\partial\mathcal{D}|} + (f, v)_\Omega, \tag{2a}$$

$$(d_s U, d_s V)_{\Lambda,|\mathcal{D}|} + (\kappa U, V)_{\Lambda,|\partial\mathcal{D}|} = (\kappa\overline{u}_\oplus, V)_{\Lambda,|\partial\mathcal{D}|} + (\overline{\overline{g}}, V)_{\Lambda,|\mathcal{D}|}, \tag{2b}$$

where $(\cdot,\cdot)_X$ denotes the standard L^2 scalar product in X and $(\cdot,\cdot)_{X,|w|} = (|w|\cdot,\cdot)_X$.

3 A-posteriori Model Error Analysis

In [7] an a-priori analysis of the model error is performed. More precisely, the model error, e, is defined as the difference of the reference and reduced solutions, namely $e = (u_\oplus + u_\ominus) - (u + U)$. The error is split in three components $e^{(i)}$, $i = 1, 2, 3$, representing the errors arising from assumptions $\mathbf{A}\boldsymbol{i}$, $i = 1, 2, 3$. It can be proved that each $e^{(i)}$ goes to zero with ε. Therefore, the smaller is the radius of the inclusion, the better the reduced problem approximates the original one. The proof is based on the theory developed in [1]. In particular, with the aim of analyzing the error $e^{(i)}$ arising from the application of assumption $\mathbf{A}\boldsymbol{i}$, the related reference and reduced problems are introduced,

$$\text{find } u_{\text{ref}}^{(i)} \in X^{(i)} : a_{\text{ref}}^{(i)}(u_{\text{ref}}^{(i)}, v) = \mathcal{F}_{\text{ref}}^{(i)}(v),\ \forall v \in X^{(i)}, \tag{3}$$

$$\text{find } u^{(i)} \in X^{(i)} : a^{(i)}(u, v) = \mathcal{F}^{(i)}(v),\ \forall v \in X^{(i)}. \tag{4}$$

For the sake of simplicity, from now on we omit the superscript (i) where there is no ambiguity of notation. However, all the quantities must be interpreted as referred to the i-th component of the error. The bilinear form a_{ref} and the functional $\mathcal{F}_{ref}$ can be expressed as a modification of a and $\mathcal{F}$ as follows

$$a_{\text{ref}}(u, v) = a(u, v) + d(u, v),\ \forall u, v \in X, \qquad \mathcal{F}_{\text{ref}}(v) = \mathcal{F}(v) + l(v),\ \forall v \in X$$

where $d(u, v)$ and $l(v)$ can be seen as perturbation operators, related to the small parameter ϵ. Let $j(\cdot) : X \to \mathbb{R}$ be a linear output functional. In order to estimate the modeling error measured by $j(e)$, the reference dual problem

$$\text{find } z_{\text{ref}} \in X : a_{\text{ref}}(v, z_{\text{ref}}) = j(v), \ \forall v \in X \,, \tag{5}$$

is introduced and the error output functional is represented as follows,

$$j(e) = l(z_{\text{ref}}) - d(u, z_{\text{ref}}) \,. \tag{6}$$

Using (6), it can be proved that the error e vanishes for infinitesimal ε, see [7].

3.1 Localization of the Model Error

Starting from the definitions of the difference operators $d^{(i)}(\cdot, \cdot)$ and $l^{(i)}(\cdot)$, $i = 1, 2, 3$, in [7] the error output functional $j(e)$ is computed, through the error representation formula (6). However, this requires solving the dual reference problem. On one side it is less expensive than the primal one since it is uncoupled, but on the other side it can be still demanding from a computational point of view. Therefore, when the radius ε is small enough we aim to replace the reference dual solution with the reduced one, using an *approximated representation formula*.

Approximated Representation Formula

For the sake of simplicity, let us neglect the role of $l(\cdot)$ in (6) as the following considerations will apply similarly when $l(\cdot) \neq 0$. Our goal is to separate the contribution of the *representation formula* (6) depending on the reference solutions u_{ref} and z_{ref} from the contribution depending on the reduced solutions u and z. To this purpose, we follow the general theory developed in [1] that is briefly summarized here for the sake of clarity. Let z be the solution of the following dual reduced problem

$$a(v, z) = j(v), \ \forall v \in X. \tag{7}$$

The functional $j(\cdot)$ applied to the error e can be written as:

$$j\left(e\right) = -d\left(u, z_{ref}\right) = -d\left(u, z\right) - d\left(u, z_{ref} - z\right). \tag{8}$$

Let us assume that $d(\cdot, \cdot)$ is continuous; therefore, with $\|\cdot\|$ denoting the usual norms on X and $X \times X$, it holds $|d(u, z)| \leq \|d\|\|u\|\,\|z\|$. If the mapping $A : X \to X'$, $A(u) = a(u, \cdot)$ is bijective, the adjoint A^* is bijective too. Using the open mapping theorem, there exists a constant $\alpha > 0$ such that:

$$\|z\| \leq \alpha \sup_{v \in X} \frac{a\left(v, z\right)}{\|v\|}, \ \forall z \in X. \tag{9}$$

The last result we need is the *dual perturbed Galerkin orthogonality*: subtracting (7) from (5), we obtain $a\left(v, z_{ref} - z\right) = -d\left(v, z_{ref}\right)$, $\forall v \in X$. Combining the previous equality with (9), we deduce the *a-priori* estimate:

$$\left\|z_{ref} - z\right\| \leq \alpha \sup_{v \in X} \frac{a\left(v, z_{ref} - z\right)}{\|v\|} \leq \alpha \sup_{v \in X} \frac{d\left(v, z_{ref}\right)}{\|v\|} \leq \alpha \|d\| \left\|z_{ref}\right\|.$$

Finally, employing the continuity of $d\left(\cdot, \cdot\right)$, we find:

$$\left|d\left(u, z_{ref} - z\right)\right| \leq \|d\| \|u\| \left\|z_{ref} - z\right\| \leq \alpha \|d\|^2 \|u\| \left\|z_{ref}\right\|. \tag{10}$$

In conclusion, the representation (8) can be bounded estimated as:

$$|j\left(e\right)| \leq \|d\| \|u\| \|z\| + \alpha \|d\|^2 \|u\| \left\|z_{ref}\right\|. \tag{11}$$

Equation (11) achieves the goal to separate the contributions depending on the reduced and the reference solutions. More importantly, Eq. (11) shows that the model error is characterized by a first order term with respect to $\|d\|$, which depends on $\|z\|$, combined with a second order term on $\|d\|$ that depends on $\|z_{ref}\|$. Reminding that $\|d\|$ is the norm of the perturbation operator, such that $\|d\| \longrightarrow 0$ when ϵ vanishes, (11) shows that the model error can be reasonably estimated using u and z solely for ε sufficiently small. Repeating the same estimate for $l\left(\cdot\right)$, at first order in $\|d\|$ and $\|l\|$, the model error can be approximated as following using the *approximated representation formula*:

$$j\left(e\right) \approx l\left(z\right) - d\left(u, z\right). \tag{12}$$

Local Error Estimator

In what follows we introduce a local error estimator based on the approximated representation formula (12). Let T_h^{Ω} be a triangulation of Ω and $X_h \subset X$ be a suitable finite element space of dimension N. We denote with u_h and z_h the discrete solutions of the reduced problem (4) and the reduced dual problem (7), respectively. Considering the Lagrangian nodal basis $\{\varphi_k\} \subset X_h$, we define the vector of residuals, $\boldsymbol{\varrho} = \{\varrho_k\}_{k=1}^{N}$ as $\varrho_k = l\left(\varphi_k\right) - d\left(u_h\right)\left(\varphi_k\right)$ and the local weights $\tilde{\boldsymbol{\omega}} = \{\tilde{\omega}_k\}_{k=1}^{N}$ correspond to the degrees of freedom of the discrete dual reduced solution, namely $z_h = \sum_{k=1}^{N} \tilde{\omega}_k \varphi_k$. Let $\langle \cdot, \cdot \rangle$ be the Euclidean scalar product in $\mathbb{R}^N$. According to (12), the model error output functional $j\left(e\right)$ can be approximated by

$\langle \boldsymbol{\varrho}, \tilde{\boldsymbol{\omega}} \rangle$ and the components of the local error estimator $\tilde{\boldsymbol{\eta}} \in \mathbb{R}^N$ are $\tilde{\eta}_k = \varrho_k \tilde{\omega}_k$. The vector $\tilde{\boldsymbol{\eta}}$ can be represented on Ω as a finite element function $\tilde{\eta}_h = \sum_{k=1}^{N} \tilde{\eta}_k \varphi_k$. Each nodal component of $\tilde{\boldsymbol{\eta}}$ represents the localization of the approximated error on $\mathcal{T}^h_\Omega$.

Using this general approach we explicitly calculate the estimators $\tilde{\boldsymbol{\eta}}_1$, $\tilde{\boldsymbol{\eta}}_2$, $\tilde{\boldsymbol{\eta}}_3$ of the model errors $e^{(1)}$, $e^{(2)}$, $e^{(3)}$ related to the assumptions **A1**, **A2**, **A3**. To this aim, we must compute the residuals $\boldsymbol{\varrho}^{(i)}$ and the weights $\tilde{\boldsymbol{\omega}}^{(i)}$ for $i = 1, 2, 3$. Since the different components of the error are defined on different domains, in particular Σ and Ω, we should introduce different finite element spaces over Σ, $\Omega_\oplus$, Ω. However, for computational convenience, we define all the estimators using the basis functions φ_k of the finite element space $V_h^\Omega \subset H_0^1(\Omega)$, under the additional assumption that the mesh $\mathcal{T}^h_\Omega$ nodally conforms with the interface Γ. Let $N_h^\Omega = \dim V_h^\Omega$ be the degrees of freedom of such space. As a result, all the vectors of weights and residuals belong to $\mathbb{R}^{N_h^\Omega}$. We first build the residuals $\boldsymbol{\varrho}^{(i)}$ for $i = 1, 2, 3$, depending on the discrete reduced solutions u_h and U_h, computed by means of the discretization of (2). Following [7], we decompose the perturbation operators $d(\cdot, \cdot)$ and $l(\cdot)$ into their contributions related to the assumptions **A1**, **A2**, **A3**. For $k = 1, \ldots, N_h^\Omega$ the nodal components of the residuals are,

$$\varrho_k^{(1)} = l^{(1)}(\varphi_k) - d^{(1)}(U_h, \varphi_k) = ((\mathcal{I} - \overline{\overline{(\cdot)}})g, \varphi_k)_\Sigma + (\kappa(\mathcal{I} - \overline{(\cdot)})u_h, \varphi_k)_\Gamma, \tag{13a}$$

$$\varrho_k^{(2)} = l^{(2)}(\varphi_k) - d^{(2)}(u_h, \varphi_k) = (\nabla u_h, \nabla \varphi_k)_\Sigma - (f, \varphi_k)_\Sigma, \tag{13b}$$

$$\varrho_k^{(3)} = l^{(3)}(\varphi_k) - d^{(3)}(u_h, \varphi_k) = -\, (\kappa(\mathcal{I} - \overline{(\cdot)})u_h, \varphi_k)_\Gamma, \tag{13c}$$

Concerning the weights, we recall that in [7] they are computed using the solutions of the following reference dual problems, defined on Σ, $\Omega_\oplus$ and Ω respectively,

$$\text{find } z^{(1)}_{ref,h} \in V_h^\Sigma : (\nabla v_h, \nabla z^{(1)}_{ref,h})_\Sigma + (\kappa v_h, z^{(1)}_{ref,h})_\Gamma = j^{(1)}(v_h), \quad \forall v_h \in V_h^\Sigma$$

$$\text{find } z^{(2)}_{ref,h} \in V_h^{\Omega_\oplus} : (\nabla v_h, \nabla z^{(2)}_{ref,h})_{\Omega_\oplus} + (\kappa v_h, z^{(2)}_{ref,h})_\Gamma = j^{(2)}(v_h), \quad \forall v_h \in V_h^{\Omega_\oplus}$$

$$\text{find } z^{(3)}_{ref,h} \in V_h^\Omega : (\nabla v_h, \nabla z^{(3)}_{ref,h})_\Omega + (\kappa v_h, z^{(3)}_{ref,h})_\Gamma = j^{(3)}(v_h), \quad \forall v_h \in V_h^\Omega,$$

being $j^{(i)}$, $i = 1, 2, 3$ suitable output functionals. Conversely, we calculate the weights using the reduced dual solutions z_h and Z_h arising from the discretization of the reduced dual problem consisting to find $z_h \in V_h^\Omega$, $Z_h \in V_h^\Lambda$ such that

$$(\nabla v_h, \nabla z_h)_\Omega + (\kappa \overline{v}_h, \overline{z}_h)_{\Lambda, |\partial \mathcal{D}|} = j_\Omega^{red}(v_h), \qquad \forall v_h \in V_h^\Omega, \tag{14a}$$

$$(d_s V_h, d_s Z_h)_{\Lambda, |\mathcal{D}|} + (\kappa V_h, Z_h)_{\Lambda, |\partial \mathcal{D}|} = J_\Lambda^{red}(V_h), \qquad \forall V_h \in V_h^\Lambda, \tag{14b}$$

where the functionals $j_\Omega^{red}(\cdot) : H_0^1(\Omega) \to \mathbb{R}$ and $J_\Lambda^{red}(\cdot) : H_0^1(\Lambda) \to \mathbb{R}$ are

$$j_\Omega^{red}(v) = \int_\Omega v \, d\Omega, \qquad J_\Lambda^{red}(v) = \int_\Lambda v \, d\Lambda \tag{15}$$

More precisely, since the residual $\varrho^{(1)}$ corresponds to the model error on Λ, due to the assumption **A1**, we choose $\tilde{\omega}^{(1)}$ as the extension on Σ of the function Z_h. Such extension is called $\mathcal{E}_\Sigma Z_h$. For $\tilde{\omega}^{(2)}$ and $\tilde{\omega}^{(3)}$ we use z_h which is defined on the entire Ω because the corresponding model error is defined on $\Omega_\oplus$ and Ω respectively, as shown in full details in [7]. Therefore,

$$\tilde{\omega}^{(1)} = \begin{cases} \mathcal{E}_\Sigma Z_h & \text{in } \Sigma \\ 0 & \text{in } \Omega_\oplus \end{cases} \qquad \tilde{\omega}^{(2)} = z_h \quad \text{in } \Omega \qquad \tilde{\omega}^{(3)} = z_h \quad \text{in } \Omega. \tag{16}$$

Finally, we combine the weights and the residuals in order to compute the local estimators,

$$\tilde{\eta}^{(1)} = \sum_{k=1}^{N_h^\Omega} \tilde{\omega}_k^{(1)} \varrho_k^{(1)}, \qquad \tilde{\eta}^{(2)} = \sum_{k=1}^{N_h^\Omega} \tilde{\omega}_k^{(2)} \varrho_k^{(2)}, \qquad \tilde{\eta}^{(3)} = \sum_{k=1}^{N_h^\Omega} \tilde{\omega}_k^{(3)} \varrho_k^{(3)}. \tag{17}$$

4 Results

We solve the primal reduced coupled problem (2) on a segment Λ from $(-0.51, 0, 0)$ to $(0.51, 0, 0)$ completely embedded in the parallelepiped $\Omega = (-1, 1)^2 \times (-0.51, 0.51)$; the tessellation of Ω is a quasi-uniform regular mesh, with characteristic length $h = 1/32$. The other parameters are: $R = 0.25$, $k = 1$, $f = 1$ and $g = 1$. The discrete solutions of (2), u_h and U_h, are computed using piecewise linear finite elements. We calculate the residuals $\varrho^{(i)}$ by means of (13). We then solve the reduced dual problem (14), using as right hand sides the output functionals (15), and we compute the weights $\tilde{\omega}^{(i)}$ as in (16). In Fig. 1 we show the estimators given by (17) on a slice of the domain. As expected, the error is localized in Σ and at the interface Γ. We notice that from the definition of the residual $\varrho^{(1)}$ and the weight $\tilde{\omega}^{(1)}$, we would expect a contribution in $\tilde{\eta}^{(1)}$ also inside Σ. However, since we are considering $g = 1$, the term $(g - \overline{\overline{g}}, \varphi_k)_\Sigma$ in $\varrho^{(1)}$ vanishes and the residual results to be located only at the interface Γ. Consequently, also $\tilde{\eta}^{(1)}$ is zero inside the cylinder. In [7] it is shown that for non constant right hand side g, the residual $\varrho^{(1)}$ has non zero values in Σ.

Moreover, we perform a second test in which an inclusion with a smaller radius is considered ($R = 0.1$). In Table 1 we show a comparison between the results obtained in the two cases, highlighting that the error decreases for a thinner cylinder.

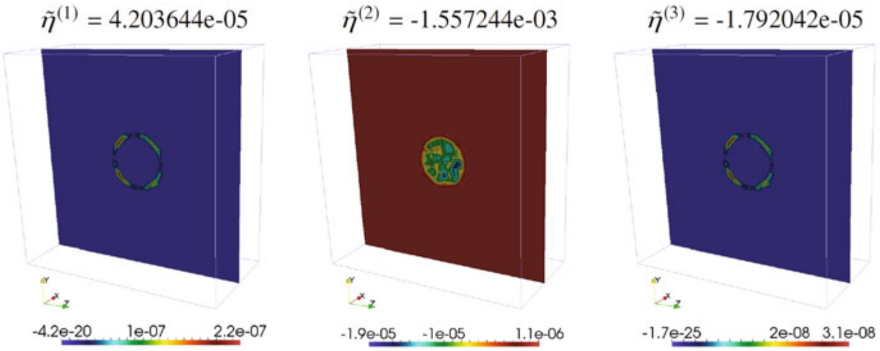


Fig. 1 The reduced estimators $\tilde{\eta}^{(1)}$, $\tilde{\eta}^{(2)}$, $\tilde{\eta}^{(3)}$

Table 1 Variation of the error output functionals $j^{(k)}(e)$ when the radius of the inclusion Σ decreases from $R = 0.25$ to $R = 0.1$

	$j^{(1)}(e) = \sum_i \eta_i^{(1)}$	$j^{(2)}(e) = \sum_i \eta_i^{(2)}$	$j^{(3)}(e) = \sum_i \eta_i^{(3)}$
$R = 0.25$	4e–05	−1e–03	−1e–05
$R = 0.1$	3e–05	−1e–03	−1e–06

Table 2 Comparison of the error output functionals $j^{(i)}(e)$ computed using the estimator $\eta^{(i)}$ based on (6) and the estimator $\tilde{\eta}^{(i)}$ based on (12)

	$j^{(1)}(e)$	$j^{(2)}(e)$	$j^{(3)}(e)$
$\eta^{(i)}$	2e–04	−1e–02	−1e–05
$\tilde{\eta}^{(i)}$	4e–05	−1e–03	−1e–05

Finally, we discuss the usage of the estimators $\tilde{\eta}^{(i)}$ based on the reduced weights instead of the estimators $\eta^{(i)}$ based on the reference ones adopted in [7]. From the previous section and the theory developed in [1], we know that with the approximated representation formula of the modeling error we neglect higher-order terms, depending on the reference dual solution. Although we cannot a priori infer that the reduced estimators are smaller than the reference ones, Table 2 shows that that $\eta^{(i)} > \tilde{\eta}^{(i)}$. The larger difference between the reference and residual estimators lays in $\eta^{(2)}$, which depends on the extension in Σ of the dual solution. However, from the comparison with the results presented in [7], one can notice that, even though the values of $\eta^{(i)}$ are different from the values of $\tilde{\eta}^{(i)}$, their distribution on the domain is almost the same.

References

1. Braack, M., Ern, A.: A posteriori control of modeling errors and discretization errors. Multiscale Model. Simul. **1**(2), 221–238 (2003). https://doi.org/10.1137/S1540345902410482
2. Cerroni, D., Laurino, F., Zunino, P.: Mathematical analysis, finite element approximation and numerical solvers for the interaction of 3d reservoirs with 1d wells. GEM - International Journal on Geomathematics **10**(1) (2019). https://doi.org/10.1007/s13137-019-0115-9
3. D'Angelo, C., Quarteroni, A.: On the coupling of 1d and 3d diffusion-reaction equations: application to tissue perfusion problems. M3AS **18**(08), 1481–1504 (2008)
4. Gjerde, I. G., Kumar, K., Nordbotten, J. M., Wohlmuth, B.: Splitting method for elliptic equations with line sources. ESAIM: M2AN **53**(5), 1715–1739 (2019). https://doi.org/10.1051/m2an/2019027
5. Köppl, T., Vidotto, E., Wohlmuth, B., Zunino, P.: Mathematical modeling, analysis and numerical approximation of second-order elliptic problems with inclusions. M3AS **28**(05), 953–978 (2018). https://doi.org/10.1142/S0218202518500252
6. Kuchta, M., Mardal, K.A., Mortensen, M.: Preconditioning trace coupled $3d$-$1d$ systems using fractional Laplacian. NMPDE **35**(1), 375–393 (2019). https://doi.org/10.1002/num.22304
7. Laurino, F., Zunino, P.: Derivation and analysis of coupled pdes on manifolds with high dimensionality gap arising from topological model reduction. ESAIM: M2AN **53**(6), 2047–2080 (2019). https://doi.org/10.1051/m2an/2019042

CG Variants for General-Form Regularization with an Application to Low-Field MRI

M. L. de Leeuw den Bouter, M. B. van Gijzen, and R. F. Remis

Abstract In an earlier paper, we generalized the CGME (Conjugate Gradient Minimal Error) algorithm to the ℓ_2-regularized weighted least-squares problem. Here, we use this Generalized CGME method to reconstruct images from actual signals measured using a low-field MRI scanner. We analyze the convergence of both GCGME and the classical Generalized Conjugate Gradient Least Squares (GCGLS) method for the simple case when a Laplace operator is used as a regularizer and indicate when GCGME is to be preferred in terms of convergence speed. We also consider a more complicated ℓ_1-penalty in a compressed sensing framework.

1 Introduction

In Magnetic Resonance Imaging (MRI), the measured signal $\mathbf{b}$ is related to $\mathbf{x}$, the object being imaged, by a Fourier Transform:

$$\mathbf{b} = \mathcal{F}\mathbf{x} + \mathbf{v}. \tag{1}$$

Here, $\mathbf{v}$ denotes a noise vector. Based on measurements $\mathbf{b}$, we will reconstruct $\mathbf{x}$, which makes this an inverse problem. In this work, we will assume the object of interest to be 2D, which means that $\mathcal{F}$ is a 2D Fourier Transform operator. However, all the results can be extended to 3D.

In conventional MRI, the signal-to-noise ratio (SNR) is so high that applying an Inverse Fourier Transform usually results in an image of very good quality.

M. L. de Leeuw den Bouter (✉) · M. B. van Gijzen
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands
e-mail: M.L.deLeeuwdenBouter-1@tudelft.nl; M.B.vanGijzen@tudelft.nl

R. F. Remis
Circuits and Systems, Delft University of Technology, Delft, The Netherlands
e-mail: R.F.Remis@tudelft.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_66

This is because superconducting magnets are used to generate strong magnetic field strengths of several tesla and the SNR is higher in the case of a stronger magnetic field. In [8], O'Reilly et al. describe a low-field MRI scanner based on a configuration of permanent magnets. The magnetic field strength inside this scanner is 50 mT, whereas conventional scanners have background fields of several teslas. For very noisy signals, it can be useful to minimize a regularized least-squares problem of the form

$$\arg\min_{\mathbf{x}} \frac{1}{2}||\mathbf{b} - \mathcal{F}\mathbf{x}||^2_{\mathbf{C}^{-1}} + \frac{1}{2}\tau||\mathbf{x}||^2_{\mathbf{R}}, \tag{2}$$

instead of disregarding the noise $\mathbf{v}$ and solving Eq. (1) for $\mathbf{x}$. In Eq. (2), the regularization parameter τ determines the tradeoff between the least-squares term $||\mathbf{b} - \mathcal{F}\mathbf{x}||^2_{\mathbf{C}^{-1}}$ and the regularization term $||\mathbf{x}||^2_{\mathbf{R}}$. In the least-squares term, $\mathbf{C}$ denotes the covariance matrix of the noise, and in the regularization term, $\mathbf{R}$ is a regularizing matrix, which we will assume to be Hermitian positive definite (HPD). Regularization allows us to enforce prior information we have about the solution. For a thorough exploration of the regularization of inverse problems, the reader is referred to [5].

2 GCGLS and GCGME

In [2], we introduced the Generalized Conjugate Gradient Minimal Error (GCGME) method for general form regularization. In this section we will review the main ideas. We are interested in solving minimization problems of the form

$$\arg\min_{\mathbf{x}} \frac{1}{2}||\mathbf{b} - \mathbf{A}\mathbf{x}||^2_{\mathbf{C}^{-1}} + \frac{1}{2}\tau||\mathbf{x}||^2_{\mathbf{R}}. \tag{3}$$

Note that Eq. (3) is of the same form as Eq. (2), but we have replaced $\mathcal{F}$ by a general forward model matrix $\mathbf{A}$.

Usually, minimization problem (3) is solved using the Generalized Conjugate Gradient Least-Squares (GCGLS) method. (We add the word "generalized" because CGLS is often used to denote the CG variant that solves the normal equations $\mathbf{A}^*\mathbf{A}\mathbf{x} = \mathbf{A}^*\mathbf{b}$ of the minimized least-squares problem without regularization.) By taking the gradient of Eq.,(3) and setting it equal to zero, we find

$$\left(\mathbf{A}^*\mathbf{C}^{-1}\mathbf{A} + \tau\mathbf{R}\right)\mathbf{x} = \mathbf{A}^*\mathbf{C}^{-1}\mathbf{b}. \tag{4}$$

Equation (4) can be solved using the conjugate gradient (CG) method. Some adjustments can be made to improve stability, see for example [1], leading to the GCGLS method. By rewriting Eq. (3) as a constrained minimization problem, we can find another set of equations that can be used to find the solution $\mathbf{x}$. We define

$\mathbf{r} = \mathbf{C}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x})$ and rewrite minimization problem (3):

$$\min_{\mathbf{r},\mathbf{x}} \frac{1}{2}||\mathbf{r}||_{\mathbf{C}}^2 + \frac{1}{2}\tau||\mathbf{x}||_{\mathbf{R}}^2 \qquad (5)$$
$$\text{s.t. } \mathbf{r} = \mathbf{C}^{-1}(\mathbf{b} - \mathbf{A}\mathbf{x}).$$

We will assume $\tau > 0$. By applying the method of Lagrange multipliers and eliminating $\mathbf{x}$, we get

$$\left(\frac{1}{\tau}\mathbf{A}\mathbf{R}^{-1}\mathbf{A}^* + \mathbf{C}\right)\mathbf{r} = \mathbf{b}. \qquad (6)$$

Additionally, the following relationship between $\mathbf{r}$ and $\mathbf{x}$ holds:

$$\mathbf{x} = \frac{1}{\tau}\mathbf{R}^{-1}\mathbf{A}^*\mathbf{r}. \qquad (7)$$

So by applying CG to Eq. (6) and subsequently solving Eq. (7) for $\mathbf{x}$, we can obtain our solution. The resulting algorithm, which we call Generalized Conjugate Gradient Minimal Error (GCGME), is given below.

Algorithm 1 GCGME

```
Require: $\mathbf{A} \in \mathbb{C}^{M\times N}, \mathbf{C} \in \mathbb{C}^{M\times M}, \mathbf{R} \in \mathbb{C}^{N\times N}, \mathbf{r}_0 \in \mathbb{C}^M, \mathbf{b} \in \mathbb{C}^M, \tau \in \mathbb{R}_{>0}$;
Ensure: Approximate solution $\mathbf{x}_k$ such that $\|\mathbf{b} - \mathbf{A}\mathbf{x}_k - \mathbf{C}\mathbf{r}_k\| \leqslant TOL$.
$\mathbf{x}_0 = \frac{1}{\tau}\mathbf{R}^{-1}\mathbf{A}^H\mathbf{r}_0$
$\mathbf{s}_0 = \mathbf{b} - \mathbf{A}\mathbf{x}_0 - \mathbf{C}\mathbf{r}_0$, $\mathbf{p}_0 = \mathbf{s}_0$, $\mathbf{q}_0 = \mathbf{A}^H\mathbf{p}_0$, $\gamma_0 = \mathbf{s}_0^H\mathbf{s}_0$, $k = 0$
while $\sqrt{\gamma_k} > TOL$ and $k < k_{max}$ do
    $\xi_k = \frac{1}{\tau}\mathbf{q}_k^H\mathbf{R}^{-1}\mathbf{q}_k + \mathbf{p}_k^H\mathbf{C}\mathbf{p}_k$
    $\alpha_k = \frac{\gamma_k}{\xi_k}$
    $\mathbf{r}_{k+1} = \mathbf{r}_k + \alpha_k\mathbf{p}_k$
    $\mathbf{x}_{k+1} = \mathbf{x}_k + \frac{\alpha_k}{\tau}\mathbf{R}^{-1}\mathbf{q}_k$
    $\mathbf{s}_{k+1} = \mathbf{s}_k - \alpha_k(\frac{1}{\tau}\mathbf{A}\mathbf{R}^{-1}\mathbf{q}_k + \mathbf{C}\mathbf{p}_k)$
    $\gamma_{k+1} = \mathbf{s}_{k+1}^H\mathbf{s}_{k+1}$
    $\beta_k = \frac{\gamma_{k+1}}{\gamma_k}$
    $\mathbf{p}_{k+1} = \mathbf{s}_{k+1} + \beta_k\mathbf{p}_k$
    $\mathbf{q}_{k+1} = \mathbf{A}^H\mathbf{p}_{k+1}$
    $k = k + 1$
end while
```

2.1 Comparison of the Condition Numbers: A Simple Case

In this section we consider a very simple but illustrative case that allows us to analyze the condition numbers, and hence the convergence speed, of GCGME and

GCGLS. We demonstrate, depending on the regularization parameter, which method is to be preferred. We set $\mathbf{A} = \mathcal{F}$ and the noise is assumed to be white noise, so $\mathbf{C} = \mathbf{I}$. We define the regularization matrix to be the discretized 2D Laplacian $\mathbf{L}$ complemented with Dirichlet boundary conditions. Choosing the regularization matrix in this way means that large jumps in the reconstructed image $\mathbf{x}$ are discouraged. In that case, GCGLS solves

$$(\mathbf{I} + \tau\mathbf{L})\,\mathbf{x} = \mathcal{F}^{*}\mathbf{b}, \tag{8}$$

where $\mathcal{F}^{*} = \mathcal{F}^{-1}$ is the inverse 2D Fourier Transform. GCGME solves

$$\left(\frac{1}{\tau}\mathcal{F}\mathbf{L}^{-1}\mathcal{F}^{*} + \mathbf{I}\right)\mathbf{r} = \mathbf{b}\ , \tag{9}$$

$$\mathbf{x} = \frac{1}{\tau}\mathbf{L}^{-1}\mathcal{F}^{*}\mathbf{r}.$$

The convergence speed of GCGLS and GCGME depends on the condition number of the matrices $\mathbf{I} + \tau\mathbf{L}$ and $\frac{1}{\tau}\mathcal{F}\mathbf{L}^{-1}\mathcal{F}^{*} + \mathbf{I}$, respectively. The eigenvalues of the Laplacian $\mathbf{L}$ are well-known and hence we can find explicit expressions for the condition numbers. For GCGLS, we have

$$\kappa_2\left(\mathbf{I} + \tau\mathbf{L}\right) = \frac{1 + 8\tau\cos^2\left(\frac{\pi}{2}\frac{1}{N+1}\right)}{1 + 8\tau\sin^2\left(\frac{\pi}{2}\frac{1}{N+1}\right)}. \tag{10}$$

Here, we assume that our image consists of $N \times N$ pixels. For GCGME, we make use of the fact that $\mathcal{F}\mathbf{L}^{-1}\mathcal{F}^{*}$ is a similarity transformation and therefore has the same eigenvalues as $\mathbf{L}^{-1}$, yielding

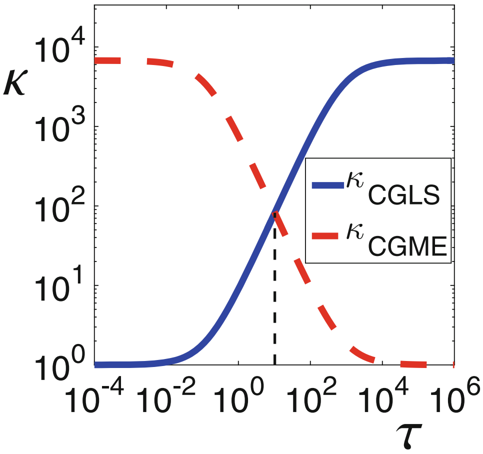
$$\kappa_2\left(\frac{1}{\tau}\mathcal{F}\mathbf{L}^{-1}\mathcal{F}^{*} + \mathbf{I}\right) = \frac{1 + \frac{1}{8\tau\sin^2\left(\frac{\pi}{2}\frac{1}{N+1}\right)}}{1 + \frac{1}{8\tau\cos^2\left(\frac{\pi}{2}\frac{1}{N+1}\right)}}. \tag{11}$$

These condition numbers can be shown to be equal when

$$\tau^{*} = \frac{1}{8\cos\left(\frac{\pi}{2(N+1)}\right)\sin\left(\frac{\pi}{2(N+1)}\right)}. \tag{12}$$

Figure 1 shows a plot of the condition numbers as a function of the value of the regularization parameter τ, in case $N = 128$. We observe that when $\tau < \tau^{*}$, GCGLS has a smaller condition number, whereas GCGME has a smaller condition number when $\tau > \tau^{*}$. Therefore, we expect GCGME to attain faster convergence for large τ.

Fig. 1 Condition numbers of the GCGLS matrix $\mathbf{I} + \tau\mathbf{L}$ and the GCGME matrix $\frac{1}{\tau}\mathcal{F}\mathbf{L}^{-1}\mathcal{F}^* + \mathbf{I}$ as a function of the value of the regularization parameter τ



2.2 GCGLS and GCGME for IRLS

The ℓ_2-penalty tends to lead to overly blurry images, due to the quadratic penalty term. Therefore, we are more interested in the ℓ_p-regularized least squares problem with $p \in (0, 1]$:

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{Ax} - \mathbf{b}||_2^2 + \frac{1}{p}\tau||\mathbf{Fx}||_p^p, \tag{13}$$

For the ℓ_p-penalty with $p \in (0, 1]$, the blurring effect is less pronounced. Additionally, the ℓ_p-penalty induces sparsity in $\mathbf{Fx}$, see for example [4]. However, solving minimization problem (13) is not as straightforward as Eq. (3). One way of solving it is by using Iterative Reweighted Least Squares (IRLS). This means that we replace minimization problem (13) by a sequence of ℓ_2-regularized problems of the same form as Eq. (3). Given an estimate $\mathbf{x}_k$ of the solution $\mathbf{x}$, the matrix $\mathbf{R}_k$ in the penalty term is recalculated based on $\mathbf{x}_k$:

$$\mathbf{R}_k = \mathbf{F}^*\mathbf{D}_k\mathbf{F}, \qquad \mathbf{D}_k = \text{diag}\left(\frac{1}{|\mathbf{Fx}_k|^{2-p}}\right). \tag{14}$$

So in each IRLS step, one minimization problem of the form (3) is solved. We will compare GCGLS and GCGME for this step. In case $\mathbf{F}$ is an invertible matrix, we have $\mathbf{R}_k^{-1} = \mathbf{F}^{-1}\mathbf{D}_k^{-1}(\mathbf{F}^H)^{-1}$, with

$$\mathbf{D}_k^{-1} = \text{diag}\left(|\mathbf{Fx}_k|^{2-p}\right). \tag{15}$$

When GCGME is used, we can take advantage of this structure, instead of calculating $\mathbf{R}_k$ and working with its inverse. Moreover, when $\mathbf{F}$ is an orthogonal matrix, no additional computations are necessary to compute inverses.

In [2], we showed that when

$$\kappa_2(\mathbf{R}) \gg \kappa_2(\mathbf{C}), \tag{16}$$

GCGME is expected to exhibit faster converge than GCGLS. When the sparsifying ℓ_p-penalty with $p \in (0, 1]$ is used, some elements of D_k will tend to infinity. Therefore, $\mathbf{R}$ is expected to become increasingly ill-conditioned, in which case Eq. (16) holds. Therefore, we expect GCGLS to be outperformed by GCGME in terms of convergence speed.

3 Experiments

Experiments were carried out using the low-field MRI scanner described in [8], a picture of which is shown in Fig. 2a. Inside the scanner, the magnetic field generated by the configuration of magnets is approximately homogeneous. Linear gradient fields are applied before and during readout for phase and frequency encoding. These steps ensure that the resulting signal is essentially equal to the Fourier Transform of the object inside the scanner. For an introduction to the principles of MRI, the reader is referred to [6]. The object being imaged, see Fig. 2b, is a real-life version of the Shepp-Logan phantom, which was introduced in [9]. It is approximately 10 cm in diameter. This phantom is often used to test reconstruction algorithms for tomographic imaging. The sampling rate was set to

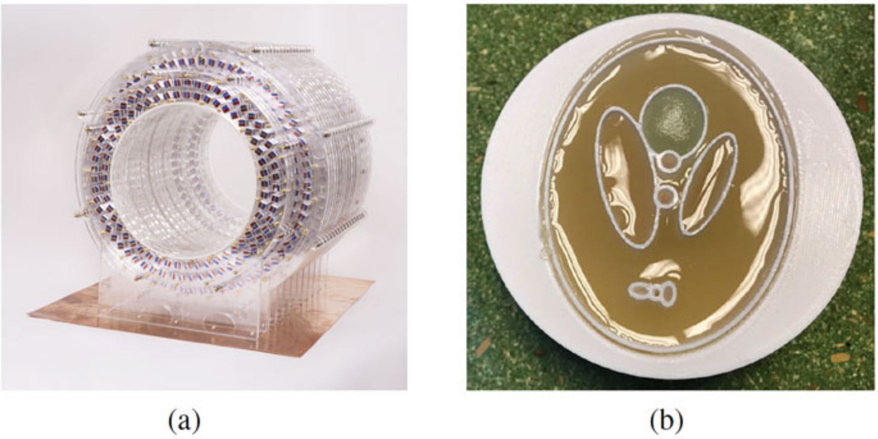
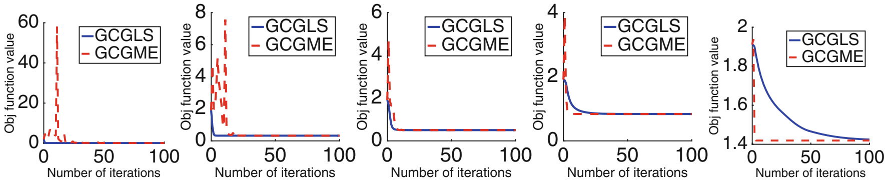


Fig. 2 Experimental setup. (**a**) Low-field MRI scanner. (**b**) Object being imaged

20 μs, corresponding to a bandwidth of 50 kHz. A spin echo pulsing sequence was used with an echo time T_E of 10 ms and a repetition time T_R of 500 ms. The length of the RF pulse was 100 μs. The Field of View (FoV) was 12 × 12 cm^2, with the target image having 128 × 128 pixels. No slice selection was carried out.

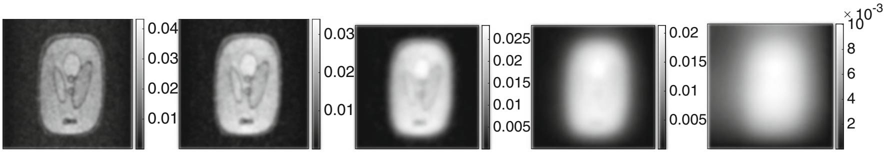
4 Numerical Results

First, we solve minimization problem (3) with $\mathbf{A} = \mathcal{F}$, $\mathbf{C} = \mathbf{I}$ and $\mathbf{R} = \mathbf{L}$, which is the scenario we reviewed earlier. GCGLS and GCGME solve different normal equations, so a comparison using a stopping criterion based on residuals would not be fair. Instead, we use a fixed number of CG iterations for both methods. For the ℓ_2 case, we use 100 iterations. Figure 3 shows plots of the value of objective function (2) with $\mathbf{R} = \mathbf{L}$ as a function of the iteration number for 5 different values of the regularization parameter τ. We observe that in all cases, both methods lead to the same objective function value, as expected. For $\tau = 10$, which is approximately equal to τ^*, we note that both methods converge equally fast. For smaller values of τ, GCGLS converges faster while for larger values, GCGME shows faster convergence. The corresponding images are shown in Fig. 4. Both methods need the same amount of time per iteration.



(a) (b) (c) (d) (e)

Fig. 3 Objective function value as a function of the iteration number for different values of the regularization parameter τ. (**a**) $\tau = 0.1$. (**b**) $\tau = 1$. (**c**) $\tau = 10$. (**d**) $\tau = 100$. (**e**) $\tau = 1000$

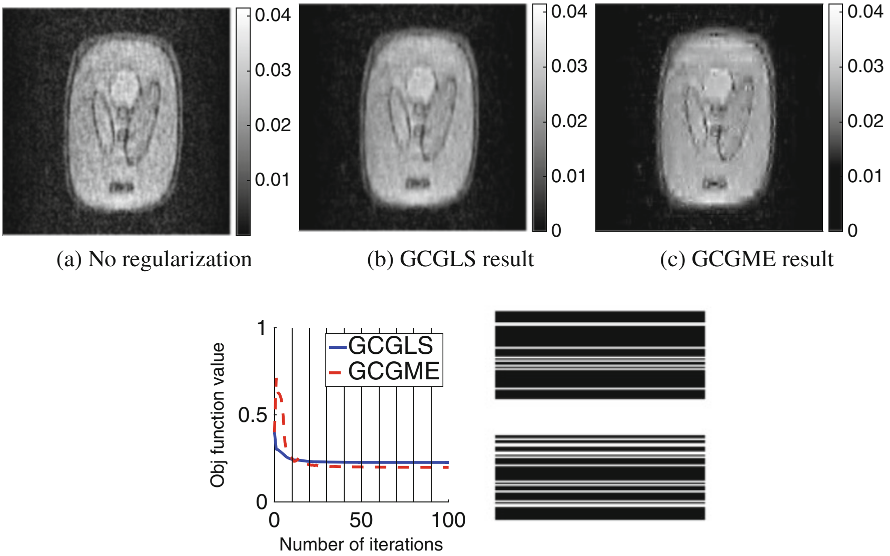


(a) (b) (c) (d) (e)

Fig. 4 Reconstructed images for different values of the regularization parameter τ. (**a**) $\tau = 0.1$. (**b**) $\tau = 1$. (**c**) $\tau = 10$. (**d**) $\tau = 100$. (**e**) $\tau = 1000$

In MRI, scan times tend to be long. They can be reduced by using compressed sensing. In compressed sensing, the number of data points acquired is reduced, compared to traditional scans. This can be done by measuring a subset of the lines in k-space, or the frequency domain. For more information about compressed sensing in MRI, [7] can be consulted. We will use the notation $\mathcal{F}_u$ to denote the Fourier Transform of the undersampled measurements. One of the assumptions made in compressed sensing is that the image is sparse in some known transform domain, for example a wavelet transform. We also investigate the two CG variants in a compressed sensing framework with an undersampling factor of 3.

We solve minimization problem (13) with $\mathbf{A} = \mathcal{F}_u$, $\mathbf{C} = \mathbf{I}$, $\mathbf{F} = \mathbf{W}$, $\tau = 6 \times 10^{-3}$ and $p = 1$. The regularization parameter is chosen heuristically. Here, $\mathbf{W}$ is the 2D Daubechies wavelet transform [3]. We choose $\mathcal{F}_u\mathbf{b}$, which is shown in Fig. 5a, as our initial guess. We use 10 IRLS iterations and in each of these, 10 CG iterations are carried out. Figure 5 shows the reconstructed images and the value of the objective function as a function of the iteration number. GCGME shows rapid convergence, whereas the convergence of GCGLS is so slow that it seems that GCGLS has converged to a higher objective function value than GCGME. However, both methods converge to the same value if the number of GCGLS iterations is increased significantly, see [2]. GCGLS and GCGME need the same amount of time per iteration.



(a) No regularization (b) GCGLS result (c) GCGME result

(d) Objective function (e) Lines in k-space

Fig. 5 Reconstructed images using (**a**) only the inverse Fourier Transform, (**b**) and (**c**) the two different CG variants. (**d**) shows a plot of the objective function value as a function of the iteration number for both methods. The vertical black lines indicate the start of a new IRLS iteration. (**e**) shows the lines in k-space that were used for reconstruction

5 Conclusion

We analyzed the condition numbers of the matrices used in GCGME and GCGLS in the simple but illustrative case where the discretized Laplacian is used as the regularization matrix. The value of the regularization parameter τ^* determines which method is to be preferred in terms of convergence speed. We can easily calculate τ^*, the value for which both methods have the same condition number. For $\tau < \tau^*$, GCGLS is expected to converge faster and for $\tau > \tau^*$, GCGME is to be preferred. We applied both methods to data measured using a low-field MRI scanner and our numerical results show that the two methods behave as expected.

We also considered the more relevant case of an ℓ_1-regularization penalty in a compressed sensing framework and used IRLS to solve this problem. Inside each IRLS iteration, GCGLS or GCGME can be used as a building block. Due to the sparsifying properties of the ℓ_p-penalty with $p \in (0, 1]$, the reweighting of the regularization matrix leads to an increasingly ill-conditioned matrix, which corresponds to the regime in which GCGME is expected to show rapid convergence. Our numerical results show that indeed, GCGME converges much faster than GCGLS for this problem.

References

1. Björck, Å.: Numerical methods for least squares problems. SIAM (1996)
2. de Leeuw den Bouter, M.L., van Gijzen, M.B., Remis, R.F.: Conjugate gradient variants for ℓ_p-regularized image reconstruction in low-field MRI. SN Applied Sciences **1**(12), 1736 (2019)
3. Daubechies, I.: Ten lectures on wavelets, vol. 61. Siam (1992)
4. Elad, M.: Sparse and redundant representations: from theory to applications in signal and image processing. Springer Science & Business Media (2010)
5. Engl, H.W., Hanke, M., Neubauer, A.: Regularization of inverse problems, vol. 375. Springer Science & Business Media (1996)
6. Liang, Z.P., Lauterbur, P.C.: Principles of Magnetic Resonance Imaging: A Signal Processing Perspective. SPIE Optical Engineering Press (2000)
7. Lustig, M., Donoho, D.L., Santos, J.M., Pauly, J.M.: Compressed sensing MRI. IEEE signal processing magazine **25**(2), 72 (2008)
8. O'Reilly, T., Teeuwisse, W., Webb, A.: Three-dimensional MRI in a homogenous 27 cm diameter bore Halbach array magnet. Journal of Magnetic Resonance **307**, 106578 (2019)
9. Shepp, L.A., Logan, B.F.: The Fourier reconstruction of a head section. IEEE Transactions on Nuclear Science **21**(3), 21–43 (1974). https://doi.org/10.1109/TNS.1974.6499235

Data-Driven Modeling for Wave-Propagation

Tristan van Leeuwen, Peter Jan van Leeuwen, and Sergiy Zhuk

Abstract Many imaging modalities, such as ultrasound and radar, rely heavily on the ability to accurately model wave propagation. In most applications, the response of an object to an incident wave is recorded and the goal is to characterize the object in terms of its physical parameters (e.g., density or soundspeed). We can cast this as a joint parameter and state estimation problem. In particular, we consider the case where the inner problem of estimating the state is a weakly constrained data-assimilation problem. In this paper, we discuss a numerical method for solving this variational problem.

1 Introduction

Many imaging modalities, such as ultrasound, geophysical exploration, and radar, rely heavily on the ability to accurately model wave propagation. In most applications, the response of an object to an incident wave is recorded and the goal is to characterize the object in terms of its physical parameters (e.g., density or soundspeed). We can capture this setup in terms of a process and measurement model

$$\mathcal{L}(c)u = q, \tag{1}$$

$$d = \mathcal{P}u, \tag{2}$$

T. van Leeuwen (✉)
Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
e-mail: van.leeuwen@cwi.nl

P. J. van Leeuwen
Colorado State University, Fort Collins, CO, USA
e-mail: Peter.vanLeeuwen@colostate.edu

S. Zhuk
IBM Research, Dublin, Ireland

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_67

where u denotes the wavefield, $\mathcal{L}(c) = \partial_t^2 - c^2\nabla^2$ represents the wave equation depending on the physical parameters, c, and $\mathcal{P}$ is the sampling operator that models the measurement process. In particular, consider the case where we are given a *finite* number of samples of the state. We are ultimately interested in estimating the parameters c from the measurements d. There are different ways to go about this;

PDE-constrained optimization: Eliminate the process model and set up a non-linear data-fitting problem to match the solution of (1) to the data [13].

Equation-error approach: Estimate the state directly from the measurements by solving (2) and subsequently solve for c from (1) [3, 9].

Joint parameter and state estimation: Find the parameter and state that satisfy both (1) and (2) approximately [14].

The former two can be thought of as limiting cases of the latter where the state is estimated either completely determined by the process model or determined solely from the data. The joint approach gives rise to a data-driven modelling problem, where one aims to estimate a state that satisfies both the data and the physics to some extent. We can formally express this as a variational problem

$$\min_u \|\mathcal{P}u - d\|^2 + \rho\|\mathcal{L}(c)u - q\|^2, \tag{3}$$

where ρ is a parameter that controls the trade-off between the two terms. How well we are able to approximate the *true* state by solving (3) depends on how many measurements are available, the observability of the system, how close c is to the true parameter and requires an appropriate choice of ρ. We will not address this issue here and focus solely on solving (3) for given c and ρ.

A straightforward approach to solving (3) would be to derive the Euler-Lagrange equations, and solve the resulting PDE numerically. Alternatively, one could discretize the wave-equation first and set up a large sparse system of equations for the state [4, 5, 8, 10]. In both cases, the dimensionality of the problem is governed by the numerical discretization. In this paper, we cast the problem in a reproducing kernel Hilbert space, allowing us to express the solution of (3) as a finite linear combination of kernel functions [6, 12]. This leads to a system of linear equations involving a kernel matrix. The dimension of this system is given by the number of measurements and is thus independent of the underlying numerical discretisation of the PDE. We discuss a preconditioned iterative method for solving this system. Finally, we present some numerical examples and conclude the paper.

2 Theory

We consider a scalar wave-equation in $[0, T] \times \mathbb{R}^d$ of the form $\mathcal{L}u = q$ with $\mathcal{L}(c) = \partial_t^2 - c^2\nabla^2$ and initial conditions $u(0, x) = \partial_t u(0, x) = 0$. Without elaborating on the details, we assume that this problem is well-posed for all parameters c of interest and that the solution is given by $u = \mathcal{G}q$ with

$$\mathcal{G}q(t, x) = \int_0^t \int_{\mathbb{R}^d} g(t - s, x, y)q(s, y)\mathrm{d}y\mathrm{d}s,$$

where g is the Green's function. The corresponding adjoint problem $\mathcal{L}^* v = r$ (with $u(T, x) = \partial_t u(T, x) = 0$) has solution $v = \mathcal{G}^* r$, where

$$\mathcal{G}^* r(t, x) = \int_t^T \int_{\mathbb{R}^d} g(s - t, x, y)r(s, y)\mathrm{d}y\mathrm{d}s.$$

For further details regarding the well-posedness of variable-coefficient wave-equations we refer to [2].

The measurements are obtained by sampling the state at given locations $\{(t_i, x_i)\}_{i=1}^M$:

$$\mathcal{P}u = \{u(t_i, x_i)\}_{i=1}^M.$$

We introduce a Hilbert space $\mathcal{U}$ with inner product:

$$\langle u, v\rangle_{\mathcal{U}} = \langle \mathcal{L}u, \mathcal{L}v\rangle_{L^2(\mathbb{R}^{d+1})}.$$

We can think of this as the space of solutions of the wave equation with square integrable source term. The space $\mathcal{U}$ is a *Reproducing Kernel Hilbert Space* (RKHS) [7]. A special property of an RKHS is that point-evaluation is a bounded linear functional with Riesz representation $k_{t,x} \in \mathcal{U}$ so that $\langle k_{t,x}, u\rangle_{\mathcal{U}} = u(t, x)$.[1] The reproducing kernel of $\mathcal{U}$ is given by $k(t, x, t', x') = \langle k_{t,x}, k_{t',x'}\rangle_{\mathcal{U}}$. It is the Green's function of $\mathcal{L}^*\mathcal{L}$ and is thus given by

$$k(t, x, t', x') = \int_0^T \int_{\mathbb{R}^d} g(t - s, x, y)g(t' - s, y, x')\mathrm{d}y\mathrm{d}s.$$

We can think of k as a spline that is tailored to represent solutions of the wave equation.

The aim is to solve a variational problem of the form

$$\min_{u\in\mathcal{U}} \sum_{i=1}^M (u(t_i, x_i) - d_i)^2 + \rho\|\mathcal{L}u - q\|^2_{L^2(\mathbb{R}^{d+1})}. \tag{4}$$

[1] This requires that the solution of $\mathcal{L}u = q$ can be bounded point-wise as $|u(t, x)| \leq C\|q\|_{L^2}$. While this is possible in general for $d = 1$, it perhaps requires more regularity of the source function for $d > 1$.

By splitting the solution as $u = \mathcal{G}q + w$ and introducing $r = d - \mathcal{G}q$, we can re-write this as

$$\min_{v \in \mathcal{U}} \sum_{i=1}^{M} (w(t_i, x_i) - r_i)^2 + \rho \|w\|_{\mathcal{U}}^2. \tag{5}$$

Utilizing the Representer Theorem [1, 11], we know that the solution to this variational problem has the following form

$$w(t, x) = \sum_{i=1}^{M} w_i k(t_i, x_i, t, x).$$

We can use this finite-dimensional representation of the solution to express (4) as a finite-dimensional least-squares problem

$$\min_{\mathbf{w} \in \mathbb{R}^M} \|K\mathbf{w} - \mathbf{r}\|_2^2 + \rho \mathbf{w}^T K \mathbf{w}, \tag{6}$$

where K is the kernel matrix with elements $k_{ij} = k(t_i, x_i, t_j, x_j)$. The kernel matrix is guaranteed to be positive definite, ensuring a unique solution given by

$$\widehat{\mathbf{w}} = (K + \rho I)^{-1} \mathbf{r}.$$

2.1 Example: Constant Coefficients

With $c(x) = 1$ we can express all quantities in the spatial Fourier domain. The Green's function is then given by

$$\widehat{g}(t, \xi) = \frac{\sin((t - s)|\xi|)}{|\xi|},$$

giving

$$\widehat{k}(t, t', \xi) = |\xi|^{-2} \int_0^{\min(t,t')} \sin((t - s)|\xi|) \sin((t' - s)|\xi|) \mathrm{d}s,$$

which yields

$$\widehat{k}(t, t', \xi) = \begin{cases} t|\xi|^{-2} \cos((t - t')|\xi|) - |\xi|^{-3} \cos(t'|\xi|) \sin(t|\xi|) & t \le t' \\ t'|\xi|^{-2} \cos((t - t')|\xi|) - |\xi|^{-3} \cos(t|\xi|) \sin(t'|\xi|) & t > t'. \end{cases} \tag{7}$$

To get some insight into the properties of the continuous kernel operator defined by

$$\mathcal{K}\,\widehat{u}(t) = \int_0^\infty \widehat{k}(t, t', \xi)\widehat{u}(t')\mathrm{d}t',$$

we take $\widehat{u}(t) = \sin(\omega t)$ and find that this is an eigenfunction with eigenvalue $\lambda = (\omega^2 - |\xi|^2)^{-2}$. The continuous operator can thus have an arbitrarily large norm due to modes with $\omega \approx \pm\|\xi\|$. The corresponding kernel matrix, K, can thus be extremely ill-conditioned.

3 Algorithm

We discretize all quantities on a regular grid and introduce the notation $\mathbf{u}_k = (u(k \cdot \Delta t, x_1), u(k \cdot \Delta t, x_2), \ldots, u(k \cdot \Delta t, x_{n_x}))$. A second order finite-difference discretization of $\mathcal{L}$ on $[0, T] \times [-D, D]$ with Dirichlet boundary conditions leads to forward and adjoint systems of the form $L\mathbf{u} = \mathbf{q}$, $L'\mathbf{v} = \mathbf{r}$ with $\mathbf{u} = (\mathbf{u}_1, \ldots, \mathbf{u}_{n_t})$, $\mathbf{q} = (\mathbf{q}_0, \mathbf{q}_1, \ldots, \mathbf{q}_{n_t-1})$ and

$$L = (\Delta t)^{-2}\begin{pmatrix} 2I & & & & \\ S & I & & & \\ I & S & I & & \\ & & \ddots & & \\ & & I & S & I \end{pmatrix}, \quad L' = (\Delta t)^{-2}\begin{pmatrix} I & S^T & I & & \\ & & \ddots & & \\ S^T & I & S^T & I & \\ & & I & S^T & \\ & & & 2I & \end{pmatrix}, \tag{8}$$

where $S = -2I - (\Delta t)^2 A$ and $A \in \mathbb{R}^{n_x \times n_x}$ is a second order central finite-difference discretization of $c^2\nabla^2$. Note that $L' \neq L^T$. The sampling operator is discretized using piecewise linear interpolation, yielding a matrix $P \in \mathbb{R}^{M\times N}$. Using adjoint interpolation is an appropriate way to represent the point source [15]. The combination of a second-order finite-difference approximation of $\mathcal{L}$ and linear interpolation ensures an overall second order approximation of the elements of K.

Storing the full matrix $K = P(L'L)^{-1}P^T$ may not be very attractive, but we can compute matrix-vector products with K by solving one forward and one adjoint problem;

$$K\mathbf{w} = P\mathbf{u},$$

with $L\mathbf{u} = \mathbf{v}$ and $L'\mathbf{v} = P^T\mathbf{w}$. Since the matrix is symmetric we can apply CG to solve the system $(K + \rho I)\,\mathbf{w} = \mathbf{r}$. As this system becomes increasingly ill-conditioned as ρ decreases, preconditioning is of paramount importance. Due to the specific form of K, we propose a preconditioner of the form $K^{-1} \approx M = QL'LQ^T$, where Q is chosen so that $Q^TP\mathbf{u} \approx \mathbf{u}$ for solutions of $L\mathbf{u} = \mathbf{q}$. When

P samples the solution on a grid we can take Q to be a high-order interpolation operator (e.g. cubic splines).

4 Numerical Results

4.1 Harmonic Oscillator

For a single spatial Fourier mode, the kernel is given by (7). Figure 1a shows an example of $\widehat{k}(t, t')$ for $\xi = 20$, $t' = \frac{1}{3}$ and $t' = \frac{1}{2}$. We take samples on a regular grid with $m = 20$ samples in $(0, 1)$. The spectrum of the corresponding kernel matrix is shown in Fig. 1b. Also shown is the Fourier approximation of the spectrum. The kink in the spectrum is due to the singularity in the spectrum of the continuous operator at $\omega = \|\xi\|$. Figure 1c shows the absolute error as a function of Δt when using the numerical approximation described above. The effect of the preconditioner is shown in Fig. 2. We see that when using 1D spline interpolation, most of the eigenvalues of MK are clustered around one.

4.2 1D Wave-Equation

We generate data by solving the *non-constant* coefficient wave equation with soundspeed $c(x)$ and sampling the solution on a regular grid. We then solve the variational problem for a *constant* reference soundspeed, c_0. The grid and velocity profiles are depicted in Fig. 3. The true state, reference state and reconstructed state for $\rho = 10^{-4}\|K\|_2$ are depicted in Fig. 4. As preconditioner for small ρ we use $(K + \rho I)^{-1} \approx K^{-1} \approx M$ while for large ρ we use $(K + \rho I)^{-1} \approx \rho^{-1}I - \rho^{-2}K$

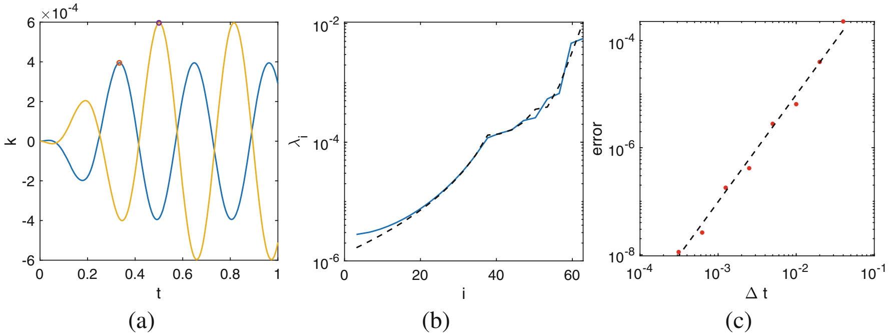


Fig. 1 (**a**) Kernel $\widehat{k}(t, t')$ for $\xi = 20$, $t' = \frac{1}{3}$ and $t' = \frac{1}{2}$. (**b**) Spectrum of K (solid) and its Fourier-approximation (dashed). (**c**) Absolute error as a function of Δt

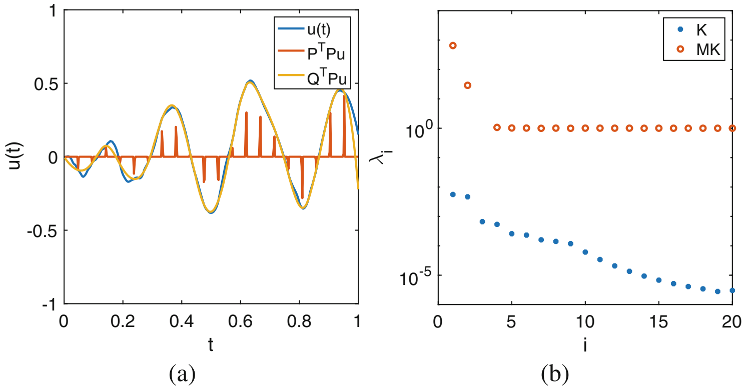


Fig. 2 (**a**) Illustration of the action of the interpolation operator Q. The solution to $L\mathbf{u} = \mathbf{q}$ with random source term is depicted, as well as its interpolated result $P^T P\mathbf{u}$ and $Q^T P\mathbf{u}$. (**b**) Eigenvalues of K and MK. The preconditioner nicely clusters most of the eigenvalues

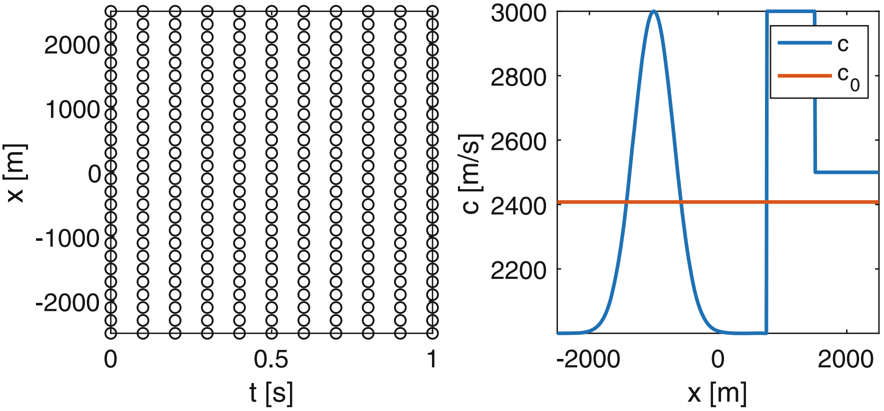


Fig. 3 Left: Grid used to define the sampling operator. Right: velocity profiles used to generate data and estimate state

with $M = QL'LQ^T$ and Q is a 2D spline interpolation. The convergence history of CG, with and without preconditioner, for various values of ρ is shown in Fig. 5.

5 Conclusion and Discussion

We presented a numerical method for solving a variational data-assimilation problem involving the wave equation. By casting the problem in a Reproducing Kernel Hilbert Space, we derived a finite-dimensional system of equations involving a kernel matrix. Computing the action of this kernel on a given vector involves

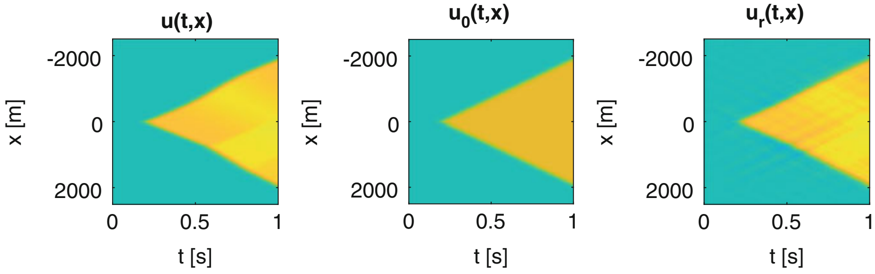


Fig. 4 True, reference and estimated states for $\rho = 10^{-6}$

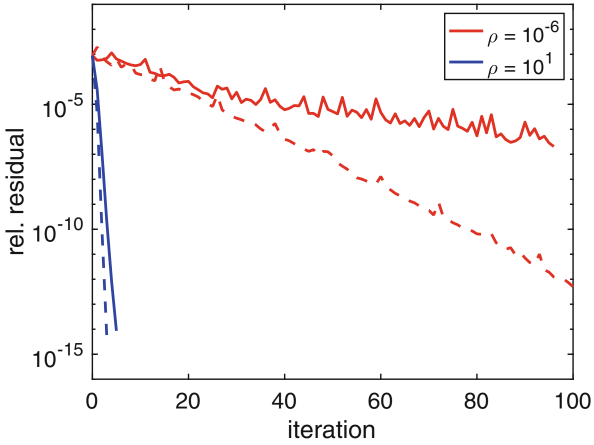


Fig. 5 Convergence of CG when solving the system for various values of ρ with (solid) and without (dashed) preconditioner

numerically solving a forward and adjoint wave equation and we described a non-self-adjoint second-order finite difference scheme for the wave equation to approximate the kernel. Using a simple Fourier analysis we show that the kernel matrix can be arbitrarily ill-conditioned. A simple preconditioner was proposed that appears to perform reasonably well in practice. The numerical examples presented in this paper involved a 1D wave equation and a relatively dense measurement grid. While the methodology described here can be easily extended to higher dimensions, the simple preconditioner will probably not perform as well on coarser measurement grids. Further analysis of the kernel for non-constant coefficients may shed some light on this issue.

Acknowledgments The first author acknowledges W.W. Symes for pointing out some issues with the RKHS framework for wave-equations in $d > 1$ dimensions.

References

1. Andrew F Bennett. *Inverse methods in physical oceanography*. Cambridge university press, 1992.
2. Kirk D Blazek, Christiaan Stolk, and William W Symes. A mathematical framework for inverse wave problems in heterogeneous media. *Inverse Problems*, 29(6):065001, Nov 2013.
3. S A L De Ridder and A Curtis. Seismic gradiometry using ambient seismic noise in an anisotropic Earth. *Geophysical Journal International Geophys. J. Int*, 209:1168–1179, 2017.
4. Ron Estrin, Dominique Orban, and Michael A. Saunders. LSLQ: An Iterative Method for Linear Least-Squares with an Error Minimization Property. *SIAM Journal on Matrix Analysis and Applications*, 40(1):254–275, Jan 2019.
5. Melina A. Freitag and Daniel L.H. Green. A low-rank approach to the solution of weak constraint variational data assimilation problems. *Journal of Computational Physics*, 357:263–281, 2018.
6. Javier González, Ivan Vujačić, and Ernst Wit. Reproducing kernel Hilbert space based estimation of systems of ordinary differential equations. *Pattern Recognition Letters*, 45:26–32, Aug 2014.
7. Vern I Paulsen and Raghupathi Mrinal. *An Introduction to the Theory of Reproducing Kernel Hilbert Spaces*. 2016.
8. Bas Peters, Felix J Herrmann, et al. A numerical solver for least-squares sub-problems in 3d wavefield reconstruction inversion and related problem formulations. In *SEG International Exposition and Annual Meeting*. Society of Exploration Geophysicists, 2019.
9. C. Poppeliers, P. Punosevac, and T. Bell. Three-Dimensional Seismic-Wave Gradiometry for Scalar Waves. *Bulletin of the Seismological Society of America*, 103(4):2151–2160, aug 2013.
10. Gabrio Rizzuti, Mathias Louboutin, Rongrong Wang, Emmanouil Daskalakis, Felix Herrmann, et al. A dual formulation for time-domain wavefield reconstruction inversion. In *SEG International Exposition and Annual Meeting*. Society of Exploration Geophysicists, 2019.
11. Bernhard Schölkopf, Ralf Herbrich, and Alex J Smola. A Generalized Representer Theorem. *COLT/EuroCOLT*, 2111(2000 - 81):416–426, 2001.
12. Florian Steinke and Bernhard Schölkopf. Kernels, regularization and differential equations. *Pattern Recognition*, 41(11):3271–3286, 2008.
13. A Tarantola and A Valette. Generalized nonlinear inverse problems solved using the least squares criterion. *Reviews of Geophysics and Space Physics*, 20(2):129–232, 1982.
14. T van Leeuwen and Felix J Herrmann. A penalty method for PDE-constrained optimization in inverse problems. *Inverse Problems*, 32(1):015007, jan 2016.
15. Johan Waldén. On the approximation of singular source terms in differential equations. *Numerical Methods for Partial Differential Equations*, 15(4):503–520, 1999.

Numerical Simulation of Coupled Electromagnetic and Thermal Problems in Permanent Magnet Synchronous Machines

A. Lotfi, D. Marcsa, Z. Horváth, C. Prudhomme, and V. Chabannes

Abstract The main objective of our task is to develop mathematical models, numerical techniques to analyse the thermal effects in electric machines, to implement the developed algorithm in multiprocessor or multi-core environments and to apply them to industrial use cases. In this study, we take into account coupled character of the electromagnetic and thermal features of the physical process. Both thermal and electromagnetic processes are considered transient, solved by means of the FEM method on independent meshes and the time-discretization is realized using time operator splitting. Two examples are presented to assess the accuracy of the developed coupled solvers and the numerical results are compared with the experimental ones, which are obtained from a prototype machine.

1 Introduction

The main objective of our task is to develop a finite element model to analyse the thermal effects in electric machines during its various operating conditions. In electrical machines the permanent magnets and the insulation in the stator windings are sensitive to temperature variations. In order to control the temperature rise and to avoid overheating, the prediction of the temperature distribution is required at the machine design stage.

Several numerical studies have been developed in order to study the thermal-magnetic coupling present in PM motors. In these studies, the approach employed is often based on 2D or 3D finite elements simulations, considering either strong

A. Lotfi (✉) · D. Marcsa · Z. Horváth
Department of Mathematics and Computational Science, Széchenyi István University, Győr, Hungary
e-mail: lotfi@sze.hu; marcsa@maxwell.sze.hu; horvathz@sze.hu

C. Prudhomme · V. Chabannes
Cemosis, IRMA UMR 7501, CNRS, University of Strasbourg, Strasbourg, France
e-mail: christophe.prudhomme@cemosis.fr; vincent.chabannes@cemosis.fr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_68

or weak coupling between the magnetic and thermal models. In strong coupling schemes, the electromagnetic and thermal computations are solved simultaneously in one global system of nonlinear equations. These procedures are not efficient in terms of the required computing time. By contrast, the weak coupling, used here, is based on solving independently and successively each sub-problem and then transferring the results between the two problems. This allows the use of appropriate numerical schemes to solve each sub-problem. Efficient implementation of this method might reduce the amount of computation time and memory requirements. The electromagnetic-thermal coupling proposed in this study is based on a coupling between the 2D transient electromagnetic computation with temperature-dependent material properties and the 3-D thermal computation using as heat sources the estimated losses from the electromagnetic model.

We start by introducing some standard notations. Let $[0, t_f]$ be the time interval of interest. The problem is set up in a bounded domain $\Omega = \cup\Omega_i \subset R^3$ consisting of a finite number of connected sub-domains. We first denote by $L^2(\Omega)$ the space of real valued measurable functions which are square integrable on Ω: $L^2(\Omega) = \{v : \Omega \longrightarrow \mathbb{R}, \int_\Omega v^2 d\Omega \leqslant \infty\}$. We define the function spaces: $H^1(\Omega) = \{v \in L_2(\Omega), \frac{\partial v}{\partial x_i} \in L_2(\Omega), 1 \leqslant i \leqslant d\}$, $H_1^0(\Omega) = \{v | v \in H^1(\Omega), v_{|\partial\Omega} = 0\}$ and $L^\infty(\Omega)$ is the space of bounded functions defined on Ω. Finally, let V be a function space and let $L^2([0, t_f]; V)$ be the Bochner space of functions in L^2 defined on $I = [0, t_f]$ with values in V.

2 Thermal Analysis of Electrical Machines

The thermal analysis of electric motors is essential today to prevent overheating problems. The power loss due to eddy current is converted into heat, resulting temperature rise. The properties of the materials change as their temperature changes and thus also the electromagnetic field. The accuracy of the thermal model depends on the material properties and the knowledge of losses in electrical machine. To achieve that, a coupled mathematical models of magneto-thermal problem is required.

The thermal model considers the transient heat transfer between all components of the PM motor including the air-gap, [1, 2]. During the simulation the air gaps inside the machine are defined as a solid domain, the heat is mostly transferred by conduction. The effective conductivity is calculated considering the state of the air flow in the air gap from empirical correlations, [2, 5, 7]. For stator and windings, the thermal and physical properties are considered as anisotropic and homogenisation technique is used in order to simplify the thermal model [5, 6]. Afterwards, the model does not take into account the effect of temperature on thermal properties.

The electromagnetic problem is performed under the conditions that the displacement currents are neglected and the electromagnetic field is two-dimensional, i.e. the magnetic vector potential has only one component in z-direction, this component does not depend on z, and the magnetization has the form $\overrightarrow{H}_c = (H_{cx}, H_{cy}, 0)$ and does not depend on z.

The mathematical model involves the solution of two submodels and the coupling is established by the temperature dependent materials and the electromagnetic losses. The two sub-models are presented as follows, [3, 4, 10, 11]:

(HT) Thermal equation	(EM) Magnetic equation
$\rho c_p \partial_t T + \nabla \cdot \Big([-\lambda]\nabla T\Big)$ $= Q(\overrightarrow{A}),$	$\sigma(T)\partial_t \overrightarrow{A} + \nabla \times \Big(\nu(\overrightarrow{A})\nabla \times \overrightarrow{A}\Big)$ $= \overrightarrow{J}_S + \nabla \times \overrightarrow{H}_{\mathrm{c}},$

where T is the temperature, c_p is the specific heat capacity, ρ represents the density, $[\lambda]$ is the thermal conductivity matrix, $\overrightarrow{A}$ is the magnetic vector potential, $Q(\overrightarrow{A})$ is the heat source, $\sigma(T)$ is the electrical conductivity which is dependent on the temperature, ν is the magnetic reluctivity which is dependent on the magnetic field intensity, $\overrightarrow{J_S}$ is the external current source and $\overrightarrow{H}_c$ is called the coercive field strength of the permanent magnet. The following boundary and initial conditions can be considered:

$\overrightarrow{n}.\Big([-\lambda]\nabla T\Big) = h(T - T_0), \quad \text{on} \quad \delta\Omega \times [0, t_f],$	$\overrightarrow{A} \times \overrightarrow{n} = \overrightarrow{0}, \quad \text{on} \quad \delta\Omega \times [0, t_f],$
$T(.,0) = T_0, \quad \text{on} \quad \Omega$	$\overrightarrow{A}(.,0) = \overrightarrow{A}_0, \quad \text{on} \quad \Omega$

where $\overrightarrow{n}$ is normal vector to the boundary, h is heat transfer coefficient and T_0, A_0 are the initial temperature distribution and the vector field at $t = 0$. The thermal and electromagnetic problems are interconnected by the following relations [12]:

$\sigma(T) = \sigma_0 \frac{1}{1+\alpha(T-T_0)},$	$Q(\overrightarrow{A}) = K_h f(B_m)^2 + K_c(fB_m)^2 + K_c(fB_m)^{1.5},$

where σ_0 is the electrical conductivity at reference temperature T_0, α is temperature coefficient, K_h, K_c, K_e are the static hysteresis loss, eddy current loss and excess loss coefficients and B_m is the peak flux density where $\overrightarrow{B} = rot(\overrightarrow{A})$.

In the following, the variational formulation of the problem is presented briefly. Using Green's formula, boundary condition and interface conditions we can derive the weak formulation of the thermo-magnetic problem:

Find $T \in L^2([0, t_f]; H^1(\Omega))$ and $A_z \in L^2([0, t_f]; H_0^1(\omega))$ such that:

$$a_1(\partial_t T, w) + b_1(T, w) = l_1(w), \forall w \in H_0^1(\Omega), \tag{1}$$

$$a_2(\partial_t A_z, w) + b_2(A_z, w) = l_2(w), \forall w \in H_0^1(\omega), \tag{2}$$

$a_1(u, v) = \int_\Omega \rho c_p u.w d\Omega,$ $b_1(u, v) = \int_\Omega (\nabla v)^T [\lambda](\nabla u) d\Omega + \int_{\partial\Omega} hu.v d\Gamma$ $l_1(v) = \int_\Omega Q(A).v d\Omega + \int_{\partial\Omega} hT_0.v,$	$a_2(u, v) = \int_\omega \sigma(T) u.w d\omega,$ $b_2(u, v) = \int_\omega (\nabla v)^T M_\nu (\nabla u) d\omega$ $l_2(v) = \int_\omega J_S w d\omega + \int_\omega rot_{2D} \overrightarrow{H}_c w d\omega$

where $M_\nu = \begin{bmatrix} \nu & 0 \\ 0 & \nu \end{bmatrix}$ is the magnetic reluctivity tensor and $rot_{2D} = \left(\frac{\partial H_{cy}}{\partial x} - \frac{\partial H_{cx}}{\partial y} \right)$.

The time scales of heat transfer and electromagnetic processes have different orders. The time scale for heat diffusion is greater than the time scale for electromagnetic effect. The coupled problem is multi-scale problem in time, therefore the time-discretization can be realized using time operator splitting. The application of this approach leads to the decomposition of the problem into two sub-problems and allows the use of appropriate numerical schemes to solve each time dependent sub-problem, [9]. To achieve this goal, the overall simulation time $[0, t_f]$ is divided into N time windows $[t_j, t_{j+1}]$ with $j = 0, 1, \ldots, N - 1$. The time t_j denotes the time where the coupling variables are exchanged between the two sub-problems. The size of each interval is determined based upon the variation of material properties in the electromagnetic problem. First, we assume that the temperature variations are not significant, meaning that the electrical conductivity remains approximately constant in each sub-interval and the parameter $\sigma(T)$ can be introduced explicitly in the electromagnetic problem (1), using the solution values from previous time step. Secondly, the problem (2) has a rapidly varying right-hand side due to variations of magnetic field produced by induced current. In this equation, for each sub-interval $[t_j, t_{j+1}]$, $Q(A)$ can be replaced by an average value over this time period as:

$$\tilde{\mathrm{Q}}(A) = \frac{1}{t_{j+1} - t_j} \int_{t_j}^{t_{j+1}} Q(A) dt. \tag{3}$$

Finally, with this two assumptions, the problem is decoupled and the vector potential equation and the heat equation can be solved independently with internal step-sizes using the following scheme:

```
0. Initialization step:
    - Set window counter n := 0 and initial values $T_0$ ,
      $A_0$ .Go to Step 1).
1. Solve the coupled problem.
    a. Find $A_z \in L^2([t_n, t_f]; H_0^1(\omega))$ such that:

           $a_2(\partial_t A_z, w) + b_2(A_z, w) = l_2(w), \forall w \in H_0^1(\omega),$      (4)
              with initial value $A_n$ and $\sigma = \sigma(T_n)$.

    b. Find $T \in L^2([t_n, t_f]; H^1(\Omega))$ such that:

           $a_1(\partial_t T, w) + b_1(T, w) = l_1(w), \forall w \in H_0^1(\Omega),$      (5)
           with initial value $T_n$ and $Q(A) = \tilde{Q}(A_z)$.

2. Set the value for $t_{n+1}$ as indicated below, If $t_{n+1} \geq t_f$
   then go to Step 3), else set $n := n+1$, $A_{n+1}$ and $T_{n+1}$ are
   the calculated solution of (4) and (5),go to Step 1)
   for the next window.
3. Stop.
```

At each step of the previous algorithm, the time discretization is performed as follow:

- We replace the time derivative with the backward difference quotient in (4) using the time step δt. Indeed, starting with $(A_z)^0 = A_n$, we can successively compute the unknowns $(A_z)^k$ at time $(t_n + k\delta t)$, $k = 1, 2, \ldots$, by repeatedly solving the obtained nonlinear equation until steady-state solution is reached.
- We replace the time derivative with the backward difference quotient in (5) using the time step $\Delta t >> \delta t$. Indeed, starting with $T^0 = T_n$, we can successively compute the unknowns T^k at time $(t_n + k\Delta t)$, $k = 1, 2, \ldots$, by repeatedly solving the obtained linear equation. During the resolution of the thermal problem, the iteration process is stopped when the variation of the relative electrical conductivity is higher than a threshold defined by the user, as:

$$\frac{|\sigma(T^{k+1}) - \sigma(T^k)|}{\sigma(T^k)} > \epsilon, \tag{6}$$

thus the new communication time is $t_{n+1} = t_n + (k+1)\Delta t$ and the EM simulation is re-launched.

3 Verification of the Proposed Coupled Model

To validate the effectiveness of the coupled thermo-magnetic model and code, two examples are presented. The first examines the transient thermal analysis of a 18-slots/16-poles PMSM with exterior rotor (BMW C1 11 kW) manufactured by the Vehicle Developing Centre of the Széchenyi István University, [8]. The second presents a numerical computation for the heat transfer in Toyota Prius 2004 electric motor.

3.1 *Case 1: The First Example: BMW C1 11 kW Electric Motor*

The measurements were performed on the BMW C1 11 kW motor presented in Fig. 1. The temperature was measured using a thermocouple type PT100 in contact with the surface of the tested motor. As shown in Fig. 1, the electric motor model has a very complicated geometry. The stator coils are supplied with a maximum current of 100 A for duration of 1800 s and then the current is turned off; the rotor is locked in a stand still position. During the temperature test, the most significant heat source in the tested motor is copper losses. The left figure in Fig. 2 shows the instantaneous Joule losses calculated from experimental data, used in thermal analysis. The computational grids used for this simulation, shown in Fig. 1, has approximately 4,000,000 tetrahedral elements. According to the model mentioned above, a transient 3D FE thermal was carried out using the FEEL++ library [13] and it was executed on 30 processors. Figure 2 shows the temperature distribution of the motor at time t = 1600 s. Figure 3 shows a comparison of the simulated and measured temperature variation obtained by a few thermocouples [8]. The simulated

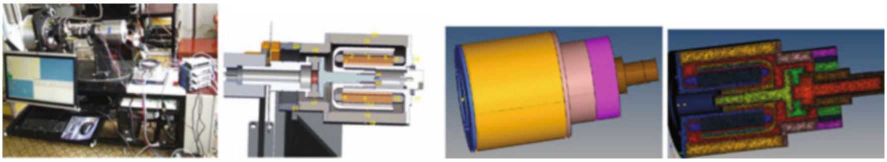


Fig. 1 Temperature test set up for the prototype machine, sensors position, geometry, 3D mesh

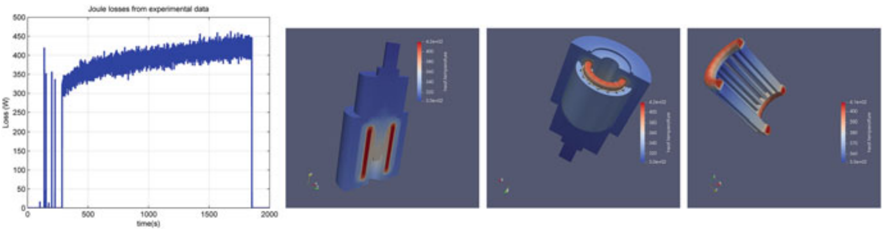


Fig. 2 Joule losses from experimental data, contours of temperature of the electric motor

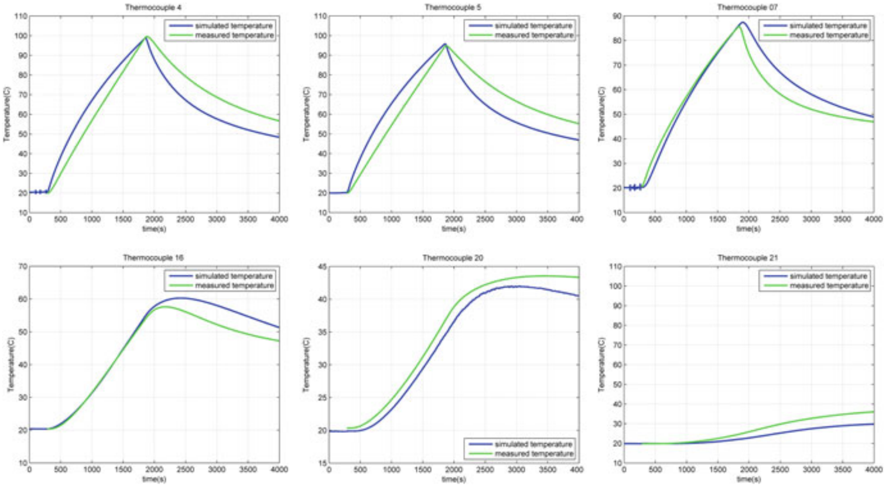


Fig. 3 Simulated and measured temperature variation at different position of PM motor

temperatures are very accurate for locations in the motor obtained by thermocouples 1, 2, 3, 4, 5, 6, 9, 10, 11, 12, but the results from the model are not in good agreement with the results obtained by the remaining thermocouples. However, the reasons for the discrepancy between transient temperature simulation and measurement need to be explained. The difference between transient temperature simulation and the results of the experiments can be attributed to the following: the glue layers between the thermocouples and the prototype motor can decrease heat transfer, the material properties were assumed to be independent of temperature.

3.2 *Case 2: The Second Example: Toyota Prius 2004 Electric Motor*

The developed numerical technique is used to simulate distributions of electromagnetic and thermal fields in Toyota Prius 2004 electric motor, as shown in the left figure in Fig. 4. The 3D mesh used for the for the thermal analysis has more than

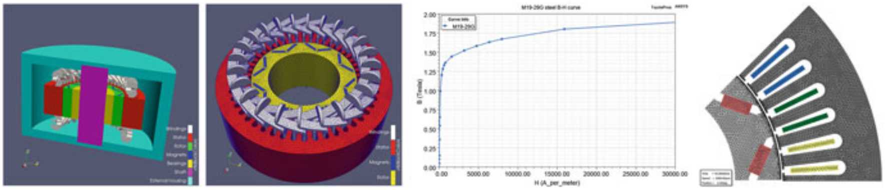


Fig. 4 Toyota Prius 2004 electric motor; 3D mesh for the thermal analysis; B(H) curve; 2D mesh

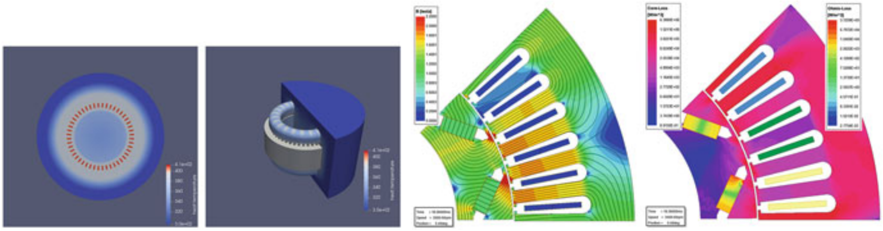


Fig. 5 Contours of temperature of the electric motor, magnetic flux density and Ohmic loss

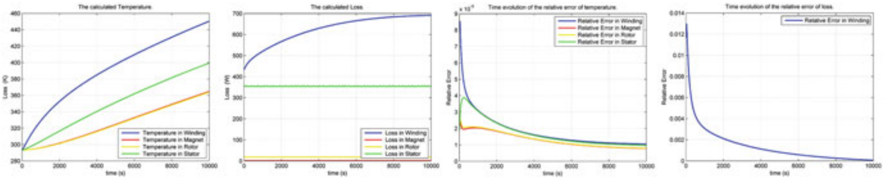


Fig. 6 Time evolution of the temperatures, losses and relative errors

7,500,000 cells and it is shown in Fig. 4. The electromagnetic analysis is performed on a periodic section of the PM considering a soft ferromagnetic material with the B(H) curve shown in Fig. 4. The thermal simulation was executed on 60 cores using the FEEL++ library, [13]. The electromagnetic simulation was performed with the help of the finite element package ANSYS Maxwell.

Figure 5 represents the temperature distribution, magnetic flux density distribution with equipotential lines and Ohmic loss in magnets with magnetic flux density vectors. These results are obtained at maximum current of 100 A and angular velocity 3000 rpm.

The first two figures in Fig. 6 show the calculated time evolution of the temperature and losses in different parts of the PM motor and the last two represent the convergence history of the relative errors.

4 Conclusions

The developed method, electromagnetic field and thermal linked analysis, gives the possibility to evaluate magnetic field intensity, the core losses in the material and the temperature distribution of PM motor for different currents and for different geometrical parameters and the result of the simulation allows a better understanding of the thermal behaviour. The developed model enables to predict temperature distribution with good accuracy of the critical parts of the electric machine such the winding, the rotor and the magnets without using the time-consuming CFD simulations. Convection heat-transfer problems are treated with dimensionless numbers and empirical correlations are used to determine heat-transfer coefficient.

Acknowledgments This work was supported by the FIEK program (Center for cooperation between higher education and the industries at the Széchenyi István University, EFOP-3.6.2-16-2017-00015). Vincent Chabannes and Christophe Prud'homme acknowledge the funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 731063.

References

1. J. Gieras, R. Wang and M. Kamper "Axial Flux Permanent Magnet Brushless Machines", Springer (2008).
2. P. H. Mellor, D. Roberts and D. R. Turner "Lumped parameter thermal model for electrical machines of TEFC design" IEE Proceedings-B, 138 (1991) 5, p. 205–218.
3. A. Arkkio, T. Jokinen, and E. Lantto, "Induction and permanent-magnet synchronous machines for high-speed applications," in Proc. 8th ICEMS, Sep. 27–29, 2005, vol. 2, pp. 871–876.
4. O. Aglén, and Å. Anderson, "Thermal analysis of a high-speed generator," in Proc. 38th IAS annual meeting, Oct. 12–16, 2003, pp. 547–554.
5. D.A Howey, P.R.N. Childs, and A.S. Holmes, "Air-gap convection in rotating electrical machines," IEEE Transactions on Industrial Electronics, vol. 59, pp. 1367–1375, 2012.
6. J. Gyselinck, P. Dular, N. Sadowski, P. Kuo-Peng, R. V. Sabariego, "Homogenization of Form-Wound Windings in Frequency and Time Domain Finite-Element Modeling of Electrical Machines", IEEE Transactions on Magnetics, vol. 46, pp. 2852–2855, Aug. 2010.
7. Incropera, Frank P.; DeWitt, David P. "Fundamentals of Heat and Mass Transfer (4th ed.)", New York: Wiley. p. 493, (2000).
8. Kuslits M. "BMW C1 motor rövidzárási hőmérsékletmérése: Mérési jegyzőkönyv", 2014. (in Hungarian language)
9. S. Clain, J. Rappaz, M. Swierkosz, and R. Touzani, Numerical modeling of induction heating for two-dimensional geometries, Mathematical models and methods in applied sciences, 3 (1993), pp. 805–8
10. Jackson J. D. (1999) Classical Electrodynamics. In: John Wiley & Sons, Inc.
11. M. Rosu, P. Zhou, D. Lin, D. Ionel, M. Popescu, F. Blaabjerg, V. Rallabandi, D. Staton, Multiphysics Simulation by Design for Electrical Machines, Power Electronics and Drives (Wiley - IEEE Press, New Jersey, 2018), p. 312.
12. G. Bertotti, General properties of power losses in soft ferromagnetic materials, IEEE Transactions on Magnetics **24**, 621–630 (1988).
13. The Feel++ Book: https://book.feelpp.org/

Parameter Robust Preconditioning for Multi-Compartmental Darcy Equations

Eleonora Piersanti, Marie E. Rognes, and Kent-Andre Mardal

Abstract In this paper, we propose a new finite element solution approach to the multi-compartmental Darcy equations describing flow and interactions in a porous medium with multiple fluid compartments. We introduce a new numerical formulation and a block-diagonal preconditioner. The robustness with respect to variations in material parameters is demonstrated by theoretical considerations and numerical examples.

1 Introduction

The multi-compartment Darcy equations[1] extend the single compartment Darcy model and describe fluid pressures in a rigid porous medium permeated by multiple interacting fluid networks. These equations have been used to model perfusion in e.g. the heart [4, 8], the brain [3] and the liver [1]. The static variant of the equations read as follows: for a given number of networks $J \in \mathbb{N}$, find the network pressures p_j for $j = 1, \dots, J$ such that

$$- K_j \operatorname{div} \nabla p_j + \sum_{i=1}^{J} \xi_{j \leftarrow i}(p_j - p_i) = g_j \quad \text{in } \Omega, \tag{1}$$

[1] In this paper, we will also refer to these equations as the multiple–network porosity (MPT) equations.

E. Piersanti · M. E. Rognes
Simula Research Laboratory, Lysaker, Norway
e-mail: eleonora@simula.no; meg@simula.no

K.-A. Mardal (✉)
Department of Mathematics, University of Oslo, Oslo, Norway
e-mail: kent-and@math.uio.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_69

where $p_j = p_j(x)$ for $x \in \Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$, and Ω is the physical domain. The scalar parameter $K_j > 0$ represents the permeability of each network j. The parameter $\xi_{j\leftarrow i} \geq 0$ is the exchange coefficient into network j from network i. These are assumed to be symmetric: $\xi_{j\leftarrow i} = \xi_{i\leftarrow j}$. The right hand side g_j can be interpreted as a source/sink term for each j. For simplicity, let $p_j = 0$ on $\partial\Omega$ for $1 \leq j \leq J$.

The system of equations is elliptic as long as $K_j > 0$, but for $K_j \ll \xi_j$ the diagonal dominance is lost for smooth components for which $\|K_j^{1/2}\nabla p_j\| \leq \|\xi_j^{1/2} p_j\|$. As diagonal dominance is often exploited in standard preconditioning algorithms such as for example multigrid, the consequence is a loss of performance. Here, we will therefore propose a transformation of the system of equations that enable the use of standard preconditioners. In detail, we propose and analyze a new approach to constructing finite element formulations and associated block–diagonal preconditioners of the system (1). The key idea is to change variables through a transformation T that gives simultaneous diagonalization by congruence of the operators involved. We preface and motivate the new approach by a demonstration of lack of robustness of a standard formulation for high exchange parameters.

2 Lack of Parameter Robustness in Standard Formulation

A standard variational formulation of (1) reads as follows: find $p_j \in H_0^1 = H_0^1(\Omega)$ for $1 \leq j \leq J$ such that:

$$\left(K_j \nabla p_j, \nabla q_j\right) + \textstyle\sum_{i=1}^{J} \left(\xi_{j\leftarrow i}(p_j - p_i), q_j\right) = \left(g_j, q_j\right) \quad \forall q_j \in H_0^1, \tag{2}$$

where $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ inner product. The system (2) can be written in the alternative form:

$$k(\mathbf{p}, \mathbf{q}) + e(\mathbf{p}, \mathbf{q}) = (\mathbf{g}, \mathbf{q})\,, \tag{3}$$

with $\mathbf{p} = (p_1, p_2, \dots, p_J)$, $\mathbf{q} = (q_1, q_2, \dots, q_J)$, $\mathbf{g} = (g_1, g_2, \dots, g_J)$, and with matrix form

$$\mathcal{A}\mathbf{p} = \mathbf{g},$$

where

$$\mathcal{A} = \mathcal{K} + E = \begin{pmatrix} K_1\Delta & 0 & \cdots & 0 \\ 0 & K_2\Delta & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & K_J\Delta \end{pmatrix} + \begin{pmatrix} \sum_{i=1}^{J}\xi_{1\leftarrow i} & -\xi_{1\leftarrow 2} & \cdots & -\xi_{1\leftarrow J} \\ -\xi_{1\leftarrow 2} & \sum_{i=1}^{J}\xi_{2\leftarrow i} & \cdots & -\xi_{2\leftarrow J} \\ \vdots & \vdots & \ddots & \vdots \\ -\xi_{1\leftarrow J} & -\xi_{2\leftarrow J} & \cdots & \sum_{i=1}^{J}\xi_{J\leftarrow i} \end{pmatrix}.$$

Taking the blocks on the diagonal of $\mathcal{A}$ we can immediately define a block diagonal preconditioner $\mathcal{B}$:

$$\mathcal{B} = \operatorname{diag}\left(-K_1\Delta + \sum_{i=1}^{J} \xi_{1\leftarrow i}, -K_2\Delta + \sum_{i=1}^{J} \xi_{2\leftarrow i}, \cdots, -K_J\Delta + \sum_{i=1}^{J} \xi_{J\leftarrow i}\right) \tag{4}$$

Alas, this formulation and preconditioner is not robust for high exchange parameters as illustrated by the following example.

Example 1 In this example we illustrate the poor performance of the block diagonal preconditioner (4) for the standard finite element discretization of the MPT equations (1) with $J = 2$. In particular, we show that the proposed preconditioner is not robust with respect to the exchange coefficient $\xi_{1\leftarrow 2}$ and mesh refinement. Let $\Omega = [0, 1]^2 \subset \mathbb{R}^2$, and let $K_1 = K_2 = 1.0$, $g_j = 0$. To discretize the pressures p_1, p_2 we consider continuous piecewise linear finite elements defined relative to a $2N \times N$ triangular mesh of Ω. The results in Table 1 show that both the number of iterations and condition numbers increase somewhat less than linearly (predicted by our theoretical analysis) in $\xi_{1\leftarrow 2}$, for $\xi_{1\leftarrow 2}$ above a threshold > 100. The number of iterations also grow for increasing N (decreasing mesh size h) in this case.

We can examine Example 1 analytically. Define the induced norm

$$\|\mathbf{p}\|_{\mathcal{B}}^2 = (\mathcal{B}\mathbf{p}, \mathbf{p}) = \textstyle\sum_{j=1}^{J} \left(K_j \nabla p_j, \nabla p_j\right) + \xi_j \left(p_j, p_j\right), \tag{5}$$

where $\xi_j = \sum_{i=1}^{J} \xi_{j\leftarrow i}$. We can show that there exists an $\alpha > 0$ such that

$$(\mathcal{A}\mathbf{p}, \mathbf{p}) \geq \alpha\, (\mathcal{B}\mathbf{p}, \mathbf{p}) \tag{6}$$

for all $\mathbf{p}$, but depending on K_j and $\xi_{j\leftarrow i}$, as follows. Note that for all $\mathbf{p}$

$$(\mathcal{A}\mathbf{p}, \mathbf{p}) = ((\mathcal{K} + E)\mathbf{p}, \mathbf{p}) \geq (\mathcal{K}\mathbf{p}, \mathbf{p}), \tag{7}$$

Table 1 Number of iterations (and condition number estimates) of a CG solver of the system (1) with an algebraic multigrid (Hypre AMG) preconditioner of the form (4) with a random initial guess. Results for $\xi_{1\leftarrow 2} = 10^{-4}, 10^{-6}$ are nearly identical to the 10^{-2} case

	N				
$\xi_{1\leftarrow 2}$	8	16	32	64	128
10^{-2}	3 (1.0)	4 (1.1)	4 (1.1)	4 (1.1)	4 (1.1)
10^{0}	4 (1.1)	4 (1.1)	5 (1.1)	5 (1.1)	5 (1.1)
10^{2}	29 (11)	30 (11)	28 (11)	25 (11)	24 (11)
10^{4}	215 (1053)	740 (1026)	1131 (1012)	1232 (1014)	1058 (1014)
10^{6}	7 (2.0)	20 (581)	84 (686)	394 (1140)	1467 (1755)

since

$$(E\mathbf{p},\mathbf{p}) = \sum_{i=1}^{J}\sum_{j=1}^{J}\left(\xi_{j\leftarrow i}(p_j - p_i), p_j\right) = \frac{1}{2}\sum_{j=1}^{J}\sum_{i=1}^{J}\xi_{j\leftarrow i}\|p_j - p_i\|^2 \geq 0.$$

By definition and by applying the Poincaré inequality, we find that there exists a constant C_Ω depending on the domain Ω, such that

$$(\mathcal{K}\mathbf{p},\mathbf{p}) = \sum_{j=1}^{J}\frac{K_j}{2}\|\nabla p_j\|^2 + \frac{K_j}{2}\|\nabla p_j\|^2 \geq \frac{1}{2}\sum_{j=1}^{J} K_j\|\nabla p_j\|^2 + \frac{C_\Omega K_j}{\xi_j}\xi_j\|p_j\|^2. \tag{8}$$

Thus, using the definition of $\mathcal{B}$, we obtain that

$$(\mathcal{K}\mathbf{p},\mathbf{p}) \geq \frac{1}{2}\min\left(1, \min_j \frac{C_\Omega K_j}{\xi_j}\right)(\mathcal{B}\mathbf{p},\mathbf{p})\,. \tag{9}$$

We observe that the coercivity constant depends on the permeability and exchange parameters and is such that it vanishes for vanishing ratios of K_j to ξ_j.

We can also show that there exists a constant β such that

$$(\mathcal{A}\mathbf{p},\mathbf{q}) \leq \beta\|\mathbf{p}\|_{\mathcal{B}}\|\mathbf{q}\|_{\mathcal{B}}. \tag{10}$$

For any $\mathbf{p}$ and $\mathbf{q}$, applying the Cauchy–Schwartz inequality twice we obtain

$$(\mathcal{A}\,\mathbf{p},\mathbf{q}) \leq \sum_{j=1}^{J}\left(K_j\|\nabla p_j\|\|\nabla q_j\| + \sum_{i=1}^{J}\xi_{j\leftarrow i}(\|p_j\| + \|p_i\|)\|q_j\|\right).$$

Applying the Cauchy–Schwartz inequality, the diffusion term is bounded as follows

$$\sum_{j=1}^{J} K_j\|\nabla p_j\|\|\nabla q_j\| \leq \left(\sum_{j=1}^{J} K_j\|\nabla p_j\|^2\right)^{1/2}\left(\sum_{j=1}^{J} K_j\|\nabla q_j\|^2\right)^{1/2}.$$

For the exchange terms, we can use the Cauchy–Schwartz inequality, the symmetry of the exchange coefficients and Chebyshev's inequality to show that

$$\sum_{j=1}^{J}\sum_{i=1}^{J}\xi_{j\leftarrow i}\|p_i\|\|q_j\| \leq J\left(\sum_{j=1}^{J}\xi_j\|p_j\|^2\right)^{1/2}\left(\sum_{j=1}^{J}\xi_j\|q_j\|^2\right)^{1/2},$$

and similarly for $\|p_j\|$ in place of $\|p_i\|$. Thus (10) holds with continuity constant β equal to $J + 1$.

The condition number of the preconditioned continuous system can be estimated as the ratio between (10) and (8), c.f. for example [7], and tends to ∞ as $\xi_{j\leftarrow i} \to \infty$. CG convergence is governed by the square root of the condition number which in Example 1, explains how the number of iterations increase as $\xi_{1\leftarrow 2}$ grows in Table 1.

3 Change of Variables Yields Parameter Robust Formulation

In this section, we present a new approach to variational formulations for the MPT equations. The key idea is to change from variables $\mathbf{p}$ to variables $\tilde{\mathbf{p}}$ via a transformation T such that the equation operators decouple. We can show that this is always possible by simultaneous diagonalization of matrices by congruence.

To this end, we define $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{q}}$ as a new set of variables such that

$$\mathbf{p} = T\tilde{\mathbf{p}}, \quad \mathbf{q} = T\tilde{\mathbf{q}}. \tag{11}$$

for a linear transformation map (matrix) $T : \mathbb{R}^J \to \mathbb{R}^J$ to be further specified. Substituting (11) into (3), we obtain a new variational formulation reading as: find $\tilde{\mathbf{p}} \in (H_0^1)^J$ such that

$$k(T\tilde{\mathbf{p}}, T\tilde{\mathbf{q}}) + e(T\tilde{\mathbf{p}}, T\tilde{\mathbf{q}}) = \left(T^T\mathbf{g}, \tilde{\mathbf{q}}\right) \quad \forall\, \tilde{\mathbf{q}} \in (H_0^1)^J. \tag{12}$$

The matrix form of the system is

$$\tilde{\mathcal{A}}\tilde{\mathbf{p}} = (\tilde{\mathcal{K}} + \tilde{E})\tilde{\mathbf{p}} = T^T\mathbf{g} = \tilde{\mathbf{g}}, \tag{13}$$

where

$$\tilde{\mathcal{K}} = (-\Delta)\tilde{K}, \quad \tilde{K} = T^TKT, \quad \tilde{E} = T^TET, \tag{14}$$

where the matrix $E \in \mathbb{R}^J \times \mathbb{R}^J$ is given in Sect. 2 and where we write $K = \operatorname{diag}(K_1, K_2, \dots, K_J)$.

The key question is now whether there exists an (invertible) transformation T that simultaneously diagonalizes (by congruence) K and E? More precisely, is there a matrix $T \in \mathbb{R}^J \times \mathbb{R}^J$ such that

$$\tilde{K} = \operatorname{diag}(\tilde{K}_1, \tilde{K}_2, \dots, \tilde{K}_J), \quad \tilde{E} = \operatorname{diag}(\tilde{\xi}_1, \tilde{\xi}_2, \dots, \tilde{\xi}_J) \quad ? \tag{15}$$

By matrix analysis theory, see e.g. [2, Theorem 4.5.17, p. 287], there exists indeed such a T since K is diagonal and non-singular and E is symmetric and thus $C = K^{-1}E$ is diagonalizable. In particular, consider the case where C has J distinct eigenvalues λ_j and eigenvectors v_j for $j = 1, \dots, J$. By taking $T = [v_1, v_2, \dots, v_J]$, (15) holds. Moreover, the eigenvalues λ_j are all real.

Example 2 To exemplify, we here show the diagonalization by congruence of a general 2-network system explicitly. Let

$$K = \begin{pmatrix} K_1 & 0 \\ 0 & K_2 \end{pmatrix}, \quad E = \begin{pmatrix} \xi_{1\leftarrow 2} & -\xi_{1\leftarrow 2} \\ -\xi_{1\leftarrow 2} & \xi_{1\leftarrow 2} \end{pmatrix}.$$

Then,

$$C = K^{-1}E = \begin{pmatrix} \xi_{1\leftarrow 2}/K_1 & -\xi_{1\leftarrow 2}/K_1 \\ -\xi_{1\leftarrow 2}/K_2 & \xi_{1\leftarrow 2}/K_2, \end{pmatrix}$$

has eigenvalues $e_1 = 0$ and $e_2 = \xi_{1\leftarrow 2}(K_1 + K_2)/(K_1K_2)$ and the eigenvectors form the columns of T:

$$T = \begin{pmatrix} 1 & K_2(\xi_{1\leftarrow 2}/K_2 - \xi_{1\leftarrow 2}(K_1 + K_2)/(K_1K_2))/\xi_{1\leftarrow 2} \\ 1 & 1 \end{pmatrix},$$

Finally, we can verify that

$$\tilde{K} = T^T K T = \begin{pmatrix} K_1 + K_2 & 0 \\ 0 & K_2(K_1 + K_2)/K_1 \end{pmatrix},$$

$$\tilde{E} = T^T E T = \begin{pmatrix} 0 & 0 \\ 0 & \xi_{1\leftarrow 2}(K_1^2 + K_1K_2 + K_2(K_1 + K_2))/K_1^2 \end{pmatrix}.$$

As the transformed system is diagonal and decoupled, a block-diagonal preconditioner is readily available. In particular, we define

$$\tilde{\mathcal{B}} = \tilde{\mathcal{A}} = \operatorname{diag}\left(-\tilde{K}_1\Delta + \tilde{\xi}_1, -\tilde{K}_2\Delta + \tilde{\xi}_2, \ldots, -\tilde{K}_J\Delta + \tilde{\xi}_J\right). \tag{16}$$

with norm

$$\|\tilde{\mathbf{p}}\|_{\tilde{\mathcal{B}}}^2 = \left(\tilde{\mathcal{B}}\tilde{\mathbf{p}}, \tilde{\mathbf{p}}\right) = \sum_{j=1}^{J} \left(\tilde{K}_j \nabla \tilde{p}_j, \nabla \tilde{p}_j\right) + \tilde{\xi}_j \left(\tilde{p}_j, \tilde{p}_j\right). \tag{17}$$

Clearly, by definition, $\tilde{\mathcal{A}}$ and $\tilde{\mathcal{B}}$ are trivially spectrally equivalent (with upper and lower bounds independent of the material parameters).

4 Numerical Examples for the New Formulation

In this section, we present numerical results supporting the theoretical considerations. All numerical experiments have been conducted using a finite element discretization, using the FEniCS library [5] and the cbc.block package [6]. To discretize the pressures p_j and the transformed variables $\tilde{p}_j$, we consider continuous piecewise linear (P_1) finite elements defined relative to each mesh $\mathcal{T}_h$ of the domain $\Omega = [0, 1]^2$. We impose homogeneous Dirichlet conditions on the whole boundary,

Table 2 Number of iterations (and condition number estimates) of a CG solver of the system (2) with an algebraic multigrid (Hypre AMG) preconditioner of the form (4)

$\xi_{1\leftarrow 2}$	K_2	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 128$
10^4	10^{-6}	277 (2139)	1178 (2135)	2395 (2035)	3001 (2034)	3001 (2034)
	10^{-4}	280 (2139)	1180 (2135)	2283 (2035)	2860 (2034)	3001 (2034)
	10^{-2}	275 (2117)	1181 (2113)	2325 (2014)	2859 (2013)	2988 (2011)
	10^0	242 (1054)	935 (1026)	1629 (1012)	1556 (1014)	1557 (1014)
	10^2	62 (21)	74 (22)	74 (22)	66 (22)	64 (22)
	10^4	12 (1.6)	11 (1.6)	11 (1.6)	11 (1.6)	10 (1.6)
	10^6	7 (1.1)	7 (1.1)	7 (1.1)	7 (1.1)	7 (1.1)
10^6	10^{-6}	138 (34,499)	692 (42,936)	2999 (45,584)	3001 (17,128)	3001 (5730)
	10^{-4}	133 (33,287)	773 (43,459)	2967 (45,532)	3001 (17,192)	3001 (5774)
	10^{-2}	141 (36,327)	695 (41,605)	2982 (45,144)	3001 (16,773)	3001 (5657)
	10^0	366 (105,246)	1816 (111,467)	3001 (22,961)	3001 (9060)	3001 (3623)
	10^2	280 (2117.4)	1110 (2113)	2608 (2014)	3001 (2013)	2979 (2011)
	10^4	65 (22)	77 (22)	74 (22)	67 (22)	64 (22)
	10^6	12 (1.6)	12 (1.6)	11 (1.6)	11 (1.6)	10 (1.6)

and zero right hand side(s). The linear systems were solved using a conjugate gradient (CG) solver, with algebraic multigrid (Hypre AMG) with the respective preconditioners, starting from a random initial guess. The tolerance is set to 10^{-9}, iterations are stopped at 3000, the condition number is just an estimation provided by the Krylov spaces involved in the iterations and will be lower than the real value.

Example 3 We first compare the performance of the preconditioners (4) and (16). We let $K_1 = 1.0$, and consider different values of the parameters $K_2, \xi_{1\leftarrow 2}$ and different mesh resolutions N. For the standard formulation (Table 2), the number of iterations (and condition number) is not bounded and increases with the ratio between $\xi_{1\leftarrow 2}$ and K_2. We see that the growth is somewhat less than the predicted linear growth. In contrast, for the new formulation (Table 3), we observe that both the number of iterations and the condition number stays nearly constant across the whole range of parameter values tested.

Example 4 In this final example, we study the performance of the preconditioner (16) for three networks. We report the results for $K_1 = 1.0$, and different values of the parameters K_2, $K_3, \xi_{1\leftarrow 2}, \xi_{1\leftarrow 3}, \xi_{2\leftarrow 3} = (10^{-4}, 10^{-2}, 10^0, 10^2, 10^4)$ and different mesh resolutions $N = (16, 32, 64)$. The results are shown in Fig. 1. We observe that the number of iterations stays between 4 and 6 across the whole range of parameters tested, with condition numbers estimated in the range 1.0–1.25.

Table 3 Number of iterations (and condition number estimates) of a CG solver of the system (12) with an algebraic multigrid (Hypre AMG) preconditioner of the form (16)

$\xi_{1\leftarrow 2}$	K_2	$N = 8$	$N = 16$	$N = 32$	$N = 64$	$N = 128$
10^4	10^{-6}	8 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)
	10^{-4}	8 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)
	10^{-2}	8 (1.2)	9 (1.2)	9 (1.2)	8 (1.2)	7 (1.1)
	10^0	8 (1.2)	8 (1.1)	6 (1.1)	6 (1.1)	6 (1.1)
	10^2	8 (1.1)	7 (1.1)	6 (1.1)	6 (1.1)	6 (1.1)
	10^4	7 (1.1)	6 (1.1)	6 (1.1)	6 (1.1)	7 (1.1)
	10^6	7 (1.1)	6 (1.1)	6 (1.1)	6 (1.1)	7 (1.1)
10^6	10^{-6}	8 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)
	10^{-4}	8 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)
	10^{-2}	8 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)
	10^0	8 (1.2)	9 (1.2)	9 (1.2)	9 (1.2)	8 (1.1)
	10^2	8 (1.2)	9 (1.2)	9 (1.2)	8 (1.2)	7 (1.1)
	10^4	8 (1.2)	9 (1.2)	9 (1.2)	8 (1.2)	7 (1.1)
	10^6	8 (1.2)	8 (1.2)	8 (1.2)	8 (1.2)	7 (1.1)

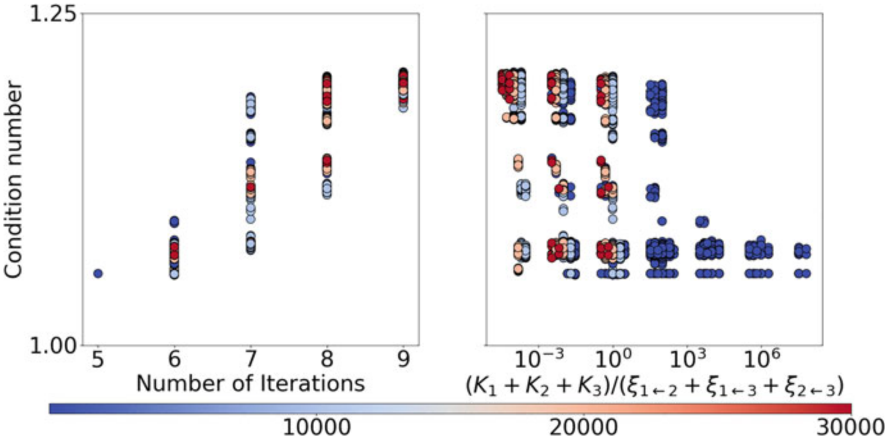


Fig. 1 Example 4: each point on the graphs represents a simulation performed with different parameters. The color represents the magnitude of $\xi_{1\leftarrow 2} + \xi_{1\leftarrow 3} + \xi_{2\leftarrow 3}$ from smaller (blue) to larger (red). Left: the condition number of the operator versus the number of iterations. Right: condition number versus the ratio between the sum of $\xi_{j\leftarrow i}$ and sum of K_j (x-axis is logarithmic y-axis is linear)

5 Conclusion

In this paper we have introduced a transformation, based on the congruence of the involved matrices, that transforms MPT systems to a form where diagonal block preconditioners are highly effective. The transformation removes a problem that elliptic systems may have when the elliptic constant is small compared to the continuity constant because of large low order terms.

Acknowledgments The authors would like to acknowledge Jeonghun J. Lee (Baylor University) and Travis Thompson (University of Oxford) for constructive discussions related to this work.

References

1. J. Brašnová, V. Lukeš, and E. Rohan. Identification of multi-compartment Darcy flow model material parameters. 2018.
2. R. A. Horn and C. R. Johnson. *Matrix Analysis*. 2nd edition, 1990. Cambridge University press.
3. T. Józsa, W. El-Bouri, R. Padmos, S. Payne, and A. Hoekstra. A cerebral circulation model for in silico clinical trials of ischaemic stroke. pages 25–27, 2019. CompBioMed Conference 2019.
4. J. Lee, A. Cookson, R. Chabiniok, S. Rivolo, E. Hyde, M. Sinclair, C. Michler, T. Sochi, and N. Smith. Multiscale modelling of cardiac perfusion. In *Modeling the heart and the circulatory system*, pages 51–96. Springer, 2015.
5. A. Logg, K.-A. Mardal, and G. Wells. *Automated solution of differential equations by the finite element method: The FEniCS book*, volume 84. Springer Science & Business Media, 2012.
6. K.-A. Mardal and B. H. J. Block preconditioning of systems of PDEs. pages 643–655. Heidelberg, Springer, Berlin, 2012.
7. K.-A. Mardal and R. Winther. Preconditioning discretizations of systems of partial differential equations. *Numerical Linear Algebra with Applications*, 18:1–40, 2011.
8. C. Michler, A. Cookson, R. Chabiniok, E. Hyde, J. Lee, M. Sinclair, T. Sochi, A. Goyal, G. Vigueras, D. Nordsletten, et al. A computationally efficient framework for the simulation of cardiac perfusion using a multi-compartment Darcy porous-media flow model. *International journal for numerical methods in biomedical engineering*, 29(2):217–232, 2013.

Scaling of the Steady-State Load Flow Equations for Multi-Carrier Energy Systems

A. S. Markensteijn, J. E. Romate, and C. Vuik

Abstract Coupling single-carrier networks (SCNs) into multi-carrier energy systems (MESs) has recently become more important. Steady-state load flow analysis of energy systems leads to a system of nonlinear equations, which is usually solved using the Newton-Raphson method (NR). Due to various physical scales within a SCN, and between different SCNs in a MES, scaling might be needed to solve the nonlinear system. In single-carrier electrical networks, per unit scaling is commonly used. However, in the gas and heat networks, various ways of scaling or no scaling are used. This paper presents a per unit system and matrix scaling for load flow models for a MES consisting of gas, electricity, and heat. The effect of scaling on NR is analyzed. A small example MES is used to demonstrate the two scaling methods. This paper shows that the per unit system and matrix scaling are equivalent, assuming infinite precision. In finite precision, the example shows that the NR iterations are slightly different for the two scaling methods. For this example, both scaling methods show the same convergence behavior of NR in finite precision.

1 Introduction

Multi-carrier energy systems (MESs) have become more important over the years, as the need for efficient, reliable and low carbon energy systems increases. In these energy systems, different energy carriers, such as gas, electricity, and heat, interact

A. S. Markensteijn (✉) · C. Vuik
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands
e-mail: A.S.Markensteijn@tudelft.nl

J. E. Romate
Shell Global Solutions International B.V., The Hague, The Netherlands

Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_70

with each other leading to one integrated energy network. An important tool for the design and operation of energy systems is steady-state load flow (LF) analysis of the energy networks. LF models for single-carrier networks (SCNs) have been widely studied, but only recently LF models for MESs have been proposed.

Steady-state LF analysis leads to a system of nonlinear equations, which is usually solved using the Newton-Raphson method (NR). The quantities in the LF equations can be several orders of magnitude apart, such that scaling might be needed to solve the nonlinear system.

In single-carrier electricity networks, per unit scaling is generally used (e.g. [1]). In the per unit system, every variable and parameter is scaled to obtain dimensionless equations. In gas and heat networks, a more ad hoc approach to scaling is used. In MESs, the SCN variables, having various scales, are combined. This requires a consistent way to scale the LF equations for MES. In [2], the per unit system is extended to the heat network for consistency throughout an example MES. To the best of the authors knowledge, there is no equivalent of the per unit system for a gas network.

Another option to scale the system of nonlinear LF equations is by scaling the equations and variables using scaling matrices. Even though this method is a well known scaling method, it is not generally used for LF analysis in any of the SCNs.

We introduce a per unit scaling for MESs consisting of gas, electricity, and heat, by extending the per unit scaling of an electricity network to gas and heat. We compare the per unit scaling with matrix scaling for NR, and show that they are equivalent when using the same base values. The advantages and disadvantages of both methods are discussed.

Using a small MES consisting of gas, electricity, and heat, we investigate the effect of the two scaling methods on the convergence of NR. Despite numerical (round-off) errors, both scaling methods show the same convergence behavior.

2 Steady-State Load Flow

An important tool for the design and operation of energy systems is steady-state LF analysis. The inflow and outflow of the energy system are assumed constant, and the network flows and potentials are determined by the LF equations. For instance, in a gas pipeline network, the gas inflow and outflow are assumed constant, and the gas flow in the pipes and the pressures at the start and end of the pipes are determined.

Energy systems are mathematically represented by a network or graph, which is a collection of nodes, connected by (directed) links. Flow enters the network through sources and leaves the network through sinks. This is represented by an open link connected to a single node only, called terminal link and terminal node respectively. For steady-state LF, the variables of interest are associated with the network nodes, links, or terminal links. Conservation of energy holds in all the single-carrier (SC) nodes. All SC (terminal) links representing a physical component have a link equation that relates link and nodal variables. SC nodes and coupling nodes can have additional node equations that relate the (terminal) link variables of the links connected to that node. There are generally more variables than nodal and link equations. Therefore, some variables are assumed known, called the boundary conditions (BCs) of the network. Collecting all the nodal and link equations, some of which are nonlinear, into one system, and substituting the BCs, gives the system of LF equations:

$$\mathbf{F}(\mathbf{x}) = \mathbf{0} \tag{1}$$

with $\mathbf{F} \in \mathbb{R}^n$ the vector of (nonlinear) LF equations and $\mathbf{x} \in \mathbb{R}^n$ the vector of variables. For specific LF models, see for instance [1] for electricity, [3] for gas, and [4] for MESs.

3 Scaling

The parameters and the dependent and independent variables in the LF equations can be several orders of magnitude apart, even within one SCN. For instance, gas flow $\sim 1\,\text{kg}\,\text{s}^{-1}$ whereas pressure $\sim 10^5\,\text{Pa}$. These different scales might result in issues with solving the system of nonlinear equations, see Sect. 4. Normalizing or scaling the variables and parameters for electricity networks is commonly done, and is called the per unit system (e.g. [1]). Another option is to scale the system of equations and the independent variables by scaling matrices, without scaling the equation parameters. To investigate the effect of scaling on the system of equations, we consider dimensional analysis.

The LF equations are a mathematical representation of a physical phenomenon. Physical quantities are not just numerical values, they also have a dimension and a unit measure associated with them. For instance, the diameter D of a gas pipe has dimension 'length', and could have a unit measure of 1 cm and a value of 15. Denoting the unit measure of length by l and the value of D by k, we can write $D = kl$. We can scale D by changing the unit measure with a scaling factor $k_l \in \mathbb{R}$, and generally $k_l > 0$, such that $l \to k_l l$. Using this new unit measure for D will change the unit measure and the value (to k/k_l), but not the dimension.

Based on the logic as laid out for dimensional analysis in for instance [5], quantities can only be combined in limited ways. Quantities can be multiplied, which multiplies the dimension in the same way. To add two quantities, they must have the same dimension and the same unit measure. Other functional relations are only possible if all arguments are dimensionless. For instance, if $f(x) = \sin(x)$, then both $f(x)$ and x must be dimensionless. Using these concepts recursively, a function of multiple arguments can be made. An equation that satisfies these properties is called 'complete' in [5]. A consequence is that the algebraic form of the equation is unit independent. That is, if the unit measure of any dimension is changed, the algebraic form of the equation remains the same. However, the value of the function might be changed, just like the value of some of the quantities is changed. This can be seen as follows.

Since two (or more) terms can only be added if the terms have the same dimension and unit measure, we can limit ourselves to functions consisting of only one term. Furthermore, for dimensionless quantities, or for a dimensionless group consisting of the power product of some quantities, the changes in unit measures cancel out. Hence, we only need to consider the change in value of functions of the form $f(y_1, \dots, y_n) = y_1^{a_1} \cdots y_n^{a_n}$. We can assume that all y_i have a single (primary) dimension. Scaling each y_i by changing the unit measures of the primary dimensions by a factor k_i gives

$$\begin{aligned} f(y_1, \dots, y_n) \rightarrow f(k_1 y_1, \dots, k_n y_n) &= \left(k_1^{a_1} \cdots k_n^{a_n}\right)\left(y_1^{a_1} \cdots y_n^{a_n}\right) \\ &= \left(k_1^{a_1} \cdots k_n^{a_n}\right) f(y_1, \dots, y_n) \end{aligned} \tag{2}$$

such that f is scaled by a power product of the unit measure scaling factors.

An equation describing a physical model does not need to be complete for the model to be valid. In fact, the commonly used form of the link equation for a transmission line in an electrical network is not a complete equation. It contains terms $\sin \delta_k$ and $\cos \delta_k$, with δ_k the voltage angles difference of link k. Based on the logic provided above, δ_i and δ_j should be dimensionless. However, they have dimension ‘plane angle’. The link equation can be turned into a complete equation by using ${}^{\delta_k}/_{\delta_0}$ instead of δ_k, with a δ_0 reference angle.

3.1 Per Unit System

The per unit system is commonly used in electricity networks, and extended in [2] to the heat network. We consider a more general extension of the per unit system to heat and gas networks. In the per unit system, a quantity x is scaled by a base value:

$$x_{\text{p.u.}} = \frac{x_a}{x_b} \tag{3}$$

Here, x_a is the unscaled or actual quantity, usually in S.I. units, x_b is a chosen base value with the same dimension as x_a, and $x_{\text{p.u.}}$ is the scaled quantity. The scaled quantity is dimensionless but is given p.u. as unit. Hence, the scaled quantity is also called the per unit quantity or value.

There are two main differences between the per unit system and changing the unit measures. The first is that the base value has a dimension, unlike the scaling factor of the unit measure. Second, only the unit measure scaling factors of the primary dimensions are chosen, whereas in the per unit system, the base value for derived quantities might be chosen. The first point has no consequence for the argumentation resulting in (2). However, the second point can lead to some difficulties. Since derived quantities are combinations of other quantities, and applying the same logic that resulted in a complete equation, only a limited set of base values can be specified. The base values for the other quantities then follow from dimensional analysis. The set of base values that can be specified is not unique, neither are the resulting base values of the other quantities. However, it is possible to find a set of base values such that the equation remains a complete equation. For such a set of base values, the argumentation resulting in (2) is still valid. We can now look at the effect of the per unit system of the equation.

Suppose we have a (complete) equation of the form $f\left(\mathbf{x_a},\ \mathbf{p_a}\right)$, with $\mathbf{x_a} \in \mathbb{R}^n$ all the variables, and $\mathbf{p_a} \in \mathbb{R}^m$ all other quantities, dimensionless or not, appearing in the algebraic form of f. We take a set of base values $b_1, \ldots, b_k$, with $k \leq n + m$, and scale each $x \in \mathbf{x_a}$ and $p \in \mathbf{p_a}$ according to (3), with x_b and p_b power products of the base values $b_1, \cdots, b_k$. If the base values are chosen such that the equation f remains a complete equation after scaling, the equation is scaled according to:

$$f\left(\mathbf{x_a},\ \mathbf{p_a}\right) = \left[b_1^{\alpha_1} \cdots b_k^{\alpha_k}\right] f\left(\mathbf{x}_{\text{p.u.}},\ \mathbf{p}_{\text{p.u.}}\right) \tag{4}$$

Usually, only the variables are explicitly denoted as arguments for the function, such that $f\left(\mathbf{x_a},\ \mathbf{p_a}\right)$ is written as $f\left(\mathbf{x_a}\right)$ and $f\left(\mathbf{x}_{\text{p.u.}},\ \mathbf{p}_{\text{p.u.}}\right)$ as $f_{\text{p.u.}}\left(\mathbf{x}_{\text{p.u.}}\right)$. For the scaled equation we then find

$$f\left(\mathbf{x_a}\right) = \left[b_1^{\alpha_1} \cdots b_k^{\alpha_k}\right] f\left(\mathbf{x}_{\text{p.u.}},\ \mathbf{p}_{\text{p.u.}}\right) := f_b f_{\text{p.u.}}\left(\mathbf{x}_{\text{p.u.}}\right) \tag{5}$$

where $f_b = \left[b_1^{\alpha_1} \cdots b_k^{\alpha_k}\right]$ is called the base value of the function f. That is, for a suitable set of base values, the same expression of the LF equations can be used for both the unscaled and per unit quantities, and all independent variables and all LF equations can be scaled to similar orders of magnitude.

For an electricity network, the base values of the voltage amplitude and the power are chosen. The base values of the other variable (current amplitude) and of the parameters of the LF equations (admittance) are determined by the requirement that the LF equations remain a complete equation, using dimensional analysis (e.g. [1]).

The per unit system is then easily extended to the gas and heat SCN, and to a MES. We choose the base values for pressure and flow in the gas network, and for pressure, mass flow, temperature, and power in the heat network. The base values of the other variables and parameters are determined based on dimensional analysis. For the couplings in a MES, we choose the base values of the power of every carrier involved in the coupling, and again determine the base values of the other quantities according to dimensional analysis.

The advantage of scaling derived quantities instead of scaling primary dimensions becomes clear when considering transformers in an electrical network, or compressors in a gas network. These components change the voltage or pressure level, and their link equation has the general form $f(x_1,\ x_2,\ r) = x_1 - rx_2 = 0$, with x_1 and x_2 the voltages or pressures, and r some ratio. Since x_1 and x_2 have the same dimension, r must be dimensionless. Hence, changing the unit measures will scale the values of x_1 and x_2 with the same factor, and will leave r unscaled.

In practice, x_1 and x_2 might be orders of magnitude apart when using the same unit measure. In the per unit system, it is possible to use a different base value for x_1 and x_2, such that both $x_1 \sim 1\,\text{p.u.}$ and $x_2 \sim 1\,\text{p.u.}$ Note that the scaled x_1 and x_2 now have different unit measures, despite both of their units being denoted by p.u. Due to the requirement for addition of dimensional quantities, r needs to scaled with $r_b = {}^{(x_1)_b}/_{(x_2)_b}$.

3.2 Matrix Scaling

Another option is to scale the independent variables and the equations only, using scaling matrices [6]. Taking non-singular matrices $T_x,\ T_F \in \mathbb{R}^{n\times n}$, the scaled variables $\hat{\mathbf{x}}$ and scaled equations $\hat{\mathbf{F}}$ are given by:

$$\hat{\mathbf{x}} = T_x\mathbf{x} \tag{6}$$

$$\hat{\mathbf{F}}\left(\hat{\mathbf{x}}\right) = T_F\mathbf{F}\left(T_x^{-1}\hat{\mathbf{x}}\right) = T_F\mathbf{F}\left(\mathbf{x}\right) \tag{7}$$

Unlike the per unit scaling, scaling with matrices requires us to also choose the scaling for the equations instead of only for the variables. However, per unit scaling requires base values for all parameters in every equation. Furthermore, matrix scaling is generally easier to implement than per unit scaling.

If we take T_x as a diagonal matrix with $(T_x)_{ii} = (x_b)_i$, where $(x_b)_i$ the base value of $x_i \in \mathbf{x}$ used in per unit scaling, it follows from (5) that T_F is a diagonal matrix with $(T_F)_{ii} = (f_b)_i$, where $(f_b)_i$ the base value of $f_i \in \mathbf{F}$ found in per unit scaling. Therefore, in infinite precision, the per unit scaling and matrix scaling will result in the same scaled system of equations $\hat{\mathbf{F}}$ and the same scaled variables $\hat{\mathbf{x}}$. Hence, the per unit system and matrix scaling are said to be equivalent.

4 Newton-Raphson

We use the Newton-Raphson method (NR) to solve the system of non-linear LF equations (1). The iteration scheme in multiple dimensions is given by [6]:

$$J\left(\mathbf{x}^k\right)\mathbf{s}^k = -\mathbf{F}\left(\mathbf{x}^k\right),\ \text{with}\ \mathbf{x}^{k+1} = \mathbf{x}^k + \mathbf{s}^k \tag{8}$$

$J\left(\boldsymbol{x}^k\right)$ is the Jacobian matrix. We take $e^k = ||\mathbf{F}\left(\mathbf{x}^k\right)||_2$ as error of NR at iteration k, with $||\cdot||_2$ the 2-norm. For the stopping criterion we take $e^k \leq \tau$ for some chosen tolerance τ. If the equations in $\mathbf{F}$ are several orders of magnitudes apart, the smaller ones might be ignored during NR, or NR might not convergence to a solution since the larger ones will never reach the required tolerance. Therefore, we scale all equations in $\mathbf{F}$ to be of same order of magnitude.

Since per unit scaling and matrix scaling are equivalent, we only consider matrix scaling. The iteration scheme of NR is adjusted to:

$$\hat{J}\left(\hat{\mathbf{x}}^k\right)\hat{\mathbf{s}}^k = -\hat{\mathbf{F}}\left(\hat{\mathbf{x}}^k\right), \text{ with } \hat{\mathbf{x}}^{k+1} = \hat{\mathbf{x}}^k + \hat{\mathbf{s}}^k \tag{9}$$

It is straightforward to show that $\hat{J}\left(\hat{\mathbf{x}}\right) = T_F J\left(\mathbf{x}\right) T_x^{-1}$. Then, for the scaled step it holds that $\hat{\mathbf{s}}^k = -T_x J\left(\mathbf{x}\right)^{-1} \mathbf{F}\left(\mathbf{x}^k\right) = T_x \mathbf{s}^k$, meaning that scaling does not affect the NR iterations. We take $\hat{e}^k = ||\hat{\mathbf{F}}^k||_2 = ||T_F \mathbf{F}\left(T_x^{-1}\hat{\mathbf{x}}\right)||_2$ as error of the scaled NR. Since all $\hat{f}_i \in T_F\mathbf{F}$ are roughly of the same order of magnitude, we take $\hat{e}^k \leq \tau$ as stopping criterion.

5 Numerical Results

The previous analysis only holds in infinite precision. In finite precision, an NR step might be affected. In the per unit system, the scaled variables and parameters are plugged into (1) to obtain the scaled system of equations, denoted by $\mathbf{F}_{\text{p.u.}}$. With matrix scaling, the unscaled variables and parameters are used in (1). Then, the scaled system of equations is given by $\hat{\mathbf{F}} = T_F \mathbf{F}\left(T_x^{-1}\hat{\mathbf{x}}\right)$. Due to round-off errors, generally $\mathbf{F}_{\text{p.u.}} \neq \hat{\mathbf{F}}$, even though $\mathbf{F}_{\text{p.u.}}$ and $\hat{\mathbf{F}}$ will be close. Similarly, $J_{\text{p.u.}} \neq \hat{J}$, such that $\hat{\mathbf{s}}^k \neq \mathbf{s}^k_{\text{p.u.}} \neq T_x\mathbf{s}^k$. We model a small MES to investigate the effect of finite precision on NR for the two different scaling options.

We consider the small MES shown in Fig. 1, and use the LF model as described in [4]. The resulting system of nonlinear equations is scaled using the per unit system and using matrix scaling. The resulting scaled systems $\mathbf{F}_{\text{p.u.}}$ and $\hat{\mathbf{F}}$ are solved using NR as described in Sect. 4, with a tolerance of $\tau = 10^{-6}$. For comparison, we also solve the unscaled system using NR. Denoting the unscaled system by $\mathbf{F}$, the (unscaled) error at each NR iteration is given by $e^k = ||\mathbf{F}^k\left(\mathbf{x}\right)||_2$. To compare with the error of NR for the scaled systems, we calculate the scaled error of the unscaled NR iteration by $\tilde{e}^k = ||T_F\mathbf{F}^k\left(\mathbf{x}\right)||_2$. Note that $\tilde{e}^k$ is different from the error $\hat{e}^k = ||\hat{\mathbf{F}}^k\left(\hat{\mathbf{x}}\right)||_2 = ||T_F\mathbf{F}^k\left(T_x^{-1}\hat{\mathbf{x}}\right)||_2$ of scaled NR, since scaled NR uses the scaled update $\hat{\mathbf{s}}^k$ instead of $\mathbf{s}^k$.

Table 1 gives the errors for NR. We can see that the errors for the per unit scaling and the matrix equals are unequal, but close, to each other and to the error of unscaled NR. Hence, scaling affects NR in finite precision. In this example, this effect does not result in a significant difference between the solutions to the LF problem.

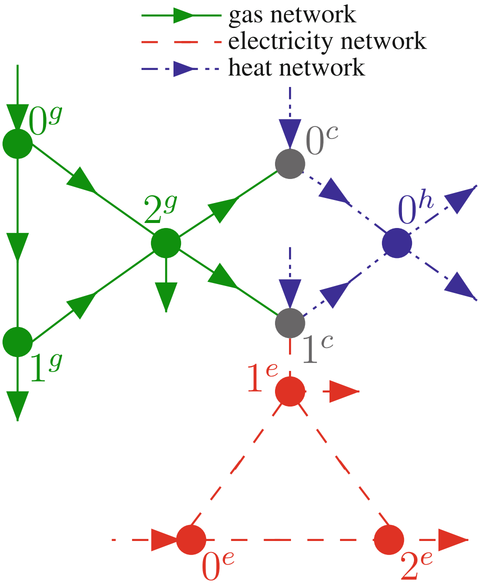


Fig. 1 Network representation of a small MES. Each SCN consists of three nodes. The gas and electricity networks have an external source connected at nodes 0^g and 0^e, the heat network has no external sources. In each SCN, nodes 1 and 2 are sinks. The SCNs are coupled by a gas-boiler, node 0^c, and a combined heat and power plant (CHP), node 1^c. The links show defined direction of flow, the terminal links show actual direction of flow

Table 1 Errors of NR for each iteration k, using a tolerance of $\tau = 10^{-6}$. Here $\tilde{e}^k = ||T_F \mathbf{F}^k||_2$, $\hat{e}^k = ||\hat{\mathbf{F}}^k||_2$ and $e^k_{\text{p.u.}} = ||\mathbf{F}^k_{\text{p.u.}}||_2$. The last column gives the relative difference between the errors of scaled NR and unscaled NR

k	$\tilde{e}^k$	$\hat{e}^k$	$e^k_{\text{p.u.}}$	$\frac{\lvert\tilde{e}^k-\hat{e}^k\rvert}{\lvert\tilde{e}^k\rvert}$	$\frac{\lvert\tilde{e}^k-e^k_{\text{p.u.}}\rvert}{\lvert\tilde{e}^k\rvert}$
0	1.0310×10^6	1.0310×10^6	1.0310×10^6	0.0000	0.0000
1	1.3081×10^3	1.3081×10^3	1.3081×10^3	2.6421×10^{-14}	1.0951×10^{-14}
2	5.7417×10^{-1}	5.7417×10^{-1}	5.7417×10^{-1}	1.5071×10^{-12}	9.6527×10^{-13}
3	7.0379×10^{-4}	7.0379×10^{-4}	7.0379×10^{-4}	6.5244×10^{-10}	7.7472×10^{-10}
4	3.2883×10^{-9}	3.2890×10^{-9}	3.2886×10^{-9}	1.8566×10^{-4}	7.4581×10^{-5}
5	6.6172×10^{-11}	–	–	–	–

6 Conclusion

We extended the per unit system used in electrical networks for scaling the load flow (LF) equations to gas networks, heat networks, and multi-carrier energy networks (MCNs). The per unit system scales the equations by scaling all variables and parameters. The base values are determined by dimensional analysis, such that the scaled system is also dimensionless. Another option is to use scaling matrices, which explicitly scales the equations. We showed that base values can be chosen such that the per unit system is equivalent to using scaling matrices, in infinite precision.

Newton-Raphson's method (NR) is used to solve the (scaled) system of nonlinear LF equations. In infinite precision NR is unaffected by scaling. Using the LF equations for a small MCN, we showed that both scaling methods lead to slightly different NR steps, meaning that NR is affected by scaling in finite precision. However, the difference in the solution found for the LF problem is small. Hence, for this example, the per unit system and scaling matrices are equivalent in finite precision.

References

1. Schavemaker, P., Van der Sluis, L.: Electrical power system essentials. Wiley, Chichester (2008)
2. Ayele, G. T., Haurant, P., Laumert, B., Lacarrieère, B.: An extended energy hub approach for load flow analysis of highly coupled district energy networks: Illustration with electricity and heating. Applied Energy. **212**, 850–867 (2018)
3. Osiadacz, A. J.: Simulation and analysis of gas networks. Spon, London (1987)
4. Markensteijn, A. S., Romate, J. E., Vuik, C.: On the Solvability of Steady-State Load Flow Problems for Multi-Carrier Energy Systems. IEEE Milan PowerTech 2019 (2019)
5. Gibbings, J. C.: Dimensional Analysis. Springer, London (2011)
6. Dennis, J. E. Jr., Schnabel, R. B.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM, Philadelphia (1996)

A Semismooth Newton Method for Regularized L^q-quasinorm Sparse Optimal Control Problems

Pedro Merino

Abstract *A semismooth Newton method* (refered as DC–SSN) is proposed for the numerical solution of a class of nonconvex optimal control problems governed by linear elliptic partial differential equations. The nonconvex term in the cost functional arises from a Huber-type local regularization of the L^q-quasinorm ($q \in (0, 1)$), therefore it promotes sparsity on the solution. The DC–SSN method solves the optimality system of the regularized problem resulting from the application of *difference-of-convex functions* programming tools.

1 Introduction

Sparse optimal controls are attractive in many applications due their parsimony. For instance, L^1-norm penalized controls in the cost function is a common approach in applications for promoting sparsity on the solutions, see [3]. On the other hand, L^q-quasinorm ($q \in (0, 1)$) penalizer differs from L^1-norm qualitatively in virtue of the induced sparsity on the solution might involve discontinuities on the boundary of its support as discussed in [2]. However, the fact that L^q-quasinorm is a non convex nor differentiable function, makes its theoretical and numerical studies challenging. Indeed, the lack of weak lower semincontinuity of the cost function does not allow the use of well known techniques used for convex problems. Still, we are motivated to use the L^q-quasinorm since it is a natural approximation for the Donoho's counting "norm", referred as the L^0-norm and interpreted as a penalization for the cost of the volume of the control support, as an analogy to the discrete measure of a vector in finite dimensions.

The theory involving this kind of problems is far from being complete. In particular, in [1] the authors considered this type of penalizers and established

P. Merino (✉)
MODEMAT Research Center on Mathematical Modeling, Departamento de Matemática, Escuela Politécnica Nacional, Quito, Ecuador
e-mail: pedro.merino@epn.edu.ec

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_71

a *maximum principle,* provided that the controls belong to the space $H_0^1(\Omega)$. By contrast, when the control space is $L^2(\Omega)$, existence of solutions cannot be proven using the direct method.

Although the question of the existence of solutions is not fully explained, several aspects of the solutions have been estudied. For instance, an optimality system was deduced in [2] for a regularized version of the optimal control problem with L^q-quasinorms. Here, we consider this kind of regularized L^q-quasinorm penalized optimal control problem which captures the nonconvexity and nondifferentiability of the L^q terms. In this paper, we exploit the corresponding necessary optimality conditions based on the DC-programming approach from [2], by proposing a semismooth Newton method for computing numerical solutions of the nonconvex optimal control problem.

2 The Optimal Control and Its Optimality Conditions

Consider Y, U and W reflexive Banach spaces, with $U \hookrightarrow L^2(\Omega)$. Let $E : Y \to U$. We assume that E is a linear and continuous operator. Moreover, we assume that for every u, the equation $Ey - u = 0$ has a unique solution $y = y(u)$ which depends (afine) linearly on u. For instance, e can represent linear elliptic equations of second-order with $Y = W = H_0^1(\Omega)$ and $U = L^2(\Omega)$. Let us denote by B, a linear and continuous operator from U into $L^2(\Omega)$. Also, let us consider two constants a and b in $\mathbb{R}$, such that $a < 0 < b$. For $q \in (0, 1)$, and $\gamma > 0$ we consider the optimal control problem:

$$(P)\begin{cases} \min\limits_{(y,u)\in Y\times U} J(y,u) := \dfrac{1}{2}\|y - y_d\|^2_{L^2(\Omega)} + \dfrac{\alpha}{2}\|Bu\|^2_{L^2(\Omega)} + \beta \displaystyle\int_\Omega |u(x)|^q_{q,\gamma}\, dx \\ \text{subject to:} \\ \qquad Ey - u = 0, \qquad \text{and } u \in U_{ad} := \{u \in U : a \le u \le b\}. \end{cases} \tag{P}$$

where $|\cdot|_{q,\gamma}$ stands for the Huber-type local regularization of the absolute value introduced in [2] (with $q = 1/p$) and given by

$$|t|_{q,\gamma} := \begin{cases} q\gamma^{\frac{1-q}{q}}|t|^{\frac{1}{q}}, & \text{if } t \in [-\frac{1}{\gamma}, \frac{1}{\gamma}], \\ |t| - \dfrac{1-q}{\gamma}, & \text{otherwise.} \end{cases}$$

Notice that, for a fixed $q \in (0, 1)$, $|\cdot|_{q,\gamma} \to |\cdot|$ uniformly as $\gamma \to \infty$.

Remark 1 Existence of solutions for problem (P) when $B = \nabla$ was proved in [1]. Indeed, compactness arguments can be used since controls are in $U = H^1(\Omega)$. The case $U = L^2(\Omega)$ is more delicate. Although, under strong conditions the controls in $L^2(\Omega)$ can be approximated by controls in $H^1(\Omega)$, see [2]. The lack of compactness in $L^2(\Omega)$ and the absence of convexity make the direct method fail. In the following, we will assume existence of a solution for problem (P).

2.1 The Optimality System

Necessary optimality conditions for problem (P) were derived in [2] by formulating the reduced optimal control problem and then splitting the cost function in the form of difference-of-convex functions (DC). This representation is obtained by replacing the control-to-state operator $S : U \to Y$, which is assumed (afine) linear and continuous, and given by $y = y(u) = Su + y_f$, with some $y_f \in Y$. Furthermore, the control constraints are included using the indicator function $I_{U_{ad}}$. Hence, problem (P) is reformulated as the following optimization problem:

$$\min_u f(u) = G(u) - H(u), \tag{1}$$

where G and H are defined by:

$$\begin{aligned} &G : L^2(\Omega) \to \mathbb{R} \\ &\quad G(u) := \tfrac{1}{2}\|Su + y_f - y_d\|^2_{L^2(\Omega)} + \tfrac{\alpha}{2}\|Bu\|^2_{L^2(\Omega)} + \beta\kappa_\gamma \|u\|_{L^1(\Omega)} + I_{U_{ad}}, \\ &H : L^2(\Omega) \to \mathbb{R} \\ &\quad H(u) := \beta\left(\kappa_\gamma \|u\|_{L^1(\Omega)} - \textstyle\int_\Omega |u(x)|^q_{q,\gamma}\,dx\right) = \beta \textstyle\int_\Omega \kappa_\gamma |u(x)| - |u(x)|^q_{q,\gamma}\,dx. \end{aligned} \tag{2}$$

Here, a constant $\kappa_\gamma := q^q\gamma^{1-q}$ is chosen for inducing Gâteaux differentiability of the function H, see [2]. Indeed, H is defined through the real function $t \mapsto \kappa_\gamma|t| - |t|^q_{q,\gamma}$, which is bounded by κ_γ and it is continuously differentiable. Let us denote its first derivative by j. Then, j takes the form

$$j(t) = \begin{cases} \left[\kappa_\gamma - q\left(|t| + \dfrac{q-1}{\gamma}\right)^{q-1}\right]\operatorname{sign}(t), & \text{if } |t| > \frac{1}{\gamma}, \\ 0, & \text{otherwise.} \end{cases}$$

Using DC-programming theory, the solution $\bar{u}$ of the minimization problem is characterized by the inclusion $\partial H(\bar{u}) \subset \partial G(\bar{u})$. Applying this relation, the following result can be proven.

Theorem 1 *Let B^* be the adjoint of B. Let $\bar{u}$ be a solution of (P), then there exist: $\bar{y} = S\bar{u} + y_f$ in Y, an adjoint state $\phi \in W^*$, a sparsity multiplier ζ and function $\bar{w}$, both in $L^\infty(\Omega)$ and nonnegative multipliers λ_a and λ_b in U^* such that the following* *optimality system is satisfied:*

$$S\bar{u} + y_f - \bar{y} = 0, \tag{3a}$$

$$S^*(\bar{y} - y_d) - \phi = 0, \tag{3b}$$

$$\phi + \alpha B^* B\bar{u} + \beta\,(\kappa_\gamma\,\zeta - \bar{w}) + \lambda_b - \lambda_a = 0, \tag{3c}$$

$$\lambda_a(\bar{u} - a) = \lambda_b(\bar{u} - b) = 0, \tag{3d}$$

$$\zeta(x) \begin{cases} = 1, & \textit{if}\ \bar{u}(x) > 0, \\ = -1, & \textit{if}\ \bar{u}(x) < 0, \\ \in [-1, 1], & \textit{if}\ \bar{u}(x) = 0, \end{cases} \tag{3e}$$

$$\bar{w}(x) = j(\bar{u}(x)), \tag{3f}$$

for almost all $x \in \Omega$.

Proof This system follows from the inclusion $\partial H(\bar{u}) \subset \partial G(\bar{u})$, which takes the form $H'_G(\bar{u}, \cdot) \subset \partial G(\bar{u})$ since H has Gâteaux derivative $H'_G(\bar{u}, \cdot) = (\beta\bar{w}, \cdot)_{L^2(\Omega)}$. □

As in [3], we collect multipliers ζ, λ_a and λ_b into a single multiplier $\mu = \kappa_\gamma\beta\zeta + \lambda_b - \lambda_a$. Then, we define the superposition operator $u \mapsto w$ by $w(u)(x) = j(u(x))$, for almost all $x \in \Omega$. In addition, for a positive constant c, we introduce

$$\begin{aligned} C(u, \mu) = u &- \max(0, u + c(\mu - \kappa_\gamma\beta)) - \min(0, u + c(\mu + \kappa_\gamma\beta)) \\ &+ \max(0, u - b + c(\mu - \kappa_\gamma\beta)) + \min(0, u - a + c(\mu + \kappa_\gamma\beta)). \end{aligned}$$

Hence, in view of the complementarity condition (3d) we reduce the problem of finding $(\bar{y}, \phi, \mu, \bar{u}, w) \in Y \times W^* \times U^* \times U \times L^\infty(\Omega)$ satisfying the following system:

$$E\bar{y} - \bar{u} = 0, \tag{4a}$$

$$E^*\phi - \bar{y} + y_d = 0, \tag{4b}$$

$$\alpha B^* B\bar{u} + \phi + \mu - \beta w = 0, \tag{4c}$$

$$C(\bar{u}, \mu) = 0, \tag{4d}$$

$$w - j(\bar{u}) = 0. \tag{4e}$$

Theorem 2 *Let* $B = I$ *and let* $\mathcal{F} : Y \times W^* \times U^* \times U \times L^\infty(\Omega) \rightarrow$ *a function of variables* y, ϕ, μ, u, w *defined by the left hand side of* (4)*, and let us assume that* $Y \hookrightarrow L^s(\Omega)$ *for some* $s > 2$*, then we have that* $\mathcal{F} : U \rightarrow L^2(\Omega)$ *is semismooth in the sense of Definition 3.1 in [4].*

Proof Let us consider the first case $C = I$. By using the solution operator we can replace $y = Su + y_f$, $\phi = Tu := S^*(Su + y_f - y_d)$ and $\mu = \beta w - \phi - \alpha u = j(u) - Tu - \alpha u$ in (4d). Moreover, we can chose $c = 1/\alpha$, thus the system (4) can be witten as a function depending on u only:

$$F(u) = u - \frac{\beta}{\alpha}\Big(\max(0, j(u) - \frac{1}{\beta}Tu - \kappa_\gamma) - \min(0, j(u) - \frac{1}{\beta}Tu - \kappa_\gamma) \\ + \max(0, j(u) - \frac{1}{\beta}Tu - \kappa_\gamma - \frac{\alpha}{\beta}b) + \min(0, j(u) - \frac{1}{\beta}Tu - \kappa_\gamma - \frac{\alpha}{\beta}a)\Big). \tag{5}$$

Using similar arguments as in [3], we observe that the terms involved in F are expressed as the composition of the operator $u \mapsto j(u) - \frac{1}{\beta}Tu$ (plus a constant term) and the max (or min) function. Moreover, by our assumptions, $T : U \to L^s(\Omega)$ is an afine and continuous operator from U into $L^s(\Omega)$. Hence, it is Fréchet differentiable. On the other hand, due to the fact that $j : \mathbb{R} \to \mathbb{R}$ is piecewise continuously differentiable, it is also semismooth. Moreover, j is bounded by κ_γ, see [2]. Hence, it is not difficult to check Lipschitz continuity of j with associated Lipschitz constant $L_\gamma = 2\gamma\kappa_\gamma$. Therefore, by applying Theorem 3.49 in [4], the superposition operator $u \to j(u(\cdot))$ defined from U to $L^\infty(\Omega)$ is semismooth from U into $L^2(\Omega)$. Then, we have that the mapping $u \mapsto j(u(\cdot)) - \frac{T}{\beta}u$ is semismooth from U into $L^2(\Omega)$. Finally, it is well known that the composition with the $\max(0, \cdot)$ function is also semismooth. □

Remark 2 For the case $B = \nabla$, it is clear that we cannot chose a suitable value of the constant c as in the previous proof. We do not discuss the semismoothness of $\mathcal{F}$ for this case in this paper, since it is rather technical. However, it can be proved that it is semismooth.

3 The DC: Semismooth Newton Method

In order to solve system (4), whose right-hand side is denoted by $\mathcal{F} = \mathcal{F}(z)$ with $z = (y, \phi, \mu, u, w)$, we apply the Semismooth Newton Method that we refer as DC–SSN algorithm. The DC–SNN solves equation $\mathcal{F}(z) = 0$ and it is based on the iterative step:

$$z_{k+1} = z_k + \delta_{z_k}, \qquad \text{with} \qquad \mathcal{F}'(z_k)\delta_{z_k} = -\mathcal{F}(z_k)$$

We introduce the active sets associated with the sparsity multiplier and the constraints multipliers given by

$$A_S = \{x : u + c(\mu - \kappa_\gamma \beta) \geq 0\} \cup \{x : u + c(\mu + \kappa_\gamma \beta) \geq 0\}, \text{ and} \tag{6}$$

$$A_C = \{x : u - b + c(\mu - \kappa_\gamma \beta) \geq 0\} \cup \{x : u - a + c(\mu + \kappa_\gamma \beta) \geq 0\}, \tag{7}$$

respectively. Then, the active set is given by:

$$A = A_S \cup A_C.$$

Then, we can compute the generalized derivative $\mathcal{F}'(z)$ and formulate the Newton system $\mathcal{F}'(y, \phi, \mu, u, w)[\delta_y, \delta_\phi, \delta_\mu, \delta_u, \delta_w] = -\mathcal{F}(y_k, \phi_k, \mu_k, u_k, w_k)$, which is given by

$$\begin{pmatrix} E & 0 & 0 & -I & 0 \\ -I & E & 0 & 0 & 0 \\ 0 & I & I & \alpha C^* C & -\beta I \\ 0 & 0 & -c\chi_A & I - \chi_A & 0 \\ 0 & 0 & 0 & -j'(u_k) & I \end{pmatrix} \begin{pmatrix} \delta_y \\ \delta_\phi \\ \delta_\mu \\ \delta_u \\ \delta_w \end{pmatrix} = - \begin{pmatrix} Ey_k - u_k \\ E^* \phi_k - y_k + y_d \\ \alpha C^* C u_k + \phi_k + \mu_k - \beta w_k \\ C(u_k, \mu_k) \\ w_k - j(u_k) \end{pmatrix} \tag{8}$$

In addition, by noticing that the function $\mathcal{F}$ can be written in the form $\mathcal{F}(z) = z - \mathcal{S}(z)$, we can incorporate a *smoothing step* as described in [4, Section 5.2.4]. Thus, we propose the following algorithm:

DC–SNN Algorithm

1. Initialization: $z_0 = (y_0, \phi_0, \mu_0, u_0, w_0)$.
2. Solve Newton system: Solve (8) for $\delta = \delta_k$.
3. Updating: $\hat{z}_{k+1} = z_k + \delta_k$.
4. Smoothing: $z_{k+1} = \mathcal{S}(\hat{z}_{k+1})$.
5. Repeat for $k + 1$ until convergence.

4 Numerical Examples

We show numerical evidence of the performance of the method DC–SSN by solving two optimal control problems of the form (*P*) numerically. For this purpose, we approximate the differential operator using a standard finite-difference scheme. Other methods of approximation might be also considered, e.g. FEM. The following examples are tested in $\Omega = (0, 1) \times (0, 1)$ with state space $Y = H_0^1(\Omega)$

Table 1 Numerical convergence for Example 1 for increasing values of γ

γ	Cost ($\times 10^{-6}$)	Residual	$\|\mathcal{F}(z_k)\|$	Null entries	Iterations	Time(s)
1000	52158.176	3.874e−12	3.9093e−10	1417	212	94.6
1200	52158.177	6.5569e−12	6.6158e−10	1409	198	89.5
1500	52158.180	4.4189e−15	4.8743e−12	1396	82	44.18
2000	52158.184	6.8007e−12	6.862e−10	1378	53	29.9
3000	52158.184	1.0683e−12	1.0788e−10	1376	173	80.7
5000	52158.187	9.4537e−15	4.9981e−12	1360	71	38.9
6000	52158.187	8.2468e−15	4.8575e−12	1360	84	44.8
7000	52158.187	3.2544e−11	3.2834e−09	1360	69	37.3
8000	52158.187	6.7549e−14	8.4028e−12	1360	90	47.0
9000	52158.187	7.4883e−15	4.9687e−12	1360	102	52.9
10,000	52158.187	8.1208e−11	8.1925e−09	1360	194	89.2
12,000	52158.187	1.0878e−14	5.1632e−12	1360	186	85.3
13,000	52158.187	2.0902e−11	2.1087e−09	1360	107	54.4

Example 1 For this example we chose operators $E = -\Delta$, $C = I$. The state and control spaces are $Y = H_0^1(\Omega)$ and $U = L^2(\Omega)$, with $U_{ad} = L^2(\Omega)$. Moreover, we chose the parameters $\alpha = 1/4$, $\beta = 4e - 4$, $q = 2$ and desired state $y_d = e^{-\cos(2\pi x_1 x_2)}$. The discretization mesh is of size 100 × 100, and the parameter $c = \alpha^{-1}$. The admissible set is given by

$$U_{ad} = \{u \in L^2(\Omega) : 0 \leq u \leq 0.08\}.$$

We observe the numerical convergence of the solutions corresponding to different values of γ in Table 1. For higher values of γ we notice that the cost value of the unregularized cost function remains steady. Figure 1 shows the optimal control where: its sparse structure, the action of the upper constraint and the discontinuity on its support are depicted.

Example 2 For this example we chose operators $E = -\Delta$, $B = \nabla$. The state and control spaces $Y = H_0^1(\Omega)$ and $U = L^2(\Omega)$ with $U_{ad} = L^2(\Omega)$. Moreover, we chose the parameters $\alpha = 1/4$, $\beta = 2^{-4}$, $q = 1/2$ and desired state $y_d = e^{-\cos(2\pi x_1 x_2)}$. No constraints have been considered for this example. We notice that the optimal control does not present discontinuities since this contribute to the cost of its gradient. Table 2 shows information for every iteration of the algorithm. See also Fig. 2.

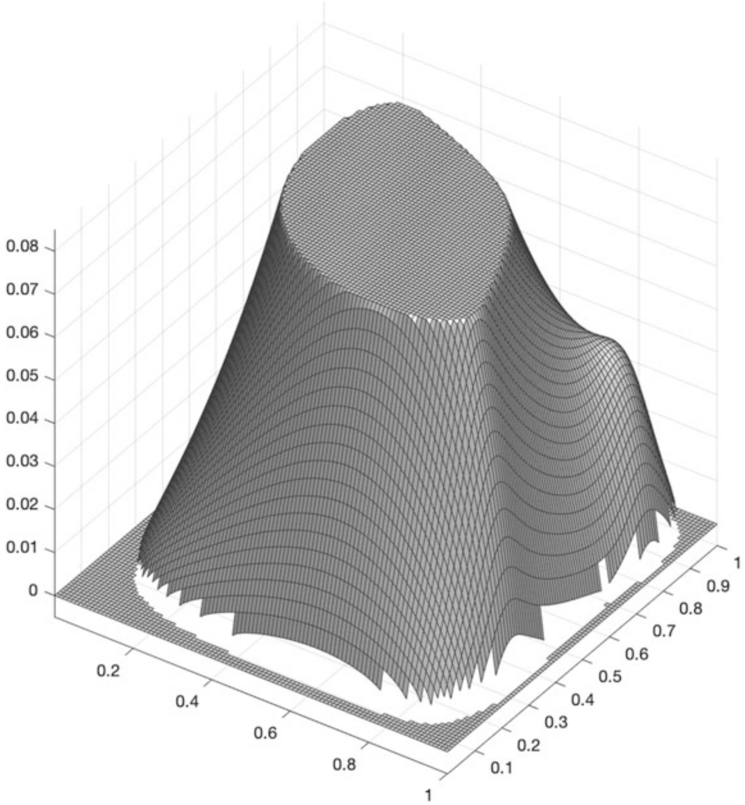


Fig. 1 Computed optimal control for $\gamma = 4000$ in Experiment 1

Table 2 Iterative information of the algorithm for Example 2 with $\gamma = 5000$

Null elements	Residual	Cost ($\times 10^{-6}$)	$\|\mathcal{F}(z_k)\|$
0	9.8006	0.052417	515148.6076
0	0.00051106	0.052416	2392.926
581	5.782e−05	0.052416	637.551
531	1.0405e−06	0.052416	86.12
481	9.3423e−07	0.052416	0.67903
449	4.3183e−07	0.052416	0.45016
434	1.4349e−07	0.052416	0.10871
434	9.2923e−11	0.052416	0.0016909
434	6.1521e−17	0.052416	8.5332e-10

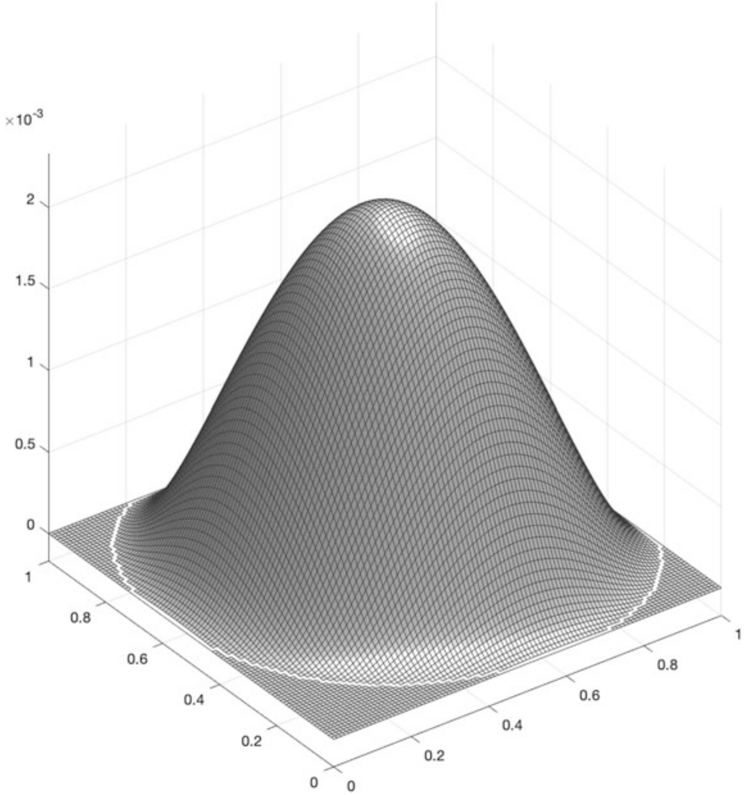


Fig. 2 Computed optimal control for $\gamma = 5000$ in Experiment 2

Acknowledgments This research was funded by project PIMI-17-01 granted by Escuela Politécnica Nacional, Quito–Ecuador. Also, our research was carried out using the research computing facilities and services offered by Scientific Computing Laboratory of the Research Center on Mathematical Modeling: HPC–MODEMAT (http://www.hpcmodemat.epn.edu.ec), Escuela Politécnica Nacional, Quito–Ecuador.

References

1. Kazufumi Ito and Karl Kunisch. Optimal control with $L^p(\Omega)$, $p \in [0, 1)$, control cost. *SIAM Journal on Control and Optimization*, 52(2):1251–1275, 2014.
2. Pedro Merino. A difference-of-convex functions approach for sparse pde optimal control problems with nonconvex costs. *Computational Optimization and Applications*, 74(1):225–258, 2019.
3. Georg Stadler. Elliptic optimal control problems with L^1-control cost and applications for the placement of control devices. *Computational Optimization and Applications*, 44(2):159, 2006.
4. Michael Ulbrich. Semismooth newton methods for operator equations in function spaces. *SIAM Journal on Optimization*, 13(3):805–841, 2002.

Monotone and Second Order Consistent Scheme for the Two Dimensional Pucci Equation

Joseph Frédéric Bonnans, Guillaume Bonnet, and Jean-Marie Mirebeau

Abstract We introduce a new strategy for the design of second-order accurate discretizations of non-linear second order operators of Bellman type, which preserves degenerate ellipticity. The approach relies on Selling's formula, a tool from lattice geometry, and is applied to the Pucci equation, discretized on a two dimensional Cartesian grid. Numerical experiments illustrate the robustness and the accuracy of the method.

1 Introduction

Degenerate Ellipticity (DE) is a property of a class of partial differential operators, often non-linear and of order at most two. When satisfied, it implies a generalized comparison principle, from which can be deduced the existence, uniqueness and stability of a viscosity solution to the Partial Differential Equation (PDE), under mild additional assumptions [CIL92]. Discrete degenerate ellipticity is the corresponding property for numerical schemes, see Definition 2, which has similarly strong implications and often turns the convergence analysis of solutions into a simple verification [Obe06]. It is therefore appealing to design PDE discretizations preserving the DE property, yet a strong limitation of the current approaches [BS91, Obe08, FJ17] is their low consistency order, usually below one. Filtered schemes [FO13] attempt to mitigate this issue by combining a DE scheme of low consistency order with a non-DE scheme of high consistency order, but their use

J. F. Bonnans
INRIA Saclay, Ecole Polytechnique CMAP, Palaiseau, France

G. Bonnet
U. Paris-Sud, U. Paris-Saclay, Orsay, France

J.-M. Mirebeau (✉)
U. Paris-Sud, CNRS, U. Paris-Saclay, Orsay, France

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_72

requires careful parameter tuning, and theoretical results are lacking regarding their effective accuracy.

In this paper, we propose a new approach to develop second order accurate DE schemes, which is the highest achievable consistency order [Obe06], on two dimensional Cartesian grids. The operator must be given in Bellman form as follows

$$\Lambda u(x) = \sup_{\alpha \in \mathcal{A}} a_\alpha + b_\alpha u(x) - \mathrm{Tr}(D_\alpha \nabla^2 u(x)), \tag{1}$$

where $\mathcal{A}$ is an abstract set of parameters, and the coefficients $a_\alpha \in \mathbb{R}$, $b_\alpha \geq 0$, and symmetric positive definite matrix D_α may additionally depend on the position x. A specific feature of our approach, that is tied to the structure of the addressed problems, is that the parameter space $\mathcal{A}$ is not discretized. We apply this approach to the two dimensional Pucci equation:

$$\lambda_{\min}(\nabla^2 u(x)) + \mu\, \lambda_{\max}(\nabla^2 u(x)) = f(x), \tag{2}$$

with Dirichlet boundary conditions, where $\lambda_{\min}$ and $\lambda_{\max}$ denote the smallest and largest eigenvalue of a symmetric matrix, and where $\mu > 0$. This PDE admits the following Bellman form, when $\mu \leq 1$, which we assume for simplicity:

$$\max_{\theta \in [0,\pi]} - \mathrm{Tr}(D(\theta,\mu)\nabla^2 u(x)) = -f(x), \qquad \text{where } D(\theta,\mu) := R_\theta \begin{pmatrix} 1 & 0 \\ 0 & \mu \end{pmatrix} R_\theta^T, \tag{3}$$

and where $R_\theta := \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$ denotes the rotation matrix of angle $\theta \in \mathbb{R}$. Our approach also applies in the case $\mu \geq 1$, with only the slight modification that the max in (3) is replaced with a min. Note that the optimization space in (3) is $\mathcal{A} = [0,\pi]$, which is compact and one dimensional, thus easing the theoretical study and the numerical implementation.

Motivation for this Study The Pucci equation interpolates between two fundamental problems in analysis: the Poisson problem when $\mu = 1$, and the (lower-)convex envelope of the boundary conditions when $\mu = 0$ and $f = 0$. It is also an excellent representative of the class of Pucci extremal operators, a.k.a. operators which can be written in the form (1), perhaps replacing the inf with a sup. This class also encompasses the Monge–Ampère operator, known for its applications in optimal transport and optics, to which similar techniques may be applied [BCM16].

2 Discretization

We rely on a tool from algorithmic lattice geometry, known as Selling's formula Sect. 2.1, which is particularly adequate for discretizing degenerate elliptic PDEs on Cartesian grids of dimension two [BOZ04] or three [Mir18, Mir19, FM14]. Throughout this section $\Omega \subset \mathbb{R}^2$ denotes a bounded domain, and $h > 0$ a grid scale. Define

$$\Omega_h := h\mathbb{Z}^2 \cap \Omega, \qquad \Delta_h^e u(x) := \frac{u(x+he) - 2u(x) + u(x-he)}{h^2}, \tag{4}$$

the discrete domain and the second order finite difference of a map $u : \Omega_h \cup \partial\Omega \to \mathbb{R}$ at $x \in \Omega_h$ in the direction $e \in \mathbb{Z}^2$. When x is adjacent to $\partial\Omega$ the latter formula becomes

$$\Delta_h^e u(x) := \frac{2}{h_+ + h_-}\Big(\frac{u(x+h_+e) - u(x)}{h_+} + \frac{u(x-h_-e) - u(x)}{h_-}\Big), \tag{5}$$

where $h_\pm > 0$ is the least value such that $x \pm h_\pm e \in \Omega_h \cup \partial\Omega$. Note that (4, right) is a second order consistent approximation of $\langle e, \nabla^2 u(x) e\rangle$, whereas (5) is only first order consistent. Thus

$$\mathrm{Tr}(ee^T \nabla^2 u(x)) = \langle e, \nabla^2 u(x) e\rangle = \Delta_h^e u(x) + O(h^r), \tag{6}$$

where $r = 1$ if x is adjacent to $\partial\Omega$, and $r = 2$ otherwise.

2.1 *Selling's Formula*

Selling's decomposition of an element of the set S_2^{++} of symmetric positive definite 2×2 matrices, see Proposition 1, can be regarded as a variant of the eigenvector/eigenvalue decomposition, but whose vectors have *integer entries*. We rely on it to discretize non-divergence form linear (8) and non-linear (11) operators, in a manner that achieves discrete degenerate ellipticity, see Definition 2.

Definition 1 A superbase of $\mathbb{Z}^2$ is a triplet $(e_0, e_1, e_2) \in (\mathbb{Z}^2)^3$ such that $e_0 + e_1 + e_2 = 0$ and $|\det(e_1, e_2)| = 1$. It is D-obtuse, where $D \in S_2^{++}$, iff $\langle e_i, De_j\rangle \le 0$ for all $i \neq j$.

Proposition 1 (Selling [Sel74]) *For each $D \in S_2^{++}$ there exists a D-obtuse superbase (e_0, e_1, e_2) of $\mathbb{Z}^2$, which can be obtained from Selling's algorithm. Furthermore one has Selling's formula*

$$D = \sum_{0 \le i \le 2} \rho_i v_i v_i^\top \quad \textit{with } \rho_i := -\langle e_{i-1}, De_{i+1}\rangle \ge 0, \quad v_i := e_i^\perp \in \mathbb{Z}^2, \tag{7}$$

where $e^\perp := (-b, a)^\top$ *if* $e = (a, b)^\top \in \mathbb{R}^2$. *The set* $\{(\rho_i, \pm v_i);\ 0 \le i \le 2,\ \rho_i > 0\}$ *is uniquely determined. In* (7), *the indices* $i-1$ *and* $i+1$ *are understood modulo* 3.

Based on this formula, one can consider the following finite differences operator:

$$\Delta_h^D u(x) := \sum_{0 \le i \le 2} \rho_i \Delta_h^{v_i} u(x). \tag{8}$$

Using (6), (7), and (8) and the linearity of the trace operator on matrices, we obtain

$$\mathrm{Tr}(D\nabla^2 u(x)) = \sum_{0 \le i \le 2} \rho_i \,\mathrm{Tr}(v_i v_i^T \nabla^2 u(x)) = \Delta_D^h u(x) + \mathcal{O}(h^r),$$

where again $r = 1$ if x is adjacent to $\partial\Omega$, and $r = 2$ otherwise.

We illustrate on Fig. 1 the relation between the anisotropy defined by a symmetric positive definite matrix $D \in S_2^{++}$, and the corresponding offsets $\pm v_0, \pm v_1, \pm v_2 \in \mathbb{Z}^2$ in Selling's formula. (The weights ρ_i are illustrated on Fig. 2.) For that purpose, we rely on a parametrization $\mathbf{D}$ of the 2×2 symmetric positive definite matrices of unit trace, by the points (x, y) of the open unit ball:

$$\mathbf{D}(x, y) := \frac{1}{2}\begin{pmatrix} 1+x & y \\ y & 1-x \end{pmatrix}, \qquad \text{where } x^2 + y^2 < 1. \tag{9}$$

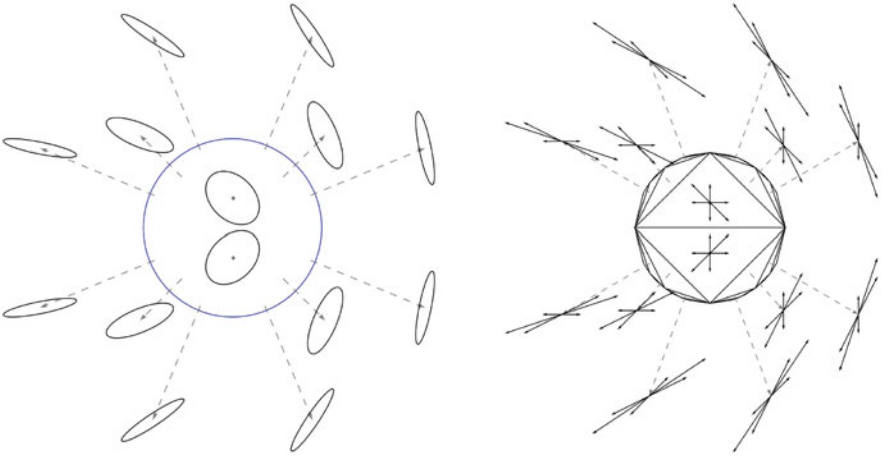


Fig. 1 (Left) Ellipsoid $\{v \in \mathbb{R}^2;\ v^T\mathbf{D}(z)v = 1\}$ for some points z of the unit disc, see (9). Anisotropy degenerates as z moves toward the unit circle, shown blue. (Right) $\mathbf{D}(z)$-obtuse superbase, and opposites, for the same points z. This superbase is piecewise constant on an infinite triangulation of the unit disk [Sch09], shown black

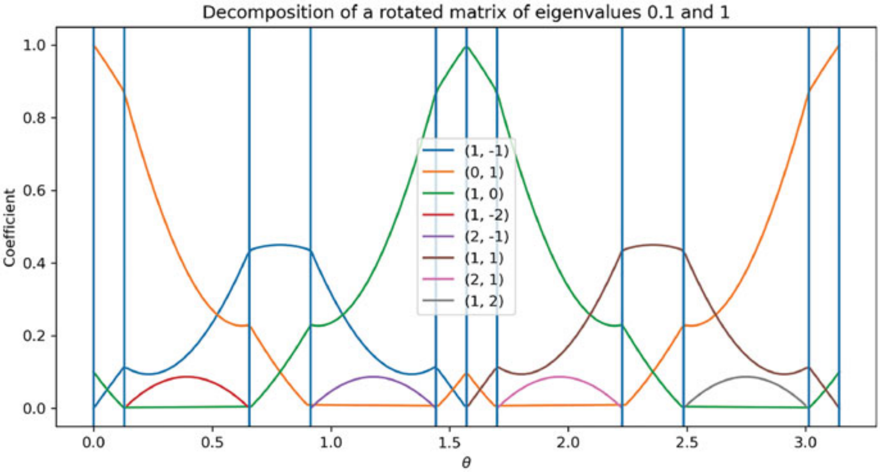


Fig. 2 Coefficients of Selling's decomposition (7) of the matrix $D(\theta, \mu)$ for $\theta \in [0, \pi]$ and $\mu = 0.1$, see (13). The vertical bars correspond to the angles $0 = \theta_0 < \cdots < \theta_N = \pi$ where the support $e_0, e_1, e_2 \in \mathbb{Z}^2$ of the decomposition changes, see legend

This parametrization is closely related to the Pauli matrices in quantum mechanics. A $\mathbf{D}(x, y)$-obtuse superbase is known explicitly, depending on a suitable triangulation of the unit disc, see Fig. 1right.

Definition 2 (Discrete Degenerate Ellipticity [Obe06]) A numerical scheme on a finite set X is a map $F : U \to U$, where $U := \mathbb{R}^X$ is the set of functions from X to $\mathbb{R}$, of the form:

$$Fu(x) := F(x,\ u(x),\ (u(x) - u(y))_{y \in X \setminus \{x\}}), \tag{10}$$

for all $u \in U$, $x \in X$. It is Discrete Degenerate Elliptic (DDE) iff F is non-decreasing w.r.t. the second argument $u(x)$, and w.r.t. each $u(x) - u(y)$, $y \in X \setminus \{x\}$.

Notation the expression $Fu(x)$ should only be regarded as a shorthand for the accurate yet more verbose (10, right). In our application $X := \Omega_h$.

The numerical scheme $-\Delta_h^D$ is DDE on Ω_h, thanks to the non-negativity of the weights $(\rho_i)_{0 \leq i \leq 2}$, and to the finite differences expression (4, right) and (5), where u is extended to $\partial\Omega$ with the provided Dirichlet boundary values. On this basis we obtain a DDE discretization of nonlinear second order operators in Bellman form (1)

$$\Lambda_h u(x) := \sup_{\alpha \in \mathcal{A}} a_\alpha + b_\alpha u(x) - \Delta_h^{D_\alpha} u(x), \quad \Lambda_h u(x) = \Lambda u(x) + O(h^r), \tag{11}$$

where again $r = 1$ if x is adjacent to $\partial\Omega$, and $r = 2$ otherwise, at least if $\mathcal{A}$ is compact—which is the case for the Pucci operator. As shown in the next section, the supremum in (11, left) can be computed analytically in closed form, for the Pucci PDE, so that the numerical scheme Λ_h is explicit in terms of the unknown u.

Efficient Construction of the Jacobian Matrix of the Numerical Scheme We use a Newton method to solve the discretized PDE, which requires assembling the sparse Jacobian matrix of the numerical scheme (11). In order to describe this essential step, let us rewrite the scheme in the following form (omitting the scale h for readability)

$$\max_{\alpha\in\mathcal{A}} \mathcal{F}(\alpha, x, u(x),\ (u(x)-u(y_1(x)))_{i=1}^{I}) = 0. \tag{12}$$

In comparison with (10), the expression (12) emphasizes (1) that F is defined as a maximum over a parameter set $\mathcal{A}$, and (2) that the active stencil $y_1(x), \cdots, y_I(x)$ of a point $x \in \Omega_h$ only involves a small number of neighbors. The Jacobian matrix construction, at a given $u : \Omega_h \to \mathbb{R}$, involves the following steps:

1. Compute the maximizer $\alpha^*(x)$ in (12), for each $x \in \Omega_h$.
2. Differentiate the function $\mathcal{F}(\alpha^*(x), x, \delta,\ (\eta_i)_{i=1}^{I})$ w.r.t. parameters δ and $\eta_1, \cdots, \eta_I$, at the values $u(x)$ and $u(x) - u(y_i(x))$, $1 \le i \le I$, respectively.
3. Fill the corresponding entries of the sparse Jacobian matrix. More precisely, omitting the arguments of $\mathcal{F}$ for readability

$$J_{x,x} = \frac{\partial\mathcal{F}}{\partial\delta} + \sum_{1\le i\le I} \frac{\partial\mathcal{F}}{\partial\eta_i}, \qquad J_{x,y_i(x)} = -\frac{\partial\mathcal{F}}{\partial\eta_i},\ 1 \le i \le I.$$

A custom automatic differentiation toolbox, open source and developed by the third author, makes these operations transparent. The above computations rely on the envelope theorem [Car01], which states that the value function to an optimization problem, here (12), over a compact set, here $\mathcal{A}$, is differentiable w.r.t. the parameters, here δ and $(\eta_i)_{i=1}^{I}$, whenever the problem solution, here $\alpha^*(x)$, is single valued (which is a generic property). In addition the first order derivatives have the expression used above, obtained by freezing the optimization parameter $\alpha \in \mathcal{A}$ to the optimal value $\alpha^*(x)$.

2.2 The Pucci Operator

The Bellman form of the Pucci operator (3) involves a family of matrices $D(\theta, \mu)$, parameterized by the inverse $0 < \mu \le 1$ of their condition number, and by an angle $0 \le \theta \le \pi$. As a starter, we rewrite those in the form (9)

$$D(\theta, \mu) = (1+\mu)\,\mathbf{D}\Big(\frac{1-\mu}{1+\mu}\ \mathrm{n}(2\theta)\Big), \tag{13}$$

where $\mathrm{n}(\varphi) := (\cos\varphi, \sin\varphi)$. Note that the argument of $\mathbf{D}$ in (9) describes a circle of fixed radius $\frac{1-\mu}{1+\mu}$ within the unit disc, see Fig. 1. Thus one can find $0 = \theta_0 < \cdots < \theta_N = \pi$, where $N = N(\mu)$, such that on each interval $[\theta_n, \theta_{n+1}]$ the superbase (e_0^n, e_1^n, e_2^n) is $D(\theta, \mu)$-obtuse and the coefficients in (7) take the form

$$\rho_i(\theta) = -\langle e_{i-1}^n, D(\theta, \mu)e_{i+1}^n\rangle = \alpha_i^n + \beta_i^n \cos(2\theta) + \gamma_i^n \sin(2\theta), \tag{14}$$

for suitable constants $\alpha_i^n, \beta_i^n, \gamma_i^n \in \mathbb{R}$, $0 \le i \le 2$, $0 \le n < N$, see Fig. 2. One finds that $N(1/4) = 2$, $N(1/10) = 10$, $N(1/400) = 122$, and one can show that $N(\mu) \le C\mu^{-1}|\ln\mu|$ for some constant C independent of μ. By linearity of (8) one also has

$$\Delta_h^{D(\theta,\mu)} u(x) = \alpha^n + \beta^n \cos(2\theta) + \gamma^n \sin(2\theta) \tag{15}$$

for all $\theta \in [\theta_n, \theta_{n+1}]$, whose coefficients $\alpha^n, \beta^n, \gamma^n$ depend on ρ, h, u and x. Therefore, evaluating the discretized Bellman operator (11) associated with the Pucci equation (3) at a point $x \in \Omega_h$ amounts to solving a small number N of optimization problems, whose value is explicit. These optimization problems, and their value, take the following generic form

$$\max_{\varphi\in[\varphi_*,\varphi^*]} \alpha + \beta\cos\varphi + \gamma\sin\varphi$$
$$= \begin{cases} \alpha + \sqrt{\beta^2+\gamma^2} & \text{if } \arg(\beta + i\gamma) \in]\varphi_*, \varphi^*[, \\ \alpha + \max\{\beta\cos\varphi_* + \gamma\sin\varphi_*,\ \beta\cos\varphi^* + \gamma\sin\varphi^*\} & \text{else,} \end{cases}$$

where $\arg(\omega)$ denotes the argument of $\omega \in \mathbb{C}$, taken in $[0, 2\pi[$. In view of (15), we choose $\varphi_* = 2\theta_n$, $\varphi^* = 2\theta_{n+1}$, $\alpha = \alpha^n$, $\beta = \beta^n$, and $\gamma = \gamma^n$. Then, following (3), we take the largest value among $0 \le n < N$.

3 Numerical Experiments

We present numerical results for the Pucci equation, chosen to illustrate the qualitative behavior of the solutions, and validate the scheme robustness and accuracy on synthetic problems with known solutions. Some of the considered domains are neither smooth nor convex, and the chosen synthetic solutions range from smooth to singular.

The numerical scheme is implemented as described in the previous section, and a Newton method is used to solve the resulting coupled systems of non-linear equations. In practice, convergence to machine precision is achieved in a dozen of iterations, without damping, from an arbitrary guess. An open source Python®

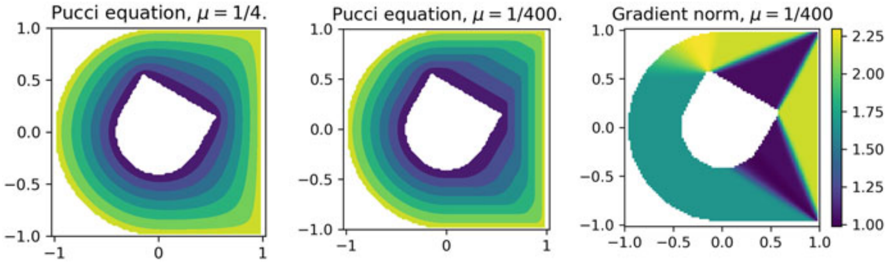


Fig. 3 Solution of the Pucci PDE with $\mu = 1/4$ (left), $\mu = 1/400$ (center, right: gradient norm)

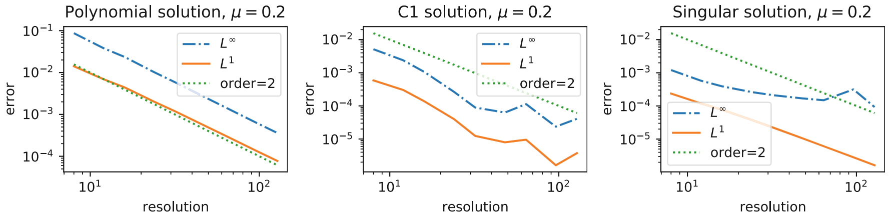


Fig. 4 Numerical error as a function of grid size, for synthetic solutions to the Pucci equation

notebook reproducing (most of) the illustrations of this paper is available on the third author's webpage[1].

We illustrate on Fig. 3 the transition of the Pucci equation from a strongly elliptic Laplacian-like PDE to a combinatorial-type convex-envelope problem, as the parameter μ takes values $1/4$ and $1/400$. The chosen domain is non-smooth and non-simply connected: $\Omega := U \setminus U'$ where $U := B(0, 1) \cup (]0, 1[\times] - 1, 1[)$ and $U' := kR_\theta(U)$ is its image under a scaling ($k = 0.4$) and a rotation ($\theta = \pi/3$). The boundary condition is 1 on ∂U, and 0 on $\partial U'$, and the r.h.s is $f \equiv 0$. The discretization grid size is 100×100, and the computation time is 1 s for $\mu = 1/4$, and 45 s for $\mu = 1/400$. The time difference is attributable to the complexity of the numerical scheme, which involves $N = 2$ pieces for in the first case and $N = 122$ in the latter, due to the larger condition number of the diffusion tensors $D(\theta, \mu)$, see Sect. 2.2. Nevertheless, the number $N = N(\mu)$ is independent of the grid scale, and both schemes are second order consistent. In the case $\mu = 1/400$, the PDE solution is quite close to the convex envelope of the boundary conditions, whose gradient is constant in some regions, and discontinuous across some lines, see Fig. 3right.

On Fig. 4, we reconstruct some known synthetic solutions from their image by the Pucci operator, with parameter $\mu = 0.2$, and their trace on the boundary. The examples are taken from the literature [FJ17, FO13], and the reconstruction errors are provided in the L^1 and L^∞ norm.

[1] Link : Github.com/Mirebeau/AdaptiveGridDiscretizations, see chapter 2.B.III.

- (Smooth example [FJ17]) $u(x) = (x^2 + y^2)^2$ on $\Omega = B(0, 1) \cup]0, 1[^2$
- (C^1 example [FO13]) $u(x) = \max\{0, \|x - x_0\|^2 - 0.2)$ on $\Omega =]0, 1[^2$.
- (Singular example [FO13]) $u(x) = \sqrt{2 - \|x\|^2}$ on $\Omega =]0, 1[^2$.

Empirically, the L^1 numerical error behaves like $\mathcal{O}(h^2)$, where h is the grid scale (inverse of resolution in images). The L^∞ error behaves like $\mathcal{O}(h^2)$ in the smooth and C^1 examples, but decays more slowly for the singular solution. *Note: we rotated the Cartesian discretization grid by $\pi/3$ in these experiments, since otherwise the perfect alignment of the domain boundary with the coordinate axes gives an unfair advantage to grid based methods (like ours).*

4 Conclusion

In this paper, we presented a new strategy for discretizing non-divergence form, fully-nonlinear second order PDEs, and applied it to the Pucci equation. The steps of this approach can be summarized as follows: (1) rewrite the problem in Bellman form, as an extremum of linear equations, (2) discretize the second order linear operators using monotone finite differences based on Selling's decomposition of positive definite matrices, (3) solve the pointwise optimization problems involved in the numerical scheme definition, either explicitly (as could be done here), or numerically.

This methodology yields finite difference schemes which are degenerate elliptic, second order consistent, and use stencils of fixed size, in contrast with existing approaches [Obe08] which cannot achieve all these desirable properties simultaneously. Numerical experiments confirm that the proposed scheme can extract smooth PDE solutions with second order accuracy, and that it remains stable and convergent for harder problems involving a singularity at a point or along a line. Future research will be devoted to extending the results to other PDEs, such as the Monge–Ampère equation and its variants.

Acknowledgments This work was partly supported by ANR research grant MAGA, ANR-16-CE40-0014. The first author acknowledges support from the FIME laboratory, and from the Labex Mathématiques Hadamard and iCODE (IDEX Paris-Saclay) for the PGMO project "Optimal control of conservation equations".

References

BCM16. Jean-David Benamou, Francis Collino, and Jean-Marie Mirebeau. Monotone and consistent discretization of the Monge–Ampere operator. *Mathematics of computation*, 85(302):2743–2775, 2016.

BOZ04. J Frédéric Bonnans, Elisabeth Ottenwaelter, and Housnaa Zidani. A fast algorithm for the two dimensional HJB equation of stochastic control. *ESAIM: Mathematical Modelling and Numerical Analysis*, 38(4):723–735, 2004.

BS91. Guy Barles and Panagiotis E Souganidis. Convergence of approximation schemes for fully nonlinear second order equations. *Asymptotic analysis*, 4(3):271–283, 1991.

Car01. Michael Carter. *Foundations of mathematical economics*. MIT Press, 2001.

CIL92. Michael G Crandall, Hitoshi Ishii, and Pierre-Louis Lions. User's guide to viscosity solutions of second order partial differential equations. *Bulletin of the American Mathematical Society*, 27(1):1–67, 1992.

FJ17. Xiaobing Feng and Max Jensen. Convergent semi-Lagrangian methods for the Monge–Ampère equation on unstructured grids. *SIAM Journal on Numerical Analysis*, 55(2):691–712, 2017.

FM14. Jérôme Fehrenbach and Jean-Marie Mirebeau. Sparse non-negative stencils for anisotropic diffusion. *Journal of Mathematical Imaging and Vision*, 49(1):123–147, 2014.

FO13. Brittany D Froese and Adam M Oberman. Convergent Filtered Schemes for the Monge–Ampère Partial Differential Equation. *SIAM Journal on Numerical Analysis*, 51(1):423–444, January 2013.

Mir18. Jean-Marie Mirebeau. Fast-marching methods for curvature penalized shortest paths. *Journal of Mathematical Imaging and Vision*, 60(6):784–815, 2018.

Mir19. Jean-Marie Mirebeau. Riemannian Fast-Marching on Cartesian Grids, Using Voronoi's First Reduction of Quadratic Forms. *SIAM Journal on Numerical Analysis*, 57(6):2608–2655, 2019.

Obe06. Adam M Oberman. Convergent Difference Schemes for Degenerate Elliptic and Parabolic Equations: Hamilton-Jacobi Equations and Free Boundary Problems. *SIAM Journal on Numerical Analysis*, 44(2):879–895, January 2006.

Obe08. Adam M Oberman. Wide stencil finite difference schemes for the elliptic Monge–Ampere equation and functions of the eigenvalues of the Hessian. *Discrete Contin Dyn Syst Ser B*, 2008.

Sch09. Achill Schürmann. Computational geometry of positive definite quadratic forms. *University Lecture Series*, 49, 2009.

Sel74. Eduard Selling. Ueber die binären und ternären quadratischen Formen. *Journal fur die Reine und Angewandte Mathematik*, 77:143–229, 1874.

A Multi-Scale Flow Model for Studying Blood Circulation in Vascular System

Ulin Nuha Abdul Qohar, Antonella Zanna Munthe-Kaas, Jan Martin Nordbotten, and Erik Andreas Hanson

Abstract In this paper, we demonstrate a multi-scale model for studying blood flow in the vascular structures of an organ. The model may be used for a tracer concentration flow simulation replicating Dynamic Contrast-Enhanced Magnetic Resonance Imaging (DCE–MRI) data. A 1D vascular graph model that represents blood flow through a vascular vessel network is coupled with a single-phase Darcy flow model for the capillary bed which is assumed as a porous media. Numerical experiments show the blood circulation in the system closely related to the structure and parameter of the vascular system, that gives qualitatively realistic tracer concentration flow. This model is a starting point for further investigation in development into clinical applications, using both real data and MRI analysis software.

1 Introduction

In the recent decades, numerical models and computational approaches have been developed intensively to study the blood circulation system [1–4]. The fundamental purpose of developing a mathematical model is to replicate the blood circulatory system in the human body. One of the main challenges for this model is the fact that vascular systems are made of a massive number of vessels at various scales [4], ranging from large arteries down to arterioles, and capillaries. These creates several limitations for conducting simulations in full vascular systems. For instance, it needs a huge computing resources and the model is not relevant for finer scale.

In the current work, a multi-scale model for blood circulation is proposed. This model is coupling a 1D flow model and single-phase Darcy flow model. We describe the flow in the vascular network (arteries and veins) using a 1D vascular graph model [1]. In line with the previous work [5], a pressure drop model was introduced at

U. N. A. Qohar (✉), A. Z. Munthe-Kaas, J. M. Nordbotten, and E. A. Hanson
University of Bergen, Bergen, Norway
e-mail: ulin.qohar@uib.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_73

vessel bifurcations which compensate the accuracy loss of the linearized model [6]. Darcy flow was used for the continuum representation of the micro-circulation in the capillary bed. The tracer concentration flow was modelled based on the pressure field from the flow model. The boundary condition for our model were defined in the inlet and outlet pressure. This tracer concentration flow model presents a possibility to generate a predicted digital MRI data that will improve the MRI analysis tools with further investigation. It may also lead to a breakthrough in the development of personalised medicine.

2 Materials and Methods

2.1 Flow Model

A system of equations was constructed based on vascular network structures for both arteries and veins. Assuming laminar flow and non-slip conditions on the vessel walls, each vessel i was described as a long cylindrical tube of length L_i with constant radius $r_i \ll L_i$. The pressure drop ΔP_i in a single vessel segment i was computed using Hagen–Poiseuille's law [1]

$$\Delta P_i^h = \frac{8\mu L_i q_i}{\pi r_i^4}, \tag{1}$$

where index h stands for hydrodynamics, μ is the blood viscosity and q_i is a volumetric blood flow. At a junction node, the pressure drop was estimated based on [6, 7],

$$\Delta P_i^b = \frac{\rho q_{dat}^2}{2\pi^2 r_{dat}^4} \left(1 + \frac{q_i^2 r_{dat}^4}{q_{dat}^2 r_i^4} - \frac{2 q_i r_{dat}^2}{q_{dat} r_i^2} \cos\left(\frac{3}{4}\theta_{(dat,i)}\right) \right), \tag{2}$$

where the upper index b stands for bifurcation and the index dat refer to the *datum* vessel, i.e. the vessel from which the flow approaches the junction. Further, $\theta_{(dat,i)}$ is the angle between the datum vessel and vessel i. Hence, the total pressure drop after a bifurcation node was computed as the sum of both Eqs. (1) and (2). The other governing equation is the conservation of mass at a node, $\sum q_{in} = \sum q_{out}$, with q_{in} representing the blood that flows into the node and q_{out} is a flow out of the node.

The capillary bed was discretized with a uniform grid and described by Darcy's single-phase flow equation. Darcy's law, which demonstrates the flow of a fluid in a porous medium, states that a fluid flows from regions of higher pressure to regions of lower pressure. Thus

$$\mathbf{v} = -\frac{\mathbf{K}}{\mu}\nabla P, \tag{3}$$

where $\mathbf{v}$ is the Darcy flux (volumetric flow rate per unit area, $\mathbf{K}$ is the permeability tensor of the porous medium and μ is the viscosity [8]. In addition, we assume conservation of mass (continuity equation),

$$\frac{\partial(\Phi\rho)}{\partial t} + \nabla \cdot (\rho\mathbf{v}) = \rho Q \tag{4}$$

where Φ is the porosity of the capillary bed and Q is the source term. In this model, the source,Q is either describing the flow in or out a terminal node arteries or veins. We assume blood to be an incompressible fluid, and by incorporating Darcy flow into the continuity equation, we obtain

$$-\nabla \cdot (\frac{\mathbf{K}}{\mu}\nabla P) = Q. \tag{5}$$

To complete the system, both the vascular network and Darcy systems were combined with the terminal nodes of the arterial and venal networks as point sources/sinks. A mollifier function is introduced [9]

$$f(x) = \begin{cases} C\exp\left(\frac{-1}{1-|x|^2}\right) & \text{if } |x| < 1 \\ 0 & \text{if } |x| \geq 1 \end{cases} \tag{6}$$

and used with a finite radius ϵ, $f^{\epsilon}(x) = f(\frac{x}{\epsilon})$ on the Darcy domain.

2.2 Tracer Concentration Flow

In this work, the contrast agent was simulated as a pulse of concentration in the input vessles. The tracer concentration influx into a small distribution volume Ω_β was determined by the product of tracer concentration (c_β) and flux (u_β)

$$-\int_{\partial\Omega_\beta} c_\beta(u_\beta \cdot n)dA, \tag{7}$$

with index β represents the numerical domain related (both vessel network and the capillary bed). The tracer concentration, $c(x,t)$ is the concentration in the blood. But, the value observed by MRI scanning is tracer indicator, $C(x,t)$. The relation between them is $C(x,t) = c(x,t)\Phi(x,t)$, which is the total tracer concentration over a tissue volume. Tracer indicator was defined as the total concentration over model layers,

$$C(x,t) = C_{cap}(x,t) + C_{VG}(x,t), \tag{8}$$

where the index cap stands for the capillary bed and VG for vascular graph network. A contrast agent was injected into the arterial root vessel as Arterial Input Function (AIF). It was simulated using a gamma variate function [9].

2.3 Model Implementation

The flow model was constructed by four main components, i.e., vascular network, Darcy model, coupling, and boundary condition. The vascular network, consisting of segments and nodes, was governed by (1) and (2), and mass conservation. The system of equation for both arterial and venous network was described by similar equations that were altered accordingly for each network. The nonlinear system of equation, $\mathbf{A}_{VG}\mathbf{x} = \mathbf{b}$ was constructed where $\mathbf{A}_{VG}$ has no more than four non-zeroes entries per row.

The Darcy equations (5) in the capillary bed were solved using a two-point flux approximation (TPFA). Applying this TPFA discretization procedure for all cells in the domain, we obtain a system of equations $\mathbf{A}_D\mathbf{x} = \mathbf{b}$, where $\mathbf{A}_D$ is a symmetric matrix. Even if the TPFA method is not a consistent discretization for general grids, it is consistent and convergent on the quadrilateral grids used in this study [10].

In our four roots terminal, the boundary conditions were constructed by assigning constant pressure on the arterial and the venal root nodes (Dirichlet BC) and conservation of mass for the whole system.

$$P_\beta = P_{ext}, \tag{9}$$

$$\sum_{i \in N_{root}} q_i = 0, \tag{10}$$

where $\beta = \{a, v\}$ is a root node on the arterial or the venal structure, $P_{ext} = \{in, out\}$ represents the constant pressure on two arterial roots (input) and two venal roots (output), and N_{root} are root segments, with $q_i > 0$ for the artery, $q_i < 0$ for the vein.

Finally, the vascular networks and the Darcy domains were combined in a (nonlinear) system of equations $\mathbf{Ax} = \mathbf{b}$, with unknown $\mathbf{x}$ consisting of the pressure and the flow rate in the model,

$$\begin{pmatrix} \mathbf{A}_{VG} & \mathbf{A}_{VG-D} \\ \mathbf{A}_{D-VG} & \mathbf{A}_D \end{pmatrix} \begin{pmatrix} \mathbf{x}_{VG} \\ \mathbf{x}_D \end{pmatrix} = \begin{pmatrix} \mathbf{b}_{VG} \\ 0 \end{pmatrix}. \tag{11}$$

The index VG refers to the vascular graph network, and index D stands for the Darcy equation discretization. The additional matrices $\mathbf{A}_{VG-D}$ and $\mathbf{A}_{D-VG}$ are the coupling equations for the vascular graph and Darcy system of equations. The unknown solution $\mathbf{x}_{VG}$ is the pressure at the nodes and flow rate in the corresponding segment and $\mathbf{x}_D$ is the pressure on the Darcy domain.

Table 1 Frog tongue model parameters

Parameter	Value	Unit
Capillary model size (2D)	515×634	pixel
Real size on simulation	$30.9 \times 38.04 \times 0.6$	mm
Porosity of capillary bed (ϕ)	0.1	–
Permeability of the capillary bed (K)	3×10^{-6}	mm^2
Resistance estimation for coupling	5×10^{-4}	$\text{kg mm}^{-4}\ \text{s}^{-1}$
Viscosity of blood (μ)	$3 \cdot 10^{-6}$	kPa·s
Pressure inlet	10	kPa
Pressure outlet	4	kPa

Despite the nonlinearity in the governing equations, it can be solved efficiently by using the Schur complement method. It was done by solving the linear system blocks and subtituting it back into the system of equations. The nonlinear system was solved using the solution of the linearized system as initial approximation for the nonlinear solver using Trust-Region-Dogleg Algorithm in MATLAB.

It was assumed that the blood flow is stationary flow and the tracer is only following the bloodstream which was generated by solving the flow model. The tracer concentration flow was computed by solving the ordinary differential equation of the tracer concentration change over time for the whole domain. It tracked the tracer movement along the bloodstream.

In this work, blood flow was simulated in the 2D frog tongue anatomy from classic biological textbook [11]. The image may be seen as a synthetic test case, but with a vascular network resembling a real vascular system in an organ. It also had small number of pixels that make it fast and easy to simulate, visualize, and analyze the result compared to a 3D data. This data is a good starting point to test and evaluate the model. The parameters for our simulations are defined in Table 1.

3 Result and Discussion

The flow model was defined to describe the pressure field in the whole domain. Figure 1 shows the original vascular anatomy of the frog tongue and the computed pressure field in the capillary bed. The network structure was based on an image segmentation of these using the method in [12]. It was observed that the pressure is higher around the top edges and lower in the middle and left bottom edge of the domain. This result was related to the vascular structure having a small number of arterial terminals in the central region and several venous terminals. For instance, two vein terminals in the lower central region are located far away from arterial terminals (see Fig. 1). In conclusion, the unbalanced pair of arterial and venous in some region caused either a higher or lower pressure in those region compared to the remaining part of domain. Based on the physiological knowledge, the vascular

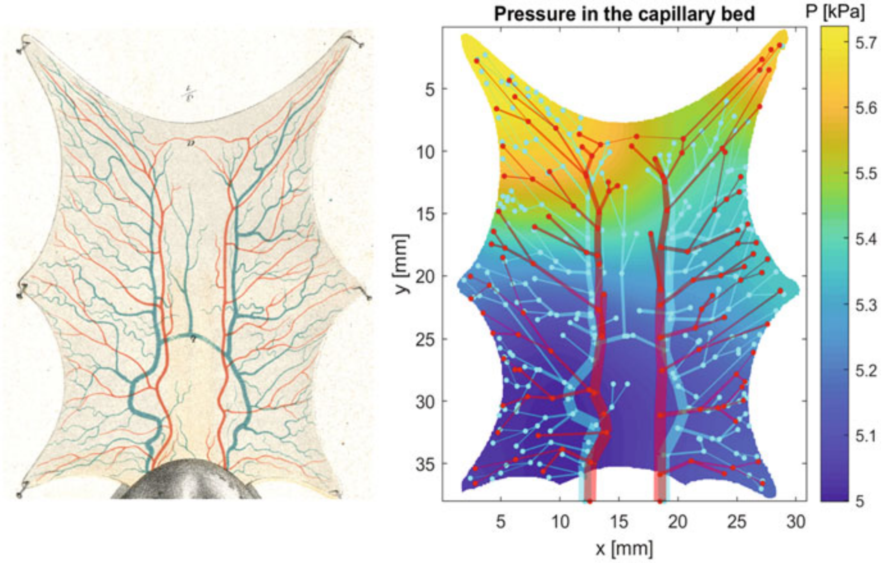


Fig. 1 Left: An anatomical frog tongue image from a classical textbook [11], with arterial (red) and venal (blue) vascular network structures. The vascular networks in this paper are obtained by segmentation of the anatomical network structures. Our capillary domain is the region inside the tongue boundary. Right: the pressure distribution in the capillary bed spreads with a high pressure at the top and decreases gradually to the lowest at the bottom.. The tracer concentration flow was computed based on this pressure field

structure in a part of an organ should have a balanced pair of arteries and veins therefore blood that flows from an arterial terminal will be absorbed by nearby venous terminals. This structures ensure the whole tissue regions oxigenated.

The total pressure drop in the system was 6 kPa (see Table 1), the arterial network being the major contributor (73.4% of total), the capillary bed 6.6%, and the vein vascular network 19.95%. These values indicated the importance of the arterial vascular structure to provide blood circulation through the whole organ, in line with the measurements and simulations from the literature [13]. If there is an alteration on the arterial vessel, its impact for blood circulation will be greater than for a similar vein alteration. The pressure drop in the capillary domain was inversely proportional to the permeability parameter (Table 1). Higher permeability allows blood flow faster in the capillary hence would decrease the pressure drop.

The tracer concentration flow on Fig. 2 has a good replication of perfusion MRI data. The tracer flows from the arterial roots into the capillary domain and is evacuated by the venous network. The arterial vessels provide faster access for blood to reach the whole domain. The area with a denser arterial terminal nodes got higher contrast compared to the other regions. The tracer concentration dispersed to nearby capillary domain afterwards. It observed that the middle-lower field was unreachable by the tracer. It caused by the nonexistence of arterial terminal nodes

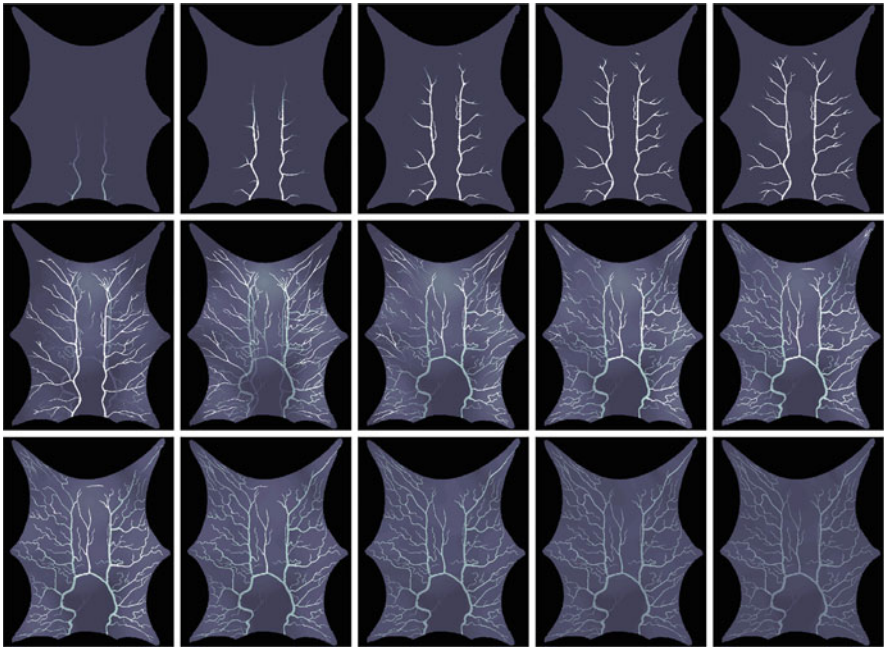


Fig. 2 Tracer concentration indicator flow in the frog tongue. The indicator shows blood flow in the vascular system in 13 s–134 s after bolus injection, with assumed delay 12 s. The tracer flow replicate the perfusion MRI data in the phantom data

in that region. The venous network absorbed the tracer gradually from the capillary bed in a balanced order. The tracer in the whole domain was evacuated slowly and periodically. The big connected vein vessel is the key for balancing the circulation in the venous system. This vessel may provide a bridge to maintain the flow in case part of the network is damaged [5].

The real data comparison is necessary for validation of our model. The tracer dispersed slowly from the capillary domain to the venous network. This condition will only occur in the extravascular flow (i.e. blood leakage in the tissue). The 2D image has an unrealistic physiology structures causing un-expected spatial artefacts that may obstruct the blood to spread. Thus the tracer concentration stays too long in the organ. The geometry refinement is outside the scope of this paper. The homogeneous permeability also contributes to this result as the real physiological permeability is not isotropic. The further research will be required to define anisotropic permeability to produce more realistic simulation result.

4 Conclusion

We have presented a multi-scale flow model to simulate blood circulation in a vascular system. In the numerical experiment, our model gives a realistic simulation result for blood circulation. The pressure field has some abnormalities due to the location of nodes, which comes from vessel segmentation input on the simulation. The tracer concentration flows according to the pressure field in the model. The unbalanced area of the arterial terminals in the capillary bed caused the tracers to spread unevenly throughout the organ tissue. In contrast to that, the existence of a big connected vessel in venous networks has a vital role in maintaining blood circulation evenly in the left and right side of the frog tongue.

This model has a systematic mathematical structure. Although constructed by a system of nonlinear equations, the flow model is solved in an efficient scheme by utilizing the construction of its components. Furthermore, the simulation result shows reliable physiology of the system and the simulation can be performed in regular PC. The systematic structure of the flow model makes it easy to use and develop further. Another application is to study the blood circulation in the local vascular system (of an organ) based on vessel segmentation results and study the blood circulation in the altered vascular network. The parameter input model has to be adjusted to replicate more realistic DCE–MRI data for further works.

References

1. J. Reichold, M. Stampanoni, A. L. Keller, A. Buck, P. Jenny, and B. Weber, “Vascular graph model to simulate the cerebral blood flow in realistic vascular networks,” *Journal of Cerebral Blood Flow and Metabolism*, vol. 29, no. 8, pp. 1429–1443, 2009, pMID: 19436317. [Online]. Available: https://doi.org/10.1038/jcbfm.2009.58
2. T. Passerini, M. d. Luca, L. Formaggia, A. Quarteroni, and A. Veneziani, “A 3d/1d geometrical multiscale model of cerebral vasculature,” *Journal of Engineering Mathematics*, vol. 64, no. 4, p. 319, Mar 2009. [Online]. Available: https://doi.org/10.1007/s10665-009-9281-3
3. P. Perdikaris, L. Grinberg, and G. E. Karniadakis, “Multiscale modeling and simulation of brain blood flow,” *Physics of Fluids*, vol. 28, no. 2, p. 021304, 2016. [Online]. Available: https://doi.org/10.1063/1.4941315
4. A. Quarteroni, A. Veneziani, and C. Vergara, “Geometric multiscale modeling of the cardiovascular system, between theory and practice,” *Computer Methods in Applied Mechanics and Engineering*, 2016.
5. U. N. Qohar, A. Z. Munthe-Kaas, J. M. Nordbotten, and E. Hanson, “A multi-scale flow model for blood regulation in a realistic vascular system,” *Theoretical Biology and Medical Modelling (in review)*, 02 2020. [Online]. Available: https://dx.doi.org/10.21203/rs.2.23112/v1
6. C. Chnafa, K. Valen-Sendstad, O. Brina, V. Pereira, and D. Steinman, “Improved reduced-order modelling of cerebrovascular flow distribution by accounting for arterial bifurcation pressure drops,” *Journal of Biomechanics*, vol. 51, pp. 83–88, 2017. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929016312672

7. J. P. Mynard and K. Valen-Sendstad, “A unified method for estimating pressure losses at vascular junctions,” *International Journal for Numerical Methods in Biomedical Engineering*, vol. 31, no. 7, pp. n/a–n/a, 2015, cnm.2717. [Online]. Available: http://dx.doi.org/10.1002/cnm.2717
8. H. Darcy, “Les fontaines publique de la ville de dijon,” p. 570, 1856.
9. E. Hodneland, E. Hanson, O. Sævareid, G. Nævdal, A. Lundervold, V. Šoltészová, A. Z. Munthe-Kaas, A. Deistung, J. R. Reichenbach, and J. M. Nordbotten, “A new framework for assessing subject-specific whole brain circulation and perfusion using MRI-based measurements and a multi-scale continuous flow model,” *PLOS Computational Biology*, vol. 15, no. 6, pp. 1–31, 06 2019. [Online]. Available: https://doi.org/10.1371/journal.pcbi.1007073
10. I. Aavatsmark, “Interpretation of a two-point flux stencil for skew parallelogram grids,” *Computational Geosciences*, vol. 11, no. 3, pp. 199–206, Sep 2007. [Online]. Available: https://doi.org/10.1007/s10596-007-9042-1
11. J. F. Cohnheim, *Untersuchungen über die embolischen Prozesse*, Berlin: Hirschwald, 1872.
12. E. A. Hanson and A. Lundervold, “Local/non-local regularized image segmentation using graph-cuts,” *International Journal of Computer Assisted Radiology and Surgery*, vol. 8, no. 6, pp. 1073–1084, Nov 2013.
13. A. R. Pries, T. W. Secomb, and P. Gaehtgens, “Design principles of vascular beds,” *Circulation Research*, vol. 77, no. 5, pp. 1017–1023, 1995.

The 8T-LE Partition Applied to the Barycentric Division of a 3-D Cube

Miguel A. Padrón and Ángel Plaza

Abstract The barycentric partition of a 3D-cube into tetrahedra is carried out by adding a new node to the body at the centroid point and then, new nodes are progressively added to the centroids of faces and edges. This procedure generates three types of tetrahedra in every single step called, Sommerville tetrahedron number 3 (ST3), *isosceles trirectangular* tetrahedron and *regular right-type* tetrahedron. We are interested in studying the number of similarity classes generated when the 8T-LE partition is applied to these tetrahedra.

1 Introduction

The problem of subdviding meshes containing hexahedra, tetrahedra, pyramids and prisms into a consistent set of tetrahedra, appears in many fields of engineering, such as, Computer Graphics and CAD, Geometrical Modelling, Geometric and Engineering Design and the Finite Element Methods [4].

One of the main applications is in Finite Element Method, when an unstructured tetrahedral solver is used to tackle a problem on an hybrid mesh, in which non tetrahedral elements must be subdivided into tetrahedra. This problem is also a major concern in Compurter Graphics when a non tetrahedral mesh must be subdivided into tetrahedra for example to use efficient algorithms for volume rendering, iso-contouring and particle advection that exist for mesh topologies including only tetrahedra [4].

M. A. Padrón (✉) · Á. Plaza
Division of Mathematics, Graphics and Computation (MAGiC), IUMA, Information and Communication Systems, University of Las Palmas de Gran Canaria, Canary Islands, Spain
e-mail: miguel.padron@ulpgc.es; angel.plaza@ulpgc.es

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_74

Without additional nodes, a hexahedron can be subdivided either into five or six tetrahedra. By adding the hexahedron centroid as a data point, we can generate a subdivision into 12 tetrahedra, all of them Sommerville tetrahedra number 3 (ST3) similar to one another, where each face is still split by a single diagonal. From here, we can progressively add face centroids, dividing a face into four triangles to produce, 14, 16, 18, 20, 22 or 24 isosceles trirectangular tetrahedra [1]. Finally, a further subdivision is carried out by adding new vertices on the cube's edges to obtain 26, 28, 30, 32, 34, 36, 38, 40, 42, 44, 46 or 48 regular right-type tetrahedra [3].

In this work we prove that the iterative application of the 8T-LE partition to those tetrahedra mentioned before, yields into a sequence of tetrahedra where the number of similarity classes is bounded, hence the non-degeneracy of the tetrahedral meshes follows, and the minimun angle condition is also satisfied. Although some tetrahedra generated are obtuse type, for the Sommerville number three and regular right-type tetrahedra, all the descendants are similar to the original one, saving many CPU time and moreover, some superconvergence phenomena can be achived. For the isosceles trirectangular tetrahedron we will prove that asymptotically, most of the tetrahedra generated are isosceles right-type tetrahedra.

2 The Barycentric Partition of a 3D-Cube

The barycentric partition of a 3D-cube is achieved by adding a node at the centroid of the cube, followed hierarchically, by new nodes to the centroids of the faces and edges. Although it is logically possible to construct subdivisions following another order, but we are unaware of any cases where this has been done [3].

Figure 1 shows the different steps until the complete division of the 3D-cube is carried out. For the first case, each face of the 3D-cube is joined to the center of the cube, resulting in 6 square pyramids and a Body Centered Cubic is achieved (BCC). Each of these is then subdivided into two tetrahedra by adding an arbitrary diagonal across the face (Face Divided), generating 12 Sommerville tetrahedra number 3 (ST3) [3], and all of them similar to each other, see Fig. 1a.

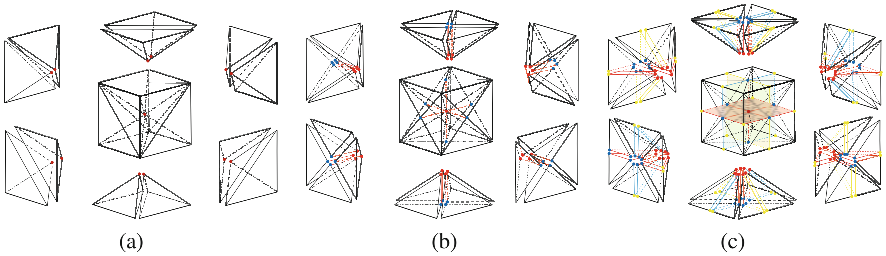


Fig. 1 Different steps for the subdivison of a 3D-cube. (**a**) BCC and Face Divided (FD). (**b**) Face Centered (FC). (**c**) Edge Divided (ED)

A futher subdivision is carried out by adding progressively face centroids (Face Centered) [1], splitting every single triangle into two sub-triangles generating up to 24 isosceles trirectangular tetrahedra congruent to one another, see Fig. 1b.

Finally, the complete barycentric partition of a 3D-cube is achieved by adding progressively new edge centroids (Edge Divided), dividing each triangle into two sub-triangles generating up to 48 regular right-type tetrahedra similar between them, see Fig. 1c, and it is equivalent to the Freudenthal partition of a 3D-cube.

3 The 8T-LE Partition

The 8T-LE partition can be seen as the natural extension to 3D of the 4T-LE partition introduced by Rivara [12]. This partition was introduced and developed by Plaza and Carey [6], and it has also been widely studied [5, 8–11].

For this partition, the tetrahedra must be previously classified depending on, the relative positions of their longest-edges of the triangular faces as type 1, type 2 and type 3 [6]. The 8T-LE partition applied to any tetrahedron t can be described algorithmically as follows:

```
8T-LE partition
/* Input variables: $t$ tetrahedron
   Output variables: new sub-tetrahedra */
   1.- Procedure classification
         $t$ is classified as type 1, 2 or 3
   2.- Division of the skeleton
         4T-LE partition is applied to the triangular faces
   3.- Division of t
         $t$ is subdivided into 8 sub-tetrahedra according to
         the division of the skeleton
/* Output: Division of $t$
```

Since there are three different similitary classes of tetrahedra in the barycentric partition of the 3D-cube, we will study the 8T-LE of each one of these classes.

3.1 The 8T-LE Partition of the Sommerville Tetrahedron Number 3 (ST3)

The tetrahedron chosen has as coordinates $0 = (0, 0, 0)$, $A = (0.5, 0.5, 0.5)$, $B = (0, 1, 0)$ and $C = (0, 1, 1)$, see Fig. 2. We just have to study two types of sub-tetrahedra depicted in Fig. 2 (on the right hand side). For the first one, just doing a $\pi/2$ clockwise rotation around the axis in blue colour, and then, two mirror reflections through planes $\pi_3 \equiv y - 1 = 0$ and $\pi_4 \equiv x = 0$, respectively, it is

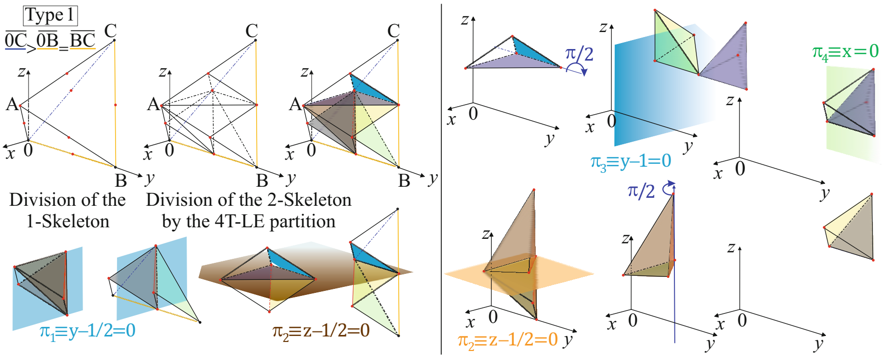

Type 1
0C > 0B = BC
C
A
B
z
x
0
y
Division of the 1-Skeleton
Division of the 2-Skeleton by the 4T-LE partition
π1≡y−1/2=0
π2≡z−1/2=0
π/2
π3≡y−1=0
π4≡x=0
π/2
π2≡z−1/2=0

Fig. 2 The 8T-LE partition of the ST3 tetrahedron through the skeleton. The sub-tetrahedra generated have been depicted in pairs, where each one is similar to the other by a mirror reflection through planes $\pi_1 \equiv y - 0.5 = 0$ and $\pi_2 \equiv z - 0.5 = 0$, respectively. The sub-tetrahedron on the top with one of its vertices C, and the other sub-tetrahedron on the bottom left corner with one of its vertices 0, are similar to the original one

proved that this sub-tetrahedron is similar to the original one. Finally, for the last case, a mirror reflection through the plane $\pi_2 \equiv z - 0.5 = 0$ and then a $\pi/2$ clockwise rotation around the axis in blue colour, it is proved that this tetrahedron is also congruent to the original one.

Figure 2 shows that no new classes of similarity are generated when this partition is applied to this tetrahedron.

3.2 The 8T-LE Partition of the Regular Right-Type Tetrahedron

This tetrahedron is an ortho-simplex and also a path-tetrahedron [2, 7] and it has been widely studied in [9] for the *regular*, *scalene* and *isosceles* cases. *Regular* means, that the three mutually perpendicular edges (*legs*) are of the same length $a = b = c$ [7, 9].

Theorem 1 *Two right-type tetrahedra t(a,b,c) and* $t'(a', b', c')$ *are similar to each other if and only if their extreme legs are in the same ratio as their central legs. That is, either* $\frac{b}{b'} = \frac{a}{a'} = \frac{c}{c'}$, *or* $\frac{b}{b'} = \frac{a}{c'} = \frac{c}{a'}$ [9].

According to Theorem 1 and Fig. 3, the 8T-LE partition of a regular right-type tetrahedron generates sub-tetrahedra similar to the original one.

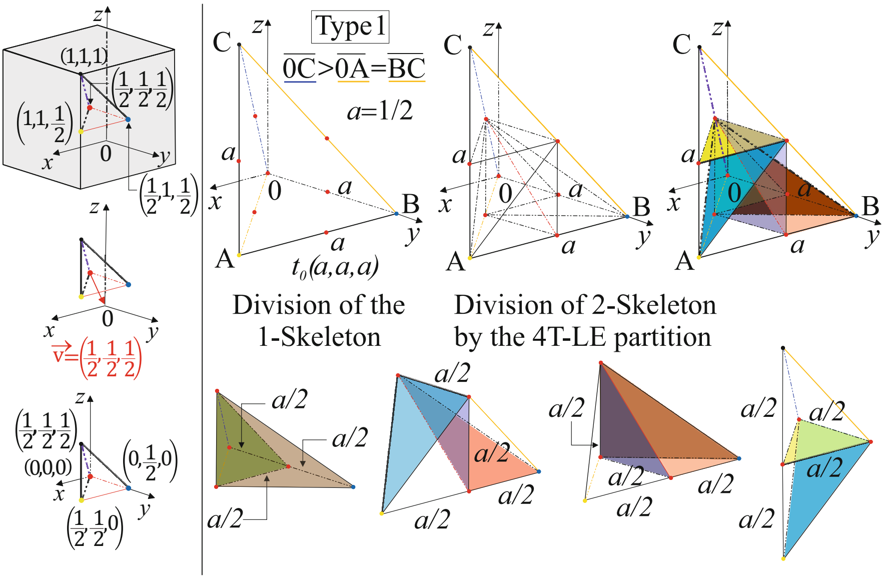

Type 1
$\overline{0C}>\overline{0A}=\overline{BC}$
$a=1/2$
$t_0(a,a,a)$
Division of the 1-Skeleton
Division of 2-Skeleton by the 4T-LE partition
(1,1,1)
$\left(\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)$
$\left(1,1,\frac{1}{2}\right)$
$\left(\frac{1}{2},1,\frac{1}{2}\right)$
$\vec{v}=\left(\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)$
(0,0,0)
$\left(0,\frac{1}{2},0\right)$
$\left(\frac{1}{2},\frac{1}{2},0\right)$
A
B
C
0
x
y
z
a
a/2

Fig. 3 The 8T-LE partition of the regular right-type tetrahedron through the skeleton. One of the 48 regular right-type tetrahedra with coordinates (1, 1, 1), (0.5, 0.5, 0.5), (0.5, 1, 0.5) and (1, 1, 0.5) is chosen to be studied, and relocated to the origin for a better visualization by a vector translation $\boldsymbol{v} = (0.5, 0.5, 0.5)$

3.3 *The 8T-LE Partition of the Isosceles Trirectangular Tetrahedron*

This tetrahedron is an ortho-simplex but not a path-tetrahedron [2, 7]. *Isosceles* means, that the three mutually perpendicular edges (*legs*) are as follows: $a = b \neq c$ [7].

According to Fig. 4, four sub-tetrahedra are congruent to the original one, so at this stage we are focussed on studying the quasi right-type sub-tetrahedra and the new two sub-tetrahedra. Both quasi right-type tetrahedra and new tetrahedra are similar to each other by a mirror reflection through the plane $\pi_5 \equiv y - z = 0$.

The Quasi Right-Type Tetrhedron and the New Tetrahedron

The last step is to study these new types of sub-tetrahedra generated when the 8T-LE partition was applied to the isosceles trirectangular tetrahedron. Figure 5 shows the different types of sub-subtetrahedra generated. Both new sub-subtetrahedra generated are also similar to one another by a mirror reflection through the plane $\pi_7 \equiv x + z = 1/2\sqrt{2}$. We leave to the reader to check that, they are of the same types of tetrahedra which already appeared studying the isosceles trirectangular

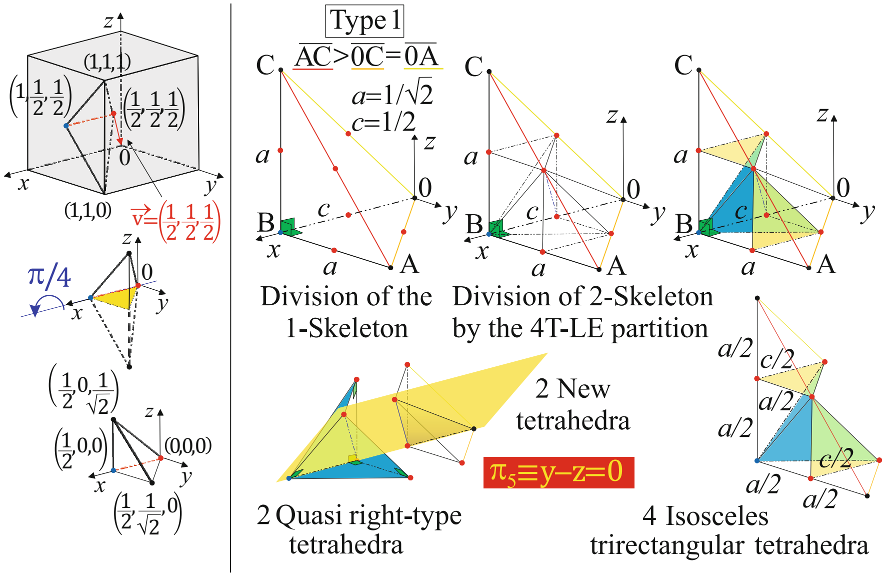

z
(1,1,1)
$\left(1,\frac{1}{2},\frac{1}{2}\right)$
$\left(\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)$
0
x
y
(1,1,0)
$\vec{v}=\left(\frac{1}{2},\frac{1}{2},\frac{1}{2}\right)$
$\pi/4$
z
0
x
y
$\left(\frac{1}{2},0,\frac{1}{\sqrt{2}}\right)$
z
$\left(\frac{1}{2},0,0\right)$
(0,0,0)
x
y
$\left(\frac{1}{2},\frac{1}{\sqrt{2}},0\right)$
Type 1
$\overline{AC}>\overline{0C}=\overline{0A}$
$a=1/\sqrt{2}$
$c=1/2$
C
a
B
x
c
z
0
y
a
A
Division of the 1-Skeleton
C
a
B
x
c
z
0
y
a
A
Division of 2-Skeleton by the 4T-LE partition
C
a
B
x
c
z
0
y
a
A
2 New tetrahedra
$\pi_5\equiv y-z=0$
2 Quasi right-type tetrahedra
a/2
c/2
a/2
a/2
c/2
a/2
a/2
4 Isosceles trirectangular tetrahedra

Fig. 4 The 8T-LE partition of the isosceles trirectangular tetrahedron through the skeleton. One of the 24 isosceles trirectangular tetrahedron with coordinates (1, 1, 1), (1, 1, 0), (1, 0.5, 0.5) and (0.5, 0.5, 0.5) is chosen to be studied, and for a better visualization, this tetrahedron is relocated to the origin by a vector translation $\boldsymbol{v} = (0.5, 0.5, 0.5)$ and a $\pi/4$ counterclockwise rotation around the axis x

tetrahedron. The quasi right-type sub-subtetrahedron has vertices $0 = (0, 0, 0)$, $A = \left(1/2\sqrt{2}, 1/4, 0\right)$, $B = \left(1/2\sqrt{2}, 1/2, 1/2\sqrt{2}\right)$ and $C = \left(1/2\sqrt{2}, 0, 1/2\sqrt{2}\right)$, after being relocated to the origin, see Fig. 6 (on the left hand side). For the new tetrahedron, its new coordinates are depicted in Fig. 6 (on the right hand side).

The different classes of similiraty generated by the 8T-LE partition applied to the new sub-subtetraedra are shown in Fig. 7. For this case, this partition only generates 4 different classes of similarity.

3.4 Classes of Similarity Generated

The classes of similarity are drawn in Fig. 8a as result of applying the 8T-LE parition to those tetrahedra. Besides, in Fig. 8b we can see the evolution of each tetrahedron class when n grows, for the case of isosceles trirectangular tetrahedron, which is the most interesting.

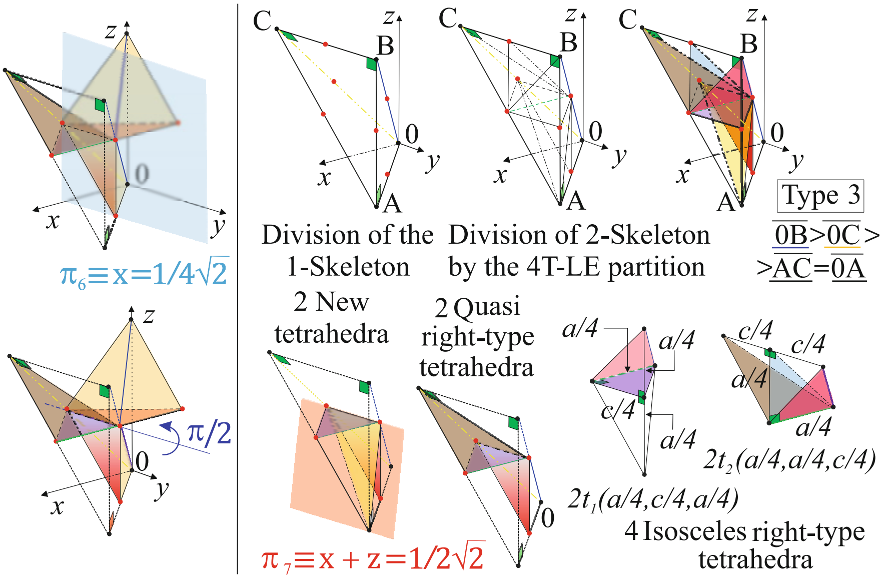

z
0
x
y
π₆≡x=1/4√2
C
B
z
0
x
y
A
Division of the 1-Skeleton
Division of 2-Skeleton by the 4T-LE partition
Type 3
0B>0C>
>AC=0A
π/2
2 New tetrahedra
2 Quasi right-type tetrahedra
a/4
c/4
2t₁(a/4,c/4,a/4)
2t₂(a/4,a/4,c/4)
4 Isosceles right-type tetrahedra
π₇≡x+z=1/2√2

Fig. 5 The 8T-LE partition of the quasi right-type tetrahedron. The quasi right-type sub-subtetrahedra on the bottom right corner with one of its vertices 0, is similar to the original one. Besides, both quasi right-type sub-subtetrahedra are similar to each other by a mirror reflection through the plane π_6 and a $\pi/2$ counterclockwise rotation around the axis in blue color. The t_1 sub-subtetrahedra generates $4t_1$ and $4t_2$, and t_2 generates $6t_2$ and $2t_1$ [9]

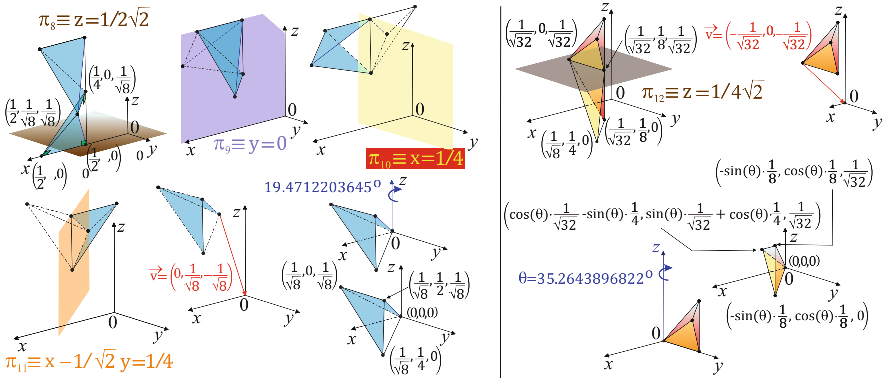


Fig. 6 The new position for the quasi right-type tetrahedron is achieved by four mirror relfection through planes $\pi_8 \equiv z = 1/2\sqrt{2}$, $\pi_9 \equiv y = 0$, $\pi_{10} \equiv x = 1/4$ and $\pi_{11} \equiv x - 1/\sqrt{2}y = 1/4$. Then a vector translation by $\boldsymbol{v}$ and a clockwise rotation around axis z is carried out. For the new tetrahedron, a mirror reflection by plane $\pi_{12} \equiv z = 1/4\sqrt{2}$ followed by a vector translation and a clockwise rotation around axis z is performed

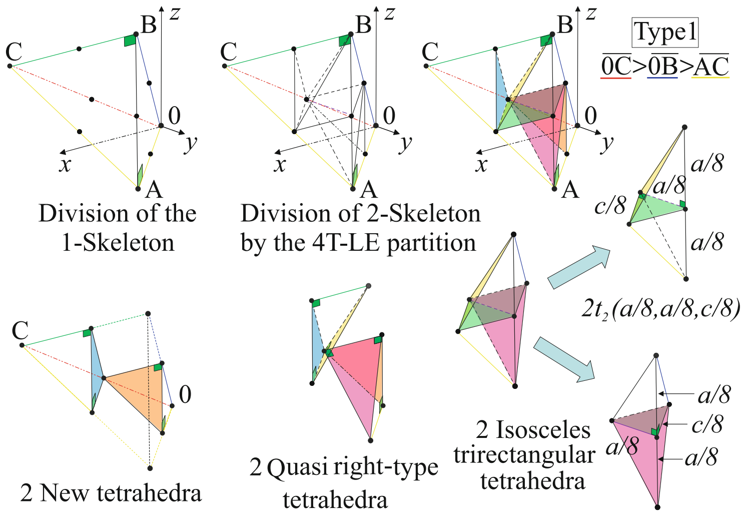
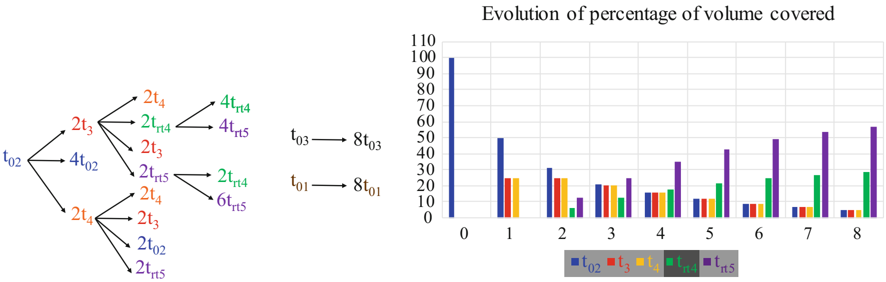

B
z
C
0
y
x
A
Division of the 1-Skeleton
Division of 2-Skeleton by the 4T-LE partition
Type1
0C>0B>AC
a/8
a/8
c/8
a/8
$2t_2(a/8,a/8,c/8)$
2 New tetrahedra
2 Quasi right-type tetrahedra
2 Isosceles trirectangular tetrahedra
a/8
c/8
a/8
a/8

Fig. 7 The 8T-LE partition of the new tetrahedron trough the skeleton. At this stage, all the tetrahedra generated have already appeared and also have been studied. It is clear that the new sub-subtetrahedra with one of its vertices 0 and C, respectively, are congruent to the original one



(a)

(b)

Fig. 8 Classes of similarity (left) and percentage of volume covered (right), generated by the 8T-LE partition. (**a**) Similarity classes generated. (**b**) Evolution of number of tetrahedra of each class

The number of tetrahedra belonging to each class is given by the following recurrence relations for $n \geq 1$, with initial conditions $t_{01}^{(0)} = 12$, $t_{02}^{(0)} = 24$ and $t_{03}^{(0)} = 48$. Obviously, $t_{01}^{(n)} = t_{03}^{(n)} = 8^n$.

$$t_{01} = t_{03} \left\{ t_{01}^{(n)} = t_{03}^{(n)} = 8^n \right. \qquad t_{02} \begin{cases} t_{02}^{(n)} = \dfrac{2^n(1+3^n)}{2} & t_{rt4}^{(n)} = \dfrac{2^{3n} + 2^{n-1}(1-3^{n+1})}{3} \\ t_3^{(n)} = \dfrac{2^n(3^n-1)}{2} & t_{rt5}^{(n)} = \dfrac{2^{3n+1} + 2^n(1-3^{n+1})}{3} \\ t_4^{(n)} = \dfrac{2^n(3^n-1)}{2} & \end{cases}$$

4 Main Results

1. The number of similarity classes is finite and so the non-degeneracy is proved.
2. From the previous result follows that the minimum angle condition is satisfied.
3. Asymptotically, for the Face Centered case, most of the tetrahedra generated are isosceles right-type.
4. For the Face Divided and Edge Divided cases, the domain is partitioned into a congruent simplices by the 8T-LE partition. This is of great important in saving CPU time and achieving super convergence phenomena.

Acknowledgments This work has been partially supported by Project Puente Cabildo 2018-01 of the Cabildo de Gran Canaria.

References

1. G. Albertelli, R. A. Crawfis, Efficient subdivision of finite-element datasets into consistent tetrahedra. In R. Yagel and H. Hagen, editors, Proc. IEEE Visualization, 213–220, Phonenix, AZ, November 1997
2. J. Brandts, S. Korotov, On nonobtuse simplicial partitions. SIAM Review. **51** (2), 317–335 (2009)
3. H. Carr, T. Möller, J. Snoeyink, Simplicial subdivision and sampling artifacts. Proc. IEEE Visualization, 99–106, 2001
4. J. Dompierre, P. Labbé, M.-G. Vallet, R. Camarero, How to subdivide pyramids, prysms and hexahedra into tetrahedra. In 8th International Meshing Roundtable, Lake Tahoe, CA, October 1999. Sandia National Laboratories
5. A. Plaza, The eight-tetrahedra longest-edge partition and Kuhn triangulations. Comp. and Math. Appli. **54**, 427–433 (2007). https://doi.org/10.1016/j.camwa.2007.01.023
6. A. Plaza, G. F. Carey, Refinement of simplicial grids based on the skeleton. App. Numer. Math. **32** (2), 195–218 (2000). https://doi.org/10.1016/S0168-9274(99)00022-7
7. M. A. Padrón, A. Plaza, The 8T-LE partition applied to the obtuse triangulation of the 3D-cube. Submitted to Mathematics and Computer in Simulation
8. A. Plaza, M. A. Padrón, J. P. Suárez, Non-degeneracy study of the 8-tetrahedra longest-edge partition. App. Numer. Math. **55** (4), 458–472 (2005). https://doi.org/10.1016/j.apnum.2004.12.003
9. A. Plaza, M. A. Padrón, J. P. Suárez, S. Falcón, The 8-tetrahedra longest-edge partition of right-type tetrahedra. Finit. Elemen. in Analy. and Desi. **41** (3), 253–265 (2004). https://doi.org/10.1016/j.finel.2004.04.005

10. A. Plaza, M. C. Rivara, Average adjacencies for tetrahedral skeleton-regular partitions. J. Comp. Appl. Math. **177** (1), 141–158 (2005). https://doi.org/10.1016/j.cam.2004.09.013
11. M. A. Padrón, J. P. Suárez, A. Plaza, A comparative study between some bisection based partitions in 3D. App. Numer. Math.**55** (4), 357–367 (2005). https://doi.org/10.1016/j.apnum.2005.04.035
12. M. C. Rivara, Algorithms for refining triangular grids suitable for adaptive and multigrid techniques. Int. J. Numer. Meth. in Eng. **20** (4), 745–756 (1987). https://doi.org/10.1002/nme.1620200412

Point Forces and Their Alternatives in Cell-Based Models for Skin Contraction

Qiyao Peng and Fred Vermolen

Abstract We consider a cell-based approach in which the balance of momentum is used to predict the impact of cellular forces on the surrounding tissue. To this extent, the elasticity equation and Dirac Delta distributions are combined. In order to avoid the singularity caused by Dirac Delta distribution, alternative approaches are developed and a Gaussian distribution is used as a smoothed approach. Based on the application that the pulling force is pointing inward the cell, the smoothed particle approach is probed as well. In one dimension, it turns out that the aforementioned three approaches are consistent. For two dimensions, we report a computational consistence between the direct and smoothed particle approach.

1 Introduction

Wound healing is a spontaneous process of the skin to cure itself after an injury. For severe traumas, due to a significant loss of soft tissue, dermal wounds may lead to various pathological problems like contractures, which are known as excessive and morbid contractions. Usually, contractures concur with disfunctioning and disabilities of the patients. The contractions of the wound appear from the third phase of wound healing, which usually starts from the second day and will continue for 2–4 weeks after wounding [1]. Wound contractions take place due to (myo)fibroblasts interacting with the environment, namely the extracellular matrix(ECM) and the formation of (permanent) stresses and strain by collagen distributions in and around the wound area. In other word, the contractions are developed by the (myo)fibroblasts exerting pulling forces on the skin. In the end, usually, the contractions will result in 5–10% reduction from the original volume of the wound [1].

Q. Peng (✉) · F. Vermolen
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, XE, The Netherlands
e-mail: Q.Peng-1@tudelft.nl; F.J.Vermolen@tudelft.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_75

According to Koppenol [2], the forces released by the (myo)fibroblasts can be categorized as temporary forces and permanent forces. Only temporary forces will be discussed in this paper, of which the formalization is described in Q Peng [4]. In the model, the elasticity equation and Dirac Delta distributions are incorporated. However, Dirac Delta distributions cause a singular solution, that is, for dimensionality exceeding one the solution is not in the same Hilbert space as the basis functions for many naive finite-element strategies. In order to circumvent this complication, the smoothed forces approach is developed, in which we use Gaussian distributions to replace Dirac Delta distributions. Especially in our healing model, the forces point towards the centre of the cell. Therefore, we use the gradient of Gaussian distribution as an alternative.

The boundary value problems for all three methods are displayed in Sect. 2 for both one and two dimensions. Section 3 shows the numerical results corresponding to the approaches investigated before. In Sect. 4, conclusions are delivered.

2 Mathematical Models

To describe the contraction of the tissue we use the equation for conservation of momentum over the computational domain Ω:

$$-\nabla \cdot \boldsymbol{\sigma} = \boldsymbol{f}. \tag{1}$$

In the above equation, inertia has been neglected. We consider a linear, homogeneous, isotropic material; hence, Hooke's Law is used here to define $\boldsymbol{\sigma}$:

$$\boldsymbol{\sigma} = \frac{E}{1+\nu}\left\{\boldsymbol{\epsilon} + tr(\boldsymbol{\epsilon})\left[\frac{\nu}{1-2\nu}\right]\boldsymbol{I}\right\}, \tag{2}$$

where E is the stiffness of the computational domain, ν is Poisson's ratio and $\boldsymbol{\epsilon}$ is the infinitesimal strain tensor:

$$\boldsymbol{\epsilon} = \frac{1}{2}\left[\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right]. \tag{3}$$

The forces exerted by a cell are modelled by Dirac Delta distributions on the midpoints of the segments of the boundary of the cell [4]:

$$\boldsymbol{f}_t = \sum_{j=1}^{N_S^i} P(\boldsymbol{x},t)\boldsymbol{n}(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{x}_j^i(t))\Delta\Gamma_N^{i,j} \tag{4}$$

$$\to \int_{\partial\Omega_N^i} P(\boldsymbol{x},t)\boldsymbol{n}(\boldsymbol{x})\delta(\boldsymbol{x}-\boldsymbol{x}_s^i(t))d\Gamma_N^i, \quad \text{as } N_S^i \to \infty, \tag{5}$$

where N_S^i is the number of line segments of cell i, $P(\boldsymbol{x}, t)$ is the magnitude of the pulling force exerted at point $\boldsymbol{x}$ and time t per length, $\boldsymbol{n}(\boldsymbol{x})$ is the unit inward pointing normal vector (towards the cell centre) at position $\boldsymbol{x}$, $\boldsymbol{x}_j^i(t)$ is the midpoint on line segment j of cell i at time t and $\Delta\Gamma_N^{i,j}$ is the length of line segment j. In Eq. (5), $\boldsymbol{x}_s^i(t)$ represents the mid point of a segment on the cell boundary of cell i at time t. Possible boundary conditions could be Dirichlet (fixed boundary) or a mixed boundary condition (spring force).

2.1 Elasticity Equation and Point Sources in One Dimension

Considering the force equilibrium in one dimension, the equations are expressed as

$$-\frac{d\sigma}{dx} = f, \quad \text{Equation of Equlibirum,} \tag{6}$$

$$\epsilon = \tfrac{du}{dx}, \quad \text{Strain-Displacement Relation,} \tag{7}$$

$$\sigma = E\epsilon, \quad \text{Constitutive Equation.} \tag{8}$$

To simplify the equation with $E = 1$ here, the equations above can be combined to Laplacian equation in one dimension:

$$-\frac{d^2u}{dx^2} = f. \tag{9}$$

According to Eq. (4), for one dimension, assume there is a biological cell with size h and centre position c in the computational domain $0 < c-h/2 < c < c+h/2 < L$. Combined with homogeneous Dirichlet boundary conditions, the boundary value problem of the direct approach is given by

$$(BVP_\delta)\begin{cases} -\dfrac{d^2u}{dx^2} = -\delta(x-(c+\dfrac{h}{2})) + \delta(x-(c-\dfrac{h}{2})), & x \in (0, L), \\ u(0) = u(L) = 0, \end{cases} \tag{10}$$

where $\delta(x-x')$ is Dirac Delta distribution. Note that in one dimension, the solution is piecewise linear and hence in $H^1(\Omega)$.

The Gaussian distribution is usually used as a replacement for Dirac Delta distributions to obtain a smoother expression. Here, we denote

$$\delta_\varepsilon(x-x') = \frac{1}{\sqrt{2\pi\varepsilon^2}} \exp\left\{-\frac{(x-x')^2}{2\varepsilon^2}\right\},$$

for the Gaussian distribution with mean x' and variance ε^2. Therefore, the boundary value problem of the smoothed approach is expressed as

$$(BVP_S)\begin{cases} -\dfrac{d^2u_\varepsilon}{dx^2} = -\delta_\varepsilon(x-(c+\dfrac{h}{2})) + \delta_\varepsilon(x-(c-\dfrac{h}{2})), \quad x \in (0,L), \\ u_\varepsilon(0) = u_\varepsilon(L) = 0, \end{cases} \tag{11}$$

In (BVP_S), since the right-hand side is smooth, we can rewrite it as

$$\begin{aligned} &\delta_\varepsilon(x-(c-\frac{h}{2})) - \delta_\varepsilon(x-(c+\frac{h}{2})) = h\frac{d\delta_\varepsilon}{dx}(x-c) \\ &+\frac{h^3}{48}(\delta_\epsilon'''(x-c+\eta_1) + \delta_\epsilon'''(x-c+\eta_2)), \quad \exists \eta_1,\eta_2 \in (-\frac{h}{2},\frac{h}{2}). \end{aligned} \tag{12}$$

In other words, the right hand side of (BVP_S) converges to right-hand side of the smoothed particle approach:

$$(BVP_{SP})\begin{cases} -\dfrac{d^2v_\varepsilon}{dx^2} = h\dfrac{d\delta_\varepsilon}{dx}(x-c), \quad x \in (0,L), \\ v_\varepsilon(0) = v_\varepsilon(L) = 0, \end{cases} \tag{13}$$

as $h \to 0$. This, in turn is combined with the boundary conditions to conclude that the difference between u_ϵ and v_ϵ satisfies

$$-\frac{d^2(u_\varepsilon - v_\varepsilon)}{dx^2} = \frac{h^3}{48}(\delta_\varepsilon'''(x-c+\eta_1) + \delta_\varepsilon'''(x-c+\eta_2)), \tag{14}$$

combined with the homogeneous boundary condition, and upon setting $\epsilon = \sqrt{h}$, and with Poincaré's Lemma, it follows that there exists $K > 0$ such that

$$\|u_\varepsilon - v_\varepsilon\|_2 \leqslant \frac{Lh^3}{24K}\|\delta_\varepsilon'''(x)\|,$$

and hence due to continuity, we have $u_\varepsilon \to v_\varepsilon$ as $h \to 0$; see Q Peng [4] for more details about the proof. In fact, we are aware that in electric dipole moments, especially in three dimensional case of potential forum, there are similar transformation occurring in potential expression of an electric dipole. A Taylor expansion is applied to bridge the potential expression of two points charge transferring to one point charge expressed with gradient; see Laud [3] for more details.

For the direct approach, the exact solution is the superposition of the Green's function in one dimension, which is known as

$$G(x,x') = \begin{cases} x'(1-\dfrac{x}{L}), & x \geqslant x', \\[2ex] x(1-\dfrac{x'}{L}), & x < x'. \end{cases} \tag{15}$$

Since the forces are inward pointing to the centre of the cell, the solution to (BVP_δ) is

$$u_\delta(x) = -G(x, c+\frac{h}{2}) + G(x, c-\frac{h}{2}). \tag{16}$$

The solutions to (BVP_S) and (BVP_{SP}), are, respectively, given by

$$\begin{aligned} u_{S_\varepsilon}(x) = \frac{x\varepsilon}{\sqrt{2}L}\left(\int_{-\frac{c-h/2}{\sqrt{2}\varepsilon}}^{\frac{L-(c-h/2)}{\sqrt{2}\varepsilon}} \operatorname{erf}(x')dx' - \int_{-\frac{c+h/2}{\sqrt{2}\varepsilon}}^{\frac{L-(c+h/2)}{\sqrt{2}\varepsilon}} \operatorname{erf}(x')dx'\right) \\ - \frac{\varepsilon}{\sqrt{2}}\left(\int_{-\frac{c-h/2}{\sqrt{2}\varepsilon}}^{\frac{x-(c-h/2)}{\sqrt{2}\varepsilon}} \operatorname{erf}(x')dx' - \int_{-\frac{c+h/2}{\sqrt{2}\varepsilon}}^{\frac{x-(c+h/2)}{\sqrt{2}\varepsilon}} \operatorname{erf}(x')dx'\right), \end{aligned} \tag{17}$$

and

$$u_{SP_\varepsilon}(x) = \frac{h}{2}\left\{(\frac{x}{L}-1)\operatorname{erf}(\frac{c}{\sqrt{2}\varepsilon}) + \frac{x}{L}\operatorname{erf}(\frac{L-c}{\sqrt{2}\varepsilon}) - \operatorname{erf}(\frac{x-c}{\sqrt{2}\varepsilon})\right\}, \tag{18}$$

where $\operatorname{erf}(x)$ is the error function defined as $\operatorname{erf}(x) = \frac{2}{\sqrt{\pi}}\int_0^x \exp(-t^2)dt$.

2.2 Elasticity Equation and Point Sources in Two Dimensions

For two dimensions, we start with analysing only one biological cell in the computational domain. According to the model described in Eq. (4), the forces released on the boundary of the cell are the superposition of point forces on the midpoint of each line segment. For example, if we use a square shape to approximate the biological cell, then the forces are depicted in Fig. 1. We only need to focus on the midpoints of the facets if we are working with the direct approach. In the meanwhile, if we work on the smoothed particle approach, we take the midpoint of the entire biological cell, since the gradient is used, which does not require an

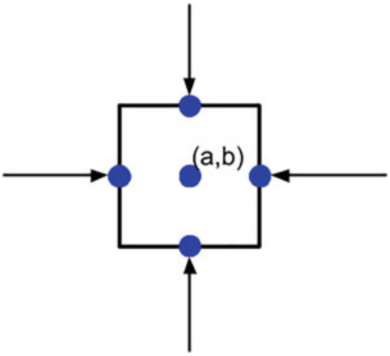


Fig. 1 We consider a square shaped biological cell, with the centre position at (a, b). The forces exerted on the boundary are indicated by arrows

explicit treatment of the corners. Therefore, in this circumstance, the forces can be rewritten as

$$\begin{aligned}
\boldsymbol{f}_t &= P\left\{-\begin{bmatrix}1\\0\end{bmatrix}\Delta y\delta(x-(a+\frac{\Delta x}{2}), y-b)+\begin{bmatrix}1\\0\end{bmatrix}\Delta y\delta(x-(a-\frac{\Delta x}{2}), y-b)\right.\\
&\quad\left.-\begin{bmatrix}0\\1\end{bmatrix}\Delta x\delta(x-a, y-(b+\frac{\Delta y}{2}))+\begin{bmatrix}0\\1\end{bmatrix}\Delta x\delta(x-a, y-(b-\frac{\Delta y}{2}))\right\}\\
&\approx P\left\{\begin{bmatrix}1\\0\end{bmatrix}\Delta y\left[-\delta_\varepsilon(x-(a+\frac{\Delta x}{2}), y-b)+\delta_\varepsilon(x-(a-\frac{\Delta x}{2}), y-b)\right]\right.\\
&\quad\left.+\begin{bmatrix}0\\1\end{bmatrix}\Delta x\left[-\delta_\varepsilon(x-a, y-(b+\frac{\Delta y}{2}))+\delta_\varepsilon(x-a, y-(b-\frac{\Delta y}{2}))\right]\right\}.
\end{aligned} \tag{19}$$

Thanks to the continuity of Gaussian distribution δ_ε, there exists $(\eta_x, \eta_y) \in (-\Delta x/2, \Delta x/2) \times (-\Delta y/2, \Delta y/2)$ such that, Eq. (19) yields into

$$\begin{aligned}
\boldsymbol{f}_t &\approx P\left\{\begin{bmatrix}1\\0\end{bmatrix}\Delta y\Delta x\frac{\partial\delta_\varepsilon}{\partial x}(x-a+\eta_x, y-b)+\begin{bmatrix}0\\1\end{bmatrix}\Delta y\Delta x\frac{\partial\delta_\varepsilon}{\partial y}(x-a, y-b+\eta_y)\right\}\\
&\to P\Delta x\Delta y\nabla\delta_\varepsilon(x-a, y-b), \quad \text{as } \Delta x, \Delta y \to 0.
\end{aligned} \tag{20}$$

The above procedure implies that as $\Delta x,\ \Delta y \to 0$, the right-hand side of the regularized Dirac Delta Distributions converges to $P\Delta x\Delta y\nabla\delta_\epsilon(x-a, y-b)$. In future research we aim at rigorously establishing that this implies that the difference between the solutions from both approaches tends to zero as $\Delta x,\ \Delta y \to 0$; see Q Peng [4] for more details.

3 Numerical Results

In this section, results in both one dimension and two dimensions are presented. Since the paper is meant to compare various modelling approaches, the results are generic in the sense that the presentation and the analysis of the modelling is done for a one-dimensional case with dimensionless parameters. Furthermore, the two-dimensional simulations only serve as an illustration of how the various approaches are related. Figure 2 shows the analytical solution of all three approaches from Eq. (16), (17), and (18). The red and blue curves, which correspond to the direct and the smoothed Delta approach, mostly overlap regardless the choices of ε. This indicates that the solutions to (BVP_S) and (BVP_{SP}) are consistent. As ε decreases, the solutions to the smoothed approach and the smoothed particle approach converge to the solution to the direct approach. In other words, Fig. 2 confirms the consistency between all three approaches, as long as ε is efficiently small.

For two dimensions, only the results that apply Eq. (4) will be compared to the smoothed particle approach. We consider only one big circular biological cell in the computational domain, and the boundary of the biological cell is split into finite line segments. Based on the special case of square (see Eq. (20)) and since the magnitude relation between the direct approach and the smoothed particle approaches is still

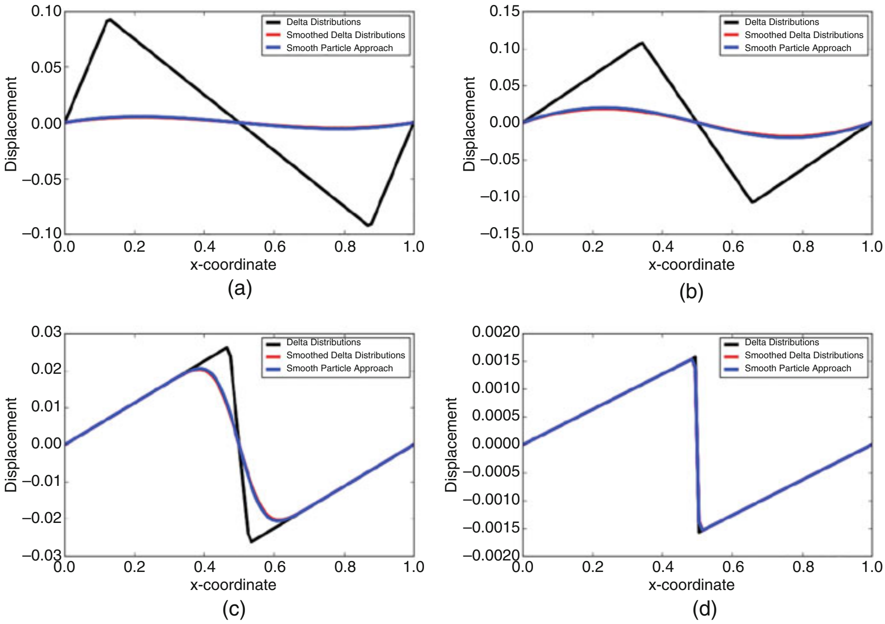


Fig. 2 For one dimension, different colours of curves show the solution to (BVP_δ), (BVP_S) and (BVP_{SP}) respectively. Black curve shows the solution to the direct approach, red curve is the smoothed approach and blue curve is the smoothed particle approach. As $h = \varepsilon$ decreases, all the results converge. **(a)** $\varepsilon = \varepsilon_0$. **(b)** $\varepsilon = (\varepsilon_0)^4$. **(c)** $\varepsilon = (\varepsilon_0)^{10}$. **(d)** $\varepsilon = (\varepsilon_0)^{20}$

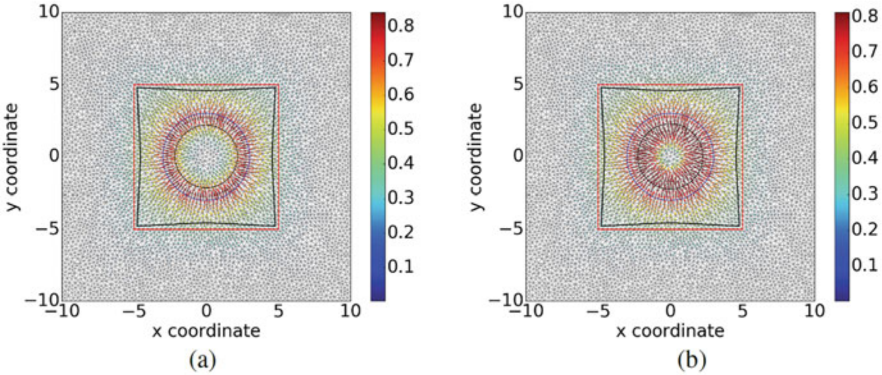


Fig. 3 Black curves show the deformed region of vicinity and the cell, and blue curve represents the cell. (**a**) Direct approach. (**b**) Smoothed particle approach

Table 1 The percentage of area change of cell and vicinity region, and time cost of the direct approach and the smoothed particle approach

	Direct approach	Smoothed particle approach
Cell area reduction ratio(%)	47.81624	43.38118
Vicinity area reduction ratio(%)	12.85195	12.88194
Time cost(s)	1.70716	1.83455

unclear, we will use the area of the biological cell as the magnitude ratio, although we realise that the transition between the two approaches is only applicable in the limit that the biological cell area tends to zero. Subsequently, we investigate the new cell area after deformation, as well as a region near the cell. Further, the computational time is compared, since in our wound healing model, there are a large number of biological cells in the computational domain. In Fig. 3, the bandwidth around the cell in the smoothed particle approach is wider than the direct approach, which is mainly because of the continuity of the smoothed particle approach. The numerical results are displayed in Table 1 to have a better insight into the performance of the smoothed forces approach. It is notable that the computation times, cell area reduction ratio and the vicinity area reduction are all more or less the same. Therefore, taking the advantage of a smooth force into consideration, the smoothed particle approach has the potential to be incorporated into the model containing multiple biological cells.

4 Conclusion

In this paper, we developed two alternative methods using Gaussian distributions to replace Dirac Delta distributions in the point forces. The first method is the smoothed approach, in which the Dirac Delta distributions at the midpoint of boundary segments of the cell are replaced by Gaussian distributions directly. The second alternative method is the smoothed particle approach, which takes into account the gradient of the Gaussian distribution at the centre of the cell, and it is based on the point forces exerted on the boundary of cells in wound healing.

In one dimension, we proved that the smoothed approach and the smoothed particle approach converge to the direct approach, and the numerical results verified consistency. In two dimensions, we are still in the process of working out the exact ratio between the direct approach and the smoothed particle approach. However, inspired by the square-shaped cell, we use the cell area to investigate the discrepancy, which turns out to be negligible. Furthermore, the smoothed particle approach costs nearly the same CPU as the direct approach, which offers the possibility to adapt it into the general healing model.

Acknowledgments The authors appreciate China Scholarship Council (CSC) for the financial support on this project.

References

1. S. Enoch and D. J. Leaper. Basic science of wound healing. *Surgery (Oxford)*, 26(2):31–37, 2008.
2. D. Koppenol. Biomedical implications from mathematical models for the simulation of dermal wound healing. 2017.
3. B. Laud. *Electromagnetics*. New Age International, 1987.
4. Q. Peng and F. Vermolen, *Point Forces and Their Alternatives in Cell-Based Models for Skin Contraction. Reports of the Delft Institute of Applied Mathematics*, vol. 19-03, Delft University, the Netherlands, 2019. ISSN: 1389-6520

Empirically Driven Orthonormal Bases for Functional Data Analysis

Hiba Nassar and Krzysztof Podgórski

Abstract In implementations of the functional data methods, the effect of the initial choice of an orthonormal basis has not been properly studied. Typically, several standard bases such as Fourier, wavelets, splines, etc. are considered to transform observed functional data and a choice is made without any formal criteria indicating which of the bases is preferable for the initial transformation of the data. In an attempt to address this issue, we propose a strictly data-driven method of orthonormal basis selection. The method uses B-splines and utilizes recently introduced efficient orthornormal bases called the splinets. The algorithm learns from the data in the machine learning style to efficiently place knots. The optimality criterion is based on the average (per functional data point) mean square error and is utilized both in the learning algorithms and in comparison studies. The latter indicate efficiency that could be used to analyze responses to a complex physical system.

1 Introduction

Functional data analysis (FDA) is the field in statistics that studies the analysis and theory of data that are in the form of functions, images, and shapes, etc, see [3]. The FD may come as a dynamical response from a physical system subject to stochastic

H. Nassar
Cognitive Systems, Department of Applied Mathematics and Computer Science,
Technical University of Denmark, Lyngby, Denmark
e-mail: hibna@dtu.dk

Department of Statistics, Lund University, Lund, Sweden

K. Podgórski (✉)
Department of Statistics, Lund University, Lund, Sweden
e-mail: krzysztof.podgorski@stat.lu.se

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_76

excitation that can be written in a generic form as

$$H(y^{(n)}, \ldots, y'', y', y, x; \theta) = F(t), \tag{1}$$

where $F(t)$ is a realization of stochastic forcing of the system whose response is given by H that involves some physical parameters given in θ. Often the response from such a system is stochastic not only because of random excitation F but also due to randomness in the parameter θ of the system. As a result, the responses $y_i(t)$ from such a system can be conceptually treated as FD depending both on θ_i and $F_i(t)$. The goal is to obtain an efficient treatment of these functional observations in order to infer about θ as well as about the functional structure of y.

FD are not observed as continuous objects, but rather as discrete data. High-frequency sampling and mathematical efficiency allow these data to be seen as samples of curves, surfaces or anything else varying over a continuum. The fundamental step in FDA is to convert this discrete recorded data to a functional form, which gives each function the possibility to be evaluated for all values of t. To utilize the topology of such data for the dimension reduction one performs the data conversion. One of the methods used is to represent a functional object as a linear combination of coefficients and a number of suitable basis functions. For the purpose, one of the standard bases such as trigonometric, wavelet, or polynomial is typically chosen.

The efficiency is accomplished by using smoothing through regression or roughness penalty for estimating the coefficients of the basis expansions. However, all such analyses are preceded by the initial choice of a functional basis used to analyze data, which is hardly objective and often driven by mathematical convenience. On the other hand, it is both theoretically and practically observed that the choice of the basis affects efficiency in retrieving the functional structure of a studied model. This motivated us to investigate this problem more thoroughly.

In the spirit of the main data analysis paradigm, for a given FD set it may be computationally effective to work with a data-driven basis. Consider, for example, the classical smoothing problem, where for a given data we want to fit a smooth function. Using the B-splines together with a regularization method, for example the Lasso method, one may selectively choose a subspace of the spline space by shrinking parameters to zero, see [4]. Such a basis can be chosen for each FD sample but a choice valid for all samples is not obvious.

We acknowledge the value of splines in FDA but we proceed differently by utilizing freedom in knots placement to algorithmically search for efficient knots patterns and utilize them in the orthonormal basis construction. In the process, we implement machine learning algorithms for the choice of basis reducing the mean square error (MSE) uniformly for all samples and study its efficiency against other choices of the basis. The optimality criterion is utilized, both in the learning algorithms and in comparison studies. This criterion allows for comparison performances of different bases in a given problem. After efficiently learning from the data about knot placements, we utilize the new construction of the orthonormal spline bases, termed splinets and introduced in [9]. There, it is demonstrated that the splinets are characterized by optimality properties that bring further benefits to our approach.

2 FD and Their Representations

Discrete observations of a single function $x(t)$, $t \in [0, 1]$, at times t_j result in

$$y_j = x(t_j) + \epsilon_j, \qquad j = 1, \ldots, p$$

where ϵ_j is an error term in the data. To account on topological features assumed to be present in $x(t)$, the function is assumed to be smooth. One of the most common ways of representing it efficiently is by using the basis function expansion, i.e. by decomposing linearly the function $x(t)$ in terms of a chosen basis system $\phi_k(t)$ consisting of $K \leq \infty$ basis functions

$$x(t) = \sum_{k=1}^{K} c_k \phi_k(t). \tag{2}$$

The most commonly used basis functions $\phi_k(t)$ are Fourier, polynomial, splines and wavelets. It is typically assumed that the observations $x_k(t)$ are random elements of $L^2[0, 1]$. In this Hilbert space, we use inner product $\langle \cdot, \cdot \rangle$ for an integral of the product of its two functional elements and which generates the norm $\| \cdot \|$. We call L^2-valued data functional observations. We use upper case and lower case letters in the context of FD in a similar manner as in the classical statistical convention, i.e. X is yet not observable random element, while $x = x(\cdot)$ stands for its particular observed functional realization, i.e. a functional outcome of random experiment carried out according to the probability model for X.

All random functions are assumed to be square integrable, i.e. $E\|X\|^2 < \infty$. In this context, one have to point to a classical result, the Karhunen–Loève expansion, see [8], which shows that the basis associated with this expansion has the optimality in the average mean square error sense, for more details see [6]:

$$X(t) = \sum_{k=0}^{\infty} \sqrt{\lambda_k} Z_k \, e_k(t),$$

where λ_k is a square summable sequence of non-negative numbers, e_k, $k \in \mathbb{N}_0$ is an orthonormal (non-random) basis in $L^2[0, 1]$ and Z_k is a sequence of zero-mean variance-one uncorrelated random variables. In the Gaussian case, Z_k are independent standard normal variables. The desired optimality of the basis (e_k) is mostly of theoretical value since except for Brownian motion ($\lambda_k = (\pi k - \pi/2)^{-2}$,

$e_k(t) = \sqrt{2}\sin((\pi k - \pi/2)t))$ and Brownian bridge ($\lambda_k = (\pi k)^{-2}$, $e_k(t) = \sqrt{2}\sin(\pi kt)$), the actual form of the optimal basis is not available. It is the central problem of the FDA to find the approximation of e_k for any specific problem. To do this one has to decompose the original data using some basis of convenience.

The most popular decomposition of a function is by the Fourier basis

$$\{\sqrt{2}\sin(2\pi nt);\ n \in \mathbb{N}\} \cup \{\sqrt{2}\cos(2\pi nt);\ n \in \mathbb{N}\} \cup \{\mathbf{1}\}.$$

Fourier functions form an orthogonal basis and have good computational properties. A Fourier decomposition is especially useful for extremely stable functions where there are no strong local features and the same curvature order everywhere. However, they are improper for data where discontinuities in the function itself or in low order derivatives are known or suspected [11, page 48]. In Fig. 1, the graphs in the second column illustrate a 40-dimensional Fourier approximation of functional signals.

Spline functions are a natural choice for approximating non-periodic FD. Splines combine the fast computation of polynomials with substantially greater flexibility. We explain some essential background of the B-splines, for more details we refer the reader to standard texts such as [2, 12].

A spline is a smooth function consisting of polynomial pieces that have the same degree, connected smoothly at points $\xi_0 < \xi_1 < \cdots < \xi_{n+1}$, referred to as knots. The splines are sensitive to the choice of the knots' position, which is behind our main idea of the basis selection since the choice of the knots can be data-driven. Once the knots are set, The spline basis (B-splines) can be effectively evaluated in a recursive way the Cox-de Boor formula [2]. The B-splines have interesting properties that characterize them. Namely, all B-splines are positive, differentiable up to a certain level (the spline order) and have minimal compact intervals for their supports. But except for the case of order zero, the B-splines are not orthogonal. Different orthogonalization methods appeared in the literature but we are using our structured orthogonalization that creates basis systems for which we coined the term *splinet*. The splinet is prioritized over other orthonormal spline systems as it preserves locality and computational efficiencies of the original splines, see Fig. 2 for two splinets used in Section 5. For a more detailed explanation, we refer the reader to [9]. In Fig. 1, the graphs in the last column illustrate 40-dimensional, third-order B-spline projection of functional signals.

3 Data Driven Choice of the Knots

The degree of a polynomial and the placement of knots defines the spline basis. We propose machine learning style techniques for the placement of the knots. The chosen knots are used to build splines basis functions $\phi_k(t)$ that are used in basis function expansion to convert the data from discrete recorded data into a functional

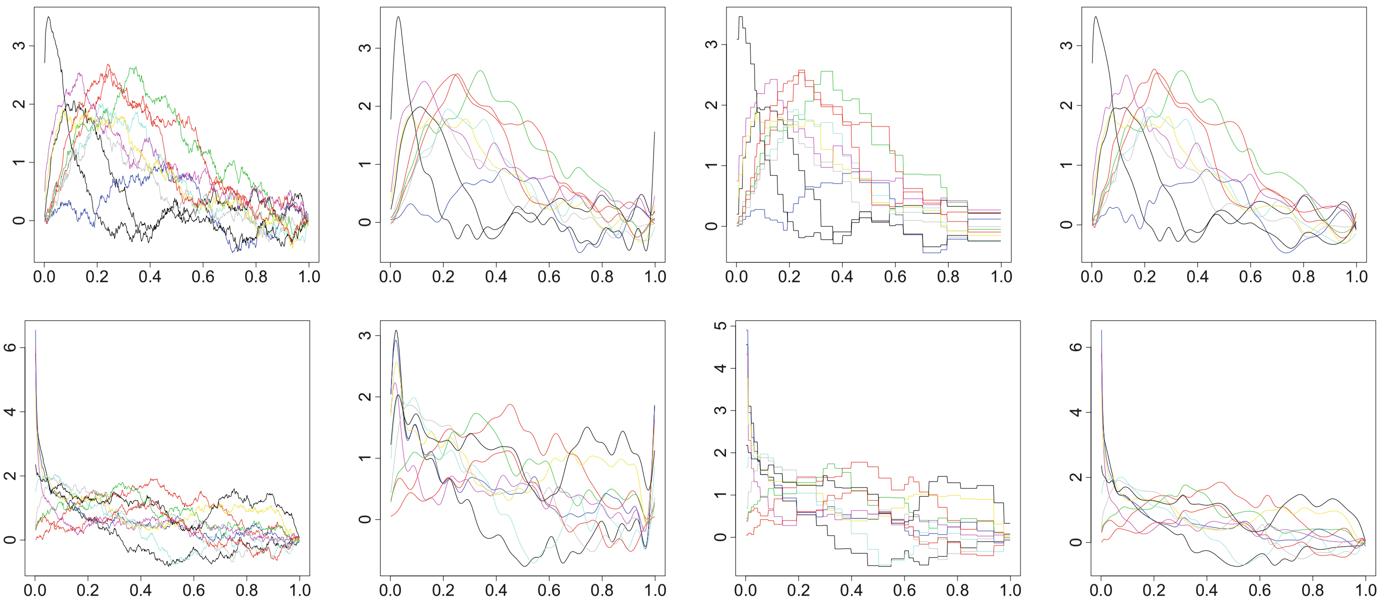


Fig. 1 Left: Ten samples of FD obtained from two different random functional of Brownian bridge. Middle-Left: Fourier approximations of the FD based on 30 Fourier basis functions. Middle-Right: Piecewise constant orthonormal basis approximations of the FD based on 30 basis functions. Right: Smooth splinet basis approximations of the FD based on 30 basis functions

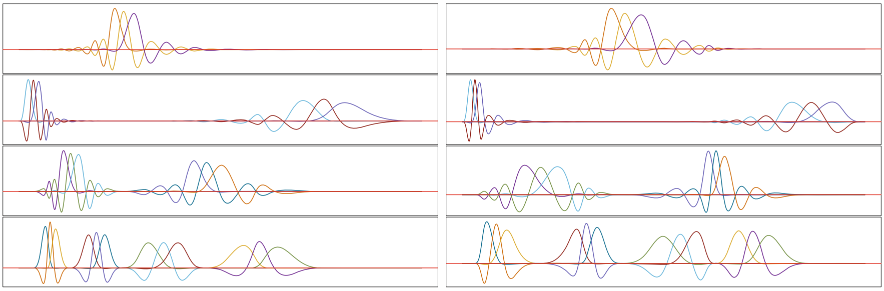


Fig. 2 Splinets for the data driven knots placement for two examples of Sect. 5

one. The method of adding knots is based on the mean square error effectiveness of approximating the FD. The method is iterative and resembles the regression tree building by which it was inspired, see [5, Chapter 9].

For any FD set $\mathcal{X} = \{x_i \in L^2, i = 1, \dots n\}$, the set of best least square constant predictors is a set of functions

$$x_i^{(0)} = \langle x_i, \mathbf{1} \rangle \mathbf{1} = \int x_i \cdot \mathbf{1}.$$

The constant functions over the entire domain [0, 1] can be viewed as 0-order splines with no internal knot points, and its one dimensional basis is given by the constant function **1**. We set the initial set of knots to an empty set, i.e. $\mathcal{K}^{(0)} = \emptyset$, the initial basis $\mathcal{B}^{(0)} = \{\mathbf{1}\}$, and the projection to the space spanned by $\mathcal{B}^{(0)}$ is given by $\mathbf{P}^{(0)} x = \langle x, \mathbf{1} \rangle \mathbf{1}$. The average mean square error (AMSE) per function of the approximations of x_i's by the optimal constant functions is given by

$$AMSE(\mathcal{Y}, \mathcal{B}^{(0)}) = \frac{1}{n} \sum_{i=1}^{n} \| x_i - \mathbf{P}^{(0)} x_i \|^2 = \frac{1}{n} \sum_{i=1}^{n} \| x_i - \langle x_i, \mathbf{1} \rangle \mathbf{1} \|^2 .$$

The method at the first step, $s = 1$, finds a knot $\xi \in [0, 1]$ such that the optimal approximation of x by a linear combination of the 0-order splines with the set of knots $\mathcal{K}^{(1)} = \mathcal{K}^{(0)} \cup \{\xi\}$ yields the smallest AMSE between the FD x_i. In other words, denote by $\mathcal{B}^{(1)}(\xi)$ the orthonormal basis of piecewise constant functions over the intervals given by the knots in $\mathcal{K}^{(1)}(\xi)$. The new knot ξ_{new} is chosen as

$$\xi_{new} = \underset{\xi \in (0,1]}{\text{argmin}}\, AMSE(\mathcal{Y}, \mathcal{B}^{(s)}(\xi)). \tag{3}$$

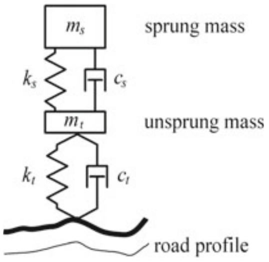
Then the new, enlarged by one function, basis $\mathcal{B}^{(1)} = \mathcal{B}^{(1)}(\xi_{new})$ is uniquely defined by the new set of knots $\mathcal{K}^{(1)} = \mathcal{K}^{(1)}(\xi_{new})$. In the recurrent process, at the step s, we start with a sequence of knots $\mathcal{K}^{(s-1)}$ and search for a new knot ξ_{new} using (3) with $\mathcal{K}^{(s)}(\xi) = \mathcal{K}^{(s-1)} \cup \{\xi\}$ and the corresponding orthonormal basis of piecewise constant functions $\mathcal{B}^{(s)}(\xi)$.

The algorithm benefits from the locality and orthogonality piecewise constant bases so that each new knot requires a removal only one base function (the constant over interval that includes the new knot) and replaces it by two new functions that remain orthonormal to all the other basis functions from the previous step. The outcome of the zero-order spline decomposition of the FD is shown in Fig. 1, the third column.

4 Application: Efficient Analysis of the Quarter Vehicle Model

The model of a damped harmonic oscillator can be utilized in studies of the durability of vehicle components in the vehicle response to the road profile, see [7, 10] for further details on the model. The road profile roughness is often quantified using the response of a quarter-vehicle model traveling at a constant velocity through road profiles, see Fig. 3. Such a simplification of a physical vehicle cannot be expected to predict loads exactly, but it will highlight the most important road characteristics as far as durability is concerned.

It is desirable to have a model of load environment that is vehicle independent and which may consist of many components, like driving habits, encountered road roughness, hilliness, curve radius, cargo loading, and others. The force acting on the sprung mass m_s (total mass of the vehicle) that is randomly distributed around some specific mean value is chosen as the response $y(t)$ from the tire which then is used to compute suitable indexes to classify the severity of road roughness.



Parameter	Mean	Unit
m_s	3400	kg
k_s	270 000	N/m
c_s	6000	Ns/m
m_t	350	kg
k_t	950000	N/m
c_t	300	Ns/m

Fig. 3 Quarter vehicle model and examples of its parameters

In a linear simplification of the problem, the entire system has the following components. The road elevation $R(t)$ that, under constant speed v of the vehicle, linearly drives two damped harmonic oscillators, one representing the tire and the other the wheel suspension system

$$m_t\ddot{u} + c_t\dot{u} + k_t u = F_t, \qquad m_s\ddot{y} + c_s\dot{y} + k_s y = F_s.$$

The parameters in the model can be set to mimic heavy vehicle dynamics as, for example, developed in SCANIA, see Fig. 3. They have the following physical interpretation: properties of the tire are described by k_t, c_t, which relate to stiffness and damping of the tire, while properties of the suspension are given by corresponding k_s, c_s. Modeling of true loads acting on components is difficult since tires filter nonlinearly the road profile and the filter parameters depend on very uncertain factors, e.g. tire's pressure, wear, etc. One way to account for the later and simplify the former is to assume that some of the parameters are random and represent properties of the tire in a concrete vehicle on a given trip.

In further simplification, the condition of the trip can be modeled by a Brownian bridge B filtered by a certain kernel r. Here the Brownian bridge model reflects small roughness of the road at the beginning and at the end of a trip and an increase of it when the vehicle enters tougher terrain in the middle of the trip. The smoothing kernel r represents road specific properties so that $R(t) = r * dB(t)$ is the road surface elevation at location t. In the literature, many models for the power spectral density S_R of road profiles have been proposed, see [1] for a review. Here, the kernel r relates to S_R through the Bochner theorem $r * \tilde{r}(t) = 2\int \cos(\omega t) S_R(\omega)\, d\omega$.

Often one chooses the force acting on the sprung mass as the response $y(t)$ which then is used to compute suitable indexes to classify the severity of road roughness. In the above simplification, this response is linearly driven by the road profile, as it is also the displacement $x(t)$ of the center of the wheel from the road. Their transfer functions, i.e. the Fourier responses to Dirac's delta, are explicit functions of the transfer functions of the two harmonic oscillators

$$H_t(\omega) = -m_t\,\omega^2 + i\,\omega\,c_t + k_t, \qquad H_s(\omega) = -m_s\,\omega^2 + i\,\omega\,c_s + k_s.$$

To recap, the model is completely defined by the vehicle related parameters: the speed of the vehicle v, the mass of the vehicle m_s, the undamped angular frequencies ω_s and ω_t, and the road related parameters, that describe S_R. All these parameters can be collectively described as θ. Some of these parameters can be considered as random and each observed journey of a vehicle produces a response $y_i(t)$, $t \in [0, 1]$, with stochastic response driven by samples of Brownian bridge B_i and random sample θ_i of the parameters, $i = 1, \dots, n$, where n is the number of trips.

5 Simulation Studies

We illustrate, through simulations, how using a data-driven orthonormal basis can improve efficiency in representing FD. The setting of Monte Carlo experiment mimics, in a simplified manner, physical systems similar to the quarter vehicle model. The data are not truly sampled from such a model since it would require more extensive study not fitting the format of this note. Instead, we use FD obtained by sampling parameters of the model to which we also inject samples of Brownian bridge

$$y_i(t) = F(t; B_i(\cdot), \theta_i), \; i = 1, \ldots, n.$$

Ten samples of FD from two such models are presented in Fig. 1Left.

We performed orthogonal projections to the three ON bases: Fourier, piecewise constant, and the splinets. In Fig. 1, we show approximations of the functions seen in the left column that uses $N = 30$ basis functions. A Monte Carlo study of the dependence of the average mean square error on the number basis elements is shown in Fig. 4. Monte Carlo samples of size 10 were drawn from the two models used in Fig. 1. For each of these sample approximations with the number of basis elements used increasing from 4 to 50 were evaluated and their average mean square error (AMSE) over all 10 elements of the data evaluated. This procedure has been repeated independently 20 times resulting in 20 AMSE's for each size of the Fourier base used. Boxplots of these data for each model and each the Fourier base size are presented in Fig. 4.

6 Conclusions

The proposed method of the data-driven orthonormal basis decomposition has been tested in through numerical simulations. The Monte Carlo simulations show clear advantages over the Fourier based method, in particular, when smoothed splines are used. The accuracy is not only exhibited in smaller errors but also in the reduced variability of the error. The improvement is greater, as expected, for the data that shows some local detail. The obtained results suggest that the method may have a great potential to improve the functional analysis of the data coming from the physical systems with random excitation and involving random parameters. This has to be confirmed by further model specific studies.

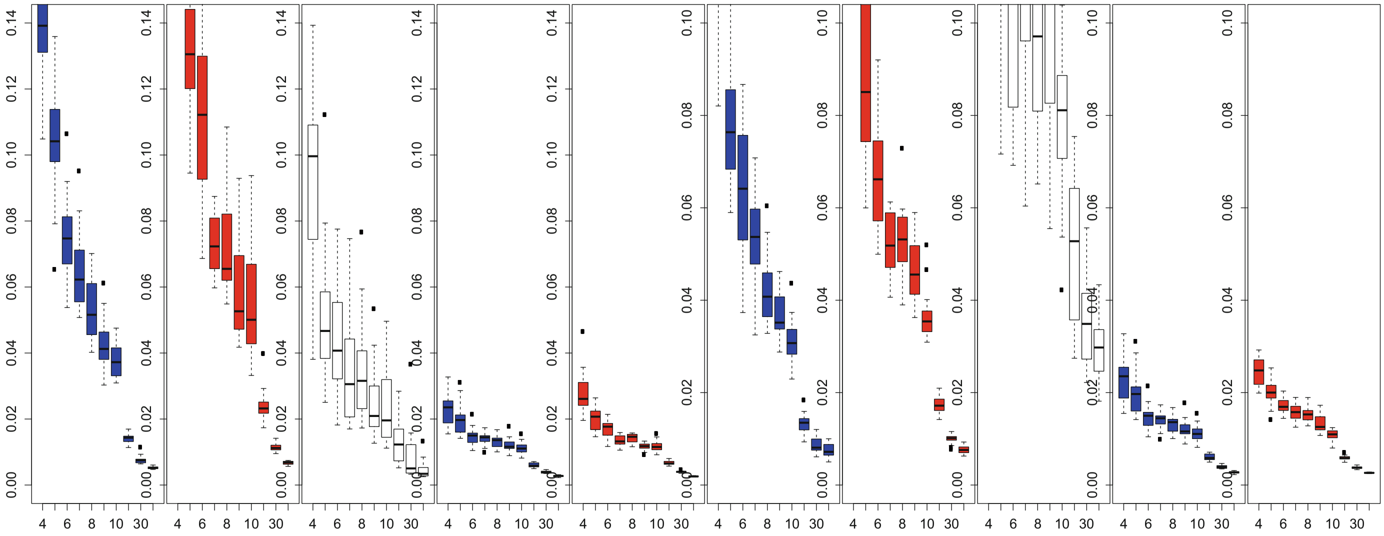


Fig. 4 The boxplots of AMSE's obtained from 20 Monte Carlo simulations for the two models used in Fig. 1 as a function of the base size, which ranges from 4 to 40. For each model, 10 FD were simulated and the orthonormal basis decomposition was run through these FD with an increasing number of basis elements. In each of the two cases grouped in five plots each, in the first and the fourth plot (blue) a new basis is selected anew for each MC sample, while in the second and the fifth (red) a basis is selected only for the original sample and then used for every new MC sample. The first two pictures in each group, correspond to the piecewise constant data-driven basis, the last two to the spline basis, the plots in the middle (black and white) corresponds to the Fourier basis applied to the MC data

References

1. Andrén, P.: Power spectral density approximations of longitudinal road profiles. Int. J. Vehicle Design **40**, 2–14 (2006)
2. De Boor, C.: A practical guide to splines, *Applied Mathematical Sciences*, vol. 27, revised edn. Springer-Verlag New York (2001)
3. Ferraty, F., Vieu, P.: Nonparametric functional data analysis: theory and practice. Springer Science and Business Media (2006)
4. Guo J., H.J.J.B.Y., Zhang, Z.: Spline-lasso in high-dimensional linear regression. Journal of the American Statistical Association **111**(513), 288–297 ((2016))
5. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction, 2 edn. Springer (2009). URL http://www-stat.stanford.edu/~tibs/ElemStatLearn/
6. Hsing, T., Eubank, R.: Theoretical foundations of functional data analysis, with an introduction to linear operators. John Wiley and Sons (2015)
7. Johannesson, P., Speckert, M. (eds.): Guide to Load Analysis for Durability in Vehicle Engineering. Wiley, Chichester. (2013)
8. Karhunen, K.: Über lineare Methoden in der Wahrscheinlichkeitsrechnung. Ann. Acad. Sci. Fennicae. Ser. A. **37**, 1–79 (1947)
9. Liu, X., Nassar, H., Podgórski, K.: Splinets—efficient orthonormalization of the b-splines. ArXiv **abs/1910.07341** (2019)
10. Podgórski, K., Rychlik, I., Wallin, J.: Slepian noise approach for gaussian and Laplace moving average processes. Extremes **18**(4), 665–695 (2015). URL https://doi.org/10.1007/s10687-015-0227-z
11. Ramsay, J.O.: Functional data analysis. Encyclopedia of Statistical Sciences **4** (2004)
12. Schumaker, L.: Spline functions: basic theory. Cambridge University Press (2007)

Uniqueness for a Second Order Gradient Flow of Elastic Networks

Matteo Novaga and Paola Pozzi

Abstract In a previous work by the authors a second order gradient flow of the p-elastic energy for a planar theta-network of three curves with fixed lengths was considered and a weak solution of the flow was constructed by means of an implicit variational scheme. Long-time existence of the evolution and convergence to a critical point of the energy were shown. The purpose of this note is to prove uniqueness of the weak solution when $p = 2$.

MSC(2010) 35K92, 53A04, 53C44

1 Introduction

In [1] we considered a second order gradient flow of the p-elastic energy for a planar theta-network of three curves with fixed lengths. We constructed a weak solution of the flow by means of an implicit variational scheme and showed long-time existence of the evolution as well as convergence to a critical point of the energy. The purpose of this short note is to show uniqueness of the long-time weak solution when $p = 2$.

For the sake of conciseness we refer to [1] for motivation and a list of relevant references. Let us here briefly recall the setting and state our new contribution.

We consider a theta-network composed of three inextensible planar curves. Each curve $\gamma_i = \gamma_i(s)$ of fixed length $L_i > 0$, $i = 1, 2, 3$, is parametrized by arc-length s over the domain $\bar{I}_i = [0, L_i]$. Without loss of generality we may assume that

$$0 < L_3 \leq \min\{L_2, L_1\}.$$

M. Novaga
Dipartimento di Matematica, Università di Pisa, Pisa, Italy
e-mail: matteo.novaga@unipi.it

P. Pozzi (✉)
Fakultät für Mathematik, Universität Duisburg-Essen, Essen, Germany
e-mail: paola.pozzi@uni-due.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_77

Since the network is a theta- network, the three curves satisfy the constraint

$$\gamma_1(0) = \gamma_2(0) = \gamma_3(0), \qquad \gamma_1(L_1) = \gamma_2(L_2) = \gamma_3(L_3).$$

Let $T^i = T^i(s) = \gamma_i'(s) = (\cos\theta^i, \sin\theta^i)$ denote the unit tangent of the curve γ_i and let $\boldsymbol{\kappa}_i = \partial_s T^i$ be the curvature vector. Letting $p \in (1, +\infty)$, the p-elastic energy of the network is defined as

$$E_p(\Gamma) = \sum_{i=1}^{3} E_p(\gamma_i),$$

where

$$E_p(\gamma_i) := \frac{1}{p}\int_{I_i} |\boldsymbol{\kappa}_i|^p ds = \frac{1}{p}\int_{I_i} |\partial_s T^i|^p ds =: F_p(T^i).$$

In [1] we studied the L^2-gradient flow of the energy

$$F_p(\Gamma) := \sum_{i=1}^{3} F_p(T^i),$$

when expressed in terms of the angles θ^i corresponding to the tangent vectors T^i. This gave rise to a second order parabolic system.

The long-time existence result presented in [1] reads as follows: We let

$$H := \Big\{\boldsymbol{\theta} = (\theta^1, \theta^2, \theta^3) \in W^{1,p}(0, L_1) \times W^{1,p}(0, L_2) \times W^{1,p}(0, L_3) \mid \int_{I_1} (\cos\theta^1, \sin\theta^1) ds = \int_{I_2} (\cos\theta^2, \sin\theta^2) ds = \int_{I_3} (\cos\theta^3, \sin\theta^3) ds\Big\}$$

where note that the above constraint accounts for the fact that the theta-network should maintain its topology along the flow.

Theorem 1 *Let $\boldsymbol{\theta}_0 \in H$ and let $T > 0$. Assume that the lengths of the three curves are such that*

$$L_3 < \min\{L_1, L_2\}. \tag{1}$$

Then, there exist functions $\boldsymbol{\theta} = (\theta^1, \theta^2, \theta^3)$, with $\theta^j \in L^\infty(0, T; W^{1,p}(I_j)) \cap H^1(0, T; L^2(I_j))$, and Lagrange multipliers $\lambda^1, \lambda^2, \mu^1, \mu^2 \in L^2(0, T)$ such that the following properties hold:

(i) for any $\boldsymbol{\varphi} = (\varphi^1, \varphi^2, \varphi^3)$ *with* $\varphi^j \in L^\infty(0, T; W^{1,p}(I_j))$, $j = 1, 2, 3$, *there holds*

$$\begin{aligned} 0 = \sum_{j=1}^{3} \int_0^T \int_{I_j} \partial_t \theta^j \, \varphi^j \, ds dt + \sum_{j=1}^{3} \int_0^T \int_{I_j} |\theta_s^j|^{p-2} \theta_s^j \cdot \varphi_s^j \, ds dt \\ - \int_0^T (\lambda^1 - \mu^1) \int_{I_1} \sin(\theta^1) \, \varphi^1 ds dt + \int_0^T (\lambda^2 - \mu^2) \int_{I_1} \cos(\theta^1) \, \varphi^1 ds dt \\ + \int_0^T \lambda^1 \int_{I_2} \sin(\theta^2) \, \varphi^2 ds dt - \int_0^T \lambda^2 \int_{I_2} \cos(\theta^2) \, \varphi^2 ds dt \\ - \int_0^T \mu^1 \int_{I_3} \sin(\theta^3) \, \varphi^3 ds dt + \int_0^T \mu^2 \int_{I_3} \cos(\theta^3) \, \varphi^3 ds dt \, ; \end{aligned} \tag{2}$$

(ii) the maps $|\partial_s \theta^j|^{p-2} \partial_s \theta^j$ *belong to* $L^\infty(0, T; L^{\frac{p}{p-1}}(I_j)) \cap L^2(0, T; H^1(I_j))$, $j = 1, 2, 3$, *and satisfy*

$$(|\partial_s \theta^1|^{p-2} \partial_s \theta^1)_s = \theta_t^1 - (\lambda^1 - \mu^1) \sin \theta^1 + (\lambda^2 - \mu^2) \cos \theta^1, \tag{3}$$

$$(|\partial_s \theta^2|^{p-2} \partial_s \theta^2)_s = \theta_t^2 + \lambda^1 \sin \theta^2 - \lambda^2 \cos \theta^2, \tag{4}$$

$$(|\partial_s \theta^3|^{p-2} \partial_s \theta^3)_s = \theta_t^3 - \mu^1 \sin \theta^3 + \mu^2 \cos \theta^3, \tag{5}$$

$$\theta_s^j(0, t) = \theta_s^j(L_j, t) = 0, \text{ for } j = 1, 2, 3 \text{ and for a.e. } t \in (0, T); \tag{6}$$

(iii) for all $t \in [0, T]$, *there holds*

$$\int_{I_1} (\cos \theta^1, \sin \theta^1) ds = \int_{I_2} (\cos \theta^2, \sin \theta^2) ds = \int_{I_3} (\cos \theta^3, \sin \theta^3) ds. \tag{7}$$

Notice that the time $T > 0$ can be chosen arbitrarily, and hence Theorem 1 provides a long-time existence result.

The behavior of the solutions as $t \to +\infty$, the possible relaxation of condition (1), as well as the treatment of triods instead of theta-networks are discussed in detail in [1].

Here we want to address the question of uniqueness of the above weak solution when $p = 2$. Our goal is to show the following statement.

Theorem 2 *Let the assumptions of Theorem 1 hold and let* $p = 2$. *Then the solution* $(\boldsymbol{\theta}, \boldsymbol{\lambda}, \boldsymbol{\mu})$ *given in Theorem 1 is unique.*

Before providing the proof let us recall some important facts about the Lagrange multipliers and the solution given in Theorem 1. First of all by Novaga and Pozzi [1, Lemma 3.5] we have that

$$\sup_{(0,T)} \|\partial_s \theta^j\|_{L^p(I_j)} \leq C, \qquad j = 1, 2, 3, \tag{8}$$

where the constant C depends on the energy of the initial data and the choice of p. By Novaga and Pozzi [1, Proposition 3.9], we have also a uniform bound

$$|\boldsymbol{\lambda}(t)| + |\boldsymbol{\mu}(t)| \leq C \tag{9}$$

for almost any $t \in (0, T)$, where $\boldsymbol{\lambda}(t) = (\lambda^1(t), \lambda^2(t))$, $\boldsymbol{\mu}(t) = (\mu^1(t), \mu^2(t))$. More precisely, the Lagrange multipliers solve the system

$$\boldsymbol{\lambda} \cdot A^2 + \boldsymbol{\mu} \cdot A^3 = G^3 - G^2 \tag{10}$$

$$-\boldsymbol{\lambda} \cdot (A^2 + A^1) + \boldsymbol{\mu} \cdot A^1 = G^2 - G^1 \tag{11}$$

for a.e. time $t \in (0, T)$ where A^i, $i = 1, 2, 3$, are the matrices

$$A^i = A^i(t) = \begin{pmatrix} \int_{I_i} \sin^2 \theta^i ds & -\int_{I_i} \sin\theta^i \cos\theta^i ds \\ -\int_{I_i} \sin\theta^i \cos\theta^i ds & \int_{I_i} \cos^2 \theta^i ds \end{pmatrix} =: A^i(\theta^i), \tag{12}$$

and G^i are the vectors

$$G^i = G^i(\theta^i) := \int_{I_i} |\partial_s \theta^i|^p (\cos\theta^i, \sin\theta^i) ds. \tag{13}$$

As discussed in [1] condition (1) yields not only the solvability of the above system, but also the bound

$$|\boldsymbol{\lambda}(t)| + |\boldsymbol{\mu}(t)| \leq C \left(|G^3 - G^2| + |G^2 - G^1|\right) \tag{14}$$

which is crucial for the analysis. The above constants C appearing in (9) and (14) depend on the initial data, initial energy, the length of the three curves, but not on time (see [1, Lemma 2.5 and Proposition 3.9] for more details).

2 Proof of Uniqueness

Here we provide the proof of Theorem 2. Let the assumptions of Theorem 1 hold and let $p = 2$. Moreover let $\boldsymbol{\theta} = (\theta^1, \theta^2, \theta^3)$ and $\hat{\boldsymbol{\theta}} = (\hat{\theta}^1, \hat{\theta}^2, \hat{\theta}^3)$ with Lagrange multipliers (λ^1, λ^2), (μ^1, μ^2) respectively $(\hat{\lambda}^1, \hat{\lambda}^2)$, $(\hat{\mu}^1, \hat{\mu}^2)$ be two solutions to the same initial data $\boldsymbol{\theta}_0 \in H$ and satisfying (2). Taking the difference of the two weak formulations tested with $\varphi = (\varphi^1, \varphi^2, \varphi^3)$, $\varphi^j = (\theta^j - \hat{\theta}^j)\eta_\epsilon$, $j = 1, 2, 3$, where $\eta_\epsilon \in C^\infty([0, T], [0, 1])$ is such that $\eta_\epsilon(t) = 1$ for $t \in [0, \tau]$, $\eta_\epsilon(t) = 0$ for

$t \in [\tau + \epsilon, T]$, $0 < \epsilon < T - \tau$, we obtain after sending $\epsilon \to 0$ the following equation

$$
\begin{aligned}
0 = \sum_{j=1}^{3} \int_0^\tau \int_{I_j} (\partial_t \theta^j - \partial_t \hat{\theta}^j)\,(\theta^j - \hat{\theta}^j) ds dt + \sum_{j=1}^{3} \int_0^\tau \int_{I_j} |(\theta_s^j - \hat{\theta}_s^j)|^2\, ds dt \\
+ \Big\{ - \int_0^\tau (\lambda^1 - \mu^1) \int_{I_1} \sin(\theta^1)\,(\theta^1 - \hat{\theta}^1) ds dt \\
+ \int_0^\tau (\lambda^2 - \mu^2) \int_{I_1} \cos(\theta^1)\,(\theta^1 - \hat{\theta}^1) ds dt \\
-\Big(- \int_0^\tau (\hat{\lambda}^1 - \hat{\mu}^1) \int_{I_1} \sin(\hat{\theta}^1)\,(\theta^1 - \hat{\theta}^1) ds dt \\
+ \int_0^\tau (\hat{\lambda}^2 - \hat{\mu}^2) \int_{I_1} \cos(\hat{\theta}^1)\,(\theta^1 - \hat{\theta}^1) ds dt \Big) \\
+ \int_0^\tau \lambda^1 \int_{I_2} \sin(\theta^2)\,(\theta^2 - \hat{\theta}^2) ds dt - \int_0^\tau \lambda^2 \int_{I_2} \cos(\theta^2)\,(\theta^2 - \hat{\theta}^2) ds dt \\
- \Big(\int_0^\tau \hat{\lambda}^1 \int_{I_2} \sin(\hat{\theta}^2)\,(\theta^2 - \hat{\theta}^2) ds dt - \int_0^\tau \hat{\lambda}^2 \int_{I_2} \cos(\hat{\theta}^2)\,(\theta^2 - \hat{\theta}^2) ds dt \Big) \\
- \int_0^\tau \mu^1 \int_{I_3} \sin(\theta^3)\,(\theta^3 - \hat{\theta}^3) ds dt + \int_0^\tau \mu^2 \int_{I_3} \cos(\theta^3)\,(\theta^3 - \hat{\theta}^3) ds dt \\
- \Big(- \int_0^\tau \hat{\mu}^1 \int_{I_3} \sin(\hat{\theta}^3)\,(\theta^3 - \hat{\theta}^3) ds dt + \int_0^\tau \hat{\mu}^2 \int_{I_3} \cos(\hat{\theta}^3)\,(\theta^3 - \hat{\theta}^3) ds dt \Big) \Big\}.
\end{aligned}
$$

This gives

$$
\begin{aligned}
\sum_{j=1}^{3} \frac{1}{2} \|(\theta^j - \hat{\theta}^j)(\tau)\|_{L^2(I_j)}^2 + \sum_{j=1}^{3} \int_0^\tau \|(\theta_s^j - \hat{\theta}_s^j)(t)\|_{L^2(I_j)}^2 dt \qquad (15) \\
= \sum_{j=1}^{3} \frac{1}{2} \|(\theta^j - \hat{\theta}^j)(0)\|_{L^2(I_j)}^2 - \{\ldots\},
\end{aligned}
$$

where the first term in the right-hand side vanishes since $\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$ have the same initial data. The terms in the bracket $\{\ldots\}$ are made up of differences that are estimated in a similar way. We give here in a exemplary manner the treatment of the term

$$
J := \int_0^\tau \lambda^1 \int_{I_2} \sin(\theta^2)\,(\theta^2 - \hat{\theta}^2) ds dt - \int_0^\tau \hat{\lambda}^1 \int_{I_2} \sin(\hat{\theta}^2)\,(\theta^2 - \hat{\theta}^2) ds dt.
$$

First of all notice that

$$\begin{aligned}|J| \leq &\left|\int_0^\tau (\lambda^1 - \hat{\lambda}^1) \int_{I_2} \sin(\theta^2)\,(\theta^2 - \hat{\theta}^2) ds dt\right| \\ &+ \left|\int_0^\tau \hat{\lambda}^1 \int_{I_2} (\sin(\hat{\theta}^2) - \sin(\theta^2))\,(\theta^2 - \hat{\theta}^2) ds dt\right| \\ &\leq C \int_0^\tau |\lambda^1(t) - \hat{\lambda}^1(t)|\, \|(\theta^2 - \hat{\theta}^2)(t)\|_{L^2(I_2)} dt \\ &+ C \int_0^\tau \|(\theta^2 - \hat{\theta}^2)(t)\|^2_{L^2(I_2)} dt\end{aligned} \tag{16}$$

where we have used the mean value theorem and the bound (9) in the last inequality.

To estimate the difference in the Lagrange multipliers we recall that they fulfill the system (10) and (11) for almost every time. Subtraction of the corresponding equations yields

$$\begin{aligned}(\boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}) \cdot A^2 + (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}) \cdot A^3 &= rhs1 \\ -(\boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}) \cdot (A^2 + A^1) + (\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}) \cdot A^1 &= rhs2\end{aligned}$$

where

$$\begin{aligned}rhs1 &= G^3 - \hat{G}^3 - (G^2 - \hat{G}^2) + \hat{\boldsymbol{\lambda}}(\hat{A}^2 - A^2) + \hat{\boldsymbol{\mu}}(\hat{A}^3 - A^3) \\ rhs2 &= G^2 - \hat{G}^2 - (G^1 - \hat{G}^1) + \hat{\boldsymbol{\lambda}}(A^2 + A^1 - \hat{A}^2 - \hat{A}^1) + \hat{\boldsymbol{\mu}}(\hat{A}^1 - A^1).\end{aligned}$$

Similarly to (14) we obtain

$$|\boldsymbol{\lambda} - \hat{\boldsymbol{\lambda}}| + |\boldsymbol{\mu} - \hat{\boldsymbol{\mu}}| \leq C(|rhs1| + |rhs2|).$$

Again we show exemplary the treatment of a few terms in the evaluation of $|rhs1| + |rhs2|$, since all remaining ones are estimated in a similar way. We have using the mean value theorem that

$$\begin{aligned}|G^3 - \hat{G}^3| &= \left|\int_{I_3} |\partial_s \theta^3|^2 (\cos\theta^3, \sin\theta^3) ds - \int_{I_3} |\partial_s \hat{\theta}^3|^2 (\cos\hat{\theta}^3, \sin\hat{\theta}^3) ds\right| \\ &\leq \left|\int_{I_3} (|\partial_s \theta^3|^2 - |\partial_s \hat{\theta}^3|^2)(\cos\theta^3, \sin\theta^3) ds\right| \\ &+ \left|\int_{I_3} |\partial_s \hat{\theta}^3|^2 (\cos\hat{\theta}^3 - \cos\theta^3, \sin\hat{\theta}^3 - \sin\theta^3) ds\right| \\ &\leq C\|(\theta^3_s - \hat{\theta}^3_s)\|_{L_2(I_3)} \|(\theta^3_s + \hat{\theta}^3_s)\|_{L_2(I_3)} + C\|\hat{\theta}^3_s\|^2_{L_2(I_3)} \|\theta^3 - \hat{\theta}^3\|_{L^\infty(I_3)}.\end{aligned}$$

Using (8) and embedding theory yields

$$|G^3 - \hat{G}^3| \leq C\|(\theta_s^3 - \hat{\theta}_s^3)\|_{L_2(I_3)} + C\|(\theta^3 - \hat{\theta}^3)\|_{L_2(I_3)}.$$

Next observe that by (9) and the mean value theorem we can compute

$$|\hat{\boldsymbol{\lambda}}(\hat{A}^2 - A^2)| \leq C \int_{I_2} |\theta^2 - \hat{\theta}^2| ds \leq C\|(\theta^2 - \hat{\theta}^2)\|_{L_2(I_2)}.$$

With similar argument as depicted above we therefore infer that

$$\begin{aligned} &|\boldsymbol{\lambda}(t) - \hat{\boldsymbol{\lambda}}(t)| + |\boldsymbol{\mu}(t) - \hat{\boldsymbol{\mu}}(t)| \\ &\leq C \sum_{j=1}^{3} \Big(\|(\theta_s^j - \hat{\theta}_s^j)(t)\|_{L_2(I_j)} + \|(\theta^j - \hat{\theta}^j)(t)\|_{L_2(I_j)} \Big) \end{aligned} \tag{17}$$

for almost every time $t \in (0, T)$. Using this estimate in (16) for the evaluation of the term J we obtain by means of a ϵ-Young inequality

$$|J| \leq \epsilon \sum_{j=1}^{3} \int_0^\tau \|(\theta_s^j - \hat{\theta}_s^j)(t)\|_{L_2(I_j)}^2 dt + C_\epsilon \sum_{j=1}^{3} \int_0^\tau \|(\theta^j - \hat{\theta}^j)(t)\|_{L_2(I_j)}^2 dt.$$

Going back to (15) and treating all remaining terms in the bracket $\{\ldots\}$ in an analogous way we finally infer

$$\begin{aligned} &\sum_{j=1}^{3} \frac{1}{2}\|(\theta^j - \hat{\theta}^j)(\tau)\|_{L^2(I_j)}^2 + \sum_{j=1}^{3} \int_0^\tau \|(\theta_s^j - \hat{\theta}_s^j)(t)\|_{L^2(I_j)}^2 dt \\ &\leq \epsilon \sum_{j=1}^{3} \int_0^\tau \|(\theta_s^j - \hat{\theta}_s^j)(t)\|_{L_2(I_j)}^2 dt + C_\epsilon \sum_{j=1}^{3} \int_0^\tau \|(\theta^j - \hat{\theta}^j)(t)\|_{L_2(I_j)}^2 dt. \end{aligned}$$

Choosing ϵ sufficiently small yields

$$\sum_{j=1}^{3} \|(\theta^j - \hat{\theta}^j)(\tau)\|_{L^2(I_j)}^2 \leq C \sum_{j=1}^{3} \int_0^\tau \|(\theta^j - \hat{\theta}^j)(t)\|_{L_2(I_j)}^2 dt$$

for any $\tau \in (0, T)$. A Gronwall argument gives $\boldsymbol{\theta} = \hat{\boldsymbol{\theta}}$ and hence by (17) also equality of the Lagrange multipliers, as claimed.

Acknowledgments MN has been supported by GNAMPA-INdAM and PP has been supported by the DFG (German Research Foundation) Projektnummer: 404870139.

Reference

1. Novaga, M., Pozzi, P.: A second order gradient flow of p-elastic planar networks. SIAM J. Math. Anal. **52**(1), 682–708 (2020). https://doi.org/10.1137/19M1262292

A Second Order Finite Element Method with Mass Lumping for Wave Equations in H(div)

Herbert Egger and Bogdan Radu

Abstract We consider the efficient numerical approximation of acoustic wave propagation in time domain by a finite element method with mass lumping. In the presence of internal damping, the problem can be reduced to a second order formulation in time for the velocity field alone. For the spatial approximation we consider $H(\mathrm{div})$-conforming finite elements of second order. In order to allow for an efficient time integration, we propose a mass-lumping strategy based on approximation of the L^2-scalar product by inexact numerical integration which leads to a block-diagonal mass matrix. A careful error analysis allows to show that second order accuracy is not reduced by the quadrature errors which is illustrated also by numerical tests.

1 Motivation

The propagation of acoustic sound in channels or ducts with a small extension in one of the spatial directions is substantially damped by friction at the walls. Averaging over the small direction then leads to systems with internal damping of the form

$$\partial_t u + \nabla p = -du \tag{1}$$

$$\partial_t p + \mathrm{div}\, u = 0 \tag{2}$$

with appropriate initial and boundary conditions. The variables u and p here denote the velocity and pressure fields, respectively, and for ease of notation, the equations are written in dimensionless form. The right hand side in (1) models the drag forces and d denotes the corresponding dimensionless damping or drag coefficient.

H. Egger · B. Radu (✉)
TU Darmstadt, Darmstadt, Germany
e-mail: egger@mathematik.tu-darmstadt.de; bradu@mathematik.tu-darmstadt.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_78

In the absence of damping, i.e., when $d = 0$, the system (1)–(2) can be reduced to the second order wave equation for the pressure

$$\partial_{tt} p - \Delta p = 0 \tag{3}$$

which results from differentiating (2) and eliminating u via equation (1). The efficient discretization of (3) can be obtained in various ways, e.g., by finite difference or finite element methods. The latter are more flexible concerning high-order approximations and the treatment of non-trivial domains but suffer from non-diagonal mass-matrices which hinder the efficient time-integration. This can be overcome by mass-lumping; we refer to [2] for an overview about various methods and to [4, 5] for some particular results concerning mass-lumping for finite element approximations.

In the presence of damping, i.e., if $d \neq 0$, the elimination of the velocity u from (1)–(2) leads to an integro-differential equation for the pressure whose time-integration is again non-trivial. Elimination of the pressure, on the other hand, again leads to a second order differential equation

$$\partial_{tt} u + d\partial_t u - \nabla \operatorname{div} u = 0 \tag{4}$$

but now for the vector valued velocity field u. The stable discretization of (4) by finite elements requires the use of $H(\mathrm{div})$-conforming spaces and novel mass lumping techniques are required for the efficient time integration. We refer to [2] for corresponding results for $H(\mathrm{curl})$-conforming finite-elements required in the context of electromagnetic wave propagation.

In a recent work [3], we considered the lowest-order discretization of the system (1)–(2) by BDM$_1$–P$_0$ finite-elements with mass-lumping as suggested by Wheeler and Yotov [6] in the context of porous medium flow. The resulting scheme is convergent of first order and super-convergence for the projected pressure can be utilized to obtain second-order convergence for the velocity by a non-local post-processing strategy. In this paper, we choose finite elements with better approximation properties which lead to second order approximations in the energy norm

$$\|\partial_t u(t) - \partial_t u_h(t)\|_{L^2} + \|\operatorname{div}(u(t) - u_h(t))\|_{L^2} \le C(u)h^2 \tag{5}$$

without the need for post-processing. A novel mass-lumping strategy is proposed to allow for the efficient time integration and a careful analysis of the quadrature error is presented in order to establish the order optimal convergence rates (5). We here consider only approximations of second order on hybrid meshes in two space dimensions. The basic arguments of our analysis however can be used to investigate approximations of higher order and in three space dimensions.

The remainder of this note is organized as follows: In Sect. 2, we formally state our model problem and basic assumptions and then introduce its finite element approximation. In Sect. 3, we present some auxiliary estimates and then formulate and prove our main result in Sect. 4. Details about the numerical implementation are given in Sect. 5 and for illustration, we present in Sect. 6 some preliminary numerical tests.

2 Problem Statement and Finite Element Approximation

Throughout the presentation, we denote by $\Omega \subseteq \mathbb{R}^2$ a bounded polygonal Lipschitz domain and by $T > 0$ a finite time horizon. We consider the system

$$\partial_{tt} u + d\partial_t u - \nabla \operatorname{div} u = 0, \qquad \text{in } \Omega \tag{6}$$

$$n \cdot u = 0, \qquad \text{on } \partial\Omega. \tag{7}$$

The existence of a unique solution u for (6)–(7) with given initial values $u(0) = u_0$ and $\partial_t u(0) = u_1$ can be established by semigroup theory; see [3] for details. Moreover, any classical solution of (6)–(7) satisfies the variational identity

$$(\partial_{tt} u(t), v) + (d\partial_t u(t), v) + (\operatorname{div} u(t), \operatorname{div} v) = 0, \tag{8}$$

for all $v \in H_0(\operatorname{div}, \Omega) = \{u \in L^2(\Omega)^2 \,:\, \operatorname{div} u \in L^2(\Omega) \text{ and } n \cdot u = 0 \text{ on } \partial\Omega\}$. Here and below, we use $(\cdot, \cdot)$ to denote the standard L^2-scalar product.

Let $T_h = \{K\}$ be a quasi-uniform mesh of Ω comprised of triangles and parallelograms and h be the mesh size. We consider local approximation spaces

$$V(K) = \begin{cases} \mathrm{RT}_1(K), & \text{if } K \text{ is a triangle,} \\ \mathrm{BDFM}_2(K), & \text{if } K \text{ is a parallelogram,} \end{cases} \tag{9}$$

with vector valued polynomial spaces $\mathrm{RT}_1(K)$ and $\mathrm{BDFM}_2(K)$ as defined in [1]; compare with Fig. 1. The global approximation spaces is then defined as

$$V_h = \{v_h \in H_0(\operatorname{div}, \Omega) \,:\, v_h|_K \in V(K)\}.$$

The scalar product on V_h will be approximated by $(u, v) \simeq (u, v)_h := \sum_K (u, v)_{h,K}$ with local contributions obtained by numerical integration according to

$$(u, v)_{h,K} = |K| \Big(\alpha(u(m_K) \cdot v(m_K)) + \sum\nolimits_i \beta\, (u(v_{K,i}) \cdot v(v_{K,i}))\Big) \tag{10}$$

Here m_K and $v_{K,i}$ represent the midpoint and vertices of the element K, respectively, while α and β are the corresponding weights. On triangles, we choose $\alpha = \frac{3}{4}$ and $\beta = \frac{1}{12}$, while on parallelograms, we choose $\alpha = \frac{2}{3}$ and $\beta = \frac{1}{12}$. For the space discretization of (8), we then consider the following inexact Galerkin scheme.

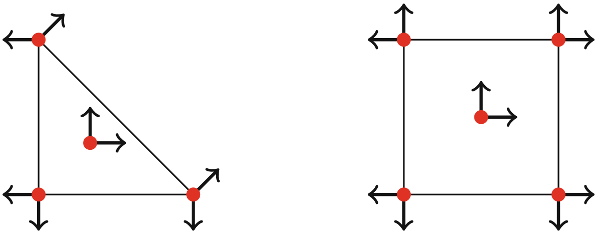


Fig. 1 Degrees of freedom for RT_1 (left) and BDFM_2 (right) and quadrature points (red dots)

Problem 1 Let $u_{h,0}, u_{h,1} \in V_h$ be given. Find $u_h : [0,T] \to V_h$ such that

$$(\partial_{tt} u_h(t), v_h)_h + (d\partial_t u_h, v_h)_h + (\operatorname{div} u_h(t), \operatorname{div} v_h) = 0 \tag{11}$$

for all $v_h \in V_h$ and all $t \in [0,T]$ and such that $u_h(0) = u_{h,0}$ and $\partial_t u_h(0) = u_{h,1}$.

The following result ensures the well-posedness of Problem 1.

Lemma 1 *The inexact scalar product $(\cdot,\cdot)_h$ induces a norm on V_h and, as a consequence, Problem 1 admits a unique solution.*

Proof Choose any basis for V_h. Then the mass matrix associated with the inexact scalar product $(\cdot,\cdot)_h$ is symmetric and positive definite; this can be verified by elementary computations on single elements. Existence of a unique solution then follows from the Picard–Lindelöf theorem. □

3 Auxiliary Results

In the following, we recall some well-known interpolation results and then derive estimates for the quadrature error which will be required below. Let us start with introducing a canonical interpolation operator which is defined locally by

$$(\Pi_h u)|_K = \begin{cases} \Pi_K^{\mathrm{RT}} u|_K, & \text{if } K \text{ is a triangle,} \\ \Pi_K^{\mathrm{BDFM}} u|_K, & \text{if } K \text{ is a parallelogram.} \end{cases} \tag{12}$$

Here Π_K^{RT} and Π_K^{BDFM} denote the standard interpolation operators for the local finite element spaces $\mathrm{RT}_1(K)$ and $\mathrm{BDFM}_2(K)$, respectively; see [1] for details. The following assertions then follow from well-known results about the local operators.

Lemma 2 *Let $K \in T_h$ and Π_h be defined as in* (12). *Then*

$$\|u - \Pi_h u\|_{L^2(K)} \le Ch^2 \|u\|_{H^2(K)},$$

for all $u \in H(div, \Omega) \cap H^2(T_h)^2$ *with constant* C *independent of* h. *Moreover*

$$(div(u - \Pi_h u), div\, v_h) = 0, \qquad \forall\, v_h \in V_h.$$

We will further require the following property of the spaces $\mathrm{RT}_1(K)$ on triangles.

Lemma 3 *Let* K *be a triangle. Then there exists a unique splitting*

$$\mathrm{RT}_1(K) = P_1(K)^2 \oplus B(K)$$

and $\dim(B(K)) = \dim(\mathrm{div}(B(K)))$. *Therefore,* $\|\mathrm{div}(\cdot)\|_{L^2}$ *defines a norm on* $B(K)$ *and* $\|\mathrm{div}\, v_h^B\|_{L^2(K)} \geq c\|\nabla v_h^B\|_{L^2(K)}$ *for any* $v_h^B \in B(K)$ *with* c *independent of* K.

These assertions can be verified by a elementary computations on the reference element and a mapping argument. As a next step, we summarize some properties of the numerical integration underlying the definition (10) of the inexact scalar product.

Lemma 4 *The quadrature rule in* (10) *is exact for polynomials of degree* $k \leq 2$ *on triangles and for polynomials of degree* $k \leq 3$ *on parallelograms.*

The validity of these claims can again be verified by elementary computations on reference elements. In the following, we will abbreviate the quadrature errors by

$$\sigma_K(u, v) := (u, v)_{h,K} - (u, v)_K \quad \text{and} \quad \sigma_h(u, v) = \sum\nolimits_{K \in T_h} \sigma_K(u, v) \tag{13}$$

Moreover, we denote by $\pi_K^k : L^2(K) \to P_k(K)^2$ the local L^2-orthogonal projections and we use $\pi_h^k : L^2(\Omega) \to P_k(T_h)^2$ to denote the corresponding global projection.

Lemma 5 *Let* $u \in L^2(\Omega)^2$ *with* $u|_K \in H^1(K)^2$ *for all* $K \in T_h$. *Then*

$$|\sigma_K(\pi_h^1 u, v_h)| \leq \begin{cases} Ch^2\|u\|_{H^1(K)}\|div\, v_h\|_{L^2(K)} & \text{if } K \text{ is a triangle,} \\ 0, & \text{if } K \text{ is a parallelogram,} \end{cases}$$

for all $v_h \in V_h$ *and all* $K \in T_h$ *with constant* C *independent of the element* K.

Proof From Lemma 4, we deduce that $|\sigma_K(\pi_h^1 u, v_h)| = 0$ on parallelograms. For triangles, on the other hand, we can estimate the quadrature error by

$$\begin{aligned} |\sigma_K(\pi_h^1 u, v_h)| &= |\sigma_K(\pi_h^1 u - \pi_h^0 u, v_h - \pi_h^1 v_h)| \\ &\leq \|\pi_h^1 u - \pi_h^0 u\|_{L^2(K)}\|v_h - \pi_h^1 v_h\|_{L^2(K)} + \|\pi_h^1 u - \pi_h^0 u\|_h \|v_h - \pi_h^1 v_h\|_h \\ &\leq Ch^3\|u\|_{H^1(K)}\|\nabla^2 v_h\|_{L^2(K)}. \end{aligned}$$

By Lemma 3, we can split $v_h = v_h^1 \oplus v_h^B$ on K and further estimate

$$\|\nabla^2 v_h\|_{L^2(K)} = \|\nabla^2 v_h^B\|_{L^2(K)} \le C' h^{-1} \|\nabla v_h^B\|_{L^2(K)} \le C'' h^{-1} \|\mathrm{div} v_h^B\|_{L^2(K)}.$$

The linear independence of the splitting also yields $\|\mathrm{div} v_h^B\|_{L^2(K)} \le C \|\mathrm{div} v_h\|_{L^2(K)}$, and a combination of the estimates already yields the bound for the triangles. □

4 Convergence Analysis

For ease of notation, we will only consider the case $d = 0$ in the sequel. As usual, we begin with splitting the error in interpolation and discrete error components by

$$u - u_h = (u - \Pi_h u) + (\Pi_h u - u_h) =: -\eta + \psi_h.$$

The discrete error component can be estimated as follows.

Lemma 6 *Let u and u_h denote the solutions of (8) and (11) with initial values linked by $u_h(0) = \Pi_h u(0)$ and $\partial_t u_h(0) = \Pi_h \partial_t u(0)$. Then the discrete error satisfies*

$$\|\partial_t(\Pi_h u - u_h)\|_{L^\infty(0,T;L^2(\Omega))} + \|div\,(\Pi_h u - u_h)\|_{L^\infty(0,T;L^2(\Omega))} \le C_1(u,T)h^2$$

with constant $C_1(u,T) = C_1'(u,T) + C_1''(u,T)$ as defined in the proof below.

Proof The discrete error $\psi_h = \Pi_h u - u_h$ can be seen to satisfy the identity

$$(\partial_{tt}\psi_h(t), v_h) + (\mathrm{div}\,\psi_h(t), \mathrm{div}\, v_h) = \\ (\partial_{tt}\eta(t), v_h) + (\mathrm{div}\,\eta(t), \mathrm{div}\, v_h) + \sigma_h(\Pi_h \partial_{tt} u(t), v_h)$$

for all $v_h \in V_h$ and $0 \le t \le T$. Moreover, $\psi_h(0) = \partial_t \psi_h(0) = 0$ by construction. Choosing $v_h = \partial_t \psi_h(t)$ as a test function followed by integrating from 0 to t leads to

$$\frac{1}{2}\left(\|\partial_t \psi_h(t)\|_h^2 + \|\mathrm{div}\,\psi_h(t)\|_{L^2(\Omega)}^2\right) \tag{14}$$
$$= \int_0^t (\partial_{tt}\eta(s), \partial_t\psi_h(s)) + (\mathrm{div}\,\eta(s), \mathrm{div}\,\partial_t\psi_h(s)) + \sigma_h(\Pi_h\partial_{tt}u(s), \partial_t\psi_h(s))\,ds$$
$$=: (i) + (ii) + (iii).$$

Using Cauchy–Schwarz and Young's inequalities, the first term can be estimated by

$$(i) \le C_1'(u)^2 h^4 + \tfrac{1}{4}\|\partial_t\psi_h\|_{L^\infty(0,t,L^2(\Omega))}^2$$

with constant $C_1'(u,t) = C\|\partial_{tt}u\|_{L^1(0,t,H^2(\Omega))}$, and by Lemma 2, we get $(ii) = 0$. The remaining third term can finally be estimated by

$$\begin{aligned}(iii) &= \int_0^t \sigma_h(\Pi_h\partial_{tt}u(s) - \pi_h^1\partial_{tt}u(s), \partial_t\psi_h(s)) + \int_0^t \sigma_h(\pi_h^1\partial_{tt}u(s), \partial_t\psi_h(s))\\ &=: (iv) + (v).\end{aligned}$$

The term (iv) can be bounded with the same arguments as (i). If K is a parallelogram, then $(v) \equiv 0$ by Lemma 5. On triangles, we use integration-by-parts in time, to get

$$\begin{aligned}(v) &= \sigma_h(\pi_h^1\partial_{tt}u(t), \psi_h(t)) - \int_0^t \sigma_h(\pi_h^1\partial_{ttt}u(s), \psi_h(s))\,ds\\ &\leq C_1''(u,t)^2h^4 + \tfrac{1}{2}\|\mathrm{div}\,\psi_h\|^2_{L^\infty(0,t,L^2(\Omega))}\end{aligned}$$

with $C_1''(u,t) = C(\|\partial_{tt}u\|_{L^\infty(0,t,H^1(\Omega))} + \|\partial_{ttt}u\|_{L^1(0,t,H^1(\Omega))})$, where we used Lemma 5 in the second step. Taking the supremum over $t \in [0,T]$ in (14) and absorbing all the terms with the test function into the left side of (14) now yields the assertion. □

Theorem 1 *Let the assumptions of Lemma 6 hold. Then*

$$\|\partial_t(u-u_h)\|_{L^\infty(0,T;L^2(\Omega))} + \|div(u-u_h)\|_{L^\infty(0,T;L^2(\Omega))} \leq C(u,T)h^2,$$

with constant $C(u,T) = C_1'(u,T) + C_1''(u,T) + C_2(u,T)$ as in the proof below.

Proof Using Lemma 2, we can estimate the interpolation error by

$$\|\partial_t\eta\|_{L^\infty(0,T;L^2(\Omega))} + \|\mathrm{div}\,\eta\|_{L^\infty(0,T;L^2(\Omega))} \leq C_2(u,T)h^2,$$

with $C_1(u,t) = C(\|\partial_t u\|_{L^\infty(0,t;H^2(\Omega)} + \|\mathrm{div}\,u\|_{L^\infty(0,t;H^2(\Omega)})$. The proof is completed by adding the bounds for the discrete error components provided by Lemma 6. □

5 Implementation and Mass Lumping

For completeness, we now briefly introduce appropriate basis functions for the spaces $\mathrm{RT}_1(K)$ and $\mathrm{BDFM}_2(K)$ which together with the inexact scalar product $(\cdot,\cdot)_h$ lead to a block-diagonal mass matrix. Let $\{\lambda_i\}$ denote the barycentric coordinates of the element K and let $\nabla^\perp f = (\partial_y f, -\partial_x f)^T$. On triangles, we define

$$\Phi_{B1} = \lambda_2(\lambda_1\nabla^\perp\lambda_3 - \lambda_3\nabla^\perp\lambda_1) \quad\text{and}\quad \Phi_{B2} = \lambda_3(\lambda_1\nabla^\perp\lambda_2 - \lambda_2\nabla^\perp\lambda_1)$$

which are the two $H(\mathrm{div})$-bubble functions associated with the element midpoint; see Fig. 1. The basis functions associated with the three vertices are given by

$$\begin{aligned} \Phi_{1,1} &= \lambda_1 \nabla^\perp \lambda_2 + \Phi_{B1} - 2\Phi_{B2}, & \Phi_{1,2} &= \lambda_2 \nabla^\perp \lambda_1 + \Phi_{B1} + \Phi_{B2}, \\ \Phi_{2,1} &= \lambda_2 \nabla^\perp \lambda_3 - 2\Phi_{B1} + \Phi_{B2}, & \Phi_{2,2} &= \lambda_3 \nabla^\perp \lambda_2 + \Phi_{B1} - 2\Phi_{B2}, \\ \Phi_{3,1} &= \lambda_1 \nabla^\perp \lambda_3 - 2\Phi_{B1} + \Phi_{B2}, & \Phi_{3,2} &= \lambda_3 \nabla^\perp \lambda_1 + \Phi_{B1} + \Phi_{B2}. \end{aligned}$$

For parallelograms, let $\xi_{ij} \in [0, 1]$ denote the local coordinate on the edge e_{ij} pointing from vertex p_i to p_j. Following the construction in [7], we define by

$$\phi_{B1} = (\lambda_1 + \lambda_4)(\lambda_2 + \lambda_3)\nabla^\perp \xi_{23} \quad \text{and} \quad \phi_{B1} = (\lambda_1 + \lambda_2)(\lambda_3 + \lambda_4)\nabla^\perp \xi_{12}$$

two $H(\mathrm{div})$-bubble functions associated with the midpoint of the element. For any of the four vertices, we further define two basis functions by

$$\begin{aligned} \phi_{1,1} &= \lambda_2 \nabla^\perp \xi_{23} + \phi_{B1}, & \phi_{1,2} &= \lambda_3 \nabla^\perp \xi_{23} + \phi_{B1}, \\ \phi_{2,1} &= \lambda_3 \nabla^\perp \xi_{34} + \phi_{B2}, & \phi_{2,2} &= \lambda_4 \nabla^\perp \xi_{34} + \phi_{B2}, \\ \phi_{3,1} &= \lambda_4 \nabla^\perp \xi_{41} - \phi_{B1}, & \phi_{3,2} &= \lambda_1 \nabla^\perp \xi_{41} - \phi_{B1}, \\ \phi_{4,1} &= \lambda_1 \nabla^\perp \xi_{12} - \phi_{B2}, & \phi_{4,2} &= \lambda_2 \nabla^\perp \xi_{12} - \phi_{B2}. \end{aligned}$$

Let us note that by construction, exactly two basis functions are associated to any of the quadrature points. Moreover, the basis functions vanish on all quadrature points except one. As a consequence, the local mass matrix corresponding to $(\cdot, \cdot)_{h,K}$ is block diagonal with 2×2 blocks. After assembling, the global mass-matrix is block-diagonal with each block corresponding to one of the quadrature points. The dimension of the individual blocks is determined by the number of degrees of freedom associated with that quadrature point; we refer to [3, 6] for details.

6 Numerical Illustration

For illustrating our results, we consider a simple test problem in two space dimensions, whose analytical solution is given by the plane wave

$$u_{ex}(x, y, t) = g(x - t)\begin{pmatrix}1\\0\end{pmatrix} \quad \text{with} \quad g(x) = 2\exp(-50(x + 1)^2),$$

See Fig. 2, we consider problem (4) with $d = 0$ on the domain $\Omega = (0, 1)^2$ with boundary and initial conditions obtained from the exact solution. In Table 1, we display the errors obtained by our second-order finite-element approximation

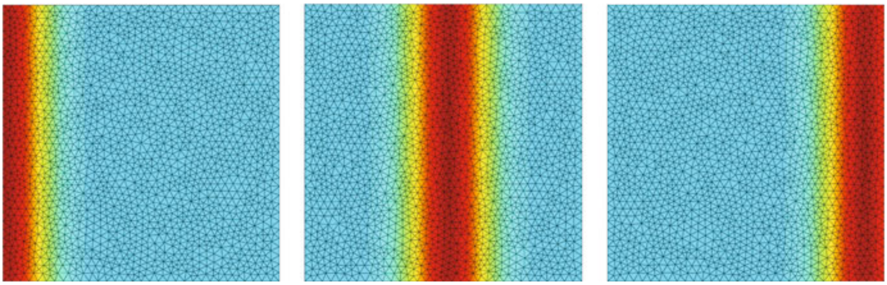


Fig. 2 Snapshots of the first component of u at different time steps

Table 1 Discrete error of the method

h	$\|\|\|\pi_h^1 u - u_h\|\|\|$	eoc
2^{-3}	0.270790	–
2^{-4}	0.060266	2.17
2^{-5}	0.016328	1.88
2^{-6}	0.004343	1.91
2^{-7}	0.001046	2.06

Time step was fixed at $\tau = 0.001$

with mass-lumping on a sequence of quasi-uniform but non-nested meshes with decreasing mesh size. As predicted by our theoretical results, we observe second order convergence.

Due to the mass lumping, time integration could be performed efficiently by the leapfrog scheme with time-step $\tau \approx h$. Since this method is second order accurate, this choice does not influence the overall convergence behavior; see [2] for details.

Acknowledgments This work was supported by the German Research Foundation (DFG) via grants TRR 146 C3, TRR 154 C4, Eg-331/1-1, and through the "Center for CE" at TU Darmstadt.

Reference

1. D. Boffi, F. Brezzi, and M. Fortin. *Mixed finite element methods and applications*, volume 44 of *Springer Series in Computational Mathematics*. Springer, 2013.
2. G. Cohen. *Higher-Order Numerical Methods for Transient Wave Equations*. Springer, 2002.
3. H. Egger and B. Radu. Super-convergence and post-processing for mixed finite element approximations of the wave equation. *Numer. Math.*, 140:427–447, 2018.
4. S. Geevers, W. A. Mulder, and J. J. W. Van der Vegt. New higher-order mass-lumped tetrahedral elements for wave propagation modeling. *SIAM J. Sci. Comp.*, 40:2830–2857, 2018.
5. W. A. Mulder. Higher-order mass-lumped finite elements for the wave equation. *J. Comput. Acoustics*, 14:671–680, 2001.
6. M. F. Wheeler and I. Yotov. A multipoint flux mixed finite element method. *SIAM J. Numer. Anal.*, 44(5):2082–2106, 2006.
7. S. Zaglmayr. *High Order Finite Elements for Electromagnetic Field Computation*. PhD thesis, 2006.

Model Order Reduction of Combustion Processes with Complex Front Dynamics

Philipp Krah, Mario Sroka, and Julius Reiss

Abstract In this work we present a data driven method, used to improve mode-based model order reduction of transport fields with sharp fronts. We assume that the original flow field $q(\mathbf{x}, t) = f(\phi(\mathbf{x}, t))$ can be reconstructed by a front shape function f and a level set function ϕ. The level set function is used to generate a local coordinate, which parametrizes the distance to the front. In this way, we are able to embed the local 1D description of the front for complex 2D front dynamics with merging or splitting fronts, while seeking a low rank description of ϕ. Here, the freedom of choosing ϕ far away from the front can be used to find a low rank description of ϕ which accelerates the convergence of $\|q - f(\phi_n)\|$, when truncating ϕ after the nth mode. We demonstrate the ability of this new ansatz for a 2D propagating flame with a moving front.

1 Introduction

Nowadays combustion systems are studied by simulating the reactive Navier Stokes equations with billions of degrees of freedom. The simulations are numerically expensive, because computational resources scale with the number of degrees of freedom. Therefore, model order reduction (MOR) techniques are desired to reduce the number of relevant parameters, which describe the system. Unfortunately, classical MOR techniques fail for these systems. We aim for an improvement in this report.

The author "Julius Reiss" is the Speaker.

P. Krah · M. Sroka · J. Reiss (✉)
Technische Universität Berlin, Institut für Strömungsmechanik und Technische Akustik (TUB), Berlin, Germany
e-mail: krah@tnt.tu-berlin.de; sroka@tnt.tu-berlin.de; reiss@tnt.tu-berlin.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_79

Our method follows a data driven approach, where a set of N snapshots, $\{q(\boldsymbol{x}, t_i)\}_{i=1,\dots N}$, gathered during a numerical simulation, is used to generate a reduced order model (ROM). Here, most ROMs rely on separation of variables:

$$q(\boldsymbol{x}, t) \approx \sum_{i=1}^{n} a_i(t)\psi_i(\boldsymbol{x}) \tag{1}$$

in which the initial high fidelity field $q(\boldsymbol{x}, t)$ is represented by a set of basis functions ψ_i and their amplitudes a_i. Based on Eq. (1), Petrov Galerkin and Galerkin methods (see for a review [1]) project the original dynamics on a n-dimensional subspace spanned by the basis ψ_i. However, the approximation error of the produced ROM crucially depends on the error made in Eq. (1). Unfortunately, in combustion systems transport dominated phenomena like moving flame kernels with sharp gradients or traveling shock waves critically slow down the convergence of Eq. (1). For instance, this was numerically investigated by Huang et al. [2] for reactive flows and is theoretically quantified with help of the Kolmogorov n-width [3, 4].

In order to handle transport dominated fields with sharp fronts, we propose a nonlinear mapping of the solution manifold:

$$q(\boldsymbol{x}, t) = f(\phi) \qquad \text{s. t. } \phi(\boldsymbol{x}, t) = \sum_{i=1}^{n} \tilde{a}_i(t)\tilde{\psi}_i(\boldsymbol{x})\,. \tag{2}$$

Here, the function $f : \mathbb{R} \to \mathbb{R}$ is simply the wave profile and ϕ describes the shift of the wave profile in time.

This approach shares similar features as [5–8], but overcomes some of the problems associated with these methods. Namely, [5, 7] depends on the choice of the shifts or their a priori knowledge and becomes cumbersome if the topology of the moving object changes. The latter is also the main drawback of [8]. Furthermore, [5, 8] are mainly applied in one spatial dimension. Although [6] does not suffer from the aforementioned drawbacks, it lacks physical interpretation and provides little insight of the underlying structure.

The report is organized as follows: In the first two sections we introduce the basic idea for 1D and 2D advective systems and explain the benefits of our concept. One possible realization is provided in Sect. 4, which is then applied to a real life application in combustion—the reduction of burning hydrogen with complex front dynamics including topological changes (Sect. 5). Finally, we come back to [5, 6, 8, 9] by comparing the concept (Sect. 6).

2 Basic Idea: 1D Example—Advective Transport

To motivate the proposed method we first consider a one dimensional problem. A field $q(x,t)$ defined on $\mathcal{V} = [0, L] \times [0, T]$, with $L, T > 0$ is given by

$$q(x,t) = f(\phi(x,t)), \tag{3}$$

where f is a non-linear function and the auxiliary field $\phi(x,t) = x - \Delta(t)$. This describes an advective transport with trajectory path $\Delta(t) \colon [0, T] \to \mathbb{R}$, in the most simple example $\Delta(t) = ct$ with transport speed c. The function f is assumed to have a large gradient near $\phi = 0$. In the examples we use

$$f_\lambda(\phi) = (\tanh(\phi/\lambda) + 1)/2\,, \tag{4}$$

where $\lambda > 0$ adjusts the front width. Snapshots of the functions q and ϕ are plotted for increasing time t in Fig. 1, left. The corresponding snapshot matrices X^ϕ, X^q are defined as usual, $X^\alpha_{i,j} = \alpha(x_i, t_j)$. A common approach to find a small representation of X^α is the truncated singular value decomposition (SVD)

$$X^\alpha_n = \sum_{k=1}^{n} \sigma_k \mathbf{u}_k \mathbf{v}_k^\mathsf{T} \mathrel{\widehat{=}} \sum_{k=1}^{n} a_k(t)\psi_k(\mathbf{x})\,, \tag{5}$$

which approximates X^α in the sense that the residuum $\mathcal{R} = X^\alpha - X^\alpha_n$ is minimized. We call the orthonormal basis $\{\psi_k(x_i) = (\mathbf{u}_k)_i\}_{k=1,\ldots,n}$ spatial modes and $\{a_k(t_i) = (\sigma_k \mathbf{v}_k)_i\}_{k=1,\ldots,n}$ temporal coefficients. As known from the Eckart Young Theorem [10], the approximation error $\left\| X^\alpha - X^\alpha_n \right\|_2$[1] is given by the singular value σ_{n+1}, when truncating after the nth spatial mode. The acceptable residuum is typically determined by the target application. A small number n is desired in model order reduction as it governs the numerical cost of the reduced model.

In Fig. 1, right we see the decay of the truncated singular value decomposition Eq. (5) of X^q and X^ϕ is fundamentally different, even though q is created from ϕ and both share the same advective transport. The failure of the SVD or POD to represent sharp transports is well known [3]. In contrast to our example, ϕ can be represented by a linear combination of two functions $\{x, 1\}$. Here, transport of the field ϕ is simply an amplitude change of the constant function. This fact is not new and exploited by Reiss et al. [5] and Rim et al. [8]).

With this ansatz we aim for a generalization of the method to higher spatial dimensions, by representing the movement of a front by an auxiliary field ϕ which is of low rank and a nonlinear mapping f to recover the original field. Thereby a locally one dimensional transport is implied. However, we abstain from a global

[1] Note that for simplicity we will also use $\|\alpha - \tilde{\alpha}\|_2$ for scalar functions $\alpha : \mathcal{V} \to \mathbb{R}$. which we actually calculate as $\left\| X^\alpha - X^{\tilde{\alpha}} \right\|_2$.

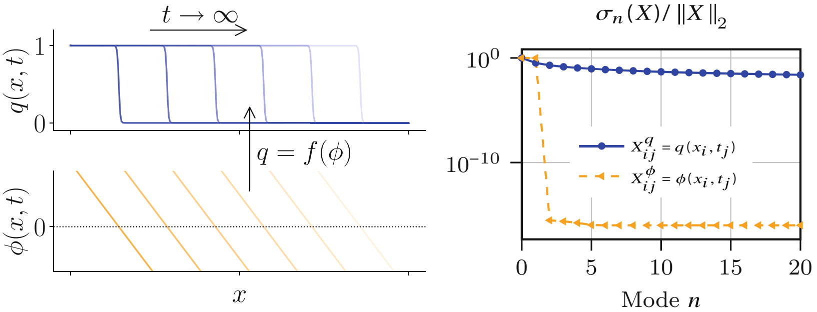


Fig. 1 Transported quantities q and ϕ and singular values of the associated snapshotmatrix X^q and X^ϕ. Both functions share the same transport, since $q = f(\phi)$. However the transport of the sharp front is not well presented by a linear ansatz, Eq. (5), and therefore the singular values decay substantially slower as for the smooth field ϕ

transport map between snapshots, as this obstructs the application for topology changes.

3 2D Example: Moving Disc

The setting is now illustrated for a two dimensional problem of a disc with radius $R = 0.15L$, moving in a circle in a $[0, L]^2$ domain. The translation of the disc is parametrized by:

$$q(\mathbf{x}, t) = f(\phi(\mathbf{x}, t)) \quad \text{and} \quad \phi(\mathbf{x}, t) = \|\mathbf{x} - \mathbf{x}_0(t)\|_2 - R \tag{6}$$

$$\text{where } \mathbf{x}_0(t) = L \begin{pmatrix} 0.5 + 1/4\cos(2\pi t) \\ 0.5 + 1/4\sin(2\pi t) \end{pmatrix}, \tag{7}$$

and f is again the step function defined in Eq. (4). We sample 60 snapshots in a time interval $[0, 1]$. ϕ is the signed distance function shown in the left of Fig. 2, which shall mimic the ϕ of the one dimensional example close to $\phi \approx 0$.

The original field q is again reconstructed applying the SVD to ϕ from which the approximation $\tilde{q} = f(\phi_n)$ is obtained. Figure 2 shows the comparison between the reconstruction using $f(\phi_n)$ and the naive POD approach using q_n for snapshot $t = 1/4$ with $n = 10$ modes. The results show not only a reduction in the overall error but also that the basic structure of the moving disc is recovered. The latter is already the case for a small number of modes. While the example shows that the concept works well for 2D problems, we show that the choice of ϕ has a strong effect on the possible reduction. By replacing ϕ with a paraboloid $\varphi(\mathbf{x}, t) = \frac{1}{2R}(\|\mathbf{x} - \mathbf{x}_0(t)\|_2^2 - R^2)$ in Eq. (6), it is possible to obtain a much smaller error with less spatial modes. A one dimensional profile of the two functions ϕ, φ is plotted in the left of Fig. 3. Note

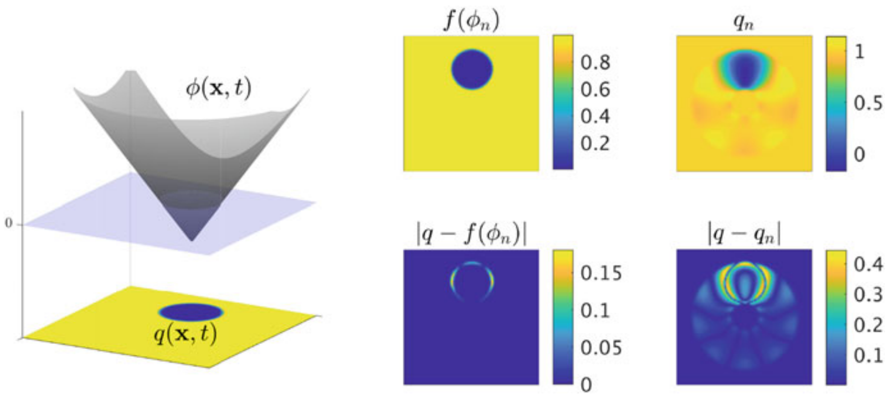


Fig. 2 Left: visualization of the signed distance function. Right: Reconstruction of q with $n = 10$ modes

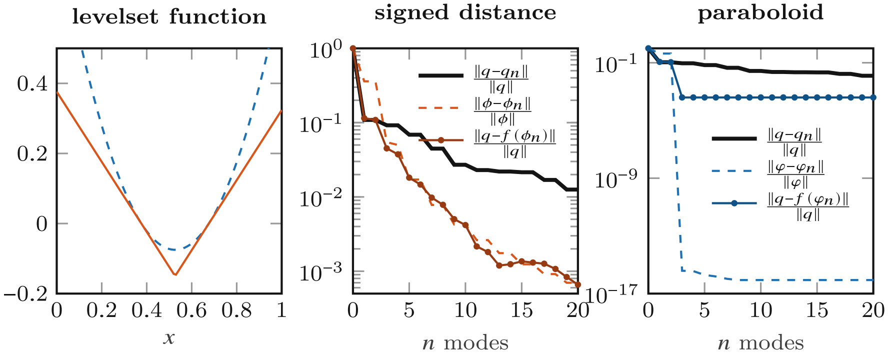


Fig. 3 Left:Graphical visualization of the smoothed signed distance function ϕ defined in Eq. (6) and the paraboloid $\varphi(\mathbf{x},t) = \frac{1}{2R}(\|\mathbf{x} - \mathbf{x}_0(t)\|_2^2 - R^2)$ Left: A slice of ϕ and φ at $t = 0$ and $\mathbf{x} = (x, y_0)$, $x \in [0, L]$. Comparison of the different truncation errors for the signed distance (center) and paraboloid (right)

that ϕ and φ have the same zero-level and their gradients are identical at the zero-level for all times t. Therefore $f(\varphi) \approx q = f(\phi)$ is a good approximation for small widths λ. In contrast to ϕ, the total error $\|q - f(\varphi_n)\|_2$ is reduced to its minimum with only three modes, because φ can be represented by $\{x^2 + y^2 + R^2, x, y\}$. This is shown in the second and third column of Fig. 3. From the ansatz $q \approx f(\varphi)$ we can deduce two different errors contributing to the total error:

$$\|q - f(\varphi_n)\| \leq \underbrace{\|q - f(\varphi)\|}_{=\Delta f} + \left\| f'(\varphi)\mathcal{R}\right\| + O\left(\left\|\mathcal{R}^2\right\|\right). \tag{8}$$

The truncation error of the SVD is $\mathcal{R} = \varphi - \varphi_n$ and the approximation error of the data is Δf. Consequently, for vanishing approximation error, the truncation error

$\left\| f'(\varphi)\mathcal{R} \right\| \leq \left\| f'(\varphi) \right\| \sigma_{n+1}$ bounds the total error in the two norm. Therefore, the total error scales with the decaying singular values of ϕ. This behaviour can be seen in Fig. 3. While for ϕ the relative truncation error aligns with the total error (middle), the error of φ in the right plot is dominated by the approximation error $\Delta f \approx 10^{-3}$.

From this example, we see that the ansatz is a good candidate for a low rank optimization of φ. In contrast to areas where $f'(\varphi) \neq 0$ and the field φ has to mimic a signed distance to the front, it can be chosen to minimize the truncation error far away from the zero level where $f'(\phi) = 0$. Additionally, in an optimization procedure one could relax the assumption of constant front width by imposing appropriate conditions on the slope of φ close to the zero level. However this does not lie within the scope of this work.

4 Front Transport Reconstruction (FTR)

Now, we proceed to extent the idea to numerical data. Here, only the field q is known and the auxiliary field ϕ and the front shape function f need to be determined. For this, we assume that the front location can be calculated using threshold search of the relevant variables.

As proof of our concept, we compute ϕ as a two dimensional signed distance function, because it is easy to compute and can be directly interpreted as a local 1D coordinate system. This is a special choice for ϕ which is likely to be sub-optimal as was shown in the previous section. The zero-level curve C_0 of ϕ is determined by a threshold search with threshold q_{C_0}. The discrete contour line C_0 was sampled at points where q had the value q_{C_0} on any vertical or horizontal gridline of our computational mesh. A linear interpolation of q between the grid points is used to determine the crossing. The distance $d_{C_0}(\boldsymbol{x})$ is calculated as the minimal distance to all sections of this curve, assumed to be linear between two points. The sign of ϕ is negative if $q(\boldsymbol{x}) < q_{C_0}$, and positive otherwise. With the described procedure we determine the signed distance function $\phi(\mathbf{x}, t_i)$ for every $t_i = i\Delta t$, $i = 1, \ldots, N$.

At this point a value of ϕ and q is available at every grid point from which the front shape function f is to be determined such that $q = f(\phi)$. This is complicated by the fact that such relation is approximate and only discrete values are available. From the computed signed distance function we choose all grid points $\hat{\phi}_l = \phi(x_{i_l}, y_{j_l}, t_{i_l})$ with $\left|\hat{\phi}_l\right| \leq \Delta\phi$ on vertical, horizontal or diagonal lines which cross C_0 as support of the samples $\hat{q}_l = q(x_{i_l}, y_{j_l}, t_{k_l})$. The sample vectors $(\hat{\phi}_l, \hat{q}_l)$ are then interpolated on a predefined support set $\phi_1, \ldots, \phi_M$ which is used to find the corresponding interpolated values $f_1, \ldots, f_M$ minimizing the difference between $\hat{q}_l$ and $f(\hat{\phi}_l)$.

5 2D Example: Application to Combustion

In this section we show that the described procedure is capable of reconstructing flow dynamics with inherent two dimensional transport including changing typology, which is difficult for methods building on a mapping between snapshots to remove transports.

The configuration of a flame kernel interacting with a vortex pair mimics turbulence flame interaction. Our data set consists of 40 snapshots derived from a 2D simulation of the reactive Navier Stokes equations. For our purpose we restrict the reconstruction on the normalized mass fraction of hydrogen Y_{H_2}. The simulation was tuned such that a vortex pair moves towards burning H_2 and mixes unburned ($Y_{\mathrm{H}_2} = 1$) with burned gas ($Y_{\mathrm{H}_2} = 0$), such that a small bubble of unburned gas detaches into the burned area. The time evolution is visualized for some selected snapshots in the left of Fig. 4. As seen from Fig. 4, the Y_{H_2} snapshots contain a very interesting structure, in which the front changes along its contour line and even the topology of the line changes—splitting from one curve at $t/\Delta t = 24$ into two curves $t/\Delta t = 29$ and then back to a single curve at $t/\Delta t = 34$.

Applying the described procedure with a threshold of $q_{C_0} = 0.14$, we achieve promising results when comparing our method with a POD approximation in Fig. 4 using 10 modes. For this specific data the threshold q_{C_0} was chosen to be rather small, in order to resolve the tail of the incoming bubble. The overall relative approximation error is decreased by a factor of three. More important, our approximation preserves the physical structure of the data. The POD does not

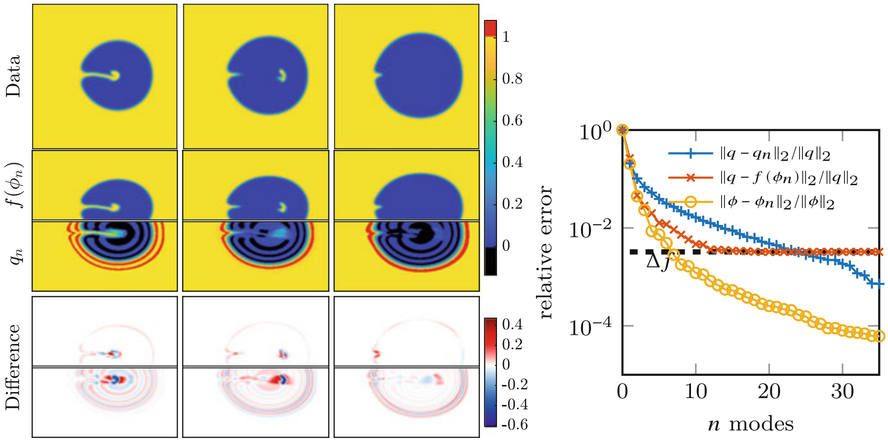


Fig. 4 Comparison of q_n (POD) and $f(\phi_n)$ (levelset method). Left: Direct comparison of the snapshots $t/\Delta t = 21, 26, 31$ with q plotted in the first row, the approximations $f(\phi_n)$, q_n in the second row and the difference between the data and its approximation in the last row. Please note that the images in the lower rows contain only the fractions of the full snapshot that are relevant for our comparison. Right: Relative error in the two norm

respect the allowed physical range of $0 \leq Y_{\mathrm{H}_2} \leq 1$ and shows staircasing, i.e. replaces a front by several fractional fronts. No sensible physical description can be expected from this structure. The new method, in contrast, has well defined fronts and respects the physical range, since f is by construction restricted to the range of the input data q.

6 Discussion and Conclusion

We presented a concept for modal decomposition of transported fronts. It builds on representing the original field q by an auxiliary field ϕ and a non-linear function f in such a way that the ϕ has a better low rank description than q. In a numerical example, ϕ was taken as a signed distance function with the front as zero level and f describing the front shape. It is evident, that this choice is in general not optimal, since moving kinks in ϕ yield a slow decay of singular values.

This approach can be interpreted as embedding a local one dimensional coordinate into a multidimensional domain, orthogonal to the front. A transport in this direction is simply an additive term for ϕ. The time dependent shift to compensate a transport in one dimension is a special case of this approach. This induces a transport map, similar to [5, 8], with the important difference that there is a local but no global one to one mapping, by which topology changes are permitted. A different perspective is the comparison with neural networks, as used for model order reduction in [6]. This linear combination to construct a low rank representation ϕ with the application of a non linear function can be seen as a one layer network with a special activation function f. A recent work uses level sets to handle geometry changes in which shares some technical aspects with the current work [7].

For full practical applicability, improvements are needed but near at hand. To improve the approximation error f and ϕ should be minimized based on Eq. (8) and a more general ansatz should be used to allow a changing front shape.

Acknowledgments This work is supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—384950143/ GRK2433 and Project 200291049—SFB 1029.

References

1. C. W. Rowley, T. Colonius and R. M. Murray, *Model reduction for compressible flows using pod and galerkin projection*, *Physica D: Nonlinear Phenomena* **189** (2004) 115–129.
2. C. Huang, K. Duraisamy and C. Merkle, *Challenges in reduced order modeling of reacting flows*, in *2018 Joint Propulsion Conference*, p. 4675, 2018.
3. M. Ohlberger and S. Rave, *Reduced basis methods: Success, limitations and future challenges*, *Proceedings of the Conference Algoritmy* (2016) 1–12.

4. C. Greif and K. Urban, *Decay of the kolmogorov n-width for wave problems*, *Applied Mathematics Letters* **96** (2019) 216–222.
5. J. Reiss, P. Schulze, J. Sesterhenn and V. Mehrmann, *The shifted proper orthogonal decomposition: A mode decomposition for multiple transport phenomena*, *SIAM Journal on Scientific Computing* **40** (2018) A1322–A1344.
6. K. Lee and K. T. Carlberg, *Model reduction of dynamical systems on nonlinear manifolds using deep convolutional autoencoders*, *Journal of Computational Physics* (2019) 108973.
7. E. N. Karatzas, F. Ballarin and G. Rozza, *Projection-based reduced order models for a cut finite element method in parametrized domains*, *Computers and Mathematics with Applications* **79** (2020) 833–851.
8. D. Rim, S. Moe and R. J. LeVeque, *Transport reversal for model reduction of hyperbolic partial differential equations*, *SIAM/ASA Journal on Uncertainty Quantification* **6** (2018) 118–150.
9. G. Welper, *Interpolation of functions with parameter dependent jumps by transformed snapshots*, *SIAM Journal on Scientific Computing* **39** (2017) A1225–A1250.
10. C. Eckart and G. Young, *The approximation of one matrix by another of lower rank*, *Psychometrika* **1** (1936) 211–218.

Modelling of the Influence of Vegetative Barrier on Particulate Matter Concentration Using OpenFOAM

Hynek Řezníček

Abstract High concentration of atmospheric dust is a well known risk factor to human health. Vegetative barriers are one of the most popular ways how to substantially reduce the high pollution concentration. Correct modelling of air flow inside the Atmospheric Boundary Layer (ABL) is essential to accurately predict concentration of the passive scalar (dust). The question whether the CFD toolbox OpenFOAM is capable of modelling of this type of problems is tested in the contribution. The results obtained from OpenFOAM were compared simultaneously with the experimental data and the CFD results of the program Atifes, developed at CTU for ABL simulations. It is shown that the recommended setting of OpenFOAM's atmospheric library has several limitations. Special attention is paid to the different wall functions used in both solvers and the differences are discussed.

1 Introduction

The pollution produced by industry has a negative effect on human health in surrounding inhabited areas, see e.g. [8]. The vegetative barriers are one of the common protective measures. For effective design and inhabitants protection we need to understand how they decrease the dust concentration [5]. CFD modelling helps to answer both cases. Because dust particles are abducted by the air flow as a passive contaminant, the essential part is proper capture of the flow.

The Atifes software was developed for simulations in ABL [2, 11] and it was successfully validated [10, 14] at CTU. Since the CFD community used the open-source OpenFOAM software also for such kind of simulation, we were curious how reliable the results are. Therefore the OpenFoam atmBoundaryLayer library was employed and tested for two cases: Full-scale unperturbed ABL problem [7] and

H. Řezníček (✉)
Czech Technical University in Prague, Faculty of Mechanical Engineering, dep. of Technical Mathematics, Prague, Czech Republic
e-mail: hynek.reznicek@fs.cvut.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_80

Flow in and around the forest canopy [3]. The results were compared to the results obtained by Atifes and to the field test experiment.

The atmBoundaryLayer library in simpleFoam was employed for simulation of the ABL with its implemented wall function analogous to [1]. The problem with the wall functions in the ABL was discussed in more detailed [12] and [7].

Furthermore some limitations of OpenFoam's atmBoundaryLayer library are shown and some future recommendations for its modifications are given.

2 Mathematical Model

Fluid Flow is described by the incompressible Reynolds-averaged Navier-Stokes (RANS) equations. The pressure p and the potential temperature θ are split into background component in hydrostatic balance and fluctuations, $p = p_0 + p'$ and $\theta = \theta_0 + \theta'$. The Boussinesq approximation is employed. The resulting set of equations reads

$$\nabla \cdot \boldsymbol{u} = 0, \tag{1}$$

$$\frac{\partial \boldsymbol{u}}{\partial t} + (\boldsymbol{u} \cdot \nabla)\boldsymbol{u} - \nabla \cdot (\nu_T \nabla \boldsymbol{u}) = -\nabla \left(\frac{p'}{\rho_0}\right) + \boldsymbol{g} + \boldsymbol{S_u}, \tag{2}$$

$$\frac{\partial \theta'}{\partial t} + \nabla \cdot (\theta' \boldsymbol{u}) = \nabla \left(\frac{\nu_T}{\mathrm{Pr}} \nabla \theta'\right), \tag{3}$$

vector $\boldsymbol{u}$ is averaged velocity, the constant ρ_0 denotes the air density at the ground level, ν_T represents the turbulent kinematic viscosity (laminar is neglected). Vector $\boldsymbol{g} = (0, 0, -g\frac{\theta'}{\theta_0})$ denotes the gravity term, $\boldsymbol{S_u}$ is the momentum sink due to the vegetation and Pr is the constant Prandtl number which equals $\mathrm{Pr} = 0.75$.

Turbulence is modelled by the standard $k - \epsilon$ model for the turbulent kinetic energy (k) and its dissipation rate (ϵ). The standard equations

$$\frac{\partial \rho k}{\partial t} + \underbrace{\nabla \cdot (\rho k \boldsymbol{u})}_{\text{Convection}} = \underbrace{\nabla \cdot \left(\left(\mu_L + \frac{\mu_T}{\sigma_k}\right) \nabla k\right)}_{\text{Diffusion}} + \underbrace{P_k}_{\text{Production}} - \underbrace{\rho \epsilon}_{\text{Dissipation}} + \underbrace{\rho S_k}_{\text{Sources}}, \tag{4}$$

$$\frac{\partial \rho \epsilon}{\partial t} + \underbrace{\nabla \cdot (\rho \epsilon \boldsymbol{u})}_{\text{Convection}} = \underbrace{\nabla \cdot \left(\left(\mu_L + \frac{\mu_T}{\sigma_\epsilon}\right) \nabla \epsilon\right)}_{\text{Diffusion}} + \underbrace{C_{\epsilon 1} \frac{\epsilon}{k} P_k}_{\text{Production}} - \underbrace{C_{\epsilon 2} \rho \frac{\epsilon^2}{k}}_{\text{Dissipation}} + \underbrace{\rho S_\epsilon}_{\text{Sources}}, \tag{5}$$

are completed with source terms acting inside the vegetation (S_k and S_ϵ on the RHS). Model for these sources is described below. According to [6] turbulent model constants were set: $\sigma_k = 1.0$, $\sigma_\epsilon = 1.167$, $C_{\epsilon_1} = 1.44$, $C_{\epsilon_2} = 1.92$ and $C_\mu = 0.09$.

Effects of Vegetation on the flow field consist of two processes. The first one, a sink of momentum inside the vegetation, is given in Eq. (2) by term $\boldsymbol{S}_u$, modelled as $\boldsymbol{S}_u = -C_d L_{AD}|\boldsymbol{u}|\boldsymbol{u}$, where $C_d = 0.3$ represents the drag coefficient [14] and L_{AD} is a function describing the vertical profile of the leaf area density. The vegetation is considered as a horizontally homogeneous block and L_{AD} represents a foliage surface area per unit volume [13].

The second process describes the influence of the vegetation on the turbulence. The turbulent source terms can be modelled according to the given L_{AD} profile as

$$S_k = C_d L_{AD}(\beta_p|\boldsymbol{u}|^3 - \beta_d|\boldsymbol{u}|k), \quad S_\epsilon = C_{\epsilon 4}\frac{\epsilon}{k}S_k, \tag{6}$$

where the constants are chosen as $\beta_p = 1.0$, $\beta_d = 5.1$ and $C_{\epsilon_4} = 0.9$, according to [6].

The model from [9] is adopted in the study. The additional terms describing the vegetation were easily implemented in OpenFoam and they were the same in both solvers.

3 Numerical Solvers

As mentioned earlier two different numerical solvers were employed and compared. The numerical schemes are summarized in the Table 1. The recommended setting for atmospherical calculations in OpenFoam from the source [1] was used.

Table 1 Comparison of numerical solvers

OpenFoam	Atifes
SimpleFoam solver (pressure correction method).	Artificial compressibility method (solved in dual time).
LinearUpwind scheme for u (with grad-based explicit correction).	FVM solver based on AUSM$^+$up with linear reconstruction.
LimitedLinear scheme for turbulent quantities.	Venkatakrishnan limiter.
Viscous terms are solved by linear diffusion scheme used on limited corrected gradient.	Viscous terms on dual (diamond type) mesh.
PCG solver (GAMG) for p and GaussSeidel smoothSolver for other quantities	Fully implicit BDF2 in time, Jacobian Free Newton-Krylov method for non-linear terms.
Standard relaxation factors.	GMRES solver (with ILU Preconditioner).

The atmBoundaryLayer library works fine with simpleFoam, therefore the simpleFoam solver is employed, neutral stratification is assumed (same as in the test cases) and Eq. (3) is omitted. The solver buoyantBoussinesqSimpleFoam for the stratified flow was also tried but it doesn't work well with the mentioned library. When the boundary conditions were prescribed manually (via code-stream utility), the computation was very slow and therefore omitted here. The implementation and correction OpenFoam atmospheric library for stratified solver is left for future work.

3.1 Wall Functions

Different wall functions for the bottom boundary were used. In OpenFoam the standard wall function developed for the flat plate boundary layer is recommended [1] for ABL calculations:

$$u_+ = \ln(Ez_+)/\kappa, \qquad\qquad k_+ = C_\mu \ln(z_+)/\kappa, \tag{7}$$

where the law of the wall for boundary layer variables was used ($u = u_* u_+,\ z_+ = \frac{zu_*}{\nu}$). u_* is a friction velocity and a constant E is chosen as 9.8.

On the other hand in Atifes the wall function for ABL proposed in [7] is implemented. The wall function is based on the assumption of equality between the dissipation and production of turbulent kinetic energy k near the ground. The additional turbulence production $P_{k,w}$ reads as

$$u_+ = \ln\left(\frac{z_+ + z_0}{z_0}\right), \qquad\qquad P_{k,w} = \frac{\kappa C_\mu^{0.25} k^{0.5} u^2}{\rho_0 \ln\left(\frac{z_+ + z_0}{z_0}\right)(z_+ + z_0)}. \tag{8}$$

4 Results

Results obtained by OpenFoam were compared to the results from Atifes software on two test cases. In the first test, the full-scale unperturbed ABL problem tests the capability of the solvers regarding conservation with respect to the prescribed boundary layer velocity and turbulent profiles. The profiles are assumed to be conserved along the whole ABL because the (planetary) boundary layer is already developed (when entering the computational domain), see [12]. (Shortly written it cannot grow around the planet). The second test case, Flow in and around the forest canopy, tests the air flow field distribution near the edge and inside the vegetative barrier. The flow results are compared to the measured data published in [3].

The Following Boundary Conditions (b.c.) were prescribed in both test cases: At the **Inlet** the logarithmic wind profile with u_{ref} value for velocity and Dirichlet b.c. for turbulent quantities and temperature were prescribed. Pressure was extrapolated from the domain. On the **Top** boundary the homog. Neumann b.c. is set for all quantities. For the **Outlet** pressure perturbation was set to $p' = 0$ and for the other quantities the homog. Neumann b.c. was employed. On the **Ground** the no-slip b.c. for velocity, the homog. Neumann b.c. for p' and Dirichlet b.c. for $\Theta = \Theta_0 = 300\,\mathrm{K}$ were utilized. Different wall functions (for turbulent quantities) were used.

The neutral ABL was assumed ($\Theta(z) = \Theta_0$) and the roughness parameter was set to $z_0 = 0.3$. The reference velocity at 10 m height was set as $u_{ref} = 5$ m/s. Minor differences applicable to individual cases are given below.

4.1 Full-Scale Unperturbed ABL

The ABL is assumed to be a fully developed and the simulation should keep the same inlet profile through all the narrow domain. Therefore a fully developed flow was prescribed on the inlet of the 5 km long domain (500 m high) and the evolution of turbulent and flow quantities was observed along the domain. The roughness parameter $z_0 = 1$ and $u_{ref} = 5$ m/s were prescribed, same as in [7], also the velocity on Top boundary was set to free-slip condition.

As can be seen in Fig. 1, Atifes preserves the velocity and k profile quite well, only the small peak of turbulent kinetic energy near the ground is noticeable, but this problem is well documented in the literature [4]. OpenFoam acts differently, the mass flow is slightly displaced to the lower levels which can be caused by different wall functions used for OpenFoam. The distribution of turbulent kinetic energy displayed in Fig. 2 indicated the loss along the domain. It is the correct behaviour for flow over a flat plate for which this kind of wall function was developed, however, it is not appropriate for the simulation of an ABL. The lost of k in the ABL should be compensated by the regeneration of k through the roughness of the surface. In order to preserve the inlet profiles for long domains, the other kind of wall function has to be used.

However in most cases the ground is not perfectly flat and the observed problem is hidden in k generation by the rugged terrain, for example [1] used the OpenFoam wall function successfully for simulations of air flow over a hilly terrain.

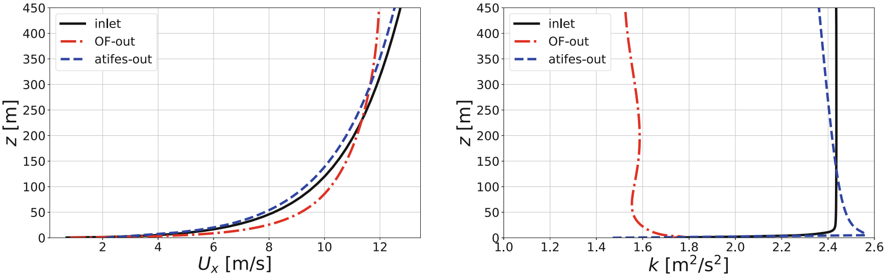


Fig. 1 Different horizontal velocity and turbulent kinetic energy (k) profiles at Outlet

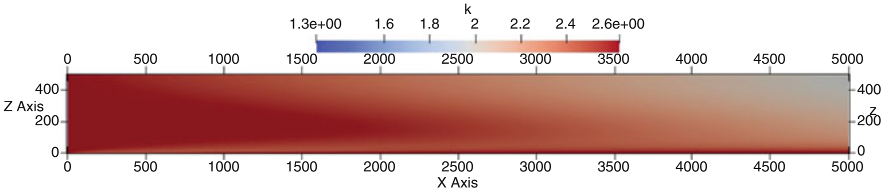


Fig. 2 Turb. kinetic energy drop across the domain for OpenFoam results

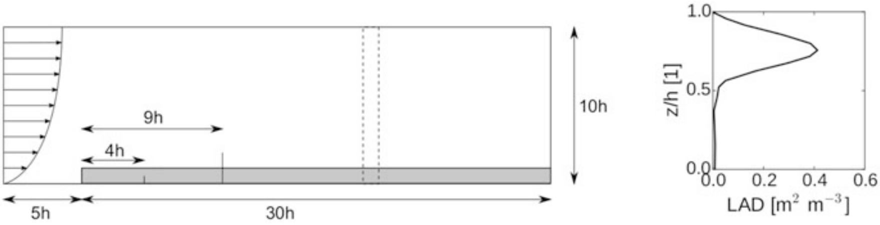


Fig. 3 2D computational domain with indicated points of measurement [14] and L_{AD} profile [3]

4.2 Flow in and Around the Forest Canopy

Dupont et al. [3] measured the flow field in and around a sufficiently long homogeneous forest composed from maritime pines. Average height of the trees with a dense crown layer was $h = 22$ m. The situation is schematically sketched in Fig. 3 where the vertical lines of the measurements are highlighted. The L_{AD} profile used in the simulations is taken from [3] and is plotted also in Fig. 3.

Both CFD solvers and their results were compared to measurements. The main effects of the vegetative barrier are the deflection of air flow near the forest edge and deceleration inside the forest. They are well captured by OpenFoam and Atifes, see Fig. 4. However the detailed view on the results for the air flow inside the forest reveal differences between the solvers.

The comparison for the 9 h-distance from the forest edge is plotted in Fig. 5 and shows still very good agreement of both simulations with the experiment.

Sadly the same agreement is not achieved when the profile is plotted in the cut far from the forest edge as can be seen in Fig. 6. The horizontal velocities near the ground are underestimated with OpenFoam results. It is probably caused by false prediction of recirculation zone which can be identified in a distance of approximately 27 h from the forest edge, see Fig. 4. The measurements and the Atifes results, both in very good agreement, do not indicate any presence of recirculation zone.

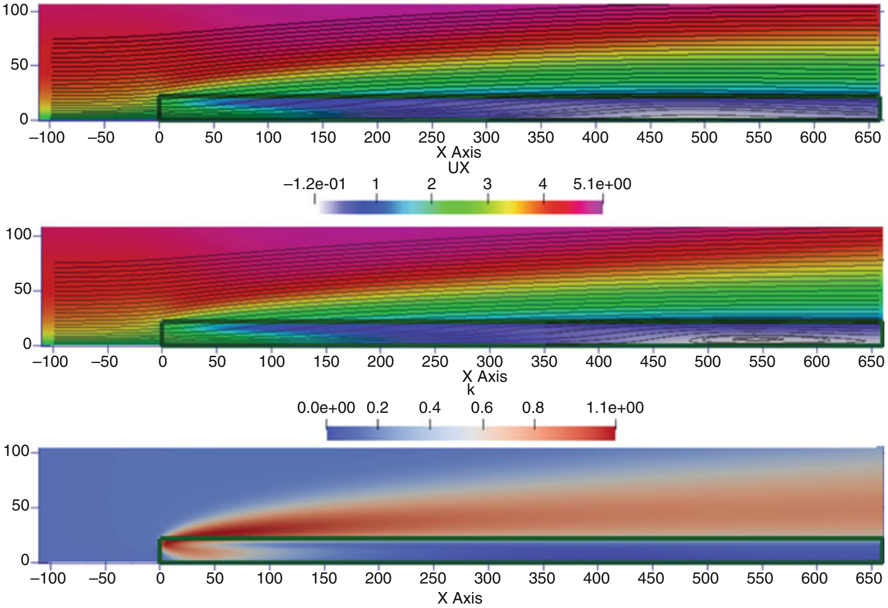


Fig. 4 Streamlines for horizontal velocity in the domain for Atifes *(top)* and OpenFoam *(middle)* and the distribution of *k* for OpenFoam *(bottom)*. The forest area is marked with green rectangle

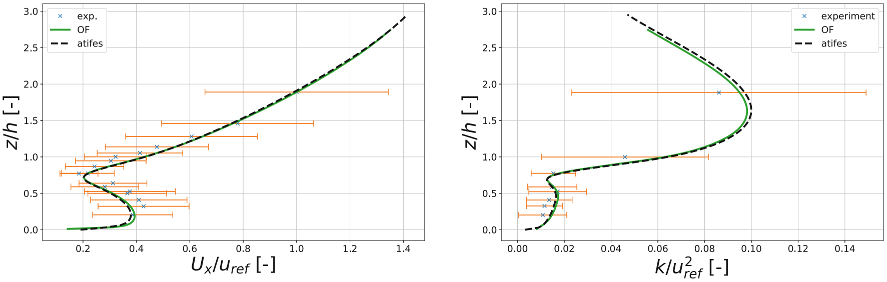


Fig. 5 Comparison for normalized horizontal velocity and *k* profiles for 9-h distance from the forest edge. The results were normalized by the reference velocity in 2 h height

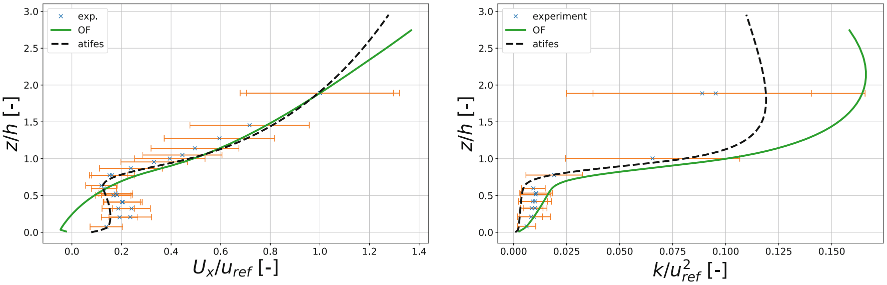


Fig. 6 Comparison for normalized horizontal velocity and *k* profiles far from the forest edge

5 Conclusions

OpenFoam with standard settings recommended for atmospheric boundary layer was tested on two selected cases. Simulations of the fully developed ABL demonstrate decrease of the turbulent kinetic energy along the domain in OpenFoam results which is not valid for flow inside the ABL. The problem could be removed by the implementation of an appropriate wall function to OpenFoam.

In the case of the Flow in and around the forest canopy good agreement is obtained for the air flow closed to the forest edge. The significant differences are obtained deeper in the forest. OpenFoam predicts a false recirculation zone and underestimates the horizontal velocity near the ground.

To simply summarize text above, the simulation of ABL with vegetation by OpenFoam can produce un-physical predictions and therefore caution is needed when interpreting them.

Acknowledgments This work was supported by grant No. *SGS19/154/OHK2/3T/12* of Grant Agency of the CTU in Prague.

References

1. M. Balogh, A. Parente, and C. Benocci, *RANS simulation of ABL flow over complex terrains applying an enhanced k-ε model and wall function formulation: Implementation and comparison for fluent and openfoam*, J. Wind Engineering and Industrial Aerodyn. **104** (2012).
2. L. Beneš, *Comparison the influence of conifer and deciduous trees on dust concentration emitted from low-lying highway by CFD*, submited.
3. S. Dupont, J.-M. Bonnefond, et al., *Long-distance edge effects in a pine forest with a deep and sparse trunk space: in situ and numerical experiments*, Agricultural and forest meteorology **151**:3 (2011), 328–344.
4. D. Hargreaves and N. Wright, *On the use of the k–ε model in commercial cfd software to model the neutral atmospheric boundary layer*, J. of wind engineering and industrial aerodyn. **95**:5 (2007), 355–369.

5. S. Janhäll, *Review on urban vegetation and particle air pollution—deposition and dispersion*, Atmos. Environ. **105** (2015), 130–137.
6. G. Katul et al., *One- and two-equation models for canopy turbulence*, Bound. Layer Meteor. **113** (2004), 81–109.
7. A. Parente et al., *Improved k–ε model and wall function formulation for the rans simulation of ABL flows*, J. of wind engineering and industrial aerodyn. **99**:4 (2011), 267–278.
8. A. K. Patra, S. Gautam, and P. Kumar, *Emissions and human health impact of particulate matter from surface mining operation—a review*, Environmental Technology and Innovation **5** (2016), 233–249.
9. A. Petroff et al., *Aerosol dry deposition on vegetative canopies*, Atmos. Environ. **42** (2008).
10. H. Řezníček and L. Beneš, *Impact of vegetation on dustiness produced by surface coal mine in north bohemia*, Computers and Mathematics with Applications **78**:9 (2019), 3175–3186.
11. V. Šíp and L. Beneš, *CFD optimization of a vegetative barrier*, Numerical Mathematics and Advanced Applications ENUMATH 2015 (2015), 471–479.
12. O. Temel and J. vanBeeck, *Two-equation eddy viscosity models based on the Monin–Obukhov similarity theory*, Applied Mathematical Modelling **42** (2017), 1–16.
13. A. Tiwary, H. Morvanb, and J. Colls, *Modelling the size-dependent collection efficiency of hedgerows for ambient aerosols*, Aerosol Science **37** (2005), 990–1015.
14. V. Šíp, *Numerical simulations of microscale atmospheric flows and pollution dispersion*, Ph.D. thesis, FME, Czech Technical Uni in Prague, 2017.

Logistic Regression for Prospectivity Modeling

Samuel Kost, Oliver Rheinbach, and Helmut Schaeben

Abstract Regression models are often employed in prospectivity modeling for the targeting of resources. Logistic regression has a well understood statistical foundation and uses an explicit model from which knowledge can be gained about the underlying phenomenon. In this paper, a model selection procedure based on logistic regression enhanced with nonlinearities is proposed. The method is designed to help the researcher in the model building process and can also be used as preprocessing step for other machine learning algorithms such as neural networks.

1 Introduction

The objective of prospectivity modeling in geoscience is to predict locations for which the estimated conditional probability of the occurrence of a target event, e.g., a resource, is maximal; see [15] for an overview of current methods. Logistic regression is a widely used tool in statistics for the classification of a two-class dependent variable. It is a parametric method, i.e., an explicit model is used for the classification. From the computed model, information about the underlying structure of the problem can be gained. Logistic regression has been extensively studied, e.g., in [2].

In potential modeling events are typically rare. Here, logistic regression can underestimate the probabilities of the rare events, e.g., [3]. As a remedy, undersampling [3] can be used, i.e., by taking a subset of the majority events.

S. Kost · O. Rheinbach (✉)
Technische Universität Bergakademie Freiberg, Institut für Numerische Mathematik und Optimierung, Freiberg, Germany
e-mail: oliver.rheinbach@math.tu-freiberg.de

H. Schaeben
Technische Universität Bergakademie Freiberg, Institut für Geophysik und Geoinformatik, Freiberg, Germany

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_81

2 Logistic Regression

Logistic regression uses the logit function

$$\operatorname{logit}(P(\mathbf{y}|\mathsf{X})) = \boldsymbol{\eta} = \ln \frac{P(\mathbf{y}|\mathsf{X})}{1 - P(\mathbf{y}|\mathsf{X})}. \tag{1}$$

Here, the matrix $\mathsf{X} \in \mathbb{R}^{n \times m+1}$ contains n data points consisting of m covariables (or predictor variables) each. The first column in X consist of ones and incorporates the intercept. Similarly, the vector $\mathbf{y} \in \{0, 1\}^n$ contains all target events with 1 indicating the presence of the target. An experiment $\mathbf{x}_i$ in X can be seen as a Bernoulli trial with the conditional expectation $E(y|\mathbf{x}_i) = \mu(\eta_i)$ with $\mathbf{x}_i$ denoting the i-th row of the data matrix X and $\eta_i = \boldsymbol{\beta}^T \mathbf{x}_i$ being the linear predictor. In order to model the relation of an experiment $\mathbf{x}_i$ and the expected outcome $\mu(\eta_i)$, one uses the logistic function, given as

$$\mu(\eta_i) = \frac{\exp(\boldsymbol{\beta}^T \mathbf{x}_i)}{1 + \exp(\boldsymbol{\beta}^T \mathbf{x}_i)}, \tag{2}$$

where $\boldsymbol{\beta} \in \mathbb{R}^{m+1}$ is the vector of parameters for the intercept and the variables. The logistic function is the inverse of the logit. The parameters are estimated by maximizing the logarithm of the maximum likelihood function,

$$\max_{\boldsymbol{\beta}} \ln \mathbb{L}(\boldsymbol{\beta}) = \max_{\boldsymbol{\beta}} \sum_{i=1}^{n} \big(y_i \ln(\mu(\boldsymbol{\beta}^T \mathbf{x}_i)) + (1 - y_i) \ln(1 - \mu(\boldsymbol{\beta}^T \mathbf{x}_i))\big). \tag{3}$$

The solution $\hat{\boldsymbol{\beta}}$ is called maximum likelihood estimate (MLE). Preferred methods for the computation of the MLE are conjugate gradients (as in [9, 11]) and truncated Newton methods using conjugate gradients (i.e., Newton-Krylov methods); see [6, 7].

Logistic regression assumes that there is little or no multicollinearity among the covariables, i.e., that they are not correlated. Otherwise the parameters of the associated covariables might grow to infinity [6]. As a remedy, one uses, e.g., L^2-regularization, also known as ridge regression,

$$\max_{\boldsymbol{\beta}} \quad \sum_{i=1}^{n} \big(y_i \ln(\mu(\boldsymbol{\beta}^T \mathbf{x}_i) + (1 - y_i) \ln(1 - \mu(\boldsymbol{\beta}^T \mathbf{x}_i))\big) - \frac{\lambda}{2} ||\boldsymbol{\beta}||_2^2 . \tag{4}$$

This prevents the parameters from growing to infinity, and furthermore it is equivalent to assuming a Gaussian prior for the parameters $\boldsymbol{\beta}$ [12].

3 General Logistic Regression with Endogenous Sampling

Weighted Likelihood Function When dealing with rare events under-sampling can be applied. This induces a class bias, affecting logistic regression. Since the goal is to gain knowledge about the complete population, not the balanced sample, corrections have to be applied, i.e., using a modified likelihood. Denoting $\mu_i = \mu(\boldsymbol{\beta}^T \mathbf{x}_i)$, we use a weighted likelihood proposed by Manski and Lerman [10],

$$\ln \mathbb{L}(\boldsymbol{\beta}) = \sum_{i=1}^{n} w_i (y_i \ln \mu_i + (1 - y_i) \ln(1 - \mu_i)), \tag{5}$$

where $w_i = \frac{\tau}{\bar{y}} y_i + \frac{(1-\tau)}{(1-\bar{y})}(1 - y_i)$ and where τ and $\bar{y}$ are the proportion of the positive events in the population and in the sample, respectively. A L^2-regularization is added again for the reasons mentioned above. Formulated as a minimization problem this results in

$$\min_{\boldsymbol{\beta}} f(\boldsymbol{\beta}) = -\ln \mathbb{L}(\boldsymbol{\beta}) + \frac{\lambda}{2} ||\boldsymbol{\beta}||_2^2. \tag{6}$$

The MLE $\hat{\boldsymbol{\beta}}$ is determined using the truncated iteratively re-weighted least squares (TR-IRLS) approach [6]. This is a truncated Newton method where the gradient and the Hessian are obtained by differentiating the objective function (6) with respect to $\boldsymbol{\beta}$. The gradient written in matrix form is $\nabla f(\boldsymbol{\beta}) = -\mathsf{X}^T \mathsf{W}(\mathbf{y} - \boldsymbol{\mu}) + \lambda \boldsymbol{\beta}$, where $\mathsf{W} = \text{diag}(w_i)$, and the Hessian is $\nabla^2 f(\boldsymbol{\beta}) = \mathsf{X}^T \mathsf{D} \mathsf{X} + \lambda \mathsf{I}$, where $\mathsf{D} = \text{diag}(w_i \mu_i (1 - \mu_i))$. After a reformulation of the Newton-Raphson update, the new iterate $\boldsymbol{\beta}^{k+1}$ can be calculated as $\boldsymbol{\beta}^{k+1} = (\mathsf{X}^T \mathsf{D} \mathsf{X} + \lambda \mathsf{I})^{-1} \mathsf{X}^T \mathsf{D} \mathbf{z}^k$, where $\mathbf{z}^k = \mathsf{X} \boldsymbol{\beta}^k + \mathsf{D}^{-1}(\mathbf{y} - \boldsymbol{\mu})$ is the adjusted dependent variable. The weighted least squares subproblem $(\mathsf{X}^T \mathsf{D} \mathsf{X} + \lambda \mathsf{I}) \boldsymbol{\beta}^{k+1} = \mathsf{X}^T \mathsf{D} \mathbf{z}^k$ is solved using the linear conjugate gradient method. Typically, the iteration is stopped early.

Conditional Independence Assumption Standard logistic regression models a linear relationship of the covariables in the logit, implicitly assuming conditional independence of the covariables given the target variable. If this assumption is fulfilled, the logistic regression model yields the true conditional probability, hence it is optimal. The same applies for a violation of the conditional independence assumption and an inclusion of the proper interaction terms [13]. The linear predictor then reads

$$\eta = \boldsymbol{\beta}^T \mathbf{x}_i + \sum_{j=m+2}^{\tilde{m}} \beta_j x_k \otimes \cdots \otimes x_l, \qquad i = 1, \ldots, n, \quad k, l \in \{2, \ldots, m+1\}. \tag{7}$$

Note that this model is now nonlinear for the data points but still linear in the parameters $\boldsymbol{\beta}$. This notation describes the inclusion of every interaction term between covariables that are not conditional independent given the target variable. Unfortunately, there is no easy way to prove conditional independence [14]. So in general, one does not know which interaction terms need to be included. Furthermore, the true model might also include other types of nonlinearities such as quadratic terms, logarithms, etc. of covariables.

4 Algorithm

Let us assume an underlying ground truth which can be approximately described by the variables collected and is not too complex. We start including only quadratic terms, two-fold, three-fold interactions of covariables and interactions of quadratic terms and covariables. The indices of the candidates are stored in the set $\mathcal{I}$. Our model selection process is then performed in two main steps. A coarse filtering of variables where the majority of nonlinearities is discarded and a fine filtering that performs a more careful selection of nonlinearities and covariables.

Coarse Filtering Starting with the large model including all nonlinearities in the set $\mathcal{I}$, we discard the majority of unimportant variables using the Wald test: We test the null hypothesis that a parameter $\hat{\boldsymbol{\beta}}_i, \forall i = m+2, \dots, \tilde{m}$ is zero. Here, original covariables are not considered. From basic statistics it follows that the parameter of interest divided by its standard error yields a standard normal distribution with mean zero and standard deviation of one, i.e.

$$W = \frac{\hat{\boldsymbol{\beta}}}{\widehat{\mathrm{SE}}(\hat{\boldsymbol{\beta}})} \sim \mathcal{N}(0, 1). \tag{8}$$

The two-tailed p-value is then calculated as

$$p = P(|z| > W), \tag{9}$$

where z is a random variable following the standard normal distribution. It is calculated for every variable in the model.

The variable can be assumed to be significant if its p-value is smaller than a threshold α, i.e. $p < \alpha$. Otherwise it is excluded from the model. In order to use the Wald test in our setting of rare events an adjustment for the covariance matrix needs to be applied. As described in [16], if the parameters are heteroscedastic, i.e. their variance is not equal, the calculation of the variance matrix is inefficient. This is also the case when the model is misspecified. A correction is proposed for linear models. In the case of generalized linear models, the authors in [4] describe the corrected estimate for the variance, also called sandwich estimate,

$$V(\boldsymbol{\beta}) = \mathsf{P}^{-1}\mathsf{M}\mathsf{P}^{-1}, \tag{10}$$

where $\mathsf{P} = \mathbb{E}(\nabla^2 f(\boldsymbol{\beta}))$ is the expectation of the Hessian and $\mathsf{M} = \mathbb{V}(\nabla f(\boldsymbol{\beta}))$ is the covariance of the gradient. The Wald test is known to have some shortcomings: Every parameter will become significant if the sample sizes increases while the current model maintains the same [8]. Furthermore, the authors in [1] reported that, even when the coefficient was significant, the Wald test often failed to reject the null hypothesis. Despite these shortcomings we use the Wald test for the coarse filtering, because it is possible to reject several variables at once without computing the MLE for every case. We furthermore apply an adaptive decrease of the threshold α in every iteration until we reach an a priori defined minimal threshold.

Fine Filtering The fine selection of our procedure uses the Bayesian information criterion (BIC) to decide whether a variable should be excluded from the model or not. The BIC is defined as

$$\text{BIC} = -2\ln(\widehat{\mathbb{L}}) + \ln(n)k, \tag{11}$$

where k is the number of variables used to obtain the maximum value $\widehat{\mathbb{L}}$ of the likelihood function $\mathbb{L}$. Let $\mathcal{N}$ be the variables that are detected not to be important for the model. Furthermore, let BIC_j denote the BIC value of the current model without variable j. We then calculate the difference between the BIC including all current variables, BIC_0, and BIC_j for all variables j, i.e. $d_j = \text{BIC}_0 - \text{BIC}_j$. Since a smaller BIC value indicates a favorable model, all variables with $d_j > 0$ are considered to be discarded. Discarding only one variable in every iteration is time consuming; instead, we calculate the BIC values for three different models: $\text{BIC}_{1/8}, \text{BIC}_{1/4}, \text{BIC}_{1/2}$ where $1/q$ describes the fraction of the largest d_j that are discarded.

5 Computational Results

Our method is first tested on fabricated data where the true underlying model is known; then it is applied to real world datasets and compared to weighted logistic regression without nonlinearities and to a neural network. Neural networks can be seen as a nested logistic regression model (when using the logistic function as activation). Including the nonlinearities in our logistic regression model can therefore be seen as an approximation of the hidden layer of a neural network. This transforms the data to a higher dimension in which it might be separable by a hyperplane. Transforming it back to the original dimension would then give a nonlinear decision boundary.

Tests on Fabricated Data In the first experiment on fabricated data, the true model can be recovered exactly by our procedure. In the second, the true model cannot be recovered exactly. The dataset consist of 23 covariables taken from a normal distribution. The true model consists of 17 covariables and 6 two-fold interactions

Table 1 Datasets

	Instances	Covariables	Class 0	Class 1	Rarity(%)
ds1.10	26,733	10	25,929	804	3
Ghana	6,091,636	30	6,055,475	36,161	0.6

Table 2 Results

	Model selection		WLR		Neural network	
	AUC-PR	F1	AUC-PR	F1	AUC-PR	F1
ds1.10	0.4656	0.3971	0.2889	0	**0.4854**	**0.4046**
Ghana	0.0864	0.0316	0.0574	0.0344	**0.0919**	**0.1244**

randomly taken from the 23 covariables. For the first setting the procedure was able to detect the true model in every run.

In the second setting for the true model, we take again 17 covariables and add 5 two-fold interactions, 3 three-fold interactions and 3 four-fold interactions. The four-fold interactions are not in the set of nonlinearities, hence cannot be found. However, the procedure finds all possible covariables and interactions and recovers some other two-fold and three-fold interactions to approximate the four-fold interactions. It is possible to rank the variables in the final model. This helps to gain confidence about the variables that are present in the true underlying model.

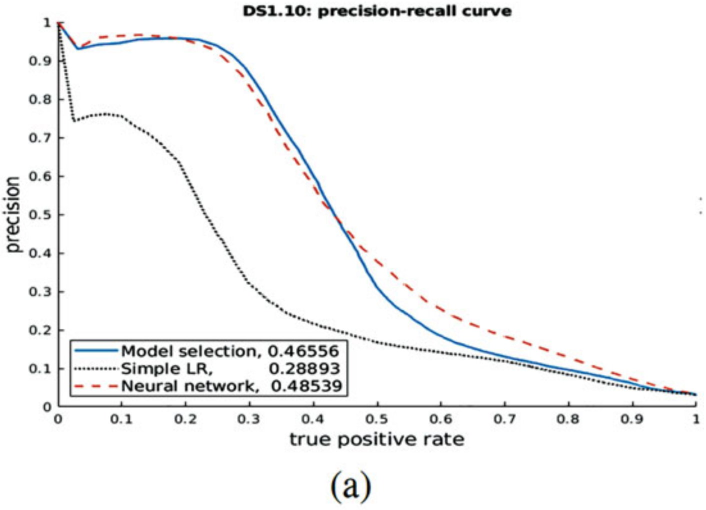
Tests on Benchmark Datasets The ds1.10 is taken from [5], a compressed life sciences dataset. Each row of the original ds1 dataset represents a chemistry or biology experiment, and the output represents the reactivity of the compound observed in the experiment. After performing a principal component analysis only the top 10 principal components are used for ds1.10. The Ghana dataset was provided by Beak Consultants GmbH, Freiberg. The dataset describes geochemical, geological, geophysical and tectonic data from a survey area in Ghana. The target variable is gold. All of the datasets are normalized with mean of zero and standard deviation of one (Table 1).

The model selection procedure is conducted ten times on each dataset. The model that gives the best F1 score is used for comparison with an artificial neural network and the weighted logistic regression (WLR) using all given covariables. The neural network structure consists of one hidden layer with twice as many hidden neurons as covariables available for every dataset. Determining the optimal number of hidden neurons is rather difficult, and we use the same setting as experts in the field of prospectivity modeling. The activation function is the logistic function. The methods are compared mainly in terms of area under the precision-recall curve (AUC-pr) and the F1-score. We further give the TPR and PREC using a threshold of 0.5.

We use bootstrapping to test the methods. From the test set we created 1000 new test sets through drawing with replacement. Table 2 shows the resulting mean of AUCpr and the F1-scores. Table 3 shows the mean of TPR and PREC. Best result in bold. The corresponding precision-recall curves are shown in Fig. 1.

Table 3 Results

	Model selection		WLR		Neural network	
	TPR	PREC	TPR	PREC	TPR	PREC
ds1.10	0.2522	**0.9412**	0	0	**0.2641**	0.9003
Ghana	0.0864	**0.4336**	0.0288	0.0429	**0.0947**	0.1812



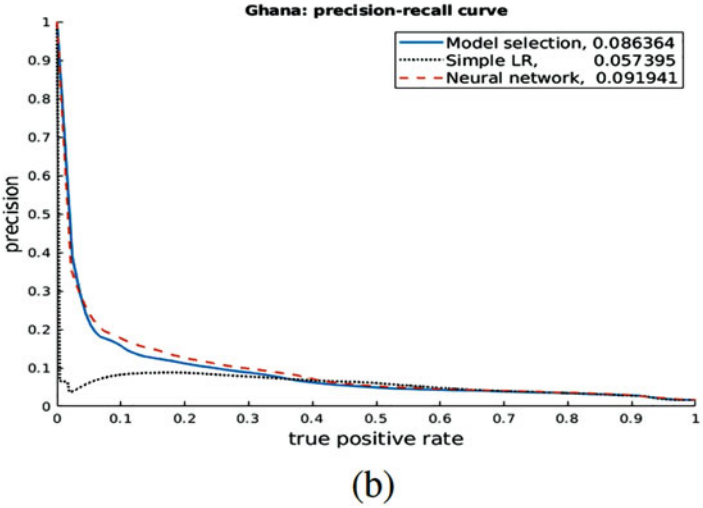


Fig. 1 Precision recall curves for all data sets. (**a**) ds1.10. (**b**) Ghana

Discussion As reported in [5], the ds1.10 dataset triggered strange behavior in the support vector machine used for training and logistic regression did not yield good results either. The same can be seen here. Weighted logistic regression has a very low F1-score and was not able to detect a single positive target event. For our model selection the F1-score is also quite low and the resulting model did highly differ every time. This indicates that the 10 covariables do not describe the target very well. Taking the best model of the 10 runs did increase the prediction compared to weighted logistic regression. Despite the bad overall performance, it did come close to the neural network result. The Ghana dataset has very few positive targets. That makes it very hard to predict the majority of positive events without having too many false positive. Both the model selection and the neural network are not capable of predicting many positive targets as can be seen in the low true positive rate. However, the model selection procedure has a precision of about 43%. This is an important feature for potential modeling, since the drilling at non-target areas is very expensive.

6 Conclusions

We suggest a new combination of automated model selection making use of the debatable Wald test only for the coarse selection and the Bayes' information criterion for a more careful selection. Introducing nonlinearities in the logit enables the final model to give improved predictions. Companies working in the field of prospectivity modeling, e.g., BEAK Consultants GmbH with their software ADVANGEO, have a vital interest in finding appropriate covariables and possible combinations of them. Additional knowledge can, e.g., help to find proper variables to feed to an artificial neural network. Because of the sampling, different runs of the model selection can result in different models. However, the test on artificial data showed that the same true model is found every time if it can be described by the variables in the starting set $\mathcal{I}$. In the case of real world datasets this is almost never the case hence one obtains different models in different runs. This is not uncommon for model selection. Because we minimize the likelihood and the number of variables, there can be several solutions which do not dominate another in terms of Pareto optimality.

Our method gives prediction results that are comparable to the neural networks used here, while, at the same time, the explicit model may increase the insight into the problem.

Acknowledgments We would like to thank BEAK Consultants GmbH and GSD/GGSA for providing the Ghana dataset.

References

1. W.W. Hauck Jr. and A. Donner. Wald's test as applied to hypotheses in logit analysis. *Journal of the American Statistical Association*, 72:851–853, 1977.
2. D.W. Hosmer, S. Lemeshow, and R.X. Sturdivant. *Applied Logistic Regression*. Wiley Series in Probability and Statistics, 3rd edition, 2013.
3. G. King and Z. Langche. Logistic regression in rare events data. *Political Analysis*, 9:137–163, 2001.
4. G. King and M.E. Roberts. How robust standard errors expose methodological problems they do not fix, and what to do about it. *Political Analysis*, 23:159–179, 2014.
5. P. Komarek. http://komarix.org/ac/ds/.
6. P. Komarek and A. Moore. Making logistic regression a core data mining tool: A practical investigation of accuracy, speed, and simplicity. Technical Report CMU-RI-TR-05-27, Carnegie Mellon University, 2005.
7. C-J. Lin, R.C. Weng, and S.S. Keerthi. Trust region newton method for large-scale logistic regression. *Journal of Machine Learning Research*, 9:627–650, 2008.
8. M. Lin, H.C. Lucas, and G. Shmueli. Too big to fail: Large samples and the p-value problem. *Information Systems Research*, 24:906–917, 2013.
9. R. Malouf. A comparison of algorithms for maximum entropy parameter estimation. In *Proc. of the Sixth Conf. on Natural Language Learning*, volume 20, pages 49–55, 2002.
10. C.F. Manski and S.R. Lerman. The estimation of choice probabilities from choice based samples. *Econometrica*, 45:1977–1988, 1977.
11. T. Minka. A comparison of numerical optimizers for logistic regression, 2003. https://www.microsoft.com/en-us/research/publication/comparison-numerical-optimizers-logistic-regression/.
12. J. Rennie. On l2-norm regularization and the Gaussian prior, 2003. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.113.1049.
13. H. Schaeben. A mathematical view of weights-of-evidence, conditional independence, and logistic regression in terms of Markov random fields. *Math. Geosci.*, 46:691–709, 2014.
14. H. Schaeben. Testing joint conditional independence of categorical random variables with a standard log-likelihood ratio test. In *Handbook of Mathematical Geoscience*, chapter 3. SpringerLink, 2018.
15. H. Schaeben, S. Kost, and G. Semmler. Popular raster-based methods of prospectivity modeling and their relationships. *Math. Geosci.*, pages 1–27, 2019.
16. H. Withe. A heteroskedasticitsy-consistent covariance matrix estimator and a direct test for heteroskedasticity. *Econometrica*, 48:817–838, 1980.

A Monge-Ampère Least-Squares Solver for the Design of a Freeform Lens

Lotte B. Romijn, Jan H. M. ten Thije Boonkkamp, and Wilbert L. IJzerman

Abstract Designing freeform optical surfaces that control the redistribution of light from a particular source distribution to a target irradiance poses challenging problems in the field of illumination optics. There exists a wide variety of strategies in academia and industry, and there is an interesting link with optimal transport theory. Many freeform optical design problems can be formulated as a generalized Monge-Ampère equation. In this paper, we consider the design of a single freeform lens that converts the light from an ideal point source into a far-field target. We derive the generalized Monge-Ampère equation and numerically solve it using a generalized least-squares algorithm. The algorithm first computes the optical map and subsequently constructs the optical surface. We show that the numerical algorithm is capable of computing a lens surface that produces a projection of a painting on a screen in the far field.

1 Introduction

The field of illumination optics has surged since the advent of LED light sources and the use of plastic materials for the optical components of LED luminaires. Our aim is to develop accurate methods to compute freeform (i.e., arbitrarily-shaped) optical surfaces that transform the energy emitted by an LED light source to a desired light intensity distribution in the near or far field. In this paper, we compute the shape

L. B. Romijn (✉) · J. H. M. ten Thije Boonkkamp
CASA, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: l.b.romijn@tue.nl; j.h.m.tenthijeboonkkamp@tue.nl

W. L. IJzerman
CASA, Department of Mathematics and Computer Science, Eindhoven University of Technology, Eindhoven, The Netherlands

Signify Research, Eindhoven, The Netherlands
e-mail: wilbert.ijzerman@signify.com

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_82

of a lens surface for an LED light source, approximated as a point in space, and a far-field intensity distribution.

Using *inverse* methods, rather than *forward* methods (e.g., ray tracing), the location of the surface can be described by a function u that satisfies a generalized Monge-Ampère equation, which is a fully nonlinear 2nd order elliptic PDE. The derivation of this equation uses concepts in geometrical optics, optimal transport theory, and energy conservation. Many optical systems can be cast in the framework of optimal transport theory by associating a cost function to the transport of light which can be derived using the principle of Fermat; i.e., the optical path taken by a ray between two given points is stationary with respect to variations of the path.

In this paper, we present a numerical algorithm to solve the generalized Monge-Ampère equation for a freeform lens. The original algorithm for the standard Monge-Ampère equation was first presented in [2, 4, 5] for a parallel source beam. The shape of the optical surface $z = u(\boldsymbol{x})$ corresponds to the solution to the standard Monge-Ampère equation $\det(\mathrm{D}^2 u) = f(x, u, \nabla u)$ with $\mathrm{D}^2 u$ the Hessian matrix of u and f a positive function. Subsequently, arbitrary orthogonal coordinate systems were included by Beltman et al. [1]. Next, the method was extended to optical systems with generalized Monge-Ampère equations, considering parallel incoming and parallel outgoing beams by Yadav et al. [9, 10] and point sources by Romijn et al. [6, 7]. Recently, the method has been further extended to polar stereographic coordinates for the source domain in [7]. In all versions of our algorithm, first the optical mapping $\boldsymbol{m}$ is computed in an iterative procedure, and subsequently the surface u is computed from the mapping.

For an extensive overview of results on existence, uniqueness and regularity of solutions, and of methods available to solve both standard and non-standard Monge-Ampère equations, we refer to [7].

In this paper, in Sect. 2 we first present the optimal transport formulation of the optical system and derive the corresponding generalized Monge-Ampère equation. In Sect. 3, we present a broad outline of the numerical approach in [6, 7, 10]. We present the performance of the algorithm on two test problems in Sect. 4.

2 Mathematical Formulation

We consider a point source, lens with refractive index n and far-field target. The point source, located at the origin O, see Fig. 1, emits beams of light radially outward in the direction $\hat{\boldsymbol{s}} = \hat{\boldsymbol{e}}_r$. The first surface is spherical and the freeform lens surface $\mathcal{L}$ is described by the parametric equation $\mathcal{L} : \boldsymbol{r}(\phi, \theta) = u(\phi, \theta)\hat{\boldsymbol{e}}_r$, where $u(\phi, \theta) > 0$ is the radial parameter that describes the location of the lens surface and (ϕ, θ) are spherical angular coordinates. The intensity of the source is given by $f(\phi, \theta)$ [lm/sr], and the required target intensity in the far field is denoted by $g(\psi, \chi)$ [lm/sr], where (ψ, χ) represents a different set of spherical coordinates with the lens surface as origin approximated as a point at O (far-field approximation).

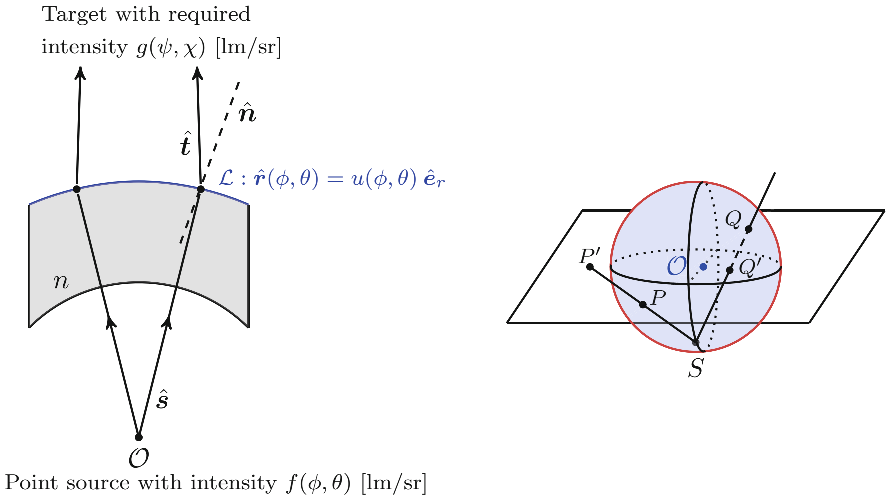

Target with required
intensity $g(\psi,\chi)$ [lm/sr]
$\hat{\boldsymbol{n}}$
$\hat{\boldsymbol{t}}$
$\mathcal{L} : \hat{\boldsymbol{r}}(\phi,\theta) = u(\phi,\theta)\,\hat{\boldsymbol{e}}_r$
n
$\hat{\boldsymbol{s}}$
$\mathcal{O}$
Point source with intensity $f(\phi,\theta)$ [lm/sr]
Q
P'
$\mathcal{O}$
Q'
P
S

Fig. 1 Schematic representations of the freeform lens (left) and stereographic projections for the source domain and target domain (if $\mathcal{O}$ replaced by $\mathcal{L}$) (right). (The point P is projected to P' and the point Q to Q')

The direction of the incident ray $\hat{\boldsymbol{s}} = \hat{\boldsymbol{e}}_r$ is refracted by the second freeform surface $\mathcal{L}$ in the direction $\hat{\boldsymbol{t}}$. We transform coordinates on the source and target domains from spherical to stereographic. We define

$$\boldsymbol{x}(\hat{\boldsymbol{s}}) = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \frac{1}{1+s_3}\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}, \quad \hat{\boldsymbol{s}}(\boldsymbol{x}) = \hat{\boldsymbol{e}}_r = \frac{1}{1+|\boldsymbol{x}|^2}\begin{pmatrix} 2x_1 \\ 2x_2 \\ 1-|\boldsymbol{x}|^2 \end{pmatrix}, \tag{1}$$

as the stereographic projection and corresponding inverse projection of the source domain. The projection is drawn schematically in Fig. 1. The stereographic projection $\boldsymbol{y}(\hat{\boldsymbol{t}})$ and inverse projection $\hat{\boldsymbol{t}}(\boldsymbol{y})$ of the target domain are defined analogously. Transforming to stereographic coordinates, we obtain bounded source and target domains which are circular for a cone-shaped incoming beam. We define our source domain $\mathcal{X}$ as the closed support of $\tilde{f}(\boldsymbol{x}) = f(\phi(\boldsymbol{x}), \theta(\boldsymbol{x}))$, and our target domain $\mathcal{Y}$ as the closed support of $\tilde{g}(\boldsymbol{y}) = g(\psi(\boldsymbol{y}), \chi(\boldsymbol{y}))$. We refer to $\boldsymbol{m} : \mathcal{X} \to \mathcal{Y}$ as the optical map $\boldsymbol{y} = \boldsymbol{m}(\boldsymbol{x})$ from the source set $\mathcal{X}$ to the target set $\mathcal{Y}$.

Using Fermat's principle (via Hamilton's characteristic functions) it was shown in [7] that the location of the lens surface can be described by the relation

$$u_1(\boldsymbol{x}) + u_2(\boldsymbol{y}) = -\log\left(n - 1 + \frac{2|\boldsymbol{x}-\boldsymbol{y}|^2}{(1+|\boldsymbol{x}|^2)(1+|\boldsymbol{y}|^2)}\right) = c(\boldsymbol{x}, \boldsymbol{y}), \tag{2}$$

where $u_1(\boldsymbol{x}) = \log(u(\hat{\boldsymbol{s}}(\boldsymbol{x})))$ describes the shape of the optical surface, and $u_2(\boldsymbol{y})$ is a measure for the optical path length. A unique solution can be found by assuming that u_1 and u_2 are either c-convex or c-concave functions [8, p. 58]. The surfaces u_1 and u_2 are c-convex if

$$u_1(\boldsymbol{x}) = \max\nolimits_{\boldsymbol{y}\in\mathcal{Y}}(c(\boldsymbol{x}, \boldsymbol{y}) - u_2(\boldsymbol{y})), \qquad \forall \boldsymbol{x} \in \mathcal{X}, \tag{3a}$$

$$u_2(\boldsymbol{y}) = \max\nolimits_{\boldsymbol{x}\in\mathcal{X}}(c(\boldsymbol{x}, \boldsymbol{y}) - u_1(\boldsymbol{x})), \qquad \forall \boldsymbol{y} \in \mathcal{Y}, \tag{3b}$$

which we call the maximum solution. The surfaces u_1 and u_2 are c-concave if we replace max by min, referred to as the minimum solution.

For a continuously differentiable function $c \in C^1(\mathcal{X} \times \mathcal{Y})$, the expression for the optical map $\boldsymbol{y} = \boldsymbol{m}(\boldsymbol{x})$ is implicitly given by the critical point of (3b) [8, p. 60], i.e.,

$$\nabla u_1(\boldsymbol{x}) = \nabla_{\boldsymbol{x}} c(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x})), \tag{4}$$

where $\nabla_{\boldsymbol{x}} c$ is the gradient of c with respect to $\boldsymbol{x}$, under the condition that the Jacobi matrix $\boldsymbol{C} = \mathrm{D}_{\boldsymbol{xy}} c$ is invertible. A sufficient condition for a maximum/minimum solution requires

$$\mathrm{D}^2 u_1(\boldsymbol{x}) - \mathrm{D}_{\boldsymbol{xx}} c(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x})) = \boldsymbol{P}, \tag{5}$$

to be positive/negative semi-definite (SPD/SND), respectively, where $\mathrm{D}^2 u_1$ is the Hessian matrix of u_1 and $\mathrm{D}_{\boldsymbol{xx}} c$ is the Hessian matrix of c with respect $\boldsymbol{x}$. Hence, for a c-convex pair we require $\mathrm{tr}(\boldsymbol{P}) \geq 0$ and $\det(\boldsymbol{P}) \geq 0$. Differentiating (4) again with respect to $\boldsymbol{x}$ we can derive the matrix equation

$$\boldsymbol{C}\mathrm{D}\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{P}, \tag{6}$$

where $\mathrm{D}\boldsymbol{m}(\boldsymbol{x})$ is the 2×2 Jacobi matrix of $\boldsymbol{m}$ with respect to $\boldsymbol{x}$.

By transferring the light from source to target we require that all light from the source ends up at the target and energy is conserved, i.e.,

$$\int_{\mathcal{A}} f(\phi, \theta) \mathrm{d}\boldsymbol{S}(\phi, \theta) = \int_{\hat{\boldsymbol{t}}(\mathcal{A})} g(\psi, \chi) \mathrm{d}\boldsymbol{S}(\psi, \chi), \tag{7}$$

for an arbitrary set $\mathcal{A} \subset \mathrm{S}^2$ and image set $\hat{\boldsymbol{t}}(\mathcal{A}) \subset \mathrm{S}^2$. Note that this image set corresponds to the far-field approximation; for more details, see [7]. Changing coordinates and substituting the mapping gives

$$\int_{\boldsymbol{x}(\mathcal{A})} \tilde{f}(\boldsymbol{x}) \frac{4}{(1 + |\boldsymbol{x}|^2)^2} \mathrm{d}\boldsymbol{x} = \int_{\boldsymbol{x}(\mathcal{A})} \tilde{g}(\boldsymbol{m}(\boldsymbol{x})) \frac{4}{(1 + |\boldsymbol{m}(\boldsymbol{x})|^2)^2} \det(\mathrm{D}\boldsymbol{m}(\boldsymbol{x})) \mathrm{d}\boldsymbol{x}, \tag{8}$$

where we omit the absolute value sign of the determinant and restrict ourselves to a positive Jacobian of the mapping.

Combining the matrix equation (6) with energy conservation (8), we arrive at the generalized Monge-Ampère equation

$$\det(\mathrm{D}\boldsymbol{m}(\boldsymbol{x})) = \frac{\tilde{f}(\boldsymbol{x})}{\tilde{g}(\boldsymbol{m}(\boldsymbol{x}))} \frac{(1+|\boldsymbol{m}(\boldsymbol{x})|^2)^2}{(1+|\boldsymbol{x}|^2)^2} = F(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x})) = \frac{\det(\boldsymbol{P})}{\det(\boldsymbol{C})}, \tag{9a}$$

where we introduce $F(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x}))$ to denote the total right hand side. We define the corresponding transport boundary condition to (9a) as

$$\boldsymbol{m}(\partial\mathcal{X}) = \partial\mathcal{Y}, \tag{9b}$$

stating that all light from the boundary of the source $\mathcal{X}$ is mapped to the boundary of the target $\mathcal{Y}$ [4, 5].

We consider two options for the coordinate system of the source domain $\mathcal{X}$: Cartesian stereographic coordinates $\boldsymbol{x}$ and polar stereographic coordinates (ρ, θ). We maintain Cartesian stereographic coordinates for the target domain. The derivation of the corresponding generalized Monge-Ampère equation in polar stereographic coordinates is presented in [7].

3 Generalized Least-Squares Algorithm

In this section, we will give a broad overview of the numerical algorithm to compute the mapping $\boldsymbol{m}$ in (9) and the surface u in (4).

The computation of $\boldsymbol{m}$ is an iterative procedure; in every iteration we minimize several functionals. First, to solve (9a), or (6), for the interior domain we minimize

$$J_I[\boldsymbol{m}, \boldsymbol{P}] = \frac{1}{2}\int_{\mathcal{X}} \parallel \boldsymbol{C}\mathrm{D}\boldsymbol{m} - \boldsymbol{P} \parallel^2 \mathrm{d}\boldsymbol{x}, \tag{10}$$

under the constraint $\det(\boldsymbol{P}) = F(\cdot, \boldsymbol{m})\det(\boldsymbol{C}(\cdot, \boldsymbol{m}))$. The norm used is the Frobenius norm. To impose the transport boundary condition in (9b) we minimize the functional

$$J_B[\boldsymbol{m}, \boldsymbol{b}] = \frac{1}{2}\oint_{\partial\mathcal{X}} |\boldsymbol{m} - \boldsymbol{b}|^2 \mathrm{d}s, \tag{11}$$

where $|\cdot|$ denotes the L_2-norm and $\boldsymbol{b} : \partial\mathcal{X} \to \mathcal{Y}$. Last, we minimize a weighted average of the functionals J_I and J_B as

$$J[\boldsymbol{m}, \boldsymbol{P}, \boldsymbol{b}] = \alpha J_I[\boldsymbol{m}, \boldsymbol{P}] + (1-\alpha) J_B[\boldsymbol{m}, \boldsymbol{b}], \tag{12}$$

with $0 < \alpha < 1$ the weighting parameter.

We iterate starting from an initial guess $\boldsymbol{m}^0$ and cost function matrix $\boldsymbol{C}(\cdot, \boldsymbol{m}^0)$:

$$\boldsymbol{b}^{n+1} = \text{argmin}_{\boldsymbol{b}\in\mathcal{B}} J_B[\boldsymbol{m}^n, \boldsymbol{b}], \tag{13a}$$

$$\boldsymbol{P}^{n+1} = \text{argmin}_{\boldsymbol{P}\in\mathcal{P}(\boldsymbol{m}^n)} J_I[\boldsymbol{m}^n, \boldsymbol{P}], \tag{13b}$$

$$\boldsymbol{m}^{n+1} = \text{argmin}_{\boldsymbol{m}\in\mathcal{M}} J[\boldsymbol{m}, \boldsymbol{P}^{n+1}, \boldsymbol{b}^{n+1}], \tag{13c}$$

where the minimization steps are performed over the spaces $\mathcal{B} = \{\boldsymbol{b} \in C^1(\partial\mathcal{X})^2 | \boldsymbol{b} \in \partial\mathcal{Y}\}$, $\mathcal{P}(\boldsymbol{m}) = \{\boldsymbol{P} \in C^1(\mathcal{X})^{2\times2} | \boldsymbol{P}\text{SPD}, \det(\boldsymbol{P}) = F(\cdot, \boldsymbol{m})\det(\boldsymbol{C}(\cdot, \boldsymbol{m}))\}$, $\mathcal{M} = C^2(\mathcal{X})^2$ and thus we obtain a c-convex u_1. After each iteration we update the matrix $\boldsymbol{C}(\cdot, \boldsymbol{m}^n)$.

The minimization steps (13a), (13b) and (13c) are described in detail in [6, 7]. The operations in (13a) and (13b) are point-wise minimization steps. In contrast, the minimization step (13c) and the subsequent calculation of the lens surface can not be performed point-wise. Using calculus of variations, we obtain two coupled elliptic equations for the components m_1 and m_2 of $\boldsymbol{m}$, which can be written as

$$\nabla \cdot (\boldsymbol{C}^T \boldsymbol{C}\mathrm{D}\boldsymbol{m}) = \nabla \cdot (\boldsymbol{C}^T \boldsymbol{P}), \qquad \boldsymbol{x} \in \mathcal{X}, \tag{14a}$$

$$(1-\alpha)\boldsymbol{m} + \alpha(\boldsymbol{C}^T \boldsymbol{C}\mathrm{D}\boldsymbol{m})\hat{\boldsymbol{n}} = (1-\alpha)\boldsymbol{b} + \alpha\boldsymbol{C}\cdot\boldsymbol{P}\hat{\boldsymbol{n}}, \qquad \boldsymbol{x} \in \partial\mathcal{X}, \tag{14b}$$

We discretize (14) using the finite volume method [7].

For the computation of the surface $u = e^{u_1}$, we minimize the functional $I[u_1] = \frac{1}{2}\int_{\mathcal{X}} |\nabla u_1 - \nabla_{\boldsymbol{x}} c(\cdot, \boldsymbol{m})|^2 \mathrm{d}\boldsymbol{x}$, cf. (4). Using calculus of variations, we obtain the Neumann problem

$$\nabla \cdot \nabla u_1 = \nabla \cdot \nabla_{\boldsymbol{x}} c(\cdot, \boldsymbol{m}), \qquad \boldsymbol{x} \in \mathcal{X}, \tag{15a}$$

$$\nabla u_1 \cdot \hat{\boldsymbol{n}} = \nabla_{\boldsymbol{x}} c(\cdot, \boldsymbol{m}) \cdot \hat{\boldsymbol{n}}, \qquad \boldsymbol{x} \in \partial\mathcal{X}. \tag{15b}$$

This BVP has a unique solution up to an additive constant. We calculate the unique solution u_1 by prescribing the average value of u_1 as a constraint.

4 Numerical Results

We apply the numerical algorithm to an example problem, using both Cartesian and polar stereographic coordinates. We compute a freeform surface that transforms the light of a point source into an image on a projection screen in the far field. Preliminary experiments motivate the choice of $\alpha = 0.1$ as weighting parameter to obtain a mapping that adheres nicely to the boundary of the target domain. The image on the screen in the far field is a self-portrait by Van Gogh [3]. The laptop used for the calculations has an Intel Core i7-7700HQ CPU 2.80 GHz with 32.0 GB of RAM. In Cartesian stereographic coordinates the source domain is given by the square $\mathcal{X} = [-0.5, 0.5]^2$. In polar stereographic coordinates, the source domain is

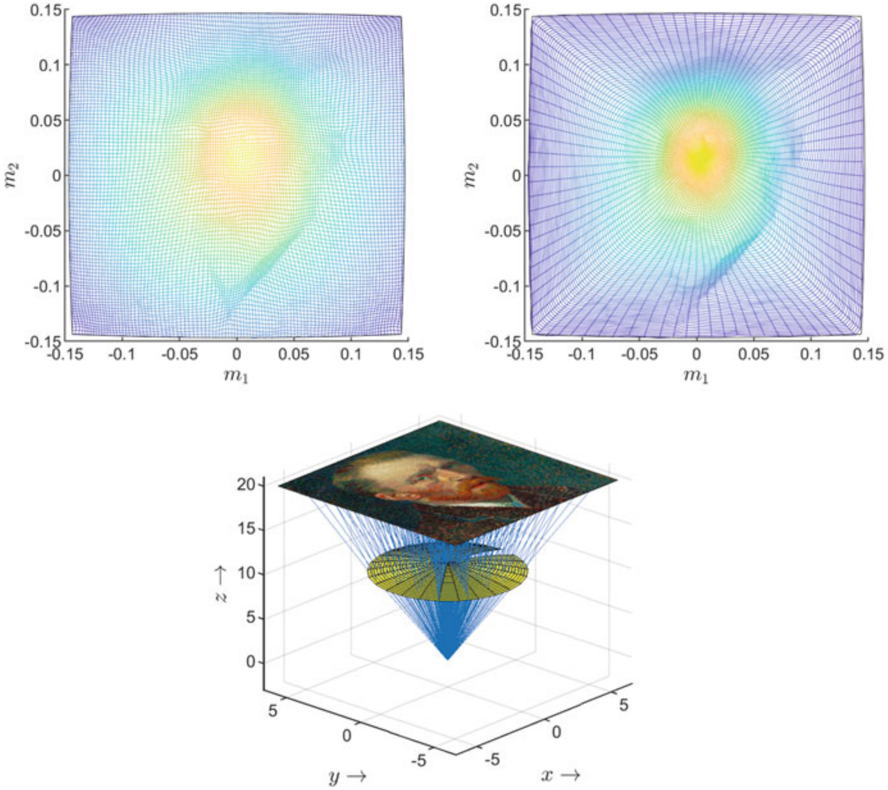


Fig. 2 The resulting Cartesian mapping (top left), polar mapping (top right), and a cartoon of the ray-tracing procedure for the polar surface (bottom center)

given by the circle $\mathcal{X} = \{(\rho, \zeta) \in \mathbb{R}^2 |\ 0 \leq \rho < 0.5, 0 \leq \zeta < 2\pi\}$. The source has a uniform light distribution $\tilde{f}(\boldsymbol{x}) = 1$. The refracted rays are projected on a screen in the far field, parallel to the xy-plane at a distance $d = 20$ above the lens. The required illumination $L(\xi, \eta)$ [lm/m^2], with (ξ, η) the local Cartesian coordinates on the projection screen, is derived from the gray scale values of the portrait of Van Gogh, see Fig. 3. The target distribution $\tilde{g}(\boldsymbol{y})$ is a deformation of the illuminance $L(\xi, \eta)$; the conversion from $L(\xi, \eta)$ to $\tilde{g}(\boldsymbol{y})$ is explained in detail in [4, p. 78]. The gray scale values of the picture prescribe the illuminance with black regions set to 5% of the maximum illuminance to avoid division by zero in (9a).

We discretize the source domain using a uniform 500 × 500 grid. The results are shown in Fig. 2. We validated the resulting lens image by using quasi-Monte Carlo ray tracing with 3000 × 3000 rays. The resulting target illuminances $L(\xi, \eta)$ are plotted in Fig. 3. The total computation time of performing 500 iterations to calculate $\boldsymbol{m}$ is 1655 (Cartesian) and 2452 (polar) seconds. The subsequent computation time of u_1 is 15.7 (Cartesian) and 19.3 (polar) seconds. The details of

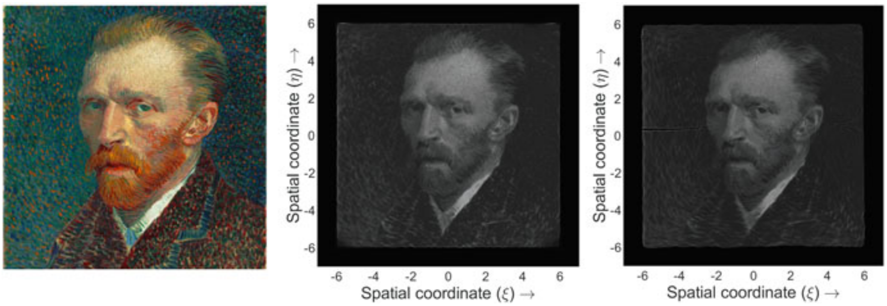


Fig. 3 The original (left) and Cartesian ray-traced (middle) and polar ray-traced (right) images

Van Gogh's beard and hair are noticeable in the ray-traced images. Starting from a square or circular source domain while using either Cartesian or polar stereographic coordinates produces similar results with a high level of detail in the far field.

Acknowledgments This work is part of the research program NWO-TTW Perspectief with project number P15-36, which is (partly) financed by the Netherlands Organisation for Scientific Research (NWO).

References

1. Beltman, R., ten Thije Boonkkamp, J., IJzerman, W.: A least-squares method for the inverse reflector problem in arbitrary orthogonal coordinates. J. Comput. Phys. **367**, 347–373 (2018)
2. Caboussat, A., Glowinski, R., Sorensen, D.: A least-squares method for the numerical solution of the Dirichlet problem for the elliptic Monge-Ampère equation in dimension two. ESAIM: Control, Optimisation and Calculus of Variations **19**(3), 780–810 (2013)
3. Encyclopædia Britannica: Atomic force microscopy (2013). URL britannica.com/biography/Vincent-van-Gogh/images-videos#/media/1/237118/59904. [Online; accessed October 9, 2019]
4. Prins, C.: Inverse Methods for Illumination Optics. Ph.D. thesis, Eindhoven University of Technology, Eindhoven (2014)
5. Prins, C., Beltman, R., ten Thije Boonkkamp, J., IJzerman, W., Tukker, T.W.: A least-squares method for optimal transport using the Monge-Ampère equation. SIAM J. Sci. Comput. **37**(6), B937–B961 (2015)
6. Romijn, L., ten Thije Boonkkamp, J., IJzerman, W.: Inverse reflector design for a point source and far-field target. J. Comput. Phys. **408**, 109–283 (2020)
7. Romijn, L., ten Thije Boonkkamp, J., IJzerman, W.: Freeform lens design for a point source and far-field target. JOSA A **36**(11), 1926–1939 (2019)
8. Yadav, N.: Monge-Ampère Problems with Non-Quadratic Cost Function: Application to Freeform Optics. Ph.D. thesis, Eindhoven University of Technology, Eindhoven (2018)
9. Yadav, N., Romijn, L., ten Thije Boonkkamp, J., IJzerman, W.: A least-squares method for the design of two-reflector optical systems. J. Phys.: Photonics **1**(3), 034001 (2019)
10. Yadav, N., ten Thije Boonkkamp, J., IJzerman, W.: A Monge-Ampère problem with non-quadratic cost function to compute freeform lens surfaces. J. Sci. Comput. **80**(1), 475–499 (2019)

Reduced Order Methods for Parametrized Non-linear and Time Dependent Optimal Flow Control Problems, Towards Applications in Biomedical and Environmental Sciences

Maria Strazzullo, Zakia Zainib, Francesco Ballarin, and Gianluigi Rozza

Abstract We introduce reduced order methods as an efficient strategy to solve parametrized non-linear and time dependent optimal flow control problems governed by partial differential equations. Indeed, the optimal control problems require a huge computational effort in order to be solved, most of all in physical and/or geometrical parametrized settings. Reduced order methods are a reliable and suitable approach, increasingly gaining popularity, to achieve rapid and accurate optimal solutions in several fields, such as in biomedical and environmental sciences. In this work, we employ a POD-Galerkin reduction approach over a parametrized optimality system, derived from the Karush-Kuhn-Tucker conditions. The methodology presented is tested on two boundary control problems, governed respectively by (1) time dependent Stokes equations and (2) steady non-linear Navier-Stokes equations.

1 Introduction

Parametrized optimal flow control problems (OFCP($\boldsymbol{\mu}$)s) constrained to parametrized partial differential equations (PDE($\boldsymbol{\mu}$)s) are a very versatile mathematical model which arises in several applications, see e.g. [6, 8, 11]. These problems are computationally expensive and challenging even in a simpler non-parametrized context. The computational cost becomes unfeasible when these problems involve time dependency [1, 14] or non-linearity [4, 5, 11], in addition to physical and/or geometrical parametrized settings that describe several configurations and phenomena. A suitable strategy to lower this expensive computational effort is to employ reduced order methods (ROMs) in the context of

M. Strazzullo · Z. Zainib · F. Ballarin · G. Rozza (✉)
SISSA mathLab, Mathematics Area, International School for Advanced Studies, Trieste, Italy
e-mail: maria.strazzullo@sissa.it; zakia.zainib@sissa.it; francesco.ballarin@sissa.it; gianluigi.rozza@sissa.it; grozza@sissa.it

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_83

OFCP($\boldsymbol{\mu}$) s, which recast them in a cheap, yet reliable, low dimensional framework [7, 12]. We exploit these techniques in order to solve boundary OFCP($\boldsymbol{\mu}$) s on a bifurcation geometry [13] which can be considered as (1) a riverbed in environmental sciences and as (2) a bypass graft for cardiovascular applications.

In the first research field, reduced parametrized optimal control framework (see e.g. [9, 10]) can be of utmost importance. It perfectly fits in forecasting and data assimilated models and it could be exploited in order to prevent possibly dangerous natural situations [15]. The presented test case is governed by time dependent Stokes equations, which are an essential tool in marine sciences in order to reliably simulate evolving natural phenomena.

Furthermore, discrepancies between computational modelling in cardiovascular mechanics and reality usually ought to high computational cost and lack of optimal quantification of boundary conditions, especially the outflow boundary conditions. In this work, we present application of the aforementioned numerical framework combining OFCP(μ) and reduced order methods in the bifurcation geometry. The aim is to quantify the outflow conditions automatically while matching known physiological data for different parameter-dependent scenarios [2, 17]. In this test case, Navier-Stokes equations will model the fluid flow.

The work is outlined as follows: in Sect. 2, the problem formulation and the methodology are summarized. Section 3 shows the numerical results for the two boundary OFCP($\boldsymbol{\mu}$) s, based on [8, 13]. Conclusions follow in Sect. 4.

2 Proper Orthogonal Decomposition for OFCP($\boldsymbol{\mu}$)s

In this section, we briefly describe the problem and the adopted solution strategy for time dependent non-linear boundary OFCP($\boldsymbol{\mu}$) s: in the cases mentioned in Sect. 1, the reader shall take the non-linear term and time-dependent terms to be zero accordingly [16, 17]. The goal of OFCP($\boldsymbol{\mu}$) s is to find a minimizing solution for a quadratic cost functional $\mathcal{J}$ under a PDE($\boldsymbol{\mu}$) constraint thanks to an external variable denoted as *control*. In the next section, we will show numerical results over a bifurcation geometry Ω with physical and/or geometrical parametrization represented by $\boldsymbol{\mu} \in \mathcal{D} \subset \mathbb{R}^d, d \in \mathbb{N}$. Thus, considering the space-time domain $Q = \Omega \times [0, T]$ with a sufficiently regular spatial boundary $\partial\Omega$,[1] let us define the Hilbert spaces $S = V \times P$, $Z = Z_V \times Z_P$ and U for state and adjoint velocity and pressure, and control variables denoted by $\boldsymbol{s} = (\boldsymbol{v}, p) \in S$, $\boldsymbol{u} \in U$ and $\boldsymbol{z} = (\boldsymbol{w}, q) \in Z$, respectively. The stability and uniqueness of the optimal solution will be guaranteed if $S \equiv Z$, which will be our assumption in this work.

[1] For the steady case, $T = 0$ and $Q \equiv \Omega$.

We introduce $X = S \times U$ such that $\mathbf{x} = (\boldsymbol{s}, \boldsymbol{u}) \in X$. Then, the problem reads: *given* $\boldsymbol{\mu} \in \mathcal{D}$, *find* $(\mathbf{x}, z; \boldsymbol{\mu}) \in X \times S$ *s.t.:*

$$\begin{cases} \begin{matrix} \mathcal{A}(\mathbf{x}, \mathbf{y}; \boldsymbol{\mu}) + \mathcal{B}(z, \mathbf{y}; \boldsymbol{\mu}) + \\ \mathcal{E}\left(\boldsymbol{v}, \boldsymbol{w}, \mathbf{y}_{\boldsymbol{v}}; \boldsymbol{\mu}\right) + \mathcal{E}\left(\boldsymbol{w}, \boldsymbol{v}, \mathbf{y}_{\boldsymbol{v}}; \boldsymbol{\mu}\right) \end{matrix} = \int_0^T \langle \mathcal{H}(\boldsymbol{\mu}), \mathbf{y} \rangle \, dt, & \forall\, \mathbf{y} \in X, \\ \mathcal{B}(\mathbf{x}, \boldsymbol{\kappa}; \boldsymbol{\mu}) + \mathcal{E}(\boldsymbol{v}, \boldsymbol{v}, \boldsymbol{\kappa}_{\boldsymbol{w}}; \boldsymbol{\mu}) = \int_0^T \langle \mathcal{G}(\boldsymbol{\mu}), \boldsymbol{\kappa} \rangle \, dt, & \forall\, \boldsymbol{\kappa} \in S, \end{cases} \tag{1}$$

where $\mathbf{y} = \left(\mathbf{y}_{\boldsymbol{v}}, \mathbf{y}_p, \mathbf{y}_{\boldsymbol{u}}\right)$ and $\boldsymbol{\kappa} = (\boldsymbol{\kappa}_{\boldsymbol{w}}, \boldsymbol{\kappa}_q)$. The bilinear forms $\mathcal{A} : X \times X \to \mathbb{R}$ and $\mathcal{H}(\boldsymbol{\mu})$ are related to the minimization cost functional $\mathcal{J}$, while the bilinear form $\mathcal{B} : X \times Z \to \mathbb{R}$ represents the linear part of state-constraints and $\mathcal{E}$ is the non-linear convection term, which will be zero for the time dependent linear case. In order to solve the optimality system (1) we exploit Galerkin Finite Elements (FE) *snapshots-based Proper orthogonal decomposition* (POD)–Galerkin (see [7]), summarized in Table 1, where the number of time steps are denoted by $\mathcal{N}_t$.

In order to guarantee the efficiency of the POD–Galerkin approach, we rely on the affine assumption over the forms, i.e. every form can be written as a linear combination of $\boldsymbol{\mu}-$dependent functions and $\boldsymbol{\mu}-$independent quantities. In this way, the system resolution is divided into parameter independent (*offline*) and dependent (*online*) phases (see Table 1 for details) such that the expensive calculations are absorbed in the former stage and only *online* stage is repeated every time the parameter $\boldsymbol{\mu}$ changes. From the perspective of the problem stability, to ensure uniqueness of pressure at the reduced order level, we enrich the state and adjoint velocity space with *supremizers* and, to guarantee the fulfillment of Brezzi's inf-sup condition [3] at the reduced level, we use aggregated equivalent state and adjoint spaces. Thus, dimension of the reduced problem reduces from $\mathcal{N} \times \mathcal{N}_t$ to $13N$.

Table 1 Algorithm: POD–Galerkin for OFCP($\boldsymbol{\mu}$)

	Input: μ_1 for lifting, N, n
Offline phase	Output: Reduced order solution spaces
1. Compute snapshots $\boldsymbol{\delta}_{\mathcal{N}_\delta \times \mathcal{N}_t}(\boldsymbol{\mu}^n)$ for $1 \le n \le \lvert\Lambda\rvert$, $\boldsymbol{\delta} = \boldsymbol{v}, p, \boldsymbol{u}, \boldsymbol{w}, q$ and state and adjoint supremizers. The global dimension of FE space discretization is $\mathcal{N} = 2\mathcal{N}_{\boldsymbol{v}} + 2\mathcal{N}_p + \mathcal{N}_{\boldsymbol{u}}$	
2. Solve eigenvalue problems $\mathbb{A}^{\delta} \boldsymbol{\rho}_n^{\delta} = \lambda_n^{\delta} \boldsymbol{\rho}_n^{\delta}$, $n = 1, \cdots, \lvert\boldsymbol{\Lambda}\rvert$, where $\mathbb{A}^{\delta}$ is correlation matrix of snapshots	
3. If relative energy of eigenvalues is greater than $1 - \epsilon_{tol}$, $0 < \epsilon_{tol} \ll 1$, keep corresponding eigenvalue-eigenvector pairs $(\lambda^n, \boldsymbol{\rho}^n)$	
4. Construct orthonormal POD basis from the retained N eigenvectors and add the POD modes of the supremizers to state and adjoint velocities	
	Input: Online parameter $\boldsymbol{\mu} \in \mathcal{D}$
Online phase	Output: Reduced order solution
1. Perform Galerkin projection to calculate reduced order coefficients such that $\boldsymbol{\delta} \approx \mathfrak{X}_{\delta} \boldsymbol{\delta}_{\mathrm{N}}$ where, $\mathfrak{X}_{\delta}$ denotes reduced bases matrices containing all the time instances	
2. Solve the reduced order version of the optimality system (1)	

3 Results

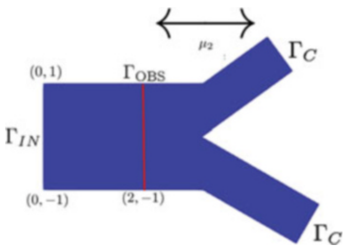
3.1 *Linear Time Dependent OFCP($\boldsymbol{\mu}$) Governed by Stokes Equations*

In this section, inspired by Negri et al. [8], Rozza et al. [13], we propose an OFCP($\boldsymbol{\mu}$) governed by a time dependent Stokes equation. First of all, let us introduce the smooth domain $\Omega(\mu_2)$. The parameter stretches the length of the reference domain shown in Fig. 1, which will be indicated with Ω from now on. We want to recover a measurement $\boldsymbol{v}_d(\mu_3) \in L^2(0, T; [L^2(\Omega)]^2)$ over the one dimensional observation domain Γ_{OBS} controlling the Neumann flux over Γ_C, with the inflow $\boldsymbol{v}_{in}(\boldsymbol{\mu}) = (10\,(x_2 - 1)\,(1 - x_2)\,, 0)$. The setting is suited for environmental applications: we control the flow in order to avoid potentially dangerous situations in an hypothetical *real time* monitoring plan on the domain, which can represent a riverbed. The space-time domain is $Q = \Omega \times [0, 1]$. Let us consider the following function spaces: $V = L^2(0, T; [H^1_{\Gamma D}(\Omega)]^2) \cap H^1(0, T; [H^1_{\Gamma D}(\Omega)^*]^2)$, $P = L^2(0, T; L^2(\Omega))$ and $U = L^2(0, T; [L^2(\Omega)]^2)$ for state and adjoint velocity, state and adjoint pressure and for control, respectively. Then, we define $X = (V \times P) \times U$. For a given $\boldsymbol{\mu} \in \mathcal{D} = [0.01, 1] \times [1, 2] \times [0.01, 1]$, we want to find the solution of time dependent Stokes equations which minimizes:

$$\mathcal{J} := \frac{1}{2}\int_0^T \int_{\Gamma_{OBS}} (\boldsymbol{v}(\boldsymbol{\mu}) - \boldsymbol{v}_d(\mu_3))^2 dsdt + \frac{\alpha_1}{2}\int_0^T \int_{\Gamma_C} \boldsymbol{u}(\boldsymbol{\mu})^2 dsdt + \frac{\alpha_2}{2}\int_0^T \int_{\Gamma_C} |\nabla \boldsymbol{u}(\boldsymbol{\mu})\boldsymbol{t}|^2 dsdt, \tag{2}$$

where $\alpha_1 = 10^{-3}, \alpha_2 = 10^{-4}$ and $\boldsymbol{t}$ is the unit tangent vector to Γ_C and $\boldsymbol{v}_d(\mu_3) = [\mu_3(8(x_2^3 - x_2^2 - x_2 + 1) + 2(-x_2^3 - x_2^2 + x_2 + 1)), 0]$. The cost functional penalizes not only the magnitude of the control, but also its rapid variations over the boundary. The constrained minimization problem (2) is equivalent to the resolution of problem

Fig. 1 Physical domain



(1) where the considered forms are defined by:

$$\mathcal{A}(\mathbf{x}, \mathbf{y}) = \int_0^T \int_{\Gamma_{OBS}} \boldsymbol{v} \cdot \mathbf{y}_{\boldsymbol{v}} \, dsdt + \alpha_1 \int_0^T \int_{\Gamma_C} \boldsymbol{u} \cdot \mathbf{y}_{\boldsymbol{u}} \, dsdt + \alpha_2 \int_0^T \int_{\Gamma_C} \nabla \boldsymbol{u} \boldsymbol{t} \cdot \nabla \mathbf{y}_{\boldsymbol{u}} \boldsymbol{t} \, dsdt,$$

$$\mathcal{B}(\mathbf{x}, z; \boldsymbol{\mu}) = \int_Q \frac{\partial \boldsymbol{v}}{\partial t} \cdot \boldsymbol{w} \, dxdt + \mu_1 \int_Q \nabla \boldsymbol{v} \cdot \nabla \boldsymbol{w} \, dxdt - \int_Q p(\nabla \cdot \boldsymbol{w}(\boldsymbol{\mu})) \, dxdt$$

$$- \int_Q q(\nabla \cdot (\boldsymbol{v}(\boldsymbol{\mu})) \, dxdt - \int_0^T \int_{\Gamma_C} \boldsymbol{u} \cdot \boldsymbol{w} \, dsdt,$$

$$\langle \mathcal{H}(\boldsymbol{\mu}), \mathbf{y} \rangle = \int_{\Gamma_{OBS}} \boldsymbol{v}_d(\mu_3) \cdot \mathbf{y}_{\boldsymbol{v}} \, ds, \qquad \langle \mathcal{G}(\boldsymbol{\mu}), q \rangle = 0, \quad \forall \mathbf{y} \in X,$$

for every $\mathbf{x}, \mathbf{y} \in X$ and $\boldsymbol{\kappa} \in S$. We built the reduced space with $N = 35$ over a training set of 70 snapshots of global dimension 131400, for $\mathcal{N}_t = 20$. In time dependent applications, ROMs are of great advantage: in Table 2 the speedup index is shown with respect to N. The speedup represents how many ROM systems one can solve in the time of a FE simulation. Nevertheless, we do not pay in accuracy as Figs. 2 and 3 show: it represents the relative error between FE and ROM variables. The relative error between FE and ROM $\mathcal{J}$ is presented in Table 2

Table 2 Speedup analysis and relative error $\mathcal{J}$

N	Speedup	Relative error $\mathcal{J}$
15	66,338	10^{-7}
20	47,579	10^{-8}
25	34,335	10^{-8}
30	22,477	10^{-9}
35	17,420	10^{-10}

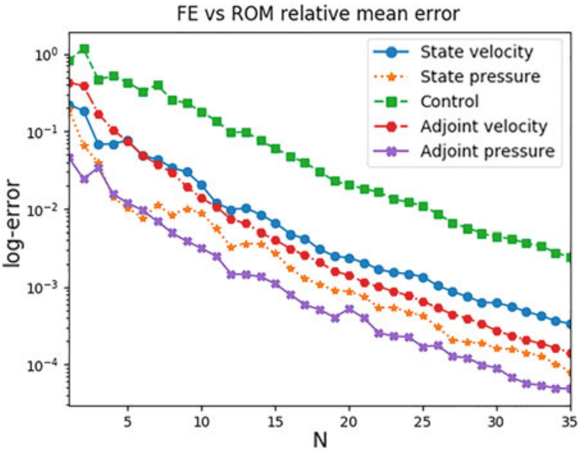


Fig. 2 FE vs ROM mean relative error over 50 parameters

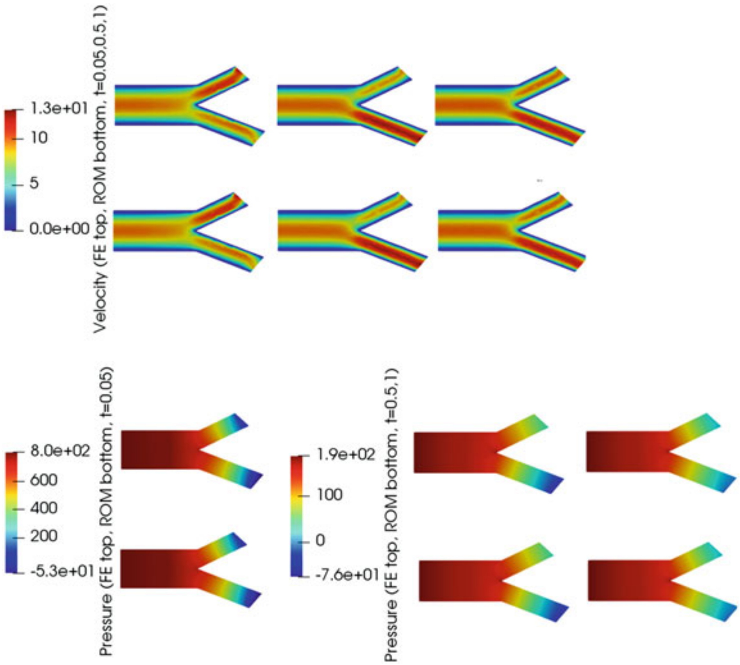


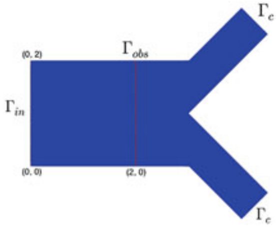
Fig. 3 FE (top) vs ROM (bottom) comparison of state velocity and state pressure, for $t = 0.05, 0.5, 1$ and $\boldsymbol{\mu} = (0.5, 1.5, 1)$

3.2 Non-linear Steady OFCP($\boldsymbol{\mu}$) Governed by Navier-Stokes Equations

In this section, we will demonstrate the numerical results for second test case with optimal boundary control problem governed by non-linear incompressible steady Navier-Stokes equations. We consider a bifurcation domain Ω as employed in the previous example (see Fig. 4), which can be considered as an idealized model of arterial bifurcation in cardiovascular problems [8, 13, 17]. Fluid shall enter the domain from Γ_{in} and shall leave through the outlets Γ_c. In this example, physical parameterization is considered for the inflow velocity given by $\boldsymbol{v}_{in}(\boldsymbol{\mu}) = 10\mu_1 \left(x_2 (2 - x_2), 0\right)$ and the desired velocity, denoted by $\boldsymbol{v}_d \in L^2(\Omega)$ and prescribed at the 1-D observation boundary Γ_{obs} through the following expression:

$$\boldsymbol{v}_d(\boldsymbol{\mu}) = \begin{pmatrix} 10\mu_1 \left(0.8\left((x_2-1)^3 - (x_2-1)^2 - (x_2-1) + 1\right) + 0.2\left(-(x_2-1)^3 - (x_2-1)^2 + x_2\right)\right) \\ 0 \end{pmatrix}.$$

Fig. 4 Domain (Ω)



The cost-functional $\mathcal{J}$ is defined as:

$$\mathcal{J}(\boldsymbol{v},\boldsymbol{u};\boldsymbol{\mu}) = \frac{1}{2}\|\boldsymbol{v}(\boldsymbol{\mu}) - \boldsymbol{v}_d(\boldsymbol{\mu})\|^2_{L^2(\Gamma_{obs})} + \frac{\alpha}{2}\|\boldsymbol{u}(\boldsymbol{\mu})\|^2_{L^2(\Gamma_c)} + \frac{0.1\alpha}{2}\|\nabla\boldsymbol{u}(\boldsymbol{\mu})\boldsymbol{t}\|^2_{L^2(\Gamma_c)}, \tag{3}$$

where $\boldsymbol{t}$ is the tangential vector to Γ_c. The mathematical problem reads: *Given* $\boldsymbol{\mu} \in \mathcal{D} = [0.5, 1.5]$, *find* $(\boldsymbol{v}(\boldsymbol{\mu}), p(\boldsymbol{\mu}), \boldsymbol{u}(\boldsymbol{\mu}))$ *that minimize* $\mathcal{J}$ *and satisfy the Navier-Stokes equations with* $\boldsymbol{v}_{in}(\boldsymbol{\mu})$ *prescribed at the inlet* Γ_{in}, *no-slip conditions at the walls* Γ_w *and* $\boldsymbol{u}(\boldsymbol{\mu})$ *implemented at* Γ_c *through Neumann conditions.*

At the continuous level, we consider $X(\Omega) = H^1_{\Gamma_{in}\cup\Gamma_w}(\Omega) \times L^2(\Omega) \times L^2(\Gamma_c)$, where

$$H^1_{\Gamma_{in}\cup\Gamma_w}(\Omega) = \left[\boldsymbol{v} \in \left[H^1(\Omega)\right]^2 : \boldsymbol{v}|_{\Gamma_{in}} = \boldsymbol{v}_{in} \text{ and } \boldsymbol{v}|_{\Gamma_w} = 0\right].$$

Thus,

$$\begin{aligned}
\mathcal{A}(\mathbf{x},\boldsymbol{y}) &= \int_\Omega \boldsymbol{v}(\boldsymbol{\mu})\cdot\boldsymbol{y}_{\boldsymbol{v}}d\Omega + \alpha\int_{\Gamma_c}\boldsymbol{u}(\boldsymbol{\mu})\cdot\boldsymbol{y}_{\boldsymbol{u}}d\Gamma_c + \frac{\alpha}{10}\int_{\Gamma_c}(\nabla\boldsymbol{u}(\boldsymbol{\mu}))\boldsymbol{t}\cdot\nabla\left(\boldsymbol{y}_{\boldsymbol{u}}\right)\boldsymbol{t}d\Gamma_c,\\
\mathcal{B}(\mathbf{x},z) &= \eta\int_\Omega\nabla\boldsymbol{v}(\boldsymbol{\mu})\cdot\nabla\boldsymbol{w}d\Omega - \int_\Omega p(\boldsymbol{\mu})(\nabla\cdot\boldsymbol{w})d\Omega - \int_\Omega q(\nabla\cdot\boldsymbol{v}(\boldsymbol{\mu}))d\Omega\\
&\quad - \int_{\Gamma_c}\boldsymbol{u}(\boldsymbol{\mu})\cdot\boldsymbol{w}d\Gamma_c,\\
\mathcal{E}(\boldsymbol{v},\boldsymbol{v},\boldsymbol{w}) &= \int_\Omega(\boldsymbol{v}(\boldsymbol{\mu})\cdot\nabla)\boldsymbol{v}(\boldsymbol{\mu})\cdot\boldsymbol{w}d\Omega \quad\text{and}\quad \langle\mathcal{H}(\boldsymbol{\mu}),\mathbf{y}\rangle = \int_\Omega\boldsymbol{v}_d(\boldsymbol{\mu})\cdot\boldsymbol{y}_{\boldsymbol{v}}d\Omega.
\end{aligned}$$

To construct the reduced order solution spaces, we consider a sample Λ of 100 parameter values and solving the problem (1) through Galerkin Finite Element method, we construct the snapshot matrices for the solution variables $\boldsymbol{v}, p, \boldsymbol{u}, \boldsymbol{w}, q$. For $N = 10$, eigenvalues energy of the state, control and adjoint variables is demonstrated in Fig. 5. Evidently, N eigenvalues capture 99.9% of the Galerkin FE discretized solution spaces and the reduced order spaces are thus built with dimensions $13\text{N} + 1 = 131$ (Fig. 6). The state velocity and control for $\mu = 0.7, 1.1, 1.4$ are shown in Fig. 7. Furthermore, we report the accumulative relative error for the solution variables and the relative error for $\mathcal{J}$ in Fig. 8. The former decreases upto 10^{-8} along with the latter decreasing upto 10^{-14}.

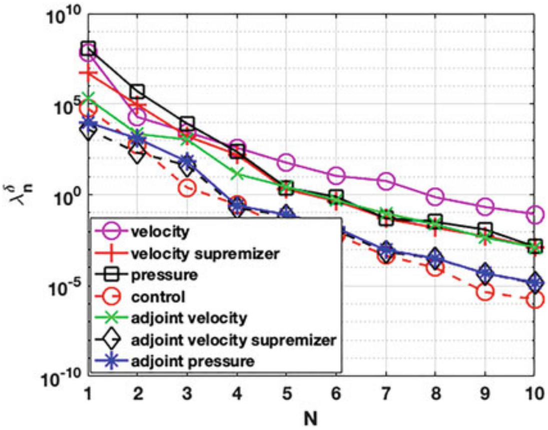


Fig. 5 Eigenvalues of $N = 10$ POD modes

Mesh size	5977
No. of reduced order bases N	131
$\mathcal{D}$	[0.7, 1.5]
$\lvert\Lambda\rvert$	100
offline phase	4.9×10^3 seconds
online phase	9×10^1 seconds

Fig. 6 Computational details of POD–Galerkin for Navier-Stokes constrained OFCP($\boldsymbol{\mu}$)

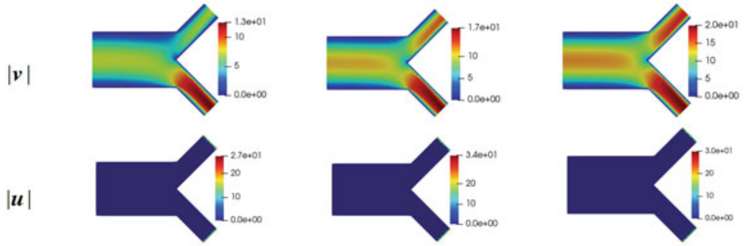
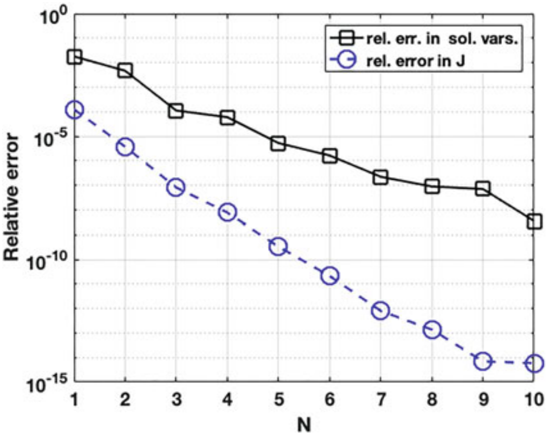


Fig. 7 State velocity and control for $\mu_1 = 0.7, 1.1, 1.4$

4 Concluding Remarks

In this work, we propose ROMs as a suitable tool to solve a parametrized boundary OFCP($\boldsymbol{\mu}$)s for time dependent Stokes equations and steady Navier-Stokes equations. The framework proposed is suited for several *many query* and *real time* applications both in environmental marine sciences and bio-engineering. The reduction of the KKT system is performed through a POD-Galerkin approach, which leads to accurate surrogate solutions in a low dimensional space. This work aims at showing how ROMs can have an effective impact in the management of

Fig. 8 Relative error for solution variables and $\mathcal{J}$



parametrized simulations for social life and activities, such as coastal engineering and cardiovascular problems. Indeed, the proposed framework deals with faster solving of parametrized optimal solutions which can find several applications in monitoring planning both in marine ecosystems and patient specific geometries.

Acknowledgments We acknowledge the support by European Union Funding for Research and Innovation—Horizon 2020 Program—in the framework of European Research Council Executive Agency: Consolidator Grant H2020 ERC CoG 2015 AROMA-CFD project 681447 "Advanced Reduced Order Methods with Applications in Computational Fluid Dynamics" (P.I. Prof. G. Rozza). We also acknowledge the INDAM-GNCS project "Advanced intrusive and non-intrusive model order reduction techniques and applications". The computations in this work have been performed with RBniCS library, developed at SISSA mathLab, which is an implementation in FEniCS of several reduced order modelling techniques; we acknowledge developers and contributors to both libraries.

References

1. Agoshkov, V., Quarteroni, A., Rozza, G.: A mathematical approach in the de-sign of arterial bypass using unsteady Stokes equations. Journal of Scientific Computing **28**, 139–165 (2006)
2. Ballarin, F., Faggiano, E., Ippolito, S., Manzoni, A., Quarteroni, A., Rozza, G., Scrofani, R.: Fast simulations of patient-specific haemodynamics of coronary artery bypass grafts based on a POD–Galerkin method and a vascular shape parametrization. Journal of Computational Physics **315**, 609–628 (2016)
3. Brezzi, F.: On the existence, uniqueness and approximation of saddle-point problems arising from Lagrangian multipliers. Revue française d'automatique, informatique, recherche opérationnelle. Analyse numérique **8**(2), 129–151 (1974)
4. Fursikov, A.V., Gunzburger, M.D., Hou, L.: Boundary value problems and optimal boundary control for the Navier–Stokes system: the two-dimensional case. SIAM Journal on Control and Optimization **36**(3), 852–894 (1998)
5. Gunzburger, M.D., Hou, L., Svobodny, T.P.: Analysis and finite element approximation of optimal control problems for the stationary Navier-Stokes equations with distributed and Neumann controls. Mathematics of Computation **57**(195), 123–151 (1991)

6. Haslinger, J., Mäkinen, R.A.E.: Introduction to shape optimization: theory, approximation, and computation. SIAM (Philadelphia, 2003)
7. Hesthaven, J.S., Rozza, G., Stamm, B.: Certified reduced basis methods for parametrized partial differential equations. Springer Briefs in Mathematics (2015, Springer, Milano)
8. Negri, F., Manzoni, A., Rozza, G.: Reduced basis approximation of parametrized optimal flow control problems for the Stokes equations. Computers & Mathematics with Applications **69**(4), 319–336 (2015)
9. Quarteroni, A., Rozza, G., Dedè, L., Quaini, A.: Numerical approximation of a control problem for advection-diffusion processes. In: IFIP Conference on System Modeling and Optimization, pp. 261–273 (Ceragioli F., Dontchev A., Futura H., Marti K., Pandolfi L. (eds) System Modeling and Optimization. CSMO 2005. vol 199. Springer, Boston, 2005)
10. Quarteroni, A., Rozza, G., Quaini, A.: Reduced basis methods for optimal control of advection-diffusion problems. In: Advances in Numerical Mathematics, pp. 193–216. RAS and University of Houston (2007)
11. De los Reyes, J.C., Tröltzsch, F.: Optimal control of the stationary Navier-Stokes equations with mixed control-state constraints. SIAM Journal on Control and Optimization **46**(2), 604–629 (2007)
12. Rozza, G., Huynh, D., Manzoni, A.: Reduced basis approximation and a posteriori error estimation for Stokes flows in parametrized geometries: Roles of the inf-sup stability constants. Numerische Mathematik **125**(1), 115–152 (2013)
13. Rozza, G., Manzoni, A., Negri, F.: Reduction strategies for PDE-constrained optimization problems in Haemodynamics pp. 1749–1768 (ECCOMAS, Congress Proceedings, Vienna, Austria, September 2012)
14. Stoll, M., Wathen, A.: All-at-once solution of time-dependent Stokes control. J. Comput. Phys. **232**(1), 498–515 (2013). DOI 10.1016/j.jcp.2012.08.039. URL http://dx.doi.org/10.1016/j.jcp.2012.08.039
15. Strazzullo, M., Ballarin, F., Mosetti, R., Rozza, G.: Model reduction for parametrized optimal control problems in environmental marine sciences and engineering. SIAM Journal on Scientific Computing **40**(4), B1055–B1079 (2018). https://doi.org/10.1137/17M1150591
16. Strazzullo, M., Ballarin, F., Rozza, G.: POD-Galerkin model order reduction for parametrized time dependent linear quadratic optimal control problems in saddle point formulation (2019). Submitted. https://arxiv.org/abs/1909.09631.
17. Zainib, Z., Ballarin, F., Rozza, G., Triverio, P., Jiménez-Juan, L., S., F.: Reduced order methods for parametric optimal flow control in coronary bypass grafts, towards patient-specific data assimilation (2019). Submitted, https://arxiv.org/abs/1911.01409.

Mathematical Assessment of the Role of Three Factors Entangled in the Development of Glaucoma by Means of the Ocular Mathematical Virtual Simulator

Lorenzo Sala, Christophe Prud'homme, Giovanna Guidoboni, Marcela Szopos, and Alon Harris

Abstract Glaucoma is a multifactorial neurodegenerative disease that involves the optic nerve head and the death of the retinal ganglion cells. The main challenge in medicine is to understand the origin of this degeneration. In this paper we present a virtual clinical study employing the Ocular Mathematical Virtual Simulator (OMVS), a mathematical model, which is able to disentangle the hidden mechanisms and to investigate the causes of this ocular neurodegeneration. In particular, we focus our attention on the influence that intraocular pressure, intracranial pressure and arterial blood pressure set on the ocular hemodynamics.

1 Introduction

One of the main challenges in medicine is to understand the main causes of diseases. Due to the increasing complexity of the problems and elevated costs needed to support scientific research, the trial-and-error technique is no longer optimal to serve this purpose. Medicine is looking for new quantitative methods, which are

L. Sala (✉)
Inria, Paris, France
e-mail: lorenzo.sala@inria.fr

C. Prud'homme
Université de Strasbourg, Strasbourg, France

G. Guidoboni
University of Missouri, Columbia, MO, USA

M. Szopos
Université de Paris, Paris, France

A. Harris
Mount Sinai School of Medicine, New York, NY, USA

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_84

able to detect abnormalities and identify underlying pathogenic mechanisms in a non-invasive way.

The goal of our research project, called *Eye2Brain*, is to develop mathematical and computational methods to study the eye and brain system, in order to aid the interpretation of ocular measurements as biomarkers of the brain status. In particular, driven by the needs in ophthalmology, we focus our attention on glaucoma, an ocular neurodegenerative disease that involves the optic nerve head and the death of the retinal ganglion cells [10]. Glaucoma, also called *the silent thief of vision*, leads to an irreversible vision loss and is the second leading cause of blindness worldwide [6].

One of the main questions arising from medicine is understanding the origin of this degeneration. Currently, the only treatable risk factor is an elevated intraocular pressure (IOP), however it is not the only one, since:

- many people with ocular hypertension do not develop glaucoma;
- many people with normal IOP develop glaucoma (normal tension glaucoma—NTG);
- 25% of IOP-treated patients progress to blindness.

Thus, several studies have shown that glaucoma is a multifactorial disease: for instance gender [12], myopia [4], faulty autoregulation [3], age [12], central corneal thickness [5], ethnicity [12], cerebrospinal fluid (CSF) pressure [7], diabetes and arterial blood pressure [22] may affect the natural behavior of the eye-brain system.

In this very compelling context, our goal is to employ a mathematical model to disentangle the hidden mechanisms and to investigate the causes of this ocular neurodegeneration beyond the IOP. Specifically, in the present contribution we analyze the influence of three main factors: the intraocular pressure, the intracranial pressure (ICP), and the arterial blood pressure. In particular this last aspect is very important from a clinical viewpoint since it is still unclear if ocular blood flow alterations are primary or secondary to cell death [21]. For what concerns the ICP, we have focused for now our attention on the eye and this quantity is an input parameter of our model, however we envision as next step to include also a CSF description *via* a 0D model [19].

The major region of interest is the lamina cribrosa (Fig. 1), a sponge tissue, which allows the retinal ganglion cells and the central retinal vessels to access the eyeball protected environment. Its biomechanical response is mainly driven by collagen beams that forms a network that gives strength to this tissue but at the same time let the nerve fibers pass through it (Fig. 2). The lamina cribrosa has, therefore, a crucial role in the connection between the eye and the brain—from a neurological viewpoint—and between the eye and the cardiovascular system—from a hemodynamical viewpoint.

The lamina cribrosa is thought to help maintain the balance between the IOP and the ICP, which may influence the ocular blood flow [7]. Thus, this tissue is a critical area in the development of glaucoma, however it is not accessible with standard investigation methods, such as the Goldmann tonometer (to measure IOP), the visual

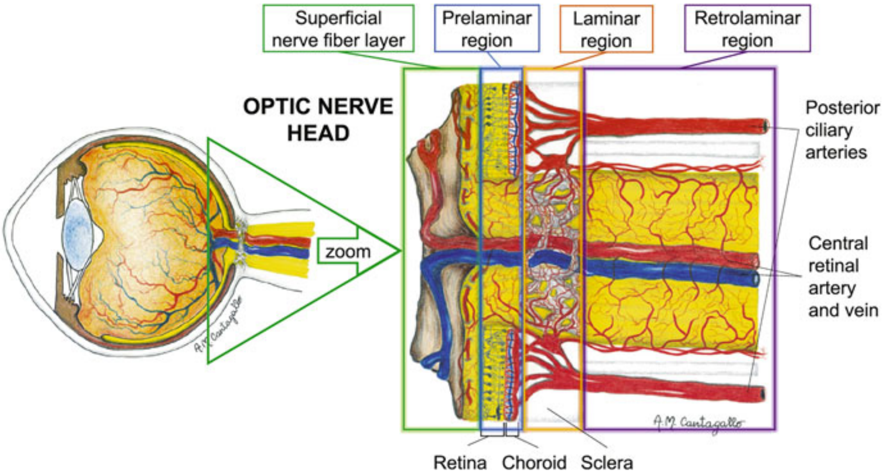


Fig. 1 Lamina cribrosa anatomy and vascular supply. Courtesy of A.M. Cantagallo [8]

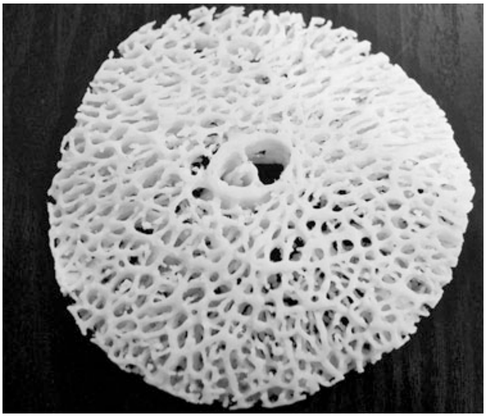


Fig. 2 Micro-structure of the lamina cribrosa. Courtesy of `https://coggle.it`

field test (to test the visual functions), and the Optic Coherence Tomography (to analyze the structure of the optic nerve head and the retina).

2 The Ocular Mathematical Virtual Simulator

We have developed an Ocular Mathematical Virtual Simulator (OMVS) [15, 18], which is able to:

- visualize and estimate in a non-invasive way ocular hemodynamics and biomechanics for tissues that are not easily accessible (e.g. lamina cribrosa);

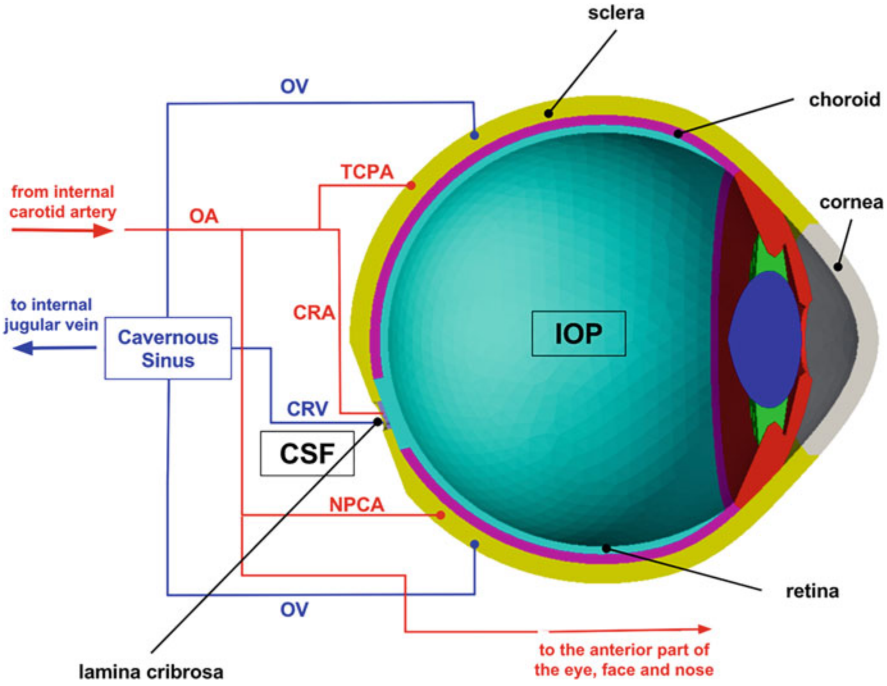


Fig. 3 OMVS framework. CRA = central retinal artery, CRV = central retinal vein, CSF = cerebrospinal fluid, IOP = intraocular pressure, NPCA = nasal posterior ciliary artery, OA = ophthalmic artery, OV = ophthalmic vein, TCPA = temporal posterior ciliary artery

- help isolate the effect of single risk factors and quantify their influence on the multifactorial disease process.

This computational model can be used as a virtual laboratory [17] where to perform virtual experiments. The OMVS (Fig. 3) describes (1) the retinal blood circulation *via* a circuit-based (0D) model; (2) the hemodynamics and biomechanics of the lamina cribrosa *via* a three-dimensional (3D) poroelastic model; (3) the biomechanics of the retina, choroid, sclera and cornea *via* a 3D linear elastic model. More details on this model can be found in [13].

The modeling choice of coupling the 0D blood circulation and the 3D zoom on the lamina has two main motivations:

- combine ocular tissue that have available quantitative data—e.g. blood flow in the central retinal artery (CRA)—and crucial ocular areas that are not accessible with clinical images—e.g. lamina cribrosa perfusion;
- provide a multiscale systemic overview of the blood flow, while maintaining a relatively accessible mathematical complexity and low computational costs.

As a consequence, this 3D-0D coupling is realized with an innovative operator splitting method [16], which is based on a semi-discretization in time inspired by the energy estimates at the continuous level; the algorithm is unconditionally stable with respect of the time-step without requiring sub-iterations between two sub-steps. Moreover, the nonlinearities in the 3D and the 0D models are solved separately with appropriate numerical methods.

For the space discretization of the 3D poroelastic system, we have employed a Hybridizable Discontinuous Galerkin formulation with an integral condition on the interface between the 3D domain and the circuit. This numerical method supports the optimal approximation for the primal and flux variables and the direct solution of this integral condition without any sub-iterations, computing the pressure and the flux on the interface at the same time. Thus, we have obtained a natural spatial coupling between the 3D and the 0D domain [1, 13]. The computational strategy to solve this model includes (1) first order in time discretization, (2) static condensation with integral conditions to get an efficient solution of the HDG system, and (3) algebraic multigrid preconditioners and postprocessing of the primal variables for higher accuracy. All these ingredients are embedded into an efficient parallel computing environment (Feel++) [9].

3 Numerical Study

We present hereafter the results obtained using the web interface of the OMVS [14]. This online tool allows the user to utilize the model without the need of an expertise in mathematics or computer science and to visualize the results of a single simulation or compare multiple outcomes.

3.1 Data

We summarize in Table 1 the virtual database with 4 subjects that we are going to use as input in our simulations. The data that are used to generate the database are derived from experimental values reported in literature, in particular we have for the:

- systolic blood pressure (SP [mmHg]) a Gaussian distribution $\mathcal{N}(116, 23.2)$ [24];
- diastolic blood pressure (DP [mmHg]) a Gaussian distribution $\mathcal{N}(69, 13.8)$ [24];
- IOP ([mmHg]) a Gaussian distribution $\mathcal{N}(17, 6)$ [2];
- ICP ([mmHg]) a Gaussian distribution $\mathcal{N}(9.5, 2.2)$ [11].

The goal of this patient selection is to vary few parameters at a time in order to understand the influence of each parameter variation and disentangle the different effects. The variation of each parameter has been based on experimental results summarized above; in particular to simulate high conditions we have selected the mean value added to the standard deviation. An exception is made for IOP, for which the high value is directly provided by the clinical literature [2].

Table 1 Virtual patient database used for the OMVS simulations

Name	Gender	Age	SP/DP	IOP	ICP	Note
Tony	Male	65 y	116/69	17	9.5	Baseline
John	Male	47 y	116/69	26	9.5	High IOP
Tina	Female	81 y	116/69	26	12.8	High IOP, ICP
Margaret	Female	55 y	150.8/89.7	26	12.8	High IOP, ICP, SP/DP

SP/DP systolic/diastolic blood pressure [mmHg], *IOP* intraocular pressure [mmHg], *ICP* intracranial pressure [mmHg]

3.2 Simulation Results

For this paper we focus our attention on central retinal artery (CRA) and central retinal vein (CRV) blood flows predicted by the OMVS.

Figure 4 displays the comparison between the simulation of Tony's and John's blood flows. In the top part of the figure we have the characteristics CRA blood flow as provided also by other clinical instruments such as the Color Doppler Imaging [23]. The differences in this case between the two subjects are relatively small. On the other hand, CRV simulation results show that the virtual patient with high IOP have a decrease up to 66% in the blood flow (green line vs red line in the bottom of Fig. 4).

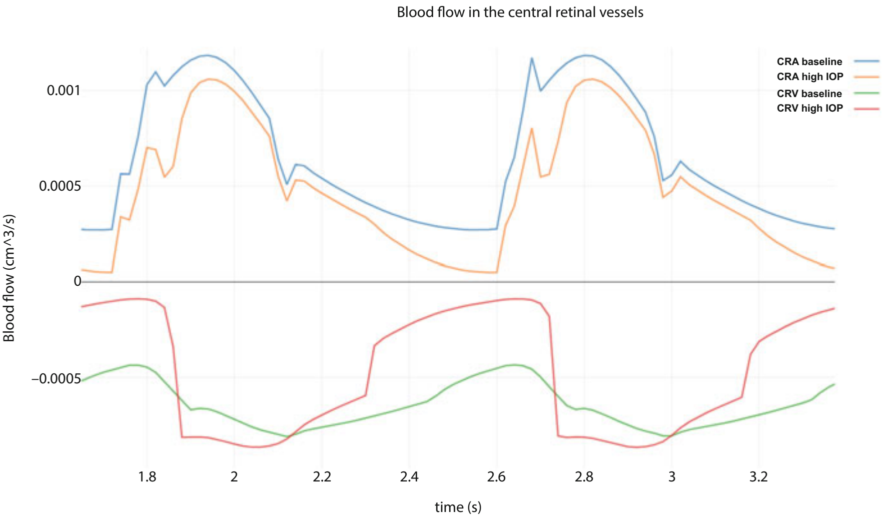


Fig. 4 OMVS simulation results: comparison Tony's and John's blood flows in central retinal vessels

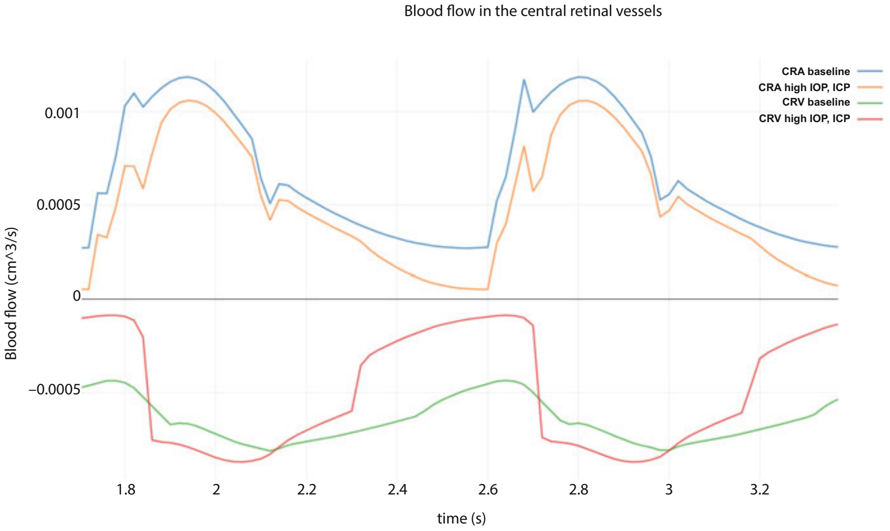


Fig. 5 OMVS simulation results: comparison Tony's and Tina's blood flows in central retinal vessels

Moreover, the OMVS predicts that this reduction is slightly less marked (64%) if we compare our baseline subject (Tony) with patients that suffer from high ICP (Tina) as pointed out by Fig. 5 (bottom, green line vs red line).

The most interesting simulation result, from the clinical viewpoint, is the one presented in Fig. 6. In this virtual clinical case, the drop in CRV blood flow between the healthy subject and the hypertensive patient with high IOP and ICP is 50% and we highlight also how this difference lasts remarkably less in time (narrower drop) than the previous two comparisons.

3.3 *Discussion*

The reduction in the CRV blood flow can be associated to collapsed veins due to high IOP, which may lead to disrupted perfusion of the ocular tissues. Figs. 4 and 5 exhibit virtually this situation. Figure 6, however, conjectures that the effect of collapsed veins, leading therefore to abnormal blood flow, is balanced by high blood pressure, obtaining a situation closer to the baseline.

Note that, concerning the validation of our simulations, these theoretical predictions on the relationship between intraocular pressure, blood pressure and ocular perfusion have been confirmed by an independent population-based study including nearly 10000 individuals [20].

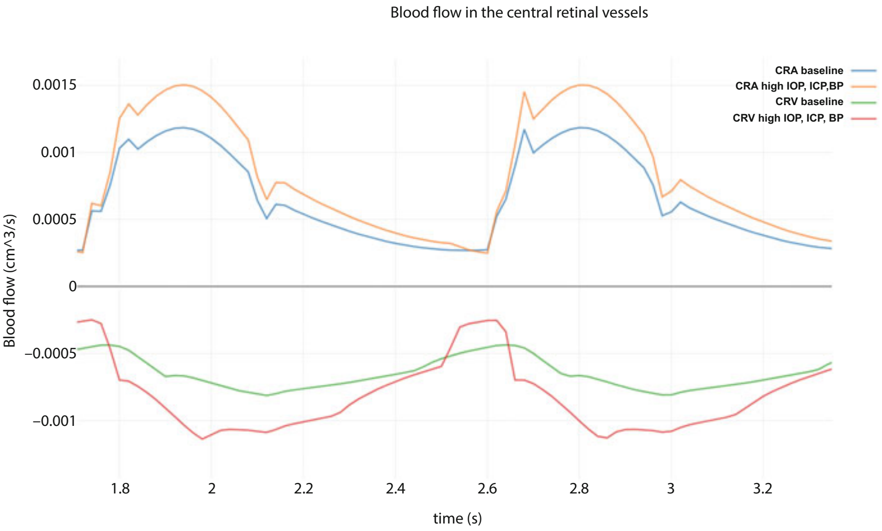


Fig. 6 OMVS simulation results: comparison Tony's and Margaret's blood flows in central retinal vessels

4 Conclusions

The virtual experiment we have described above is an example how the OMVS can be employed as a clinical tool to disentangle different factors that may affect the physiological behavior of the ocular system. In the present contribution, we focused on the simulations in the central retinal vessels, which are the result of more complex mechanisms involving other tissues of the eye. For instance, the IOP is acting directly on the intraocular segments of the CRA and CRV, but also on the biomechanics of the lamina cribrosa, which influences in turn the perfusion of this tissue that alters, in a feedback loop, the central vessels hemodynamics.

In conclusion, motivated by open questions in ophthalmology, we have developed a mathematical model that combines innovative numerical methods and high performance computing in order to propose new complementary tools of data analysis and visualization for clinical and experimental research. Moreover, this multiscale model serves as a virtual laboratory in which it is possible to simulate the behavior and the interactions between the hemodynamics and biomechanics within the eye.

References

1. S. Bertoluzza, G. Guidoboni, R. Hild, D. Prada, C. Prud'homme, R. Sacco, L. Sala, M. Szopos. *A HDG method for elliptic problems with integral boundary condition: Theory and Applications*. Submitted.

2. T. Colton, F. Ederer. *The distribution of intraocular pressures in the general population.* Survey of ophthalmology. 1980. 25(3):123-129.
3. D. W. Evans, A. Harris, M. Garrett, H. S. Chung, L. Kagemann. *Glaucoma patients demonstrate faulty autoregulation of ocular blood flow during posture change.* British Journal of Ophthalmology. 1999. 83(7):809-813.
4. F. Galassi, A. Sodi, F. Ucci, A. Harris, H.S. Chung. *Ocular haemodynamics in glaucoma associated with high myopia.* International ophthalmology. 1998. 22(5): 299-305.
5. L.W. Herndon, J. S. Weizer, S. S Stinnett. *Central corneal thickness as a risk factor for advanced glaucoma damage.* Archives of ophthalmology. 2004. 122(1):17-21.
6. S. Kingman. *Glaucoma is second leading cause of blindness globally.* Bulletin of the World Health Organization. 2004. 82:887-888, 2004.
7. B. Marek, A. Harris, P Kanakamedala, E. Lee, A. Amireskandari, L. Carichino, G. Guidoboni, L. Abrams Tobe, B. Siesky. *Cerebrospinal fluid pressure and glaucoma: regulation of trans-lamina cribrosa pressure.* British Journal of Ophthalmology. 2014. 98(6):721-725.
8. D. Prada. *A hybridizable discontinuous Galerkin method for nonlinear porous media viscoelasticity with applications in ophthalmology.* PhD thesis. IUPUI. 2016.
9. C. Prud'homme, V. Chabannes, V. Doyeux, M. Ismail, A. Samake, G. Pena. *Feel++: A computational framework for galerkin methods and advanced numerical methods.* 2012. In ESAIM: Proceedings, volume 38, pages 429-455. EDP Sciences.
10. H. A. Quigley. *Ganglion cell death in glaucoma: pathology recapitulates ontogeny.* Australian and New Zealand journal of ophthalmology. 1995. 23(2):85-91.
11. R. Ren, J. B. Jonas, G. Tian, Y. Zhen, K. Ma, S. Li, H. Wang, B. Li, X. Zhang, N. Wang. *Cerebrospinal fluid pressure in glaucoma: a prospective study.* Ophthalmology. 2010. 117(2):259-266.
12. A. R. Rudnicka, S. Mt-Isa, C. G. Owen, D. G. Cook, D. Ashby. *Variations in primary open-angle glaucoma prevalence by age, gender, and race: a Bayesian meta-analysis.* Investigative ophthalmology & visual science. 2006. 47(10):4254-4261.
13. L. Sala. *Mathematical modelling and simulation of ocular blood flows and their interactions.* PhD thesis. University of Strasbourg. 2019.
14. L. Sala, G. Guidoboni, C. Prud'homme, M. Szopos, A. C. Verticchio Vercellin, B. A. Siesky, A. Harris. *A web-based interface for ocular hemodynamics and biomechanics analysis via the ocular mathematical virtual simulator.* Investigative Ophthalmology & Visual Science. 2019. 60(9):4277-4277.
15. L. Sala, C. Prud'homme, G. Guidoboni, M. Szopos. *Ocular mathematical virtual simulator: A hemodynamical and biomechanical study towards clinical applications.* Journal of Coupled Systems and Multiscale Dynamics. 2018. 6(3):241-247.
16. L. Sala, C. Prud'homme, G. Guidoboni, M. Szopos. *An operator splitting method for the time discretization of a multi-scale model in ophthalmology.* SMAI congress. 2019.
17. L. Sala, C. Prud'homme, D. Prada, F. Salerni, C. Trophime, V. Chabannes, M. Szopos, R. Repetto, S. Bertoluzza, R. Sacco, A. Harris. *Patient-specific virtual simulator of tissue perfusion in the lamina cribrosa.* Investigative Ophthalmology & Visual Science. 2017. 58(8): 727.
18. L. Sala, R. Sacco, G. Guidoboni. *Multiscale modeling of ocular physiology.* Journal for Modeling in Ophthalmology. 2018. 2(1):12-18.
19. F. Salerni, R. Repetto, A. Harris, P. Pinsky, C. Prud'homme, M. Szopos, G. Guidoboni. *Biofluid modeling of the coupled eye-brain system and insights into simulated microgravity conditions.* PloS One. 2019. 14(8): e0216012.
20. Y.-C. Tham, S.-H. Lim, P. Gupta, T. Aung, T. Y. Wong, and C.-Y. Cheng. *Inter-relationship between ocular perfusion pressure, blood pressure, intraocular pressure profiles and primary open-angle glaucoma: the Singapore epidemiology of eye diseases study.* British Journal of Ophthalmology. 2018. 102(10):1402-1406.

21. F. Topouzis, A. L. Coleman, A. Harris, C. Jonescu-Cuypers, F. Yu, L. Mavroudis, E. Anastasopoulos, T. Pappas, A. Koskosas, M. R. Wilson. *Association of blood pressure status with the optic disk structure in non-glaucoma subjects: the thessaloniki eye study.* American journal of ophthalmology. 2006. 142(1):60-67.
22. R. N. Weinreb, A. Harris. *Ocular blood flow in glaucoma, volume 6.* Kugler Publications. 2009.
23. T. H. Williamson, A. Harris. *Color Doppler ultrasound imaging of the eye and orbit.* Survey of ophthalmology. 1996. 40(4):255-267.
24. J.D. Wright, J. P. Hughes, Y. Ostchega, S. S. Yoon, T. Nwankwo. *Mean systolic and diastolic blood pressure in adults aged 18 and over in the United States, 2001-2008.* Natl Health Stat Report. 2011. 35(1-22):24.

Well-Balanced and Asymptotic Preserving IMEX-Peer Methods

Moritz Schneider and Jens Lang

Abstract Peer methods are a comprehensive class of time integrators offering numerous degrees of freedom in their coefficient matrices that can be used to ensure advantageous properties, e.g. A-stability or super-convergence. In this paper, we show that implicit-explicit (IMEX) Peer methods are well-balanced and asymptotic preserving by construction without additional constraints on the coefficients. These properties are relevant when solving (the space discretisation of) hyperbolic systems of balance laws, for example. Numerical examples confirm the theoretical results and illustrate the potential of IMEX-Peer methods.

1 Introduction

Implicit-explicit (IMEX) Peer methods are designed to efficiently solve large systems of differential equations (ODEs)

$$u' = F_0(u) + F_1(u), \qquad u(0) = u_0 \in \mathbb{R}^m, \quad m \geq 1 \tag{1}$$

that arise in the modelling of various dynamical processes in engineering, physics, chemistry and other areas. Due to their special structure, IMEX-Peer methods treat the non-stiff part F_0 explicitly and the stiff contribution F_1 implicitly, thus combining the advantage of lower costs for explicit schemes with the favourable stability of implicit solvers to enhance the overall computational efficiency.

Peer methods are two-step methods with s internal stages and belong to the class of general linear methods that were introduced and described in detail by Butcher [2]. A specific feature of Peer methods is that all stages in each time step have the same order of consistency and, hence, order reduction is avoided.

M. Schneider (✉) · J. Lang
Technical University of Darmstadt, Darmstadt, Germany
e-mail: moschneider@mathematik.tu-darmstadt.de; lang@mathematik.tu-darmstadt.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_85

There is a wide range of literature concerning the different aspects of Peer methods and we will only give a short overview. More details can be found in the introductory chapters of [10, 11]. Peer methods were introduced by Schmitt and Weiner in 2004 [9]. The construction of IMEX-Peer methods via extrapolation has been applied by several authors [3, 7]. An alternative construction using partitioned methods is given in [12, 16]. Since the coefficient matrices of Peer methods offer many degrees of freedom, the construction of super-convergent schemes [10, 14, 15] and the adaptation to variable step sizes [11, 13] is possible.

Throughout this paper, we consider s-stage Peer methods of the form

$$w_{n+1} = Pw_n + \Delta t\hat{Q}F_0(w_n) + \Delta t\hat{R}F_0(w_{n+1}) + \Delta tQF_1(w_n) + \Delta tRF_1(w_{n+1}) \tag{2}$$

with $\hat{Q} = Q + RS_1$ and $\hat{R} = RS_2$ as given in [10]. Here, P, Q, R, S_1 and S_2 are $s \times s$ coefficient matrices. The matrix R is taken to be lower triangular with constant diagonal $\gamma > 0$ and S_2 is strictly lower triangular. The approximations in each time step are denoted by

$$w_n = \left[w_{n,1}^T, \dots, w_{n,s}^T\right]^T \in \mathbb{R}^{s\cdot m}, \qquad w_{n,i} \approx u(t_n + c_i\Delta t), \tag{3}$$

where $t_n = n\Delta t, n \geq 0$ and the nodes $c_1, \dots, c_s \in \mathbb{R}$, corresponding to the s stages, are such that $c_i \neq c_j$ if $i \neq j$ and $c_s = 1$. The application of F_i is meant component-wise, i.e. $F_i(w_n) = \left[F_i(w_{n,1})^T, \dots, F_i(w_{n,s})^T\right]^T, \; i = 0, 1$. For the sake of notation, we use for an $s \times s$ matrix M the same symbol for its Kronecker product with the $m \times m$ identity matrix $M \otimes I_m$ as a mapping from the space $\mathbb{R}^{s\cdot m}$ to itself. An extensive analysis of consistency and stability along with the construction of super-convergent methods as well as the adaption to variable step sizes is given in [10, 11].

In this paper, we show that our recently developed super-convergent IMEX-Peer methods [10] possess two additional properties that are important when dealing with hyperbolic balance and conservation laws: They are *well-balanced* and *asymptotic preserving*. We restrict the analysis to the setting of constant step sizes. However, the results hold true for Peer methods applied with variable step sizes as well.

For further investigation, we follow the approach of Boscarino and Pareschi [1] and consider the hyperbolic system of balance laws

$$U_t + F(U)_x = G(U), \tag{4}$$

where $U \in \mathbb{R}^N$ and $F, G : \mathbb{R}^N \to \mathbb{R}^N$. Usually, $F(U)$ gives the flux and $G(U)$ is the source.

The remainder is organised as follows. In Sect. 2, we show that IMEX-Peer methods are well-balanced without additional constraints on the coefficients. The same holds true for the asymptotic preservation property as analysed in Sect. 3. Numerical experiments to illustrate the theoretical results are given in Sect. 4.

2 Well-Balanced IMEX-Peer Methods

The steady-state U^* of the hyperbolic system of balance laws (4) is characterized by

$$U_t^* \equiv 0 \iff F(U^*)_x = G(U^*). \tag{5}$$

Accordingly, a numerical scheme is called *well-balanced*, if it preserves the steady-state solution U^* as characterized in (5). Since Peer methods are time integrators, we focus on the influence of time discretisation on the well-balanced property of the numerical solution. Thus, we discretise (4) in space and obtain the system of ODEs

$$u'(t) = F_0(u(t)) + F_1(u(t)) \tag{6}$$

with non-stiff (flux) function F_0 and stiff source term F_1. Analogously to (5), the steady-state u^* is now described by

$$u^{*\prime}(t) \equiv 0 \iff F_0(u^*) + F_1(u^*) = 0.$$

Assume that the numerical solution of (6) yields an approximation $v_n \approx u(t_n) \in \mathbb{R}^m$ satisfying

$$F_0(v_n) + F_1(v_n) = 0.$$

Then, in some sense $v_n' = 0$ holds and, in order to capture the steady-state, we claim

$$u(t_{n+1}) \approx v_{n+1} = v_n.$$

This concept works well for one-step methods as discussed in [1]. Since we are dealing with two-step methods, a small modification is needed to take into account all values of the previous time step. This leads to the following definition of well-balanced IMEX-Peer methods.

Definition 1 An s-stage IMEX-Peer method (2) is called *well-balanced* if

$$F_0(w_n) + F_1(w_n) = 0 \tag{7}$$

implies $w_{n+1} = w_n$, where $w_{n,1} = \cdots = w_{n,s}$.

Now, we can prove the following theorem.

Theorem 1 *IMEX-Peer methods of the form (2) with coefficient matrices that satisfy the standard consistency conditions from [10]*

$$Pe = e \qquad \text{and} \qquad S_1 = (I_s - S_2)V_0V_1^{-1},$$

where $V_0 = (c_i^{j-1})_{i,j}$ *and* $V_1 = ((c_i - 1)^{j-1})_{i,j}$, *are* well-balanced *in the sense of Definition 1, given that* (2) *has a unique solution* w_{n+1} *for* Δt *sufficiently small.*

Proof By Definition 1, we have $F_0(w_n) + F_1(w_n) = 0$ with $w_n = e \otimes w_{n,s}$ and $F_i(w_n) = e \otimes F_i(w_{n,s})$, $i = 0, 1$, where $e = (1, \ldots, 1)^T \in \mathbb{R}^s$. Under the hypotheses of Theorem 1 stated above, we prove that (2) implies $w_{n+1} = w_n$:

$$\begin{aligned} w_{n+1} &= Pw_n + \Delta t(\hat{Q}F_0(w_n) + QF_1(w_n)) + \Delta t(\hat{R}F_0(w_{n+1}) + RF_1(w_{n+1})) \\ &= (P \otimes I_m)(e \otimes w_{n,s}) + \Delta t((R(S_1 + S_2 - I_s)) \otimes I_m)(e \otimes F_0(w_{n,s})). \end{aligned}$$

Using $R(S_1 + S_2 - I_s) = R(I_s - S_2)(V_0 V_1^{-1} - I_s)$ and $V_0 V_1^{-1} e = V_0 e_1 = e$, where $e_1 = (1, 0, \ldots, 0)^T \in \mathbb{R}^s$, the second term vanishes and we obtain $w_{n+1} = w_n$. □

In practice, we cannot expect (7) to hold for all stage values of w_n but rather that when the numerical solution converges to the steady-state, we will reach a point in time when all stage values are sufficiently similar and the last stage of the time step satisfies $F_0(w_{n,s}) + F_1(w_{n,s}) = 0$. Then, the numerical scheme should reproduce the steady-state. It can be shown that if

$$F_0(w_{n,s}) + F_1(w_{n,s}) = 0 \quad \text{and} \quad w_{n,i} = w_{n,s} + \mathcal{O}(\varepsilon), \quad i = 1, \ldots, s-1,$$

we obtain for continuous F_0 and F_1, analogously to the proof of Theorem 1,

$$w_{n+1} = w_n + \mathcal{O}(\varepsilon).$$

Hence, the well-balanced property is beneficial in practical applications even if the strong condition (7) is not fulfilled exactly.

3 Asymptotic Preserving IMEX-Peer Methods

We investigate the behaviour of IMEX-Peer methods when the hyperbolic balance laws (4) are scaled with a parameter $\varepsilon > 0$. This is discussed in detail by Chen, Levermore and Liu [4] and has been adopted for IMEX Runge-Kutta methods by Boscarino and Pareschi and for multistep methods by Dimarco and Pareschi [1, 5].

Scaling the space and time variables in (4) with a parameter $\varepsilon > 0$ leads to

$$U_t^{\varepsilon} + F(U^{\varepsilon})_x = \frac{1}{\varepsilon} G(U^{\varepsilon}). \tag{8}$$

We are interested in the performance of numerical schemes that solve (8) for $\varepsilon \to 0$. Taking the limit $\varepsilon \to 0$ analytically yields the system of algebraic equations

$$G(U^0) = 0. \tag{9}$$

Following the analysis in [1, 4], we assume that $G(U)$ with $U \in \mathbb{R}^N$ is a dissipative relaxation operator, i.e., there exists an $M \times N$ matrix C with $\operatorname{rank}(C) = M < N$ and

$$CG(U) = 0 \qquad \text{for all } U \in \mathbb{R}^N. \tag{10}$$

We set $u = CU \in \mathbb{R}^M$ to be the vector of conservation quantities. Further, each such u uniquely defines a local equilibrium value

$$U = E(u) \tag{11}$$

that satisfies

$$0 = G(E(u)) = G(U) \tag{12}$$

and $u = CE(u) = CU$.

Therefore, for every solution U^0 of (9), we find a uniquely determined vector of conserved quantities u^0 such that $U^0 = E(u^0)$. Going back to (8) and multiplying with C, we obtain

$$CU_t^\varepsilon + CF(U^\varepsilon)_x = \frac{1}{\varepsilon} CG(U^\varepsilon)$$

and, hence, for $\varepsilon \to 0$, a system of M conservation laws

$$(CU^0)_t + (CF(U^0))_x = 0.$$

Using the equilibrium approximation $U^0 = E(u^0)$ with $CE(u^0) = CU^0 = u^0$ from above, we have

$$(CE(u^0))_t + (CF(E(u^0)))_x = 0$$

and, finally, obtain the typical system of conservation laws

$$u_t^0 + f(u^0)_x = 0 \tag{13}$$

with $f(\cdot) = CF(E(\cdot))$.

In the following, we verify that IMEX-Peer methods, as defined in (2) and (3), capture the asymptotic behaviour described above. To this end, we write (8) in the standard form for IMEX-Peer methods (1) and define

$$F_0(U^\varepsilon) = -F(U^\varepsilon)_x \qquad \text{and} \qquad F_1(U^\varepsilon) = \frac{1}{\varepsilon} G(U^\varepsilon), \tag{14}$$

where we identify U^ε, $F(U^\varepsilon)_x$ and $G(U^\varepsilon)$ with the corresponding spatial discretisations for the sake of enhanced readability. Applying an IMEX-Peer method (2) to (8) and (14) gives

$$U^\varepsilon_{n+1} = PU^\varepsilon_n - \Delta t\hat{Q}F(U^\varepsilon_n)_x - \Delta t\hat{R}F(U^\varepsilon_{n+1})_x + \frac{\Delta t}{\varepsilon}QG(U^\varepsilon_n) + \frac{\Delta t}{\varepsilon}RG(U^\varepsilon_{n+1}). \tag{15}$$

As in the continuous case, $\varepsilon \to 0$ yields

$$QG(U^0_n) + RG(U^0_{n+1}) = 0. \tag{16}$$

At this point, we are faced with a typical problem occurring for multi-step methods: We have to introduce an additional condition on the values of the previous time step U^ε_n. This is reasonable since the contribution of U^ε_n via $QG(U^\varepsilon_n)$ depends on the specific choice of $G(\cdot)$ and cannot be compensated by $RG(U^\varepsilon_{n+1})$ independently of $G(\cdot)$. Hence, we claim *well-prepared* initial values [1, 6].

Definition 2 The initial data U^ε_n for (15) is said to be *well-prepared* if

$$U^\varepsilon_n = E(u^\varepsilon_n) + \mathcal{O}(\varepsilon).$$

This allows us to formulate the following.

Theorem 2 *Assume the initial data is well-prepared. Then, in the limit* $\varepsilon \to 0$*, an IMEX-Peer method* (2) *applied to* (8) *becomes the explicit Peer method* ($F_1 \equiv 0$) *applied to the equilibrium system* (13)*.*

Proof Since the initial values are well-prepared, we obtain for the limit $\varepsilon \to 0$

$$RG(U^0_{n+1}) = 0 \implies G(U^0_{n+1}) = 0$$

from (12) and (16) since R is regular. Analogously to the continuous case, we define

$$u^0_{n+1} = (I_s \otimes C)U^0_{n+1} = \left[(CU^0_{n+1,1})^T, \dots, (CU^0_{n+1,s})^T\right]^T.$$

We set $C \bullet U = (I_s \otimes C)U$ for any $U \in \mathbb{R}^{N\cdot s}$. As in the continuous case (11), the local equilibrium values U^0_{n+1} are defined by u^0_{n+1} via

$$U^0_{n+1} = E(u^0_{n+1})$$

where $G(E(u^0_{n+1})) = 0$ and $C \bullet E(u^0_{n+1}) = u^0_{n+1}$.

Multiplying (15) with C and using that for any $M \in \mathbb{R}^{s\times s}$

$$C \bullet (M \otimes I_N)\left[U^{\varepsilon T}_{n,1}, \dots, U^{\varepsilon T}_{n,s}\right]^T = (M \otimes I_M)\left(C \bullet \left[U^{\varepsilon T}_{n,1}, \dots, U^{\varepsilon T}_{n,s}\right]^T\right),$$

as well as $MU_n^\varepsilon = (M \otimes I_N)U_n^\varepsilon$, we find

$$C \bullet U_{n+1}^\varepsilon = P(C \bullet U_n^\varepsilon) - \Delta t \hat{Q}\left(C \bullet F(U_n^\varepsilon)_x\right) - \Delta t \hat{R}\left(C \bullet F(U_{n+1}^\varepsilon)_x\right) + \frac{\Delta t}{\varepsilon} Q\left(C \bullet G(U_n^\varepsilon)\right) + \frac{\Delta t}{\varepsilon} R\left(C \bullet G(U_{n+1}^\varepsilon)\right). \tag{17}$$

We recall from (10) that $C \bullet G(U) = 0$ for all $U \in \mathbb{R}^{N\cdot s}$. Hence, (17) reduces to

$$C \bullet U_{n+1}^\varepsilon = P(C \bullet U_n^\varepsilon) - \Delta t \hat{Q}\left(C \bullet F(U_n^\varepsilon)_x\right) - \Delta t \hat{R}\left(C \bullet F(U_{n+1}^\varepsilon)_x\right).$$

For $\varepsilon \to 0$, we replace U_n^ε by $E(u_n^0)$ and use $C \bullet E(u_n^0) = u_n^0$ to obtain

$$u_{n+1}^0 = Pu_n^0 - \Delta t \hat{Q}(C \bullet F(E(u_n^0))_x) - \Delta t \hat{R}(C \bullet F(E(u_{n+1}^0))_x). \tag{18}$$

Setting $f(\cdot) = C \bullet F(E(\cdot))$, we observe that (18) coincides with the application of IMEX-Peer method (2) to the system of conservation laws (13)

$$u_t^0 = -f(u^0)_x, \tag{19}$$

where $F_0(u^0) = -f(u^0)_x$ and $F_1 \equiv 0$, giving an explicit scheme. □

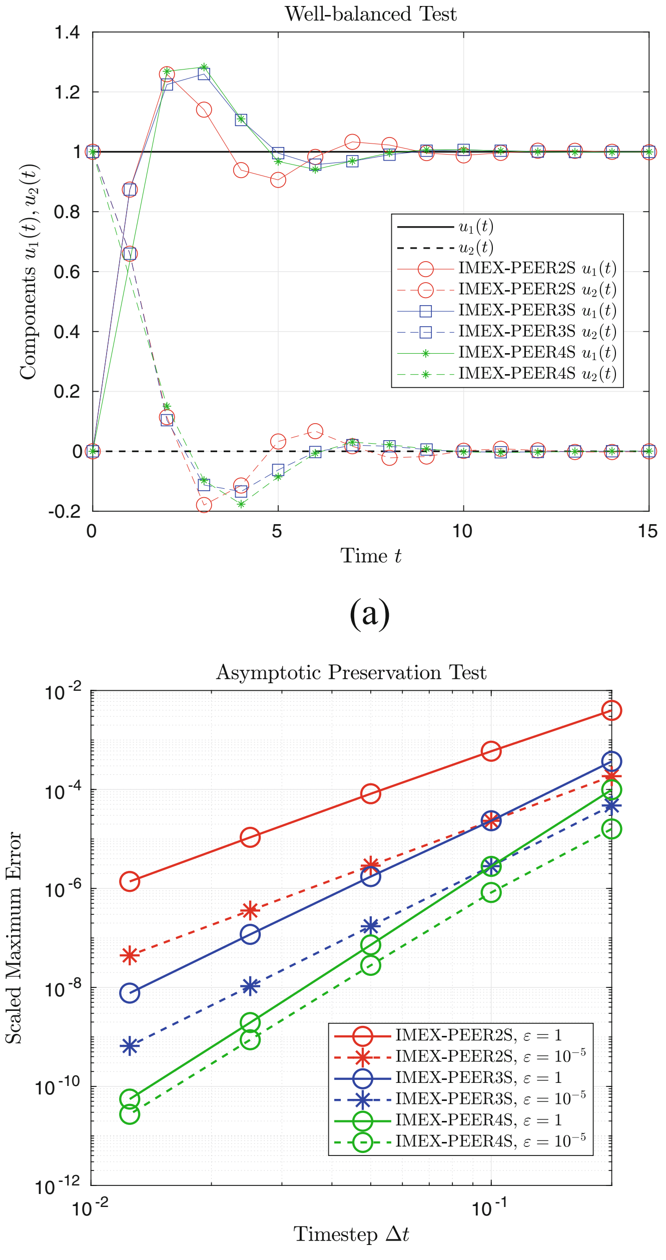
We see that instead of solving (19) explicitly, we can equivalently apply an IMEX-Peer method to the relaxed system (8), thus profiting from the stabilisation of the implicit part.

4 Numerical Examples

We present two numerical examples to illustrate that IMEX-Peer methods are well-balanced and asymptotic preserving. Since we focus on time integration, we consider systems of ODEs where no spatial discretisation is necessary and apply our recently developed super-convergent methods from [10].

Example 1 (Well-Balanced Property) We demonstrate the effect of the well-balanced property using a system of ODEs of form (1) with non-stiff part $F_0(u) = [u_2, -u_1]^T$ and stiff contribution $F_1(u) = [0, 1 - u_2]^T$ as introduced by Boscarino and Pareschi in [1]. The unique equilibrium point is $u^* = [1, 0]^T$.

In Fig. 1, the behaviour of the numerical approximation of the solution components u_1 and u_2 for $t \in [0, 15]$ is given using IMEX-Peer methods from [10]. Starting with $u_0 = [0, 1]^T$, we observe that the IMEX-Peer methods reach the equilibrium after a short time even for a large step size $\Delta t = 1$. Boscarino and Pareschi demonstrate in [1] that this is usually not the case if the time integrator is not well-balanced.



(a)

(b)

Fig. 1 Numerical results for super-convergent and A-stable s-stage IMEX-Peer methods IMEX-Peer2s, IMEX-Peer3s, and IMEX-Peer4s from [10]. (**a**) Results for the well-balanced test. All methods capture the equilibrium after short time and for a large time step $\Delta t = 1$. (**b**) Results for the asymptotic preservation test. The super-convergent methods keep their order of $s + 1$ for various choices of ε

Example 2 (Asymptotic Preservation Property) To verify that our super-convergent IMEX-Peer methods developed in [10] are asymptotic preserving, we consider a stiff system of ODEs (1) where $F_0(u) = [-u_2, u_1]^T$ and $F_1(u) = \frac{1}{\varepsilon}[0, \sin u_1 - u_2]^T$ with scaling parameter $\varepsilon > 0$, initial values $u(0) = [\pi/2, 1]^T$ and $t \in [0, 5]$ as given by Pareschi and Russo in [8].

Numerical results for $\Delta t = 0.2 \cdot 2^{-i}, i = 0, \ldots, 4$ and $\varepsilon = 1, 10^{-5}$ are given in Fig. 1. The error is computed using the scaled maximum error norm over all time steps $err = \max_{0 \le t_n \le 5} \max_{i=1,2} |U_i - u_i|/(1 + |u_i|)$, where U is the approximate solution and u is a reference solution computed using the MATLAB routine ODE15S.

We observe that the orders of convergence of the super-convergent methods IMEX-Peer2s, IMEX-Peer3s and IMEX-Peer4s with stage number $s = 2, 3, 4$ from [10], derived using a least squares fit, are 2.9, 3.9, 5.2 for $\varepsilon = 1$ and 3.0, 4.0, 4.8 for $\varepsilon = 10^{-5}$. Hence, they match the theoretical orders $s + 1$ nicely.

We remark that the order of convergence is affected in an intermediate region $\Delta t = \mathcal{O}(\varepsilon)$, see [5, 8] for further details. Nevertheless, the order of convergence is fully restored when Δt leaves the regime $\mathcal{O}(\varepsilon)$, therefore, the drawback is negligible [8].

In conclusion, we have shown that IMEX-Peer methods are well-balanced and asymptotic preserving by construction and, hence, suitable for the preservation of steady-states and the capture of asymptotic limits for space-time scaling.

Acknowledgments J. Lang was supported by the German Research Foundation within the collaborative research center TRR154 "Mathematical Modeling, Simulation and Optimisation Using the Example of Gas Networks" (DFG-SFB TRR154/2-2018, TP B01) and the Graduate Schools Computational Engineering (DFG GSC233) and Energy Science and Engineering (DFG GSC1070).

References

1. Boscarino, S., Pareschi, L.: On the asymptotic properties of IMEX Runge–Kutta schemes for hyperbolic balance laws, J. Comput. Appl. Math. **316**, 60–73 (2017)
2. Butcher, J.: General linear methods, Acta Numerica **15**, 157–256 (2006)
3. Cardone, A., Jackiewicz, Z., Sandu, A., Zhang, H.: Extrapolation-based implicit-explicit general linear methods, Numer. Algorithms **65**, 377–399 (2014)
4. Chen, G., Levermore, C., Liu, T.: Hyperbolic Conservation Laws with Stiff Relaxation Terms and Entropy, Commun. Pure Appl. Math. **47**, 787–830 (1994)
5. Dimarco, G., Pareschi, L.: Implicit-explicit linear multistep methods for stiff kinetic equations, SIAM J. Num. Anal. **55**, 664–690 (2017)
6. Filbert, F., Jin, S.: A class of asymptotic-preserving schemes for kinetic equations and related problems with stiff sources, J. Comp. Phys. **229**, 7625–7648 (2010)
7. Lang, J., Hundsdorfer, W.: Extrapolation-based implicit-explicit Peer methods with optimised stability regions, J. Comp. Phys. **337**, 203–215 (2017)
8. Trigiante, D.: Recent trends in numerical analysis. Nova Science Publishers, New York (2000)
9. Schmitt, B.A., Weiner, R.: Parallel two-step W-methods with peer variables, SIAM J. Numer. Anal. **42**, 265–282 (2004)

10. Schneider, M., Lang, J., Hundsdorfer, W.: Extrapolation-based super-convergent implicit-explicit Peer methods with A-stable implicit part, J. Comp. Phys. **367**, 121–133 (2018)
11. Schneider, M., Lang, J., Weiner, R.: Super-convergent implicit–explicit Peer methods with variable step sizes, J. Comput. Appl. Math. (2019) https://doi.org/10.1016/j.cam.2019.112501
12. Soleimani, B., Knoth, O., Weiner, R.: IMEX Peer methods for fast-wave-slow-wave problems, Appl. Numer. Math. **118**, 221–237 (2017)
13. Soleimani, B., Weiner, R.: A class of implicit Peer methods for stiff systems, J. Comput. Appl. Math. **316**, 358–368 (2017)
14. Soleimani, B., Weiner, R.: Superconvergent IMEX Peer methods, Appl. Numer. Math. **130**, 70–85 (2018)
15. Weiner, R., Schmitt, B.A., Podhaisky, H., Jebens, S.: Superconvergent explicit two-step peer methods, J. Comput. Appl. Math. **223**, 753–764 (2009)
16. Zhang, H., Sandu, A., Blaise, S.: Partitioned and implicit-explicit general linear methods for ordinary differential equations, J. Sci. Comput. **61**, 119–144 (2014)

Approximation Schemes for Viscosity Solutions of Fully Nonlinear Stochastic Partial Differential Equations

Benjamin Seeger

Abstract We develop a method for constructing convergent approximation schemes for viscosity solutions of fully nonlinear stochastic partial differential equations. Our results apply to explicit finite difference schemes and Trotter-Kato splitting formulas, and error estimates are found for schemes approximating solutions of stochastic Hamilton-Jacobi equations.

1 Introduction

We develop a general program for constructing numerical schemes to approximate pathwise viscosity solutions of the initial value problem

$$\begin{cases} du = F(D^2u, Du)\,dt + \sum_{i=1}^{m} H^i(Du)\cdot dW^i \quad \text{in } \mathbb{R}^d \times (0,T] \quad \text{and} \\ u(\cdot,0) = u_0 \quad \text{in } \mathbb{R}^d, \end{cases} \tag{1}$$

where $T > 0$, $F \in C^{0,1}(\mathbb{S}^d \times \mathbb{R}^d)$[1] is degenerate elliptic, $H \in C^2(\mathbb{R}^d)$, $W = (W^1, W^2, \ldots, W^m) \in C([0,T], \mathbb{R}^m)$, and $u_0 \in BUC(\mathbb{R}^d)$.[2]

When W is continuously differentiable or of bounded variation, (1) falls within the scope of the theory of viscosity solutions; see, for instance, [5]. However, a general continuous path W may be nowhere differentiable and have unbounded variation on every open interval, as is the case, for example, for Brownian paths

[1] $\mathbb{S}^d$ is the space of symmetric $d \times d$ matrices.

[2] $BUC(\mathbb{R}^d)$ is the space of bounded and uniformly continuous functions on $\mathbb{R}^d$.

B. Seeger (✉)
Collège de France and Université Paris – Dauphine (CEREMADE), Paris, France
e-mail: seeger@ceremade.dauphine.fr; bseeger@math.uchicago.edu

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_86

with probability one. For such paths, the study of (1) requires the theory of pathwise (or stochastic) viscosity solutions put forth by Lions and Souganidis [10–13].

In view of the robust stability properties of viscosity solutions, there is an extensive literature on the construction of approximation schemes for fully nonlinear equations, initiated by Crandall and Lions [6] and Souganidis [15, 16], who found error estimates for convergent approximations of Hamilton-Jacobi equations, and extended to second order equations by Barles and Souganidis [2] with a qualitative proof of convergence. Rates of convergence in the second order case have also been obtained in various cases, see for instance [1, 4, 8, 9, 17].

It turns out [14] that pathwise viscosity solutions are also quite amenable to various approximation schemes, although the methods are more involved due to the presence of the singular terms dW^i, as we describe in what follows.

2 A Summary of the Main Results

We discuss first the general algorithm for the construction of schemes, and we present some specific examples to illustrate its use.

2.1 The Scheme Operator

The central object to be constructed is the scheme operator, which, for $h > 0$, $0 \leq s \leq t \leq T$, and $\zeta \in C([0,T];\mathbb{R}^m)$, is a map $S_h(t,s;\zeta) : BUC(\mathbb{R}^d) \to BUC(\mathbb{R}^d)$. Then, given a partition $\mathcal{P} := \{0 = t_0 < t_1 < \cdots, t_N = T\}$ of $[0,T]$ with mesh-size $|\mathcal{P}| := \max_{n=0,1,\dots,N-1}(t_{n+1} - t_n)$ and a path $\zeta \in C([0,T];\mathbb{R}^m)$, we define the function $\tilde{u}_h(\cdot;\zeta,\mathcal{P})$ by

$$\begin{cases} \tilde{u}_h(\cdot,0;\zeta,\mathcal{P}) := u_0 \quad \text{and, for } n = 0,1,\dots,N-1 \text{ and } t \in (t_n,t_{n+1}], \\ \tilde{u}_h(\cdot,t;\zeta,\mathcal{P}) := S_h(t,t_n;\zeta)\tilde{u}_h(\cdot,t_n;\zeta,\mathcal{P}). \end{cases} \tag{2}$$

Piecewise smooth approximating paths $\{W_h\}_{h>0}$ and partitions $\{\mathcal{P}_h\}_{h>0}$ satisfying

$$\lim_{h\to 0^+} \|W_h - W\|_\infty = 0 = \lim_{h\to 0^+} |\mathcal{P}_h| \tag{3}$$

are then chosen in such a way that the function

$$u_h(x,t) := \tilde{u}_h(x,t;W_h,\mathcal{P}_h) \tag{4}$$

is an approximation of the solution of (5) for small $h > 0$.

2.2 The Main Examples

We focus here on finite difference schemes, while noting that the general convergence results apply to other approximations, for example, Trotter-Kato splitting formulas; see also [7].

To simplify the presentation, assume $d = m = 1$, $\|Du_0\|_\infty \leq L$, F and H are both smooth with bounded derivatives, and F depends only on u_{xx}, so that (1) becomes

$$du = F(u_{xx})\,dt + H(u_x)\cdot dW \quad \text{in } \mathbb{R}\times(0,T] \quad \text{and} \quad u(\cdot,0) = u_0 \quad \text{in } \mathbb{R}, \tag{5}$$

or, in the first order case, when $F \equiv 0$,

$$du = H(u_x)\cdot dW \quad \text{in } \mathbb{R}\times(0,T] \quad \text{and} \quad u(\cdot,0) = u_0 \quad \text{in } \mathbb{R}. \tag{6}$$

Below, the various specifications for $\mathcal{P}_h$ and W_h, while technical, are all made in order to ensure that, for some fixed, sufficiently small $\lambda > 0$, the following generalized CFL condition holds:

$$\sup_{h>0}\ \sup_{n=0,1,2,\dots,N-1} \frac{|W_h(t_{n+1}) - W_h(t_n)|}{h} \leq \lambda. \tag{7}$$

The reason for this is discussed further in Sect. 4.

The first scheme is defined, for some $\varepsilon_h > 0$, by

$$\begin{aligned} S_h(t,s;\zeta)u(x) := u(x) &+ H\left(\frac{u(x+h)-u(x-h)}{2h}\right)(\zeta(t)-\zeta(s)) \\ &+ \left[F\left(\frac{u(x+h)+u(x-h)-2u(x)}{h^2}\right)\right. \\ &\left. + \,\varepsilon_h\left(\frac{u(x+h)+u(x-h)-2u(x)}{h^2}\right)\right](t-s). \end{aligned} \tag{8}$$

Theorem 1 *Assume that, in addition to* (3), W_h *and* $\mathcal{P}_h$ *satisfy*

$$|\mathcal{P}_h| < \frac{h^2}{\|F'\|_\infty} \quad \textit{and} \quad \varepsilon_h := h\,\|H'\|_\infty\,\|\dot{W}_h\|_\infty \xrightarrow{h\to 0} 0.$$

Then, as $h \to 0$, *the function* u_h *defined by* (4) *using the scheme operator* (8) *converges locally uniformly to the pathwise viscosity solution* u *of* (5).

For schemes approximating solutions of the pathwise Hamilton-Jacobi equation (6), we are able to obtain explicit error estimates. We focus here on the particular scheme defined, for some $\theta \in (0, 1]$, by

$$\begin{aligned} S_h(t, s; \zeta)u(x) := u(x) + H\left(\frac{u(x+h) - u(x-h)}{2h}\right)(\zeta(t) - \zeta(s)) \\ + \frac{\theta}{2}\left(u(x+h) + u(x-h) - 2u(x)\right). \end{aligned} \tag{9}$$

Assume that $\omega : [0, \infty) \to [0, \infty)$ is the modulus of continuity of the fixed continuous path W on $[0, T]$. For $h > 0$, define ρ_h implicitly by

$$\lambda := \frac{(\rho_h)^{1/2}\omega((\rho_h)^{1/2})}{h} < \frac{\theta}{\|H'\|_\infty}, \tag{10}$$

and let the partition $\mathcal{P}_h$ and path W_h satisfy

$$\begin{cases} \mathcal{P}_h := \{n\rho_h \wedge T\}_{n \in \mathbb{N}_0}, \ M_h := \lfloor (\rho_h)^{-1/2} \rfloor, \\ \text{and, for } k \in \mathbb{N}_0 \text{ and } t \in [kM_h\rho_h, (k+1)M_h\rho_h), \\ W_h(t) := W(kM_h\rho_h) + \left(\frac{W((k+1)M_h\rho_h) - W(kM_h\rho_h)}{M_h\rho_h}\right)(t - kM_h\rho_h). \end{cases} \tag{11}$$

Theorem 2 *There exists $C > 0$ depending only on L such that, if u_h is constructed using (4) and (9) with $\mathcal{P}_h$ and W_h as in (10) and (11), and u is the pathwise viscosity solution of (6), then*

$$\sup_{(x,t) \in \mathbb{R}^d \times [0,T]} |u_h(x, t) - u(x, t)| \le C(1 + T)\omega((\rho_h)^{1/2}).$$

If $W \in C^{0,\alpha}([0, T])$, then the CFL condition (10) becomes $\rho_h = O(h^{2/(1+\alpha)})$, and the rate of convergence in Theorem 2 is $O(h^{\alpha/(1+\alpha)})$.

When W is a Brownian motion, then the CFL condition (10) can be chosen according to the Lévy modulus of continuity:

$$\lambda := \frac{(\rho_h)^{3/4}\,|\log \rho_h|^{1/2}}{h} < \frac{\theta}{\|H'\|_\infty}. \tag{12}$$

The proof of Theorem 2 can then be modified to show that, with probability one, for a deterministic constant $C > 0$ depending only on L and λ,

$$\limsup_{h \to 0} \sup_{(x,t) \in \mathbb{R}^d \times [0,T]} \frac{|u_h(x, t) - u(x, t)|}{h^{1/3}\,|\log h|^{1/3}} \le C(1 + T).$$

The final example converges in distribution. Let λ, ρ_h, W_h, and $\mathcal{P}_h$ be given, for some probability space $(\mathcal{A}, \mathcal{G}, \mathbf{P})$, by

$$\begin{cases} \lambda := \dfrac{(\rho_h)^{3/4}}{h} \leq \dfrac{\theta}{\|H'\|_\infty}, \quad M_h := \lfloor (\rho_h)^{-1/2} \rfloor, \\ \mathcal{P}_h := \{t_n\}_{n=0}^N = \{n\rho_h \wedge T\}_{n\in\mathbb{N}_0}, \\ \{\xi_n\}_{n=1}^\infty : \mathcal{A} \to \{-1, 1\} \text{ are independent and Rademacher}, \\ W_h(0) = 0, \quad \text{and} \\ W_h(t) := W_h(kM_h\rho_h) + \dfrac{\xi_k}{\sqrt{M_h\rho_h}}(t - kM_h\rho_h) \\ \text{for } k \in \mathbb{N}_0,\ t \in [kM_h\rho_h, (k+1)M_h\rho_h). \end{cases} \tag{13}$$

Donsker's invariance principle (see [3]) implies that, as $h \to 0$, W_h converges in distribution to a Brownian motion W in the space $C([0, \infty), \mathbb{R})$.

Theorem 3 *If u_h is constructed using* (4) *and* (9) *with W_h and $\mathcal{P}_h$ as in* (13)*, and u is the solution of* (5)*, then, as $h \to 0$, u_h converges to u in distribution in the topology of local uniform convergence.*

3 The Convergence Proof: Monotonicity and Consistency

We next outline the proof of the general convergence result, which is based on a generalization of the method of half-relaxed limits from the theory of viscosity solutions, used by Barles and Souganidis [2] to prove the convergence of finite difference approximations of second order equations.

We always impose the following monotonicity condition on the scheme:

$$\begin{cases} \text{if } t_n \leq t \leq t_{n+1},\ t_n, t_{n+1} \in \mathcal{P}_h, \text{ and } u, v \in BUC(\mathbb{R}^d), \text{ then} \\ u \leq v \quad \Rightarrow \quad S_h(t, t_n; W_h)u \leq S_h(t, t_n; W_h)v, \end{cases} \tag{14}$$

and also that the scheme operator commutes with constants, that is, for all $u \in BUC(\mathbb{R}^d)$, $h > 0$, $0 \leq s \leq t < \infty$, $\zeta \in C([0, T], \mathbb{R}^m)$, and $k \in \mathbb{R}$,

$$S_h(t, s; \zeta)\,(u + k) = S_h(t, s; \zeta)u + k. \tag{15}$$

In addition, the scheme operator must satisfy a consistency requirement (see (19) below). We motivate such a condition by outlining the proof next, keeping the details light.

To that end, assume that, for some $u \in BUC(\mathbb{R}^d \times [0, T])$, $\lim_{h\to 0} u_h = u$ locally uniformly,[3] and we attempt to show that u is a pathwise viscosity sub- and super-solution of (1). Recall (see [10]) that u is said to be a sub- (super)- solution of (1) if, whenever $I \subset [0, T]$ is an open interval; $\Phi \in C(I; C^2(\mathbb{R}^d))$ is a local-in-time, smooth-in-space solution of

$$d\Phi = \sum_{i=1}^{m} H^i(D\Phi) \cdot dW^i \quad \text{in } \mathbb{R}^d \times I; \tag{16}$$

$\psi \in C^1(I)$; and $u(x,t) - \Phi(x,t) - \psi(t)$ attains a strict maximum (minimum) at $(y, s) \in \mathbb{R}^d \times I$, then

$$\psi'(s) \le F(D^2\Phi(y,s), D\Phi(y,s)).$$

We show that u is a sub-solution, and the argument for super-solutions is similar.

Let I, Φ, ψ, and (y, s) be as above, and, for $h > 0$, let Φ_h be the local-in-time, smooth-in-space solution, constructed with the method of characteristics, of

$$\Phi_{h,t} = \sum_{i=1}^{m} H^i(D\Phi_h)\dot{W}_h^i \quad \text{in } \mathbb{R}^d \times I, \quad \Phi_h(\cdot, s) = \Phi(\cdot, t_0). \tag{17}$$

Since W_h converges to W uniformly as $h \to 0$, it follows that Φ_h converges in $C(I, C^2(\mathbb{R}^d))$ to Φ as $h \to 0$, with I made smaller if necessary, independently of h.

As a result, there exists $\{(y_h, s_h)\}_{h>0} \subset \mathbb{R}^d \times I$ such that $\lim_{h\to 0}(y_h, s_h) = (y, s)$ and

$$u_h(x,t) - \Phi_h(x,t) - \psi(t)$$

attains a local maximum at (y_h, s_h).

The mesh-size $|\mathcal{P}_h|$ converges to 0 as $h \to 0$, so, for sufficiently small h, there exists $n \in \mathbb{N}$ depending on h such that

$$t_n < s_h \le t_{n+1} \quad \text{and} \quad t_n, t_{n+1} \in I.$$

Then

$$u_h(\cdot, t_n) - \Phi_h(\cdot, t_n) - \psi(t_n) \le u_h(y_h, s_h) - \Phi_h(y_h, s_h) - \psi(s_h),$$

[3] In general, the existence of such a limit is not guaranteed a priori, and one must work with so-called "half-relaxed" limits. To simplify the presentation, we avoid such details here.

which leads to

$$u_h(\cdot,t_n) \le u_h(y_h,s_h) + \Phi_h(\cdot,t_n) - \Phi_h(y_h,s_h) + \psi(t_n) - \psi(s_h). \tag{18}$$

It is here that the monotonicity (14) of the scheme is used. Applying $S_h(s_h,t_n;W_h)$ to both sides of (18), using the fact that the scheme commutes with constants, and plugging in $x = y_h$, we arrive at

$$u_h(y_h,s_h) \le u_h(y_h,s_h) + S_h(s_h,t_n;W_h)\Phi_h(\cdot,t_n)(y_h) - \Phi_h(y_h,s_h) + \psi(t_n) - \psi(s_h),$$

whence

$$\frac{\psi(s_h) - \psi(t_n)}{s_h - t_n} \le \frac{S_h(s_h,t_n;W_h)\Phi_h(\cdot,t_n)(y_h) - \Phi_h(y_h,s_h)}{s_h - t_n}.$$

As $h \to 0$, the left-hand side converges to $\psi'(s)$. The right-hand side converges to $F(D^2\Phi(y,s), D\Phi(y,s))$ if we make the following consistency requirement: whenever Φ and Φ_h are as in respectively (16) and (17), we have

$$\lim_{s,t\in I,\, t-s\to 0} \frac{S_h(t,s;W_h)\Phi_h(\cdot,s) - \Phi_h(\cdot,s)}{t-s} = F(D^2\Phi, D\Phi). \tag{19}$$

4 On the Construction of the Scheme Operator

We discuss next the strategy for constructing scheme operators that satisfy the assumptions of the previous section, and, in particular, the need for regularizing the path W in general. We focus on Eq. (6) and consider the scheme operator given by

$$\begin{aligned} S_h(t,s)u(x) := u(x) + H\left(\frac{u(x+h)-u(x-h)}{2h}\right)(W(t)-W(s)) \\ + \varepsilon_h \left(\frac{u(x+h)+u(x-h)-2u(x)}{h^2}\right)(t-s), \end{aligned} \tag{20}$$

which can be seen to be monotone for $0 \le t-s \le \rho_h$ as long as, for some $\theta \in (0,1]$,

$$\varepsilon_h := \frac{\theta h^2}{2(t-s)} \quad \text{and} \quad \lambda := \max_{|t-s|\le\rho_h} \frac{|W(t)-W(s)|}{h} \le \lambda_0 := \frac{\theta}{\|H'\|_\infty}. \tag{21}$$

For any $s, t \in [0, T]$ with $|s - t|$ sufficiently small, spatially smooth solutions Φ of (6) have the expansion

$$\begin{aligned}\Phi(x,t) = \Phi(x,s) &+ H(\Phi_x(x,s))(W(t) - W(s)) \\ &+ H'(\Phi_x(x,s))^2 \Phi_{xx}(x,s)(W(t) - W(s))^2 + O(|W(t) - W(s)|^3),\end{aligned} \tag{22}$$

so that, if $0 \le t - s \le \rho_h$, then, for some $C > 0$ depending only on H,

$$\begin{aligned}\sup_{\mathbb{R}} |S_h(t,s)\Phi(\cdot,s) - \Phi(\cdot,t)| &\le C \sup_{r\in[s,t]} \left\| D^2\Phi(\cdot,r) \right\|_\infty \left(|W(t) - W(s)|^2 + h^2 \right) \\ &\le C \sup_{r\in[s,t]} \left\| D^2\Phi(\cdot,r) \right\|_\infty (1 + \lambda_0^2) h^2.\end{aligned}$$

Then (19) is satisfied if

$$\lim_{h\to 0} \frac{h^2}{\rho_h} = 0. \tag{23}$$

Both (21) and (23) can be achieved when W is continuously differentiable, or, more generally, if $W \in C^{0,\alpha}([0,T])$ with $\alpha > \frac{1}{2}$, by setting

$$\rho_h := \left(\frac{\lambda h}{[W]_{\alpha,T}} \right)^{1/\alpha}. \tag{24}$$

However, this approach fails as soon as the quadratic variation path

$$\langle W \rangle_T := \lim_{|\mathcal{P}|\to 0} \sum_{n=0}^{N-1} |W(t_{n+1}) - W(t_n)|^2$$

is non-zero, as (21) and (23) together imply that $\langle W \rangle_T = 0$. This rules out, for instance, the case where W is the sample path of a Brownian motion, or, more generally, any nontrivial semimartingale.

On the other hand, if $\{W_h\}_{h>0}$ is a family of piecewise smooth paths converging uniformly, as $h \to 0$, to W, then $\langle W_h \rangle_T = 0$ for each fixed $h > 0$, and therefore, W_h and ρ_h can be chosen so that (21) and (23) hold for W_h rather than W. As described in Sect. 2, such choices are related to the general CFL condition (7).

References

1. Barles, Guy; Jakobsen, Espen R.: Error bounds for monotone approximation schemes for Hamilton-Jacobi-Bellman equations. SIAM J. Numer. Anal. **43** (2005), no. 2, 540–558
2. Barles, G.; Souganidis, P. E.: Convergence of approximation schemes for fully nonlinear second order equations. Asymptotic Anal. **4** (1991), no. 3, 271–283
3. Billingsley, Patrick: Convergence of probability measures. John Wiley & Sons, Inc., New York-London-Sydney 1968
4. Caffarelli, Luis A.; Souganidis, Panagiotis E.: A rate of convergence for monotone finite difference approximations to fully nonlinear, uniformly elliptic PDEs. Comm. Pure Appl. Math. **61** (2008), no. 1, 1–17
5. Crandall, Michael G.; Ishii, Hitoshi; Lions, Pierre-Louis: User's guide to viscosity solutions of second order partial differential equations. Bull. Amer. Math. Soc. (N.S.) **27** (1992), no. 1, 1–67
6. Crandall, M. G.; Lions, P.-L.: Two approximations of solutions of Hamilton-Jacobi equations. Math. Comp. **43** (1984), no. 167, 1–19
7. Gassiat, Paul; Gess, Benjamin: Regularization by noise for stochastic Hamilton-Jacobi equations. Probab. Theory Related Fields **173** (2019), no. 3-4, 1063–1098
8. Jakobsen, Espen R.: On error bounds for monotone approximation schemes for multi-dimensional Isaacs equations. Asymptot. Anal. **49** (2006), no. 3-4, 249–273
9. Krylov, N. V. On the rate of convergence of finite-difference approximations for elliptic Isaacs equations in smooth domains. Comm. Partial Differential Equations **40** (2015), no. 8, 1393–1407
10. Lions, Pierre-Louis; Souganidis, Panagiotis E.: Fully nonlinear stochastic partial differential equations. C. R. Acad. Sci. Paris Sér. I Math. **326** (1998), no. 9, 1085–1092
11. Lions, Pierre-Louis; Souganidis, Panagiotis E.: Fully nonlinear stochastic partial differential equations: non-smooth equations and applications. C. R. Acad. Sci. Paris Sér. I Math. **327** (1998), no. 8, 735–741
12. Lions, Pierre-Louis; Souganidis, Panagiotis E.: Fully nonlinear stochastic pde with semilinear stochastic dependence. C. R. Acad. Sci. Paris Sér. I Math. **331** (2000), no. 8, 617–624
13. Lions, Pierre-Louis; Souganidis, Panagiotis E.: Uniqueness of weak solutions of fully nonlinear stochastic partial differential equations. C. R. Acad. Sci. Paris Sér. I Math. 331 (2000), no. 10, 783–790
14. Seeger, Benjamin: Approximation schemes for viscosity solutions of fully nonlinear stochastic partial differential equations. Ann. Appl. Probab. **30** (2020), no. 4, 1784–1823
15. Souganidis, Panagiotis E.: Approximation schemes for viscosity solutions of Hamilton-Jacobi equations. J. Differential Equations **59** (1985), no. 1, 1–43
16. Souganidis, Panagiotis E.: Max-min representations and product formulas for the viscosity solutions of Hamilton-Jacobi equations with applications to differential games. Nonlinear Anal. **9** (1985), no. 3, 217–257
17. Turanova, Olga: Error estimates for approximations of nonhomogeneous nonlinear uniformly elliptic equations. Calc. Var. Partial Differential Equations **54** (2015), no. 3, 2939–2983

Approximation Method with Stochastic Local Iterated Function Systems

Anna Soós and Ildikó Somogyi

Abstract The methods of real data interpolation can be generalized with fractal interpolation. These fractal interpolation functions can be constructed with the so-called iterated function systems. Local iterated function systems are an important generalization of the classical iterated function systems. In order to obtain new approximation methods this methods can be combined with classical interpolation methods. In this paper we focus on the study of the stochastic local fractal interpolation function in the case of a random data set.

1 Introduction

Barnsley named a function $f : I \rightarrow \mathbb{R}$ defined on the real closed interval I, a *fractal function*, if the Hausdorff dimension of the graph is noninteger. Also in [2] he introduced the notion of a fractal interpolation function (FIF). This is a fractal function which is constrained to go through on a finite number of prescribed points, so the (FIF) possess some interpolation properties. The methods of fractal interpolation methods was applied successfully in signal processing, structural mechanics, computer geometry and other fields of applied sciences. The advantage of these methods is that they can be combined with the classical methods or real data interpolation. Hutchinson and Rüschendorf [3] gave the stochastic version of the fractal interpolation function. In the construction of the iterated function systems, Wang and Yu [4] used a variable vertical scaling factor instead of a constant scaling parameter, in this way they obtained fractal functions with more flexibility. Barnsley in [1] introduced the notion of local iterated function systems which are an important generalization of the global iterated function systems. In [5], the local Hermite type fractal function was introduced. In this paper we focus on the study of local

A. Soós · I. Somogyi (✉)
Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Cluj-Napoca, Romania
e-mail: asoos@math.ubbcluj.ro; ilkovacs@math.ubbcluj.ro

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_87

type Hermite fractal function corresponding to a set of data, where these data are supposed to be random.

2 Local Iterated Function Systems and Local Fractal Functions

Let (X, d_X) be a complete metric spaces with metric d_X and $N = \{1, 2, 3, \ldots\}$ the set of positive integers.

Let $n \in N$ and $N_n = \{1, 2, \ldots, n\}$, and consider a family of nonempty subsets of X, $\{X_i | i \in N_n\}$. Assume that there exists a continuous mapping $f_i : X_i \to X, i \in N_n$, for each X_i. Then $\mathbb{F}_{loc} = \{X, (X_i, f_i)| i \in N_n\}$ is called a *local iterated function system.*
If each $X_i = X$ then this definition gives us the usual definition of a global iterated function system on a complete metric space.

A local IFS $\mathbb{F}_{loc}$ is called **contractive** if there exists a metric d' equivalent to d_X with respect to which all functions $f \in \mathbb{F}_{loc}$ are contractive, on their respective domains.

Let the power set of X be $2^X = \{S | S \subset X\}$. On this set we consider a set-valued operator using a local IFS:

$$\mathbb{F}_{loc}(S) = \cup_{i=1}^{n} f_i(S \cap X_i), \tag{1}$$

where $f_i(S \cap X_i) = \{f_i(x) | x \in S \cap X_i\}$. A subset $G \in 2^X$ is called a **local attractor** for the local IFS $\{X, (X_i, f_i)| i \in N_n\}$ if

$$G = \mathbb{F}_{loc}(G) = \cup_{i=1}^{n} f_i(G \cap X_i).$$

For example the empty set is a local attractor of the local IFS, and if G_1 and G_2 are distinct local attractors than $G_1 \cap G_2$ is also a local attractor. Hence, there exists a largest local attractor for the IFS, and this will be the so-called local attractor of the local IFS. In the case when X is compact and $X_i, i \in N_n$ are also compact in X, and the local IFS $\{X, (X_i, f_i)| i \in N_n\}$ is contractive, the local attractor may be computed in the following way. Let $L_0 = X$ and

$$L_n = \mathbb{F}_{loc}(L_{n-1}) = \cup_{i \in N_n} f_i(L_{n-1} \cap X_i), \quad n \in \mathbb{N}.$$

Then $\{L_n | n \in N_0\}$ is a decreasing nested sequence of compact sets. If each L_n is nonempty, then by the Cantor intersection theorem,

$$L = \cap_{n \in N_0} L_n \neq \emptyset,$$

we have that

$$L = \lim_{n \to \infty} L_n,$$

where the limit is taken with respect to the Hausdorff metric. This implies that

$$L = \lim_{n \to \infty} L_n = \lim_{n \to \infty} \cup_{i \in N_n} f_i(L_{n-1} \cap X_i) = \cup_{i \in N_n} f_i(L \cap X_i) = \mathbb{F}_{loc}(L).$$

It follows that, $L = G_{loc}$. Further, Barnsley introduces the local fractal functions as the local attractors which are the graphs of bounded functions.

Let X be a nonempty connected set and $\{X_i | i \in N_n\}$ are subsets of X which are nonempty and connected. We will consider a family of bijective mappings, $u_i : X_i \to X, i \in N_n$ such that $\{u_i(X_i), i \in N_n\}$ is a kind of partition of X, $X = \cup_{i=1}^{n} u_i(X_i)$ and $u_i(X_i) \cap u_j(X_j) = \emptyset, \ \ \forall i \neq j \in N_n$.

Let (Y, d_y) be also a complete metric space with the metric d_y, then a function $f : X \to Y$ is called bounded with respect to the metric d_y, if there exists $M > 0$ such that $\forall x_1, x_2 \in X, d_Y(f(x_1), f(x_2)) < M$.

Then the space $D(X, Y) = \{f : X \to Y | f \ is \ bounded\}$, with the metric $d(f, g) = \sup_{x \in X} d_Y(f(x), g(x))$ is a complete metric space, $(D(X, Y), d)$. In a similar way we can define $D(X_i, Y)$, for all $i \in N_n$ and let be $f_i = f | X_i$. We will consider now a set of functions which are uniformly contractive in the second variable $v_i : X_i \to X, \ \ i \in N_n$, and the Read-Bajactarević operator $B : D(X, Y) \to Y^X$ defined by

$$Bf(x) = \sum_{i=1}^{N} v_i(u_i^{-1}(x), f_i \circ u_i^{-1}(x)) \chi_{u_i(X_i)}(x),$$

where

$$\chi_S(x) = \begin{cases} 1, \ x \in S \\ 0, \ x \notin S. \end{cases}$$

Using the contraction properties on the second variable of the applications w_i, it follows that the operator B is also a contraction on the complete metric space $D(X, Y)$ and therefore it has a unique fixed point f^* in $D(X, Y)$. This unique fixed point will be called a **local fractal function**, generated by B.

2.1 Hermite Type Local Fractal Functions

Let be given the distinct real numbers $x_0 < x_1 < .. < x_N$ and the values $y_i^{(0)}, y_i^{(1)}, \ldots, y_i^{(r_i)}, i = 0, 1, \ldots, N$. Then there exists a unique polynomial $H(x)$, the Hermite polynomial, which satisfies the interpolation conditions

$$H^{(j)}(x_i) = y_i^{(j)}, j = 0, 1, \ldots, r_i, i = 0, 1, \ldots, N$$

and the degree of this polynomial does not exceed $n = \sum_{i=0}^{N}(r_i + 1)$.

This classical generalized Hermite interpolation polynomial is given by

$$H(x) = \sum_{i=0}^{N} \sum_{j=0}^{r_i} h_{i,j}(x) y_i^{(j)},$$

where the polynomials $h_{i,j} \in \mathbb{P}_n, n = \sum_{i=1}^{N}(r_i + 1)$ are the fundamental Hermite polynomials. These fundamental Hermite polynomials can be given in the following way too:

$$h_{ij}(x) = l_{ij}(x) - \sum_{\nu=j+1}^{r_i} l_{ij}^{\nu}(x_i) h_{i\nu}(x),$$

where

$$l_{ij}(x) = \frac{(x - x_i)^j}{j!} \prod_{s=0, s\neq i} \left(\frac{x - x_s}{x_i - x_s} \right)^{r_s}$$

are generalized Lagrange polynomials.

Let us consider a given set of interpolation data $\{(x_n, y_n) \in [a, b] \times \mathbb{R}, n = 0, 1, \ldots, N\}$, where $a < x_0 < x_1 < \ldots < x_N < b$.

Further, let Y_n be given by $Y_n = y_n + \epsilon_n,\ n = 0, 1, \ldots, N$, where ϵ_n is a stochastic perturbation term with zero expectation, $\mathbb{E}(\epsilon_n) = 0$, and finite variance, $Var(\epsilon_n) <$ inf. Each Y_n is a random variable and $\mathbb{E}(Y_n) = y_n$ Suppose that z_n is an observed value of Y_n.

Let $X = [a, b]$ and $\{X_i | i \in N_k\}, N_k = \{1, \ldots, k\}, k \in \mathbb{N}$ be a family of nonempty subsets of X such that $X_i = [x_i, x_{i+1}]$. On each subinterval X_i we consider a set of data $x_i = x_{i0} < x_{i1} < \ldots < x_{ik} = x_{i+1}$.
For each X_i there exists a contractive homeomorphism $u_{in} : X_i \to X_{in}$, where $X_{in} = [x_{in-1}, x_{in}], n = 1, 2, \ldots, k$, and $u_{in}(x_{i0}) = x_{in-1}$, $u_{in}(x_k) = x_{in}$. Let $v_{in} : X_i \times \mathbb{R} \to \mathbb{R}$ be a mapping that is continuous and contractive in the second variable.

Define $\omega_i(x, y) = (u_{in}(x), v_{in}(x, y))$ for all $n = 1, 2, \ldots, k$, than we can give the following local IFS $\{X; (X_i, \omega_i), i \in N_k\}$.

We will consider the affine IFS given by the following functions

$$u_{in}(x) = a_{in}x + b_{in}$$

$$v_{in}(x, y) = \alpha_{in} y + g_{in}(x),$$

$n = 1, 2, \ldots, k$, where α_{in} is the vertical scaling factor of the transformation ω_{in} and $g_{in}(x) = l \circ u_{in}(x) - \alpha_{in} q(x)$, where $q(x)$ is a real continuous function such that $q(x_{i0}) = z_i, q(x_{ik}) = z_{i+1}$, for l we have the condition $l(x_{in}) = z_{in}, n = 0, 1, \ldots, k$.

In order to have a differentiable fractal interpolation function we will use the proposition given by Barnsley in [2].

If we have a set of interpolation data $\{(x_{in}, z_{in}), n = 0, 1, \ldots, k\}$, with $x_{i0} < x_{i1} < \ldots x_{ik}$, the functions u_{in}, v_{in} defined before, for some integer $p > 0$, $|\alpha_{in}| < a_{in}^p$, $g_{in} \in C^p[x_i, x_{i+1}]$, and the functions

$$v_{in,t}(x, y) = \frac{\alpha_{in} y + g_{in}^{(t)}(x)}{a_{in}^t},\ z_{i0,t} = \frac{g_{i1}^{(t)}(x_{i0})}{a_{i1}^t - \alpha_{i1}},\ z_{ik,t} = \frac{g_{ik}^{(t)}(x_{ik})}{a_{ik}^t - \alpha_{ik}}, t = 1, 2, \ldots, p \tag{2}$$

with the conditions $v_{in-1,t}(x_{ik}, z_{ik,t}) = v_{in,t}(x_{i0}, z_{i0,t}), n = 2, 3, \ldots, k$ and $t = 1, 2, \ldots, p$ then $\{(u_{in}(x), v_{in}(x, y)), n = 1, 2, \ldots, k\}$ determines a fractal interpolation function $f \in C^p[x_i, x_{i+1}]$ and $f^t, t = 1, 2, \ldots, p$ is the fractal interpolation function determined by $\{(u_{in}(x), v_{in,k}(x, y)), n = 1, 2, \ldots, k\}$.

To construct the Hermite type local fractal interpolation function we will use the method introduced in [1]. We will consider a given equidistant set of data on each intervals $[x_i, x_{i+1}], i = 0, 1, \ldots, N - 1$, $(x_{ij}, z_{ij}^{(\mu)}), j = 0, 1, \ldots, k, \mu = 0, 1, \ldots, r_{ij}$ and a fixed vertical scaling parameter α_i, such that $|\alpha_i| < \frac{1}{c_i}, c_i = \max\{r_{ij}, j = 0, 1, .., k\}$.

For the given set of equidistant data we will consider the following coefficients in the functions $u_{in}^{j\mu}$

$$a_{in} = \frac{x_{in} - x_{in-1}}{x_{i0} - x_{ik}} = \frac{1}{k}, n = 1, 2 \ldots, k, i = 0, 1, \ldots, N.$$

For a fixed j and any $\mu = \{0, 1, \ldots, r_{ij}\}$ we define the functions $u_{in}, v_{in}^{j\mu}$ given by $u_{in}(x) = \frac{x}{k} + b_{in}$ and $v_{in}^{j\mu}(x, y) = \alpha_i y + g_{in}^{j\mu}(x)$, where $g_{in}^{j\mu}(x) = h_{j\mu} \circ u_{in}(x) - \alpha_i s_{j\mu}(x)$, where the functions $s_{j\mu}$ are given in such a way that the fractal functions satisfy the Hermite type interpolation conditions. Also in order to have a differentiable fractal function, in the end-point of the intervals we have the following conditions:

$$v_{in-1,t}^{j\mu}(x_{ik}, z_{ik,t}^{j\mu}) = v_{in,t}^{j\mu}(x_{i0}, z_{i0,t}^{j\mu})$$

and using relation (2), we have

$$\frac{\alpha_i z_{ik,t}^{j\mu} + g_{in-1}^{j\mu}(x_{ik})}{a_{in-1}^t} = \frac{\alpha_i z_{ik,t}^{j\mu} + g_{in}^{j\mu}(x_{ik})}{a_{in}^t}$$

also from formula (2) we obtain

$$z_{ik,t}^{j\mu}=\frac{h_{j\mu}^{(t)}(x_{ik})-k^t\alpha_i s_{j\mu}^{(t)}(x_{ik})}{1-k^t\alpha_i},\; y_{i0,t}^{j\mu}=\frac{h_{j\mu}^{(t)}(x_{i0})-k^t\alpha_i s_{j\mu}^{(t)}(x_{i0})}{1-k^t\alpha_i}$$

and it follows that

$$\frac{h_{j\mu}^{(t)}(x_{ik})-k^t\alpha_i s_{j\mu}^{(t)}(x_{ik})}{1-k^t\alpha_i}-s_{j\mu}^{(t)}(x_{ik})=\frac{h_{j\mu}^{(t)}(x_{i0})-k^t\alpha_i s_{j\mu}^{(t)}(x_{i0})}{1-k^t\alpha_i}-s_{j\mu}^{(t)}(x_{i0}),$$

this gives the following conditions for the functions $s_{j\mu}$

$$h_{j\mu}^{(t)}(x_{ik})=s_{j\mu}^{(t)}(x_{ik}),\, h_{j\mu}^{(t)}(x_{i0})=s_{j\mu}^{(t)}(x_{i0}),\, t=0,1,\ldots,p.$$

This means that we can choose this functions to be Hermite type polynomials with the nodes x_{ik} and x_{i0}, and multiplicity order p

$$s_{j\mu}(t)=\sum_{t=0}^{p}h_{j\mu}^{(t)}(x_{i0})h_{0,t}(x)+\sum_{t=0}^{p}h_{j\mu}^{(t)}(x_{ik})h_{k,t}(x),$$

where $h_{0,t}$ and $h_{k,t}$ are fundamental Hermite polynomials

$$h_{0,t}(x)=u_0(x)\frac{(x-x_{i0})^t}{t!}\sum_{l=0}^{p-t}\frac{(x-x_{i0})^l}{l!}\left(\frac{1}{u_0(x)}\right)_{x=x_{i0}}^{(l)}$$

$$h_{k,t}(x)=u_k(x)\frac{(x-x_{ik})^t}{t!}\sum_{l=0}^{p-t}\frac{(x-x_{ik})^l}{l!}\left(\frac{1}{u_k(x)}\right)_{x=x_{ik}}^{(l)}$$

with $u_0(x)=(x-x_{ik})^{p+1}$ and $u_k(x)=(x-x_{i0})^{p+1}$.

Then the iterated function system associated to $h_{j\mu,\alpha_i}^{(t)}$ is given with the following functions

$$u_{in}(x)=\frac{1}{k}x+b_{in} \tag{3}$$

$$v_{in,t}^{j\mu}(x,y)=c_i^t\alpha_i y+h_{j\mu}^{(t)}(u_{in}(x))-c_i^t\alpha_i s_{j\mu}^{(t)}(x),$$

$h_{j\mu,\alpha_i}$ is the fractal function given with the functions $s_{j\mu}$.

Theorem 1 *If we have a set of interpolation data* $\{(x_i,z_i),n=1,\ldots,N\}$, *and* $x_{i0}<x_{i1}<\ldots x_{ik}$, *an equidistant set of data on each intervals* $[x_i,x_{i+1}],i=0,1,\ldots,N$, *with a vertical scaling parameter* α_i, *then for a fixed* j *and* $\mu=1,2,\ldots,p$ *the fractal function* $h_{j\mu,\alpha_i}$ *satisfies the following conditions:*

$$h_{j\mu,\alpha_i}^{(t)}(x_i)=h_{j\mu,\alpha_i}(x_i),\quad i=1,2,\ldots,N.$$

Proof Using relation (2), we have

$$\begin{aligned}h^{(t)}_{j\mu,\alpha_i}(x_i) = h^{(t)}_{j\mu,\alpha_i}(x_{i0}) = z^{0t}_{j\mu} = \frac{(g^{j\mu}_{i1})^{(t)}(x_0)}{a^t_{i1}-\alpha_i} = \frac{1}{a^t_{i1}-\alpha_i}(h^{(t)}_{j\mu}(u_{in}(x_{i0})) - \alpha_i s^{(t)}_{j\mu}(x_{i0}))\\ = \frac{1}{1-k^t\alpha_i}(h^{(t)}_{j\mu}(x_{i0}) - \alpha_i c^t_i s^{(t)}_{j\mu}(x_{i0})) =\\ = \frac{1}{1-k^t\alpha_i}(h^{(t)}_{j\mu}(x_{i0}) - \alpha_i k^t h^{(t)}_{j\mu}(x_{i0})) =\\ = h^{(t)}_{j\mu}(x_{i0}) = h^{(t)}_{j\mu}(x_i), \quad i = 1, 2, \ldots, N.\end{aligned}$$

□

Let $D = \{f : X \to \mathbb{R} | f \ \texttt{continuous}, \ f(x_0) = z_0, f(x_N) = z_N\}$. We will consider a metric on D

$$d(f, g) = \|f - g\|_\infty = \max\{|f(x) - g(x)|, x \in X\}, \forall f, g \in D.$$

(D, d) is a complete metrixc space, than for a fixed j and any $\mu = 1, 2, \ldots, p$ we will define the Read-Bajraktarević operator on (D, d)

$$(Bf)(x) = \sum_{i=1}^{N}\sum_{n=1}^{k} v^{j\mu}_{in}(u^{-1}_{in}(x), f_i(u^{-1}_{in}(x)))\chi_{u_{in}(X_i)}(x),$$

where $f_i = f|_{X_i}$, $u_{in}(x) = \frac{x}{c_i} + b_{in}$, $v^{j\mu}_{in}(x, y) = \alpha_i y + g^{j\mu}_{in}(x)$, with $g^{j\mu}_{in}(x) = h_{j\mu} \circ u_{in}(x) - \alpha_i s_{j\mu}(x)$.

Theorem 2 *Let X be a nonempty connected set and $\{X_i, i \in N\}$ nonempty connected subsets of X and $u_{in} : X_i \longrightarrow X_{in}, i = 0, 1, \ldots, N, n = 0, 1, \ldots, k$ a family of contractive homeomorphisms, and α_i the vertical scaling parameters such that $|\alpha_i| < \frac{1}{c_i}$ and $k\sum_{i=1}^{N}\frac{1}{c_i} < 1$, then the RB operator is contractive on the complete metric space (D, d) and the unique fixed point f^* is called a local Hermite type fractal function.*

Proof

$$\begin{aligned}\|Bf - Bg\|_\infty = \max_{x\in X}|(Bf)(x) - (Bg)(x)| =\\ = \max_{x\in X}|\sum_{i=1}^{N}\sum_{n=1}^{k}(v^{j\mu}_{in}(u^{-1}_{in}(x), f_i \circ u^{-1}_{in}(x)) -\\ - v^{j\mu}_{in}(u^{-1}_{in}(x), g_i \circ u^{-1}_{in}(x)))\chi_{u_{in}(X_i)}(x)| =\end{aligned}$$

$$= \max_{x\in X} |\sum_{i=1}^{N}\sum_{n=1}^{k} \alpha_i (f_i(u_{in}^{-1}(x)) - g_i(u_{in}^{-1}(x))) \leq$$

$$\leq \sum_{i=1}^{N} |\alpha_i| \sum_{n=1}^{k} \max_{x\in X} |f_i(u_{in}^{-1}(x)) - g_i(u_{in}^{-1}(x)) \leq$$

$$\leq k \sum_{i=1}^{N} |\alpha_i| \|f - g\|_\infty \leq k \sum_{i=1}^{N} \frac{1}{c_i} \|f - g\|_\infty$$

Because $k \sum_{i=1}^{N} \frac{1}{c_i} < 1$, implies that B is a contraction on (D, d). □

Now we replace y_k by Y_k, let

$$\Delta_Y = \{(x_k, Y_k), k := 0, 1, \ldots, N\}$$

Denote f_Y^* the fractal interpolation function for Δ_Y from theorem 2.

Y_k is a random variable, so $f_Y^*(x)$ is also a random variable for all $x \in X$. In the following theorem we will give an approximation of f^* by $f_Y^*(x)$.

Theorem 3 *If*

$$\|\mathbb{E}(f_Y^*) - f^*\|_\infty < \infty$$

then

$$\|\mathbb{E}(f_Y^*) - f^*\|_\infty < \frac{k}{1-s} \|\mathbb{E}(h_{j\mu})_Y - h_{j,\mu}\|_\infty + \frac{ks}{1-s} \|\mathbb{E}(s_{j\mu})_Y - s_{j,\mu}\|_\infty \quad (4)$$

for $s = max\{|\alpha_i|, i = 1, \ldots, N\}$.

Proof By construction for $x \in X_i$ we have

$$\mathbb{E}(f_Y^*(x)) = \alpha_i \sum_{n=1}^{k} \mathbb{E}(f_Y^*(u_{in}^{-1}(x))) - \sum_{n=1}^{k} \mathbb{E}(h_{j\mu}(x)) - \alpha_i \sum_{n=1}^{k} \mathbb{E}(s_{j\mu} u_{in}^{-1}(x)).$$

Hence

$$|\mathbb{E}(f_Y^*(x)) - f^*(x)| \leq |\alpha_i||\mathbb{E}(f_Y^*(x)) - f^*(x)| + k|\mathbb{E}(h_{j\mu}(x)_Y) - h_{j\mu}(x)| +$$

$$+ k|\alpha_i||\mathbb{E}(s_{j\mu}(u_{in}^{-1}(x)))_Y - s_{j\mu}(u_{in}^{-1}(x))|.$$

Then

$$(1-s)|\mathbb{E}(f_Y^*(x)) - f^*(x)| \leq k|\mathbb{E}(h_{j\mu}(x)_Y) - h_{j\mu}(x)|-$$

$$+ks|\mathbb{E}(s_{j\mu}(u_{in}^{-1}(x)))_Y - s_{j\mu}(u_{in}^{-1}(x))|.$$

Taking the $||.||_\infty$ norm we have relation (4). □

In signal analysis, statistical models are established by observed data, in this case these data are supposed to be random. Further interpolation methods have been used in the reconstruction of random signals from samples. In the case when we have to model discrete sequences we can use iterated function systems and fractal interpolation functions. This is the reason why in this paper we investigate some statistical properties of local type fractal functions corresponding to a set of random type data. An upper bound of the error between the original function and the Hermite type local fractal function in the case of random variable data is deduced.

References

1. Barnsley, M. F., Hegeland, M., Massopust, P.: Numerics and Fractals. https://arxiv.org/abs/1309.0972,2014
2. Barnsley, M.F.: Fractals Everywhere. Academic Press (1993)
3. Hutchinson, J.E.: Fractals and Self Similarity. Indiana University Mathematics Journal, **30**, no.5, 713–747 (1981)
4. Wang, H.Y., Yu, J.S.: Fractal interpolation functions with variable parameters and their analytical properties. J. Approx. Theory **175**, 1–18 (2013)
5. Somogyi, I., Soós, A.: Interpolation using Local Iterated Function Systems, International Conference of Numerical Analysis and Approximation Methods, ICNAAM2017, 25-30 Sept. Thessaloniki, Greece (2017)

Optimal Control on a Model for Cervical Cancer

Tri Sri Noor Asih, Widodo, and Dwi Rizkiana Dewi

Abstract Cervical cancer is caused by the human Papillomavirus (HPV) that attacks the cervix. Cervical cancer globally ranks third as the most frequent cancer among women. In this research, a model of HPV infection in cervical cancer consists of five sub categories of cells, namely susceptible cells, infected cells, pre-cancer cells, cancer cells, and viruses. The study was conducted by forming a model of HPV infection with the addition of treatment controls on pre-cancerous cells. The aim is to minimize the number of pre-cancerous cells while minimizing cost. The HPV infection model with control was solved using Pontryagin's maximum principle in order to obtain optimal control. Numerical simulations are performed on the differential equations for the cell densities using the fourth order Runge-Kutta method. The simulation results indicate that a smart administration of treatment can be tailored such that the number of pre-cancer cells is minimized at minimal cost. This configuration with a minimal number of pre-cancer cells is favourable since it inhibits the development of cancer cells.

1 Introduction

Cervical cancer is abnormal cell growth that occurs in the cervix. Cervical cancer globally ranks third as the most frequent cancer among women, with estimated 569,847 new cases and 311,365 deaths in 2018 [3]. The human Papillomavirus (HPV) plays a pivotal role as a cause of cervical cancer [9]. This virus can be transmitted through sexual relations. Several types are called high risk HPV such as HPV types 16, 18, 45, and 56. The persistence of high risk HPV can cause cancer of the cervix, vagina and anus. Changes from healthy cells into cancer cells take

T. S. N. Asih (✉) · D. R. Dewi
Universitas Negeri Semarang, Semarang, Indonesia
e-mail: inung.mat@mail.unnes.ac.id

Widodo
Universitas Gadjah Mada, Daerah Istimewa Yogyakarta, Indonesia

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_88

a long time so the rate of this change can be controlled to reduce the mortality in cervical cancer cases. Cervical cancer can be controlled by applying medical treatments in various ways including chemotherapy, surgery, and radiation. Cervical cancer treatment results in healing in between 66.3–95.1% of the cases, if performed at a pre-cancer stage. The treatment renders bad results when done at an advanced stage [4].

Optimal control has been applied to control and inhibit numerous diseases in several studies. Neilan et al. [7] provide a control function of the vaccination level to Taylor vaccination schedules, to minimize the number of infected individuals, and to minimize the vaccination cost on the basis of a SEIR epidemic model. Modelling optimal control of cervical cancer has been carried out by [1, 6], who use optimal vaccination strategies to suppress HPV infection effectively and to minimize the cost of vaccination.

However, in developing countries, HPV vaccination is not common. In general, patients who are seen by doctors are already in a pre-cancer stage. This research will apply optimal control for HPV infection models in cervical cancer on the basis of the implementation by Asih et al. [2]. Application of optimal control will be done as a treatment in the pre-cancer stage. We build a new model by adding control in the pre-cancer compartment, and we will solve the model equations numerically by using Pontryagin's maximum and the fourth order Runge-Kutta time integration scheme.

2 Mathematical Model

We revise the model proposed by Asih et al. [2] by adding a control function to the pre-cancer cell density. Let S, I, P, C and V, respectively, denote the density of normal (constituent) cells, the density of infected cells, the density of pre-cancer cells, the density of cancer cells and the density of free viral particles.

$$\begin{aligned}
\frac{dS}{dt} &= rS(1-(S+I)) - \alpha SV \\
\frac{dI}{dt} &= \alpha SV - aI - \delta I \\
\frac{dV}{dt} &= nI - cV \\
\frac{dP}{dt} &= \delta pI + bP - \theta\frac{P^2}{1+P^2} - u(t)P \\
\frac{dC}{dt} &= \theta\frac{P^2}{1+P^2} - kC
\end{aligned} \tag{1}$$

where $u(t)$, $0 \leq u(t) \leq 0.9$, is a control function. The complete description of the parameter values, as well as their biophysical meaning, is given in [2]. The objective function is used to minimize the density of cancer cells and to minimize the cost of treatment over time T days. The problem is stated as

$$\min_{u \in U} \int_0^T AC(t) + u^2(t)dt.$$

where the set of control is given by

$$U = \{u : [0, T] \to [0, 0.9]\},$$

subject to (1) and the initial condition

$$S(0) = S_0, I(0) = I_0, V(0) = V_0, P(0) = P_0, C(0) = C_0,$$

Further, the corresponding values at time T, that is $S(T), I(T), V(T), P(T), C(T)$ are free. The model variable A is a weight factor representing a balancing parameter, which determines the relative importance of the two factors in the optimal control problem [5].

3 The Optimum Control Problem

The optimal control problem is solved using Pontryagin's Maximum Principle. First we will define the Hamiltonian function, which is followed by the introduction of the stationary condition. Subsequently, we define the state equation and adjoint equation.

The Hamiltonian function of this problem can be stated as

$$H(t, x, u\lambda) = f(t, x, u) + \sum_{i=1}^{5} \lambda_i(t) g_i(t, x, u),$$

with

$$\begin{aligned}
f(t, x, u) &= AC(t) + u^2(t) \\
g_1(t, x, u) &= rS(1 - (S + I)) - \alpha SV \\
g_2(t, x, u) &= \alpha SV - aI - \delta I \\
g_3(t, x, u) &= nI - cV \\
g_4(t, x, u) &= \delta pI + bP - \theta \frac{P^2}{1 + P^2} - u(t)P \\
g_5(t, x, u) &= \theta \frac{P^2}{1 + P^2} - kC.
\end{aligned}$$

Hence we obtain

$$\begin{aligned} H = {} & AC + \lambda_S\{rS(1-(S+I)) - \alpha SV\} + \lambda_I\{\alpha SV - aI - \delta I\} + \lambda_V\{nI - cV\} \\ & + \lambda_P\left\{\delta pI + bP - \theta\frac{P^2}{1+P^2} - uP\right\} + \lambda_C\left\{\theta\frac{P^2}{1+P^2} - kC\right\}, \end{aligned} \tag{2}$$

where $\lambda_S, \lambda_I, \lambda_V, \lambda_P, \lambda_C$ are the associated adjoints for the states $S,\ I,\ V,\ P,\ C$, respectively.

For the stationary condition, the optimal condition is given by

$$\frac{\partial H}{\partial u}|_{u}* = 0.$$

Solving u^* from (2) gives

$$u^*(t) = \frac{P\lambda_P}{2}.$$

Furthermore, from taking the bound of u, we conclude that

$$u^*(t) = \min\left\{0.9, \max\left(0, \frac{p\lambda_P}{2}\right)\right\}. \tag{3}$$

The state equations are given by

$$\begin{aligned} \frac{dS}{dt} &= \frac{\partial H}{\partial \lambda_S} = rS(1-(S+I)) - \alpha SV \\ \frac{dI}{dt} &= \frac{\partial H}{\partial \lambda_I} = \alpha SV - aI - \delta I \\ \frac{dV}{dt} &= \frac{\partial H}{\partial \lambda_V} = nI - cV \\ \frac{dP}{dt} &= \frac{\partial H}{\partial \lambda_P} = \delta pI + bP - \theta\frac{P^2}{1+P^2} - uP \\ \frac{dC}{dt} &= \frac{\partial H}{\partial \lambda_C} = \theta\frac{P^2}{1+P^2} - kC \end{aligned} \tag{4}$$

subject to the initial condition

$$S(0) = S_0,\ I(0) = I_0,\ V(0) = V_0,\ P(0) = P_0,\ C(0) = C_0.$$

The adjoint equations are given by

$$
\begin{aligned}
\frac{d\lambda_S}{dt} &= -\frac{\partial H}{\partial S} = \lambda_S(\alpha V + rS + r(S + I - 1)) - \lambda_I(\alpha V)) \\
\frac{d\lambda_I}{dt} &= -\frac{\partial H}{\partial I} = \lambda_S(rS) = \lambda_I(a + \delta) - \lambda_V(n) - \lambda_P(\delta p) \\
\frac{d\lambda_V}{dt} &= -\frac{\partial H}{\partial V} = \lambda_S(\alpha S) - \lambda_I(\alpha S) + \lambda_v(c) \\
\frac{d\lambda_P}{dt} &= -\frac{\partial H}{\partial P} \\
&= -\lambda_P\left(b - u - \frac{2\theta P}{P^2+1} + \frac{2\theta P^3}{(P^2+1)^2}\right) - \lambda_C\left(\frac{2\theta P}{P^2+1} + \frac{2\theta P^3}{(P^2+1)^2}\right) \\
\frac{d\lambda_C}{dt} &= -\frac{\partial H}{\partial C} = A + k\lambda_C
\end{aligned}
\tag{5}
$$

subject to the transversal condition $\lambda_i(T) = 0$.

4 Numerical Simulation

To illustrate the effect of optimum control, we perform some numerical simulations by using the set of parameter values as in [2]. The state equations and adjoint equations will be solved numerically by the use of the fourth order Runge-Kutta method [8]. The state equations will be simulated using a forward time integration method while the adjoint equations are solved using a backward time integration method, since the state equations have initial conditions and the adjoint equations have conditions at the end-time.

Since the optimum control function is only active in the pre-cancer compartment, it makes sense that the sub population of normal cells, infected cells, and free virus pathogens are not influenced by this control function. In Fig. 1, it can be seen that the pre-cancer cell density significantly decreases, and starts to increase again after 35 days.

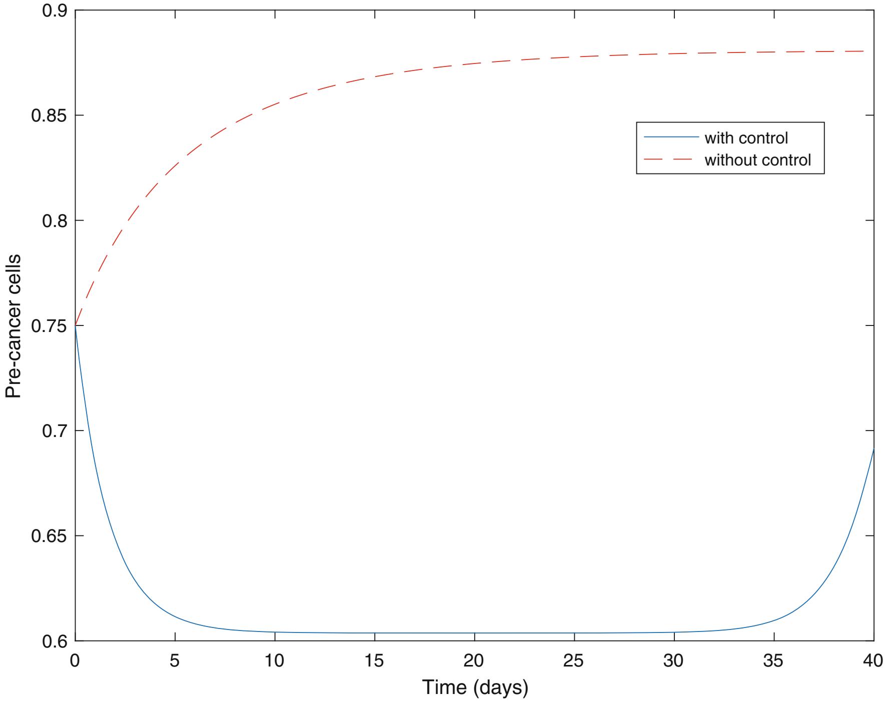


Fig. 1 The optimum control function makes the pre cancer cell density decrease significantly until about 35 days, by taking initial values for $(S_0,\ I_0,\ V_0,\ P_0,\ C_0) = (0.92,\ 0.055,\ 8.9,\ 0.75,\ 0.75)$ and $A = 0.1$

For the cancer cells, the obtained pattern is analogous, as one can see in Fig. 2. Without the application of control, the number of pre-cancer and cancer cells would increase and stabilize at the point of equilibrium. However, using control, the numbers of pre-cancer and cancer cells decrease and stabilize after approximately 35 days. This stabilization is followed by an increase.

This means that the treatment shows its effectiveness in reducing the number of pre-cancer and cancer cells until the 35th day. After that period, the effectiveness of the treatment will decrease so that the number of pre-cancer cells and cancer cells will increase again. In other words, the simulation results indicate that the therapy needs to be repeated periodically every 35 days.

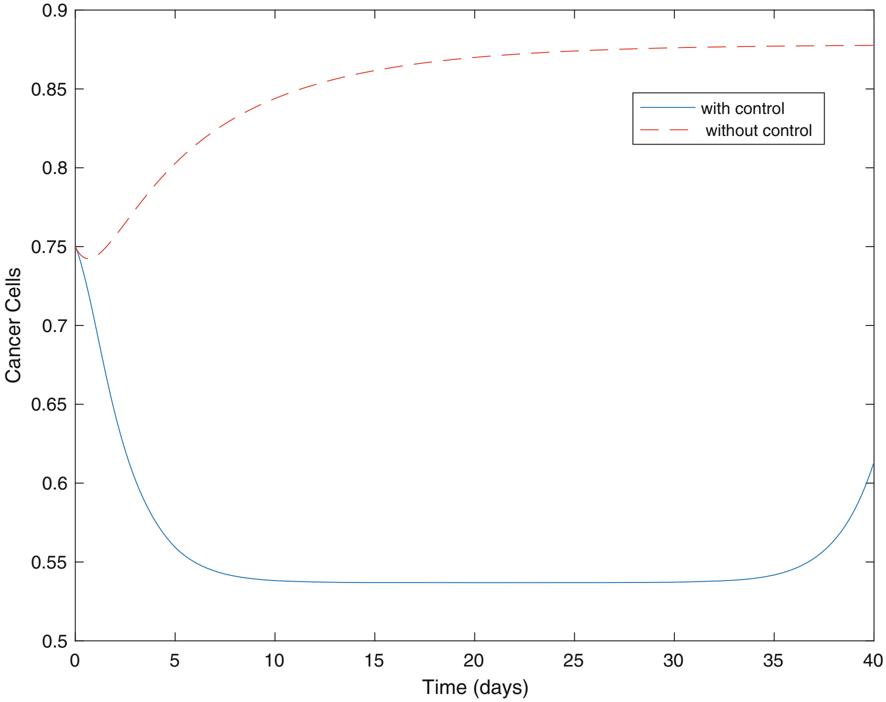


Fig. 2 Optimum control results into a significant decrease of the cancer cells

5 Conclusion

From the numerical simulations we can conclude that giving control in the pre-cancer compartment will imply a decrease of the number of pre-cancer and cancer cells. In our result the effectiveness of control is in the range of 35 days. Hence after 35 days the next treatment needs to be applied again. In other words, the current parameter setting indicates that the treatment can be given periodically with a period of 35 days. Though the result heavily depends on the parameter values and on the initial condition, we think that this model has some potential to predict optimal treatments against cervical cancer. Model calibration on the basis of medical data will be necessary in order to run more realistic simulations.

References

1. Al-Arydah, M T & Malik, T (2017) An Age-Structured Model of The Human Papillomavirus Dynamics and Optimal Vaccine Control Int J Biomath 10(6) 1750083
2. Asih, T S N, Lenhart, S, Wise, S, Aryati, L, Adi-Kusumo, F. Hardianti, M S, & Forde, J (2016). The dynamics of HPV infection and cervical cancer cells. Bull of math biol, 78(1), 4-20.
3. Bruni L, Albero G, Serrano B, Mena M, Gómez D, Muñoz J, Bosch FX, de Sanjosé S. ICO/IARC Information Centre on HPV and Cancer (HPV Information Centre). Human Papillomavirus and Related Diseases in the World. Summary Report 17 June 2019. [Date Accessed: December 15, 2019]
4. Iskandar, T M (2009). Pengelolaan Lesi Prakanker Serviks. Indones J Cancer, 3(3).
5. Lenhart, S & Workman, J T (2007). Optimal control applied to biological models. Chapman and Hall/CRC.
6. Malik, T, Imran, M, & Jayaraman, R (2016).Optimal control with multiple human papillomavirus vaccines. J theor biol 393: 179-193.
7. Neilan, R M, & Lenhart, S (2010) An Introduction to Optimal Control with an Application in Disease Modeling. In Modeling Paradigms and Analysis of Disease Transmission Models (pp. 67-82).
8. Oruh, B I, & Agwu, E U (2015) Application of Pontryagyn?s Maximum and Runge-Kutta Methods in Optimal Control Problems. IOSR Journal of Mathematics, 11(5): 43-63.
9. Farazi, P A, Siahpush, M, Michaud, T L, Kim, J, & Muchena, C (2019) Awareness of HPV and Cervical Cancer Prevention Among University Health Sciences Students in Cyprus J Cancer Educ 34(4): 685-690.

Nitsche's Master-Slave Method for Elastic Contact Problems

Tom Gustafsson, Rolf Stenberg, and Juha Videman

Abstract We survey the Nitsche's master-slave finite element method for elastic contact problems analysed in Gustafsson et al. (SIAM J Sci Comput 42:B425–B446, 2020). The main steps of the error analysis are recalled and numerical benchmark computations are presented.

1 Introduction

In a recent paper [2], we studied Nitsche's method applied to contact problems between two elastic bodies. We considered three formulations, two of which take the different material properties of the bodies into account. In the third method, which will be detailed in this paper, the body with a higher shear modulus is chosen as the master body and the slave one is mortared to it through Nitsche's method. We have the same error estimates for all three formulations but the master-slave approach appears to be the most straight-forward to implement.

Previously, the a priori estimates had been given under the assumption that the solution is in H^s, with $s > 3/2$, see e.g. [1], and the a posteriori estimates were derived using a saturation assumption. In [2], we were able to improve the error analysis and avoid the saturation assumption. The key idea was to interpret Nitsche's method as a stabilised mixed method.

The plan of this paper is the following. In the next section we recall the elastic contact problem. In Sect. 3 we present the Nitsche's formulation, the stabilised

T. Gustafsson · R. Stenberg (✉)
Department of Mathematics and Systems Analysis, Aalto University, Aalto, Finland
e-mail: tom.gustafsson@aalto.fi; rolf.stenberg@aalto.fi

J. Videman
CAMGSD/Departamento de Matemática, Universidade de Lisboa, Universidade de Lisboa, Lisbon, Portugal
e-mail: jvideman@math.tecnico.ulisboa.pt

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_89

method and show their equivalence. Then we summarise our error estimates and in the final section give some numerical results supplementing those of [2].

2 The Elastic Contact Problem

By $\Omega_i \subset \mathbb{R}^d$, $i = 1, 2$, $d = 2, 3$, we denote two elastic bodies in contact, with the common boundary $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$. The parts of $\partial\Omega_i$ on which Dirichlet and Neumann boundary conditions are imposed are denoted by $\Gamma_{D,i}$ and $\Gamma_{N,i}$, respectively. We let $\boldsymbol{u}_i : \Omega_i \to \mathbb{R}^d$ be the displacement of the body Ω_i and denote the strain tensors by $\boldsymbol{\varepsilon}(\boldsymbol{u}_i) = \frac{1}{2}(\nabla\boldsymbol{u}_i + (\nabla\boldsymbol{u}_i)^T)$. The materials will be assumed to be isotropic and homogeneous, i.e., $\boldsymbol{\sigma}_i(\boldsymbol{u}_i) = 2\mu_i\, \boldsymbol{\varepsilon}(\boldsymbol{u}_i) + \lambda_i \operatorname{tr} \boldsymbol{\varepsilon}(\boldsymbol{u}_i)\boldsymbol{I}$, where μ_i and λ_i are the Lamé parameters. We will exclude the possibility that the materials are nearly incompressible and hence it holds $\lambda_i \lesssim \mu_i$. We assume thar $\mu_1 \geq \mu_2$ and call the body Ω_1 the *master* and Ω_2 the *slave*. The outward unit normals to the boundaries are denoted by $\boldsymbol{n}_i$ and we define $\boldsymbol{n} = \boldsymbol{n}_1 = -\boldsymbol{n}_2$. Moreover, $\boldsymbol{t}$ denotes unit tangent vector satisfying $\boldsymbol{n} \cdot \boldsymbol{t} = 0$ (Fig. 1).

The traction vector $\boldsymbol{\sigma}_i(\boldsymbol{u}_i)\boldsymbol{n}_i$ is decomposed into its normal and tangential parts, viz.

$$\boldsymbol{\sigma}_i(\boldsymbol{u}_i)\boldsymbol{n}_i = \boldsymbol{\sigma}_{i,n}(\boldsymbol{u}_i) + \boldsymbol{\sigma}_{i,t}(\boldsymbol{u}_i). \tag{2.1}$$

For the scalar normal tractions we use the sign convention

$$\sigma_{1,n}(\boldsymbol{u}_1) = \boldsymbol{\sigma}_{1,n}(\boldsymbol{u}_1) \cdot \boldsymbol{n}_1, \quad \text{and} \quad \sigma_{2,n}(\boldsymbol{u}_2) = -\boldsymbol{\sigma}_{2,n}(\boldsymbol{u}_2) \cdot \boldsymbol{n}_2, \tag{2.2}$$

and note that on Γ these tractions are either both zero or continuous and compressive, i.e. it holds that

$$\sigma_{1,n}(\boldsymbol{u}_1) = \sigma_{2,n}(\boldsymbol{u}_2), \quad \sigma_{i,n}(\boldsymbol{u}_i) \leq 0, \; i = 1, 2. \tag{2.3}$$

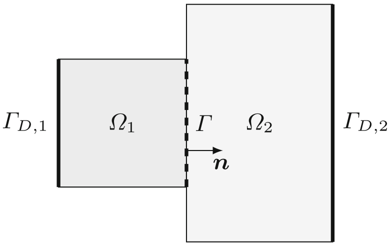


Fig. 1 Notation for the elastic contact problem

The physical non-penetration constraint on Γ reads as $\boldsymbol{u}_1 \cdot \boldsymbol{n}_1 + \boldsymbol{u}_2 \cdot \boldsymbol{n}_2 \leq 0$, which, defining $u_n = -(\boldsymbol{u}_1 \cdot \boldsymbol{n}_1 + \boldsymbol{u}_2 \cdot \boldsymbol{n}_2)$, can be written as

$$[\![u_n]\!] \geq 0, \tag{2.4}$$

where $[\![\cdot]\!]$ denotes the jump over Γ.

The *Nitsche's method* is derived from the problem in displacement variables: find $\boldsymbol{u} = (\boldsymbol{u}_1, \boldsymbol{u}_2)$, satisfying the *equilibrium equations* for the two bodies

$$-\,\mathbf{div}\, \boldsymbol{\sigma}_i(\boldsymbol{u}_i) = \boldsymbol{f}_i \quad \text{in } \Omega_i. \tag{2.5}$$

The *boundary conditions* are

$$\boldsymbol{u}_i = \mathbf{0} \quad \text{on } \Gamma_{D,i}, \quad \boldsymbol{\sigma}_i(\boldsymbol{u}_i)\boldsymbol{n}_i = \mathbf{0} \quad \text{on } \Gamma_{N,i}. \tag{2.6}$$

Next, we turn to the common boundary. Here we assume that the *tangential tractions vanish*

$$\boldsymbol{\sigma}_{i,t}(\boldsymbol{u}_i) = \mathbf{0} \ \text{ on } \Gamma, \tag{2.7}$$

and that the *normal stresses are continuous*

$$\boldsymbol{\sigma}_{i,t}(\boldsymbol{u}_i) = \mathbf{0}, \quad \sigma_{1,n}(\boldsymbol{u}_1) - \sigma_{2,n}(\boldsymbol{u}_2) = 0 \text{ on } \Gamma. \tag{2.8}$$

The contact conditions are the *non-penetration*

$$[\![u_n]\!] \geq 0 \ \text{ on } \Gamma, \tag{2.9}$$

and the *non-positivity of the normal stresses*, and the *compatibility condition*

$$\sigma_{i,n}(\boldsymbol{u}_i) \leq 0, \quad [\![u_n]\!]\, \sigma_{i,n}(\boldsymbol{u}_i) = 0 \ \text{ on } \Gamma. \tag{2.10}$$

By the *contact boundary* Γ_C we mean the subset of Γ wherein there is compression, i.e. $\sigma_{i,n}(\boldsymbol{u}_i) < 0$. On the complement $\Gamma \setminus \Gamma_C$, the normal tractions vanish. Note, however, that the contact boundary is a priori unknown as it depends on the solution.

The *stabilised method* is based on the formulation in which the normal traction is an independent unknown. Equations (2.8) and (2.10) are then replaced by

$$\lambda + \sigma_{1,n}(\boldsymbol{u}_1) = 0, \quad \lambda + \sigma_{2,n}(\boldsymbol{u}_2) = 0 \ \text{ on } \Gamma, \tag{2.11}$$

and

$$\lambda \geq 0, \quad [\![u_n]\!]\, \lambda = 0 \ \text{ on } \Gamma. \tag{2.12}$$

3 The Finite Element Methods

The continuous displacements are in $\boldsymbol{V} = \boldsymbol{V}_1 \times \boldsymbol{V}_2$, with

$$\boldsymbol{V}_i = \{\boldsymbol{w}_i \in [H^1(\Omega_i)]^d : \boldsymbol{w}_i|_{\Gamma_{D,i}} = \boldsymbol{0}\}.$$

By $\mathcal{C}_h^i$ we denote the simplicial mesh on Ω_i which induces a facet mesh $\mathcal{G}_h^i$ on Γ. The finite element solution is sought in $\boldsymbol{V}_h = \boldsymbol{V}_{1,h} \times \boldsymbol{V}_{2,h}$, with

$$\boldsymbol{V}_{i,h} = \{\boldsymbol{v}_{i,h} \in \boldsymbol{V}_i : \boldsymbol{v}_{i,h}|_K \in [P_p(K)]^d \; \forall K \in \mathcal{C}_h^i\}.$$

First, we recall the *Nitsche's master-slave method*. To this end, we define the mesh function $\mathcal{h}_2$ on Γ by $\mathcal{h}_2|_E = h_E$ for $E \in \mathcal{G}_h^2$. The method reads as follows: find $\boldsymbol{u}_h \in \boldsymbol{V}_h$ such that

$$\sum_{i=1}^{2} (\boldsymbol{\sigma}_i(\boldsymbol{u}_{i,h}), \boldsymbol{\varepsilon}(\boldsymbol{v}_{i,h}))_{\Omega_i} + \big\langle \sigma_{2,n}(\boldsymbol{u}_{2,h}), [\![v_{h,n}]\!] \big\rangle_{\Gamma_c(\boldsymbol{u}_h)} + \big\langle \sigma_{2,n}(\boldsymbol{v}_{2,h}), [\![u_{h,n}]\!] \big\rangle_{\Gamma_c(\boldsymbol{u}_h)}$$

$$+\gamma \Big\langle \frac{\mu_2}{\mathcal{h}_2} [\![u_{h,n}]\!] , [\![v_{h,n}]\!] \Big\rangle_{\Gamma_c(\boldsymbol{u}_h)} = \sum_{i=1}^{2} (\boldsymbol{f}_i, \boldsymbol{v}_{i,h})_{\Omega_i} \quad \forall \boldsymbol{v}_h \in \boldsymbol{V}_h, \tag{3.1}$$

where $\gamma > C_I^{-1}$, with $C_I > 0$ denoting the constant in the discrete trace inquality

$$C_I h_E \|\sigma_{2,n}(\boldsymbol{u}_{2,h})\|_{0,E}^2 \le \mu_2 \|\boldsymbol{\sigma}_2(\boldsymbol{u}_{2,h})\|_{0,K}^2, \quad E = K \cap \Gamma, \tag{3.2}$$

and

$$\Gamma_c(\boldsymbol{u}_h) = \{\boldsymbol{x} \in \Gamma : \sigma_{2,n}(\boldsymbol{u}_{2,h}) + \gamma \frac{\mu_2}{\mathcal{h}_2} [\![u_{h,n}]\!] < 0\}. \tag{3.3}$$

The nonlinearity of the problem stems from this dependence of the contact boundary on the solution.

To define *the stabilised method* we need some additional notation. The normal traction λ is in the space $H^{-\frac{1}{2}}(\Gamma)$, dual to the trace space $H^{\frac{1}{2}}(\Gamma)$, with the norm $\| \cdot \|_{-\frac{1}{2},\Gamma}$ defined by duality. Defining

$$\mathcal{B}(\boldsymbol{w}, \xi; \boldsymbol{v}, \eta) = \sum_{i=1}^{2} (\boldsymbol{\sigma}_i(\boldsymbol{w}_i), \boldsymbol{\varepsilon}(\boldsymbol{v}_i))_{\Omega_i} - \big\langle [\![v_n]\!] , \xi \big\rangle - \big\langle [\![w_n]\!] , \eta \big\rangle, \tag{3.4}$$

and

$$\Lambda = \{\xi \in H^{-\frac{1}{2}}(\Gamma) : \langle w, \xi \rangle \ge 0 \;\; \forall w \in H^{\frac{1}{2}}(\Gamma), \; w \ge 0 \text{ a.e. on } \Gamma\}, \tag{3.5}$$

the mixed formulation of the problem is: find $(\boldsymbol{u}, \lambda) \in \boldsymbol{V} \times \varLambda$ such that

$$\mathcal{B}(\boldsymbol{u}, \lambda; \boldsymbol{v}, \eta - \lambda) \leq \sum_{i=1}^{2} (\boldsymbol{f}_i, \boldsymbol{v}_i)_{\Omega_i} \quad \forall (\boldsymbol{v}, \eta) \in \boldsymbol{V} \times \varLambda. \tag{3.6}$$

The traction is approximated on the mesh $\mathcal{G}_h^{12}$ obtained as the intersection of $\mathcal{G}_h^1$ and $\mathcal{G}_h^2$:

$$Q_h = \{\eta_h \in H^{-\frac{1}{2}}(\Gamma) : \eta_h|_E \in P_p(E) \; \forall E \in \mathcal{G}_h^{12}\}. \tag{3.7}$$

(Note that since the approximation is discontinuous across element boundaries this is possible, even though the elements are general polygonals polyhedrons.) Moreover, we introduce a subset of $\varLambda$, denoted by $\varLambda_h$, as the positive part of Q_h, i.e.

$$\varLambda_h = \{\eta_h \in Q_h : \eta_h \geq 0\}. \tag{3.8}$$

The stabilised bilinear form $\mathcal{B}_h$ is defined through

$$\mathcal{B}_h(\boldsymbol{w}_h, \xi_h; \boldsymbol{v}_h, \eta_h) = \mathcal{B}(\boldsymbol{w}_h, \xi_h; \boldsymbol{v}_h, \eta_h) - \alpha \mathcal{S}_h(\boldsymbol{w}_h, \xi_h; \boldsymbol{v}_h, \eta_h), \tag{3.9}$$

where $\alpha > 0$ is a stabilisation parameter and

$$\mathcal{S}_h(\boldsymbol{w}_h, \xi_h; \boldsymbol{v}_h, \eta_h) = \left\langle \frac{\hbar_2}{\mu_2} (\xi_h + \sigma_{2,n}(\boldsymbol{w}_{2,h})), \eta_h + \sigma_{2,n}(\boldsymbol{v}_{2,h}) \right\rangle_\Gamma. \tag{3.10}$$

The stabilised method is: find $(\boldsymbol{u}_h, \lambda_h) \in \boldsymbol{V}_h \times \varLambda_h$ such that

$$\mathcal{B}_h(\boldsymbol{u}_h, \lambda_h; \boldsymbol{v}_h, \eta_h - \lambda_h) \leq \sum_{i=1}^{2} (\boldsymbol{f}_i, \boldsymbol{v}_i)_{\Omega_i} \quad \forall (\boldsymbol{v}_h, \eta_h) \in \boldsymbol{V}_h \times \varLambda_h. \tag{3.11}$$

Note that

$$\mathcal{S}_h(\boldsymbol{u}_h, \lambda_h; \boldsymbol{v}_h, \eta_h) = \left\langle \frac{\hbar_2}{\mu_2} (\lambda_h + \sigma_{2,n}(\boldsymbol{u}_{2,h})), \eta_h + \sigma_{2,n}(\boldsymbol{v}_{2,h}) \right\rangle_\Gamma, \tag{3.12}$$

and hence the stabilised term amounts to a symmetric term including the residual $\lambda_h + \sigma_{2,n}$.

Now, by testing with $(\mathbf{0}, \eta_h)$ in (3.11), one can infer that

$$\lambda_h = \Big(- \sigma_{2,n}(\boldsymbol{u}_{2,h}) - \frac{\mu_2}{\alpha \hbar_2} [\![u_{h,n}]\!] \Big)_+. \tag{3.13}$$

Substituting this into the first equation obtained by testing with $(\boldsymbol{v}_h, 0)$ in (3.11) we get the Nitsche's method (3.1) with $\gamma = \alpha^{-1}$.

4 Error Estimates

The error estimate will be derived in the norm

$$|||(\boldsymbol{w}, \xi)|||^2 = \sum_{i=1}^{2} \big(\mu_i \|\boldsymbol{w}\|_{1,\Omega_i}^2 + \frac{1}{\mu_i} \|\xi\|_{-\frac{1}{2},\Gamma}^2\big). \tag{4.1}$$

The stability of the continuous problem is given in the following theorem.

Theorem 1 *For every* $(\boldsymbol{w}, \xi) \in \boldsymbol{V} \times Q$ *there exists* $\boldsymbol{v} \in \boldsymbol{V}$ *such that*

$$\mathcal{B}(\boldsymbol{w}, \xi; \boldsymbol{v}, -\xi) \gtrsim |||(\boldsymbol{w}, \xi)|||^2 \ \textit{and} \ \|\boldsymbol{v}\|_V \lesssim |||(\boldsymbol{w}, \xi)|||. \tag{4.2}$$

The idea with stabilisation is that it yields a method which is always stable in a mesh-dependent norm for the Lagrange multiplier. Defining

$$|||(\boldsymbol{w}_h, \xi_h)|||_h^2 = \sum_{i=1}^{2} \mu_i \|\boldsymbol{w}\|_{1,\Omega_i}^2 + \mu_2^{-1} \sum_{E \in \mathcal{G}_h^2} h_E \|\xi_h\|_{0,E}^2, \tag{4.3}$$

we directly obtain the estimate.

Theorem 2 *Suppose that* $0 < \alpha < C_I$. *Then, for every* $(\boldsymbol{w}_h, \xi_h) \in \boldsymbol{V}_h \times Q_h$, *there exists* $\boldsymbol{v}_h \in \boldsymbol{V}_h$ *such that*

$$\mathcal{B}_h(\boldsymbol{w}_h, \xi_h; \boldsymbol{v}_h, -\xi_h) \gtrsim |||(\boldsymbol{w}_h, \xi_h)|||_h^2 \ \textit{and} \ \|\boldsymbol{v}_h\|_V \lesssim |||(\boldsymbol{w}_h, \xi_h)|||_h. \tag{4.4}$$

In view of Theorem 2, the classical Verfürth trick yields the stability estimate in the correct norms.

The error analysis then follows in a standard way, except for the additional term

$$\Big(\mu_2^{-1} \sum_{E \in \mathcal{G}_h^2} h_E \|\eta_h + \sigma_{2,n}(\boldsymbol{v}_{2,h})\|_{0,E}^2\Big)^{1/2}, \tag{4.5}$$

where $(\boldsymbol{v}_h, \eta_h)$ are the interpolants of $(\boldsymbol{u}, \lambda)$. However, by a posteriori error analysis techniques this term can be bounded by

$$|||(\boldsymbol{u} - \boldsymbol{v}_h, \lambda - \eta_h)||| + \text{HOT}, \tag{4.6}$$

where HOT stands for a higher order oscillation term. We thus arrive at the following quasi-optimality estimate of the method.

Theorem 3 *For* $0 < \alpha < C_I$ *it holds that*

$$|||(\boldsymbol{u}-\boldsymbol{u}_h, \lambda-\lambda_h)||| \lesssim \inf_{(\boldsymbol{v}_h,\eta_h)\in \boldsymbol{V}_h\times\Lambda_h} \Big(|||(\boldsymbol{u}-\boldsymbol{v}_h, \lambda-\eta_h)||| + \sqrt{\langle [\![u_n]\!], \eta_h\rangle}\Big) + \text{HOT}. \tag{4.7}$$

For the a posteriori error analysis, we define the local estimators

$$\eta_K^2 = \frac{h_K^2}{\mu_i}\|\mathbf{div}\,\boldsymbol{\sigma}_i(\boldsymbol{u}_{i,h}) + \boldsymbol{f}_i\|_{0,K}^2, \quad K \in \mathcal{C}_h^i, \tag{4.8}$$

$$\eta_{E,\Omega}^2 = \frac{h_E}{\mu_i}\left\|[\![\boldsymbol{\sigma}_i(\boldsymbol{u}_{i,h})\boldsymbol{n}]\!]\right\|_{0,E}^2, \quad E \in \mathcal{E}_h^i, \tag{4.9}$$

$$\eta_{E,\Gamma}^2 = \frac{h_E}{\mu_i}\|\boldsymbol{\sigma}_{i,t}(\boldsymbol{u}_{i,h})\|_{0,E}^2 + \frac{\mu_i}{h_E}\|([\![u_{h,n}]\!])_-\|_{0,E}^2, \qquad E \in \mathcal{G}_h^i, \tag{4.10}$$

$$\eta_{E,\Gamma_N}^2 = \frac{h_E}{\mu_i}\left\|\boldsymbol{\sigma}_i(\boldsymbol{u}_{i,h})\boldsymbol{n}\right\|_{0,E}^2, \quad E \in \mathcal{N}_h^i, \tag{4.11}$$

$$\zeta_{E,\Gamma}^2 = \frac{h_E}{\mu_2}\left\|\lambda_h + \sigma_{2,n}(\boldsymbol{u}_{2,h})\right\|_{0,E}^2, \quad E \in \mathcal{G}_h^2, \tag{4.12}$$

with $i = 1, 2$. The corresponding global estimator η is then defined as

$$\eta^2 = \sum_{i=1}^{2}\Big\{\sum_{K\in\mathcal{C}_h^i}\eta_K^2 + \sum_{E\in\mathcal{E}_h^i}\eta_{E,\Omega}^2 + \sum_{E\in\mathcal{G}_h^i}\eta_{E,\Gamma}^2 + \sum_{E\in\mathcal{N}_h^i}\eta_{E,\Gamma_N}^2\Big\} + \sum_{E\in\mathcal{G}_h^2}\zeta_{E,\Gamma}^2 . \tag{4.13}$$

In addition, we need an estimator S defined only globally as

$$S^2 = \big\langle([\![u_{h,n}]\!])_+, \lambda_h\big\rangle_\Gamma. \tag{4.14}$$

Theorem 4 (A Posteriori Error Estimate) *It holds that*

$$\eta \lesssim |||(\boldsymbol{u}-\boldsymbol{u}_h, \lambda-\lambda_h)||| \lesssim \eta + S. \tag{4.15}$$

5 Numerical Experiments

We investigate the performance of the master-slave method by solving adaptively the problem (3.1) using P_2 elements and the following geometry:

$$\Omega_1 = [0.5, 1] \times [0.25, 0.75], \quad \Omega_2 = [1, 1.6] \times [0, 1]. \tag{5.1}$$

The boundary conditions are defined on

$$\Gamma_{D,1} = \{(x, y) \in \partial\Omega_1 : x = 0.5\}, \quad \Gamma_{N,1} = \partial\Omega_1 \setminus (\Gamma_{D,1} \cup \Gamma), \tag{5.2}$$

$$\Gamma_{D,2} = \{(x, y) \in \partial\Omega_2 : x = 1.6\}, \quad \Gamma_{N,2} = \partial\Omega_2 \setminus (\Gamma_{D,2} \cup \Gamma), \tag{5.3}$$

while the material parameters are $E_1 = 1$, $E_2 = 0.1$ and $\nu_1 = \nu_2 = 0.3$. The loading is

$$\boldsymbol{f}_1 = (0, -\tfrac{1}{20}), \quad \boldsymbol{f}_2 = (0, 0), \tag{5.4}$$

which causes the left block to bend downwards. The active contact boundary Γ_c is sought by alternately evaluating the inequality condition in (3.3) and solving the linearised problem with $\alpha = 10^{-2}$.

The final meshes and the respective approximation of the contact force is given in Fig. 2a and b where we use the notation $\{\{\sigma_n(\boldsymbol{u}_h)\}\}$ for the mean normal stress over the contact boundary. The resulting global a posteriori error estimator is given as a function of the number of degrees-of-freedom in Fig. 2c. We observe, in particular, that the asymptotic rate of convergence for the total error estimator is improved from $\mathcal{O}(N^{-0.43})$ to $\mathcal{O}(N^{-1.02})$ where the latter corresponds to the rate of convergence one expects from P_2 elements and a completely smooth solution.

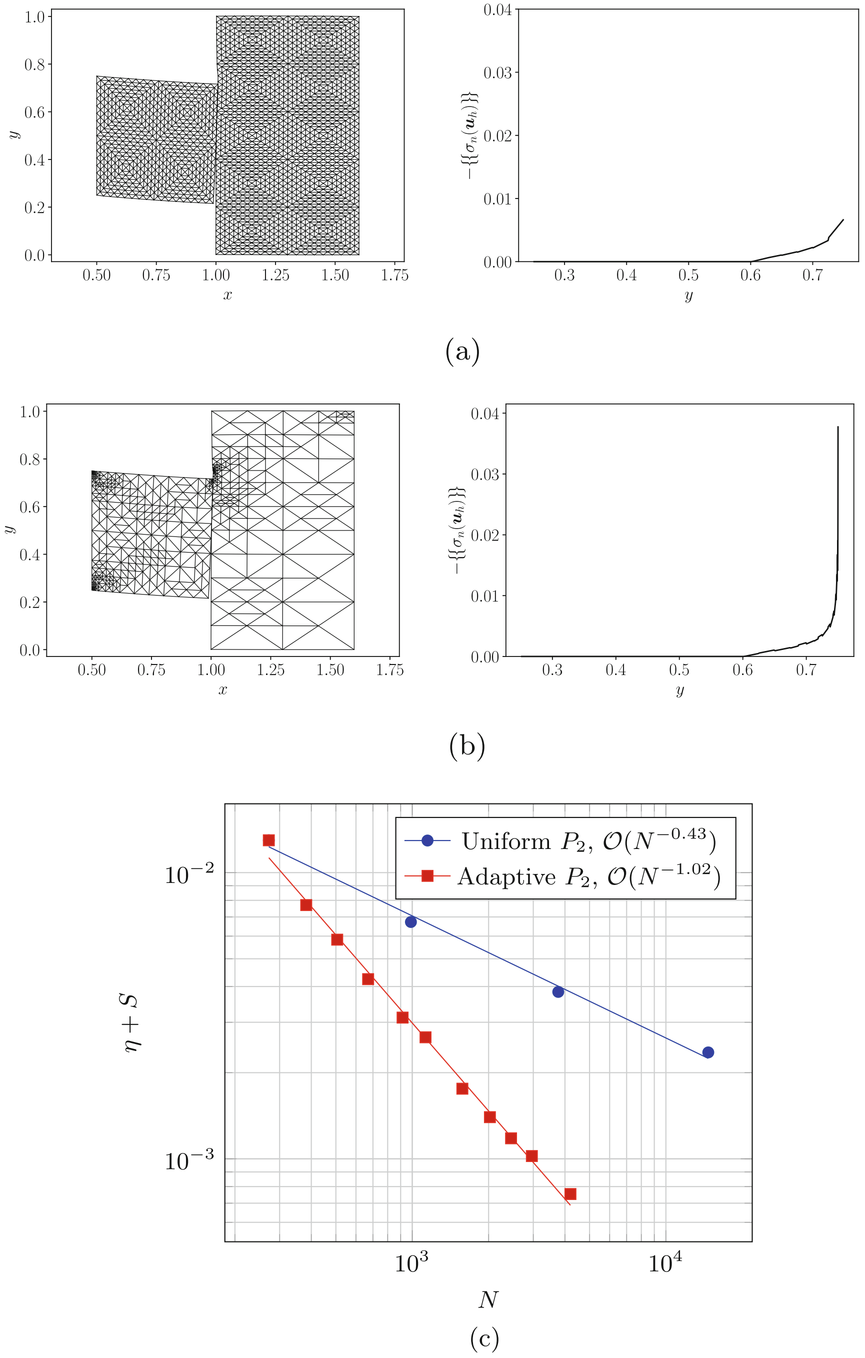


Fig. 2 The numerical example, the resulting meshes and global error estimators. (**a**) P_2 after 3 uniform refinements. (**b**) P_2 after 10 adaptive refinements. (**c**) The global a posteriori error estimator $\eta + S$ as a function of the number of degrees-of-freedom N

Acknowledgments This work was (partially) funded by the Portuguese government through FCT—Fundação para a Ciência e a Tecnologia, I.P., under the projects PTDC/MAT-PUR/28686/2017, and UID/MAT/04459/2013, and by the Academy of Finland, decision nr. 324611.

References

1. F. Chouly, M. Fabre, P. Hild, R. Mlika, J. Pousin, and Y. Renard, *An overview of recent results on Nitsche's method for contact problems*, in Geometrically Unfitted Finite Element Methods and Applications, S. Bordas, E. Burman, M. Larson, and M. Olshanskii, eds., vol. 121 of Lecture Notes in Computational Science and Engineering, Springer, 2017, pp. 93–141.
2. T. Gustafsson, R. Stenberg, J. Videman. On Nitsche's method for elastic contact problems. SIAM Journal of Scientific Computing. 42 (2020) B425–B446.

The Fixed-Stress Splitting Scheme for Biot's Equations as a Modified Richardson Iteration: Implications for Optimal Convergence

Erlend Storvik, Jakub Wiktor Both, Jan Martin Nordbotten, and Florin Adrian Radu

Abstract The fixed-stress splitting scheme is a popular method for iteratively solving the Biot equations. The method successively solves the flow and mechanics subproblems while adding a stabilizing term to the flow equation, which includes a parameter that can be chosen freely. However, the convergence properties of the scheme depend significantly on this parameter and choosing it carelessly might lead to a very slow, or even diverging, method. In this paper, we present a way to exploit the matrix structure arising from discretizing the equations in the regime of impermeable porous media in order to obtain *a priori* knowledge of the optimal choice of this tuning/stabilization parameter.

1 Introduction

Due to many applications of societal consequence, ranging from life sciences to environmental engineering, simulation of flow in deformable porous media is of great interest. A choice of a model is the quasi-static Biot equations, which couples balance of linear momentum, and volume balance. In the most basic model, allowing only small deformations and fully saturated media, the equations become linear. However, the coupled problem has a complex structure, and it is not trivial to solve the full problem monolithically. On the other hand, there are many efficient solvers available for both porous media flow and elasticity. Therefore, splitting solvers are a popular alternative, where one, often using readily available software, solves the subproblems iteratively.

In order to ensure that such splitting solvers converge, a stabilizing term is added to one or both of the equations. Choosing this term is important for the convergence properties of the scheme. Particularly, in problems with high coupling strength,

E. Storvik (✉) · J. W. Both · J. M. Nordbotten · F. A. Radu
University of Bergen, Bergen, Norway
e-mail: erlend.storvik@uib.no; jakub.both@uib.no; jan.nordbotten@uib.no; florin.radu@uib.no

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_90

the number of iterations to achieve convergence varies significantly for different stabilizations. In the fixed-stress splitting scheme, the original idea was to choose the stabilization term in order to preserve the volumetric stress over the iterations, see [1], by adding a scaled increment of pressures to the flow equation. Convergence was proved mathematically in [2] and later, using a different approach, in [3]. In [4], a range in which the optimal stabilization term resides was provided theoretically, and verified numerically. Additionally, a numerical scheme to find the parameter was proposed.

In this work, we continue the discussion on the optimal choice of the stabilization parameter of the fixed-stress splitting scheme. Particularly, for the case of impermeable porous media, where the coupling strength is known to be high [5]. Moreover, we expect it to be relevant for the more general low-permeable case. To do so, we examine the matrix structure of the linear problem that arises when we apply the fixed-stress splitting scheme for an idealistically impermeable problem and realize this as a modified Richardson iteration. Using theory for the Richardson iteration, we find the optimal stabilization parameter and discuss how to compute it. To summarize, the contributions are:

- A proof that the fixed-stress splitting scheme can be posed as a modified Richardson iteration, with a link between the optimal constant for the modified Richardson iteration and the stabilization parameter in the fixed-stress splitting scheme.
- A discussion on how to compute this stabilization parameter.

It is also worth noticing that the fixed-stress splitting scheme can be derived using a generalized gradient flow approach, [6], and can be combined with a wide range of discretizations, including space-time finite elements [7]. Moreover, it can be seen as a smoother for multigrid methods [8]. In [9], the authors derived a relation between the fixed-stress splitting scheme and the modified Richardson iteration, and applied it as a preconditioner for Krylov subspace methods for solving the monolithic problem.

2 The Quasi-Static Linear Biot Equations

The quasi-static linear Biot equations are a coupling of linear momentum balance and mass balance (see [10]):

$$-\nabla \cdot \boldsymbol{\sigma} = \boldsymbol{f}, \tag{1}$$

$$\frac{\partial_t m}{\rho} + \nabla \cdot \boldsymbol{q} = S_f, \tag{2}$$

where $\boldsymbol{\sigma}$ is the stress tensor of the medium, m is the fluid mass, ρ is the fluid density, $\boldsymbol{q}$ is the Darcy flux, and $\boldsymbol{f}$ and S_f are the body forces and sources/sinks, respectively.

Now, the St. Venant-Kirchhoff stress tensor for the effective stress and Darcy's law is applied:

$$\boldsymbol{\sigma} = 2\mu\varepsilon\left(\boldsymbol{u}\right) + \lambda\nabla\cdot\boldsymbol{u}I - \alpha pI, \tag{3}$$

$$\boldsymbol{q} = -\kappa\left(\nabla p - \boldsymbol{g}\rho\right), \tag{4}$$

where $\boldsymbol{u}$ is the displacement, $\varepsilon\left(\boldsymbol{u}\right) = \frac{1}{2}\left(\nabla\boldsymbol{u} + \nabla\boldsymbol{u}^\top\right)$ is the (linear) strain tensor, μ, λ are the Lamé parameters, α is the Biot-Willis constant, p is the fluid pressure, $\boldsymbol{g}$ is the gravitational vector and κ is the permeability. The volumetric change is asserted to be proportional to the change in pore pressure and mechanical displacement; $\frac{\partial_t m}{\rho} = \frac{\partial}{\partial t}\left(\frac{p}{M} + \alpha\nabla\cdot\boldsymbol{u}\right)$, where M is a compressibility constant. Finally, we define initial conditions $\boldsymbol{u}(t = 0) = \boldsymbol{u}_0$ and $p(t = 0) = p_0$, and boundary conditions $\boldsymbol{u}|_{\Gamma_{N,\boldsymbol{u}}} = \boldsymbol{u}_D$, $\boldsymbol{\sigma}\cdot\boldsymbol{n}|_{\Gamma_{N,\boldsymbol{u}}} = \boldsymbol{\sigma}_N$, $p|_{\Gamma_{D,p}} = p_D$ and $\boldsymbol{q}\cdot\boldsymbol{n}|_{\Gamma_{N,p}} = \boldsymbol{q}_N$, where $\partial\Omega = \Gamma_{D,\boldsymbol{u}} \cup \Gamma_{N,\boldsymbol{u}} = \Gamma_{D,p} \cup \Gamma_{N,p}$, for a Lipschitz domain, $\Omega \subset \mathbb{R}^d$, d being the spatial dimension.

3 The Discretized Biot Equations in Matrix Form

Discretizing the Biot equations by e.g., conforming finite elements and implicit Euler in a two-field formulation $(\boldsymbol{u}, p)$ (making the substitutions (3) and (4) in (1) and (2)), the resulting linear system in each time step can be written as follows

$$\begin{pmatrix} \mathbf{A} & -\mathbf{B}^\top \\ \mathbf{B} & \mathbf{C} \end{pmatrix} \begin{pmatrix} \mathbf{u}_h \\ \mathbf{p}_h \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix}, \tag{5}$$

where $\mathbf{A}$ is the linear elasticity matrix, $\mathbf{B}$ is the coupling matrix, $\mathbf{C}$ is the single-phase flow matrix, $\mathbf{f}$ and $\mathbf{g}$ correspond to the body forces and sources/sinks, respectively, and $\mathbf{u}_h$ and $\mathbf{p}_h$ are the coefficient vectors for the discretized displacement and pressure, respectively. For the rest of this paper, we consider an inf-sup stable finite element pair $(\mathbf{V}_h, Q_h)$. Furthermore, for impermeable porous media, the submatrix $\mathbf{C}$ reduces to $\frac{1}{M}\mathbf{M}$, where $\mathbf{M}$ is the mass matrix.

4 The Fixed-Stress Splitting Scheme as a Modified Richardson Iteration

As mentioned in the introduction, the fixed-stress splitting scheme adds a scaled incremental pressure to the flow equation while eliminating its dependence on the displacement. For impermeable media, $\kappa = 0$, this results in the following linear system

$$\begin{pmatrix} \mathbf{A} & -\mathbf{B}^\top \\ \mathbf{0} & \left(L+\frac{1}{M}\right)\mathbf{M} \end{pmatrix} \begin{pmatrix} \Delta\mathbf{u}_h^i \\ \Delta\mathbf{p}_h^i \end{pmatrix} = \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \end{pmatrix} - \begin{pmatrix} \mathbf{A} & -\mathbf{B}^\top \\ \mathbf{B} & \frac{1}{M}\mathbf{M} \end{pmatrix} \begin{pmatrix} \mathbf{u}_h^{i-1} \\ \mathbf{p}_h^{i-1} \end{pmatrix}, \tag{6}$$

where L is the stabilization parameter, $i \geq 1$ is the iteration index, $\Delta\mathbf{u}_h^i = \mathbf{u}_h^i - \mathbf{u}_h^{i-1}$ and $\Delta\mathbf{p}_h^i = \mathbf{p}_h^i - \mathbf{p}_h^{i-1}$.

Remark 1 (Optimality of Alternative Splitting) Notice that for impermeable media, $\kappa = 0$, and particular discretizations, e.g., $(\mathbf{P}_1, P_0)$, the undrained split [11], will converge in one iteration. Nevertheless, our experience is that the optimized fixed-stress splitting proposed here is superior for slight perturbations of the permeability.

In the original formulation of the fixed-stress splitting scheme [1] the constant $L = \frac{\alpha^2}{K_{dr}}$ was chosen, where $K_{dr} = \frac{2\mu}{d} + \lambda$, is the physical drained bulk modulus. Later, in [2], convergence was proved for $L \geq \frac{\alpha^2}{2K_{dr}}$. In [4], an interval containing the optimal stabilization parameter, including both of the two aforementioned parameters, was provided through mathematical proofs, and then verified numerically. We now show that the optimal parameter for impermeable media is a value in this domain that is possible to compute *a priori*. For this, we need the mathematical bulk modulus.

Definition 1 The mathematical bulk modulus, $K_{dr}^\star \geq K_{dr}$, is defined as the largest constant such that

$$2\mu \left\|\varepsilon\left(\boldsymbol{u}_h\right)\right\|^2 + \lambda\|\nabla\cdot\boldsymbol{u}_h\|^2 \geq K_{dr}^\star\|\nabla\cdot\boldsymbol{u}_h\|^2 \qquad \text{for all } \boldsymbol{u}_{\boldsymbol{h}} \in \mathbf{V}_h. \tag{7}$$

It is easily seen that the physical drained bulk modulus satisfies inequality (7), but generally the bound is not sharp.

Furthermore, by assuming that we have an inf-sup stable discretization we are able to define the following parameter, β, that is important in finding the optimal L.

Lemma 1 *Assume that the pair $(\boldsymbol{V}_h, Q_h)$ is inf-sup stable. There exists $\beta > 0$ such that for any $p_h \in Q_h$ there exists a $\boldsymbol{u}_h \in \boldsymbol{V}_h$ satisfying $\langle\nabla\cdot\boldsymbol{u}_h, q_h\rangle = \langle p_h, q_h\rangle$ for all $q_h \in Q_h$ and*

$$2\mu \left\|\varepsilon(\boldsymbol{u}_h)\right\|^2 + \lambda \left\|\nabla\cdot\boldsymbol{u}_h\right\|^2 \leq \beta \left\|p_h\right\|^2. \tag{8}$$

A proof of this lemma can be found in [4].

Theorem 1 *For impermeable media, $\kappa = 0$, the fixed-stress splitting scheme (6) can be interpreted as the modified Richardson iteration*

$$\mathbf{p}_h^i = \mathbf{p}_h^{i-1} + \omega\left(\mathbf{M}^{-1}\tilde{\mathbf{g}} - \mathbf{M}^{-1}\mathbf{S}\mathbf{p}_h^{i-1}\right) \tag{9}$$

where $\omega = \left(L + \frac{1}{M}\right)^{-1}$, $\tilde{\mathbf{g}} = \mathbf{g} - \mathbf{A}^{-1}\mathbf{B}\,\mathbf{f}$, *and* $\mathbf{S} = \frac{1}{M}\mathbf{M} + \mathbf{B}\mathbf{A}^{-1}\mathbf{B}^\top$ *is the Schur complement. For the error* $\mathbf{e}_p^i := \mathbf{p}_h^i - \boldsymbol{p}_h$ *it holds for all* $i \geq 1$

$$\left\langle \mathbf{M}\,\mathbf{e}_p^i, \mathbf{e}_p^i \right\rangle \leq \left\| \mathbf{I} - \omega \mathbf{M}^{-1/2}\mathbf{S}\mathbf{M}^{-1/2} \right\|_2^2 \left\langle \mathbf{M}\,\mathbf{e}_p^{i-1}, \mathbf{e}_p^{i-1} \right\rangle. \tag{10}$$

From that, the optimal choice of ω *is*

$$\omega_{\text{opt}} = \frac{2}{\lambda_{\max}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right) + \lambda_{\min}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right)} \tag{11}$$

with the identifications

$$\lambda_{\max}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right) = \frac{1}{M} + \frac{\alpha^2}{K_{dr}^\star} \quad \text{and} \quad \lambda_{\min}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right) = \frac{1}{M} + \frac{\alpha^2}{\beta},$$

where α *is the Biot-Willis coupling constant,* M *is a compressibility coefficient,* $K_{dr}^\star$ *is the mathematical bulk modulus and* β *is the constant from Lemma 1.*

Proof From (6) we have that

$$\Delta \mathbf{p}_h^i = \left(\left(L + \frac{1}{M}\right)\mathbf{M}\right)^{-1}\left(\mathbf{g} - \mathbf{B}\mathbf{u}_h^{i-1} - \frac{1}{M}\mathbf{M}\mathbf{p}_h^{i-1}\right) \tag{12}$$

and

$$\mathbf{u}_h^{i-1} = \mathbf{A}^{-1}\mathbf{f} + \mathbf{A}^{-1}\mathbf{B}^\top \mathbf{p}_h^{i-1}. \tag{13}$$

From the update of pressures in the fixed-stress splitting scheme we get the modified Richardson iteration (9)

$$\mathbf{p}_h^i = \mathbf{p}_h^{i-1} + \Delta\mathbf{p}_h^i = \mathbf{p}_h^{i-1} + \left(L + \frac{1}{M}\right)^{-1}\left(\mathbf{M}^{-1}\tilde{\mathbf{g}} - \mathbf{M}^{-1}\mathbf{S}\mathbf{p}_h^{i-1}\right).$$

To find the optimal choice of the parameter ω in (9) we modify the equation slightly by making the substitution $\mathbf{p}_h^i = \mathbf{M}^{-\frac{1}{2}}\tilde{\mathbf{p}}_h^i$ and multiply from the left by $\mathbf{M}^{\frac{1}{2}}$ to get

$$\tilde{\mathbf{p}}_h^i = \tilde{\mathbf{p}}_h^{i-1} + \left(L + \frac{1}{M}\right)^{-1}\left(\mathbf{M}^{-\frac{1}{2}}\tilde{\mathbf{g}} - \mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\tilde{\mathbf{p}}_h^{i-1}\right),$$

where $\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}$ is symmetric. By the standard theory for the modified Richardson iteration [12], we conclude the optimal tuning parameter (9) and corresponding rate (10). Now, to make the identification $\lambda_{\max}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\,\mathbf{M}^{-\frac{1}{2}}\right) = \frac{\alpha^2}{K_{dr}^{\star}} + \frac{1}{M}$ we consider Rayleigh quotients,

$$\lambda_{\max}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right) = \sup_{\mathbf{p}\neq\mathbf{0}} \frac{\mathbf{p}^{\top}\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\mathbf{p}}{\mathbf{p}^{\top}\mathbf{p}} = \frac{1}{M} + \sup_{\mathbf{p}\neq\mathbf{0},\,\mathbf{Au}=\mathbf{Bp}} \frac{\mathbf{u}^{\top}\mathbf{Au}}{\mathbf{p}^{\top}\mathbf{Mp}}$$
$$= \frac{1}{M} + \sup_{0\neq p_h\in Q_h} \frac{2\mu\,\|\varepsilon\left(\boldsymbol{u}_h\right)\|^2 + \lambda\,\|\nabla\cdot\boldsymbol{u}_h\|^2}{\|p_h\|^2},$$

where $\boldsymbol{u}_h \in \boldsymbol{V}_h$ solves

$$2\mu\,\langle\varepsilon\left(\boldsymbol{u}_h\right), \varepsilon\left(\boldsymbol{v}_h\right)\rangle + \lambda\,\langle\nabla\cdot\boldsymbol{u}_h, \nabla\cdot\boldsymbol{v}_h\rangle = \alpha\,\langle p_h, \nabla\cdot\boldsymbol{v}_h\rangle \quad \text{for all} \quad \boldsymbol{v}_h \in \boldsymbol{V}_h,$$

for given $p_h \in Q_h$. Testing with $\boldsymbol{v}_h = \boldsymbol{u}_h$ we have

$$2\mu\,\|\varepsilon\left(\boldsymbol{u}_h\right)\|^2 + \lambda\,\|\nabla\cdot\boldsymbol{u}_h\|^2 = \alpha\,\langle p_h, \nabla\cdot\boldsymbol{u}_h\rangle \leq \alpha\,\|p_h\|\,\|\nabla\cdot\boldsymbol{u}_h\|$$
$$\leq \tfrac{\alpha}{\sqrt{K_{dr}^{\star}}}\,\|p_h\|\,\sqrt{2\mu\,\|\varepsilon\left(\boldsymbol{u}_h\right)\|^2 + \lambda\,\|\nabla\cdot\boldsymbol{u}_h\|^2}$$

by the Cauchy-Schwarz inequality and the definition of $K_{dr}^{\star}$. This implies that

$$\lambda_{\max}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right) = \frac{1}{M} + \frac{\alpha^2}{K_{dr}^{\star}}.$$

For the identification $\lambda_{\min}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right) = \frac{1}{M} + \frac{\alpha^2}{\beta}$, recognize that $\lambda_{\min}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right) = \frac{1}{M} + \lambda_{\min}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{B}^{\top}\mathbf{A}^{-1}\mathbf{B}\mathbf{M}^{-\frac{1}{2}}\right)$, and consider the algebraic form of Lemma 1,

$$\exists\beta\;\forall\mathbf{p}_h\;\exists\mathbf{u}_h\;:\;\mathbf{u}_h^{\top}\mathbf{A}\,\mathbf{u}_h \leq \beta\mathbf{p}_h^{\top}\mathbf{M}\,\mathbf{p}_h \text{ and } \frac{1}{\alpha}\mathbf{B}^{\top}\mathbf{u}_h = \mathbf{M}\,\mathbf{p}_h. \tag{14}$$

Moreover, recognize that for inf-sup stable discretizations $\mathbf{B}^{\top}\mathbf{A}^{-1}\mathbf{B}$ is invertible. Let $\tilde{\mathbf{u}}_h$ be the minimizer of $\left\{\mathbf{u}_h^{\top}\mathbf{A}\,\mathbf{u}_h \;:\; \frac{1}{\alpha}\mathbf{B}^{\top}\mathbf{u}_h = \mathbf{M}\,\mathbf{p}_h\right\}$. Finding the saddle point using the Lagrangian

$$\mathcal{L}\left(\mathbf{u}_h, \boldsymbol{\Lambda}\right) = \mathbf{u}_h^{\top}\mathbf{A}\,\mathbf{u}_h + \boldsymbol{\Lambda}^{\top}\left(\frac{1}{\alpha}\mathbf{B}^{\top}\mathbf{u}_h - \mathbf{M}\,\mathbf{p}_h\right)$$

we get the solutions $\mathbf{\Lambda} = 2\alpha^2\big(\mathbf{B}^\top\mathbf{A}^{-1}\mathbf{B}\big)^{-1}\mathbf{M}\mathbf{p_h}$ and $\tilde{\mathbf{u}}_h = \frac{1}{2\alpha}\mathbf{A}^{-1}\mathbf{B}\mathbf{\Lambda} = \alpha\mathbf{B}^{-\top}\mathbf{M}\mathbf{p}_h$. Hence, the minimizer satisfies

$$\tilde{\mathbf{u}}_h^\top\mathbf{A}\tilde{\mathbf{u}}_h = \alpha^2\mathbf{p}_h^\top\mathbf{M}\Big(\mathbf{B}^\top\mathbf{A}^{-1}\mathbf{B}\Big)^{-1}\mathbf{M}\mathbf{p}_h.$$

From (14) we get, with $\mathbf{u}_h$ depending on $\mathbf{p}_h$ as above,

$$\beta = \sup_{\mathbf{p}_h\neq\mathbf{0}}\frac{\tilde{\mathbf{u}}_h^\top\mathbf{A}\tilde{\mathbf{u}}_h}{\mathbf{p}_h^\top\mathbf{M}\,\mathbf{p}_h} = \alpha^2\sup_{\mathbf{p}_h\neq\mathbf{0}}\frac{\mathbf{p}_h^\top\mathbf{M}\big(\mathbf{B}^\top\mathbf{A}^{-1}\mathbf{B}\big)^{-1}\mathbf{M}\mathbf{p}_h}{\mathbf{p}_h^\top\mathbf{M}\,\mathbf{p}_h}$$

$$= \alpha^2\sup_{\mathbf{p}_h\neq\mathbf{0}}\frac{\mathbf{p}_h^\top\mathbf{M}^{\frac{1}{2}}\big(\mathbf{B}^\top\mathbf{A}^{-1}\mathbf{B}\big)^{-1}\mathbf{M}^{\frac{1}{2}}\mathbf{p}_h}{\mathbf{p}_h^\top\mathbf{p}_h} = \alpha^2\lambda_{\min}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{B}^\top\mathbf{A}^{-1}\mathbf{B}\mathbf{M}^{-\frac{1}{2}}\right)^{-1}.$$

□

As a result of Theorem 1 we get the optimal stabilization parameter $L_{\text{opt}} = \frac{\alpha^2}{2}\left(\frac{1}{K_{dr}^\star} + \frac{1}{\beta}\right)$, which includes information on the boundary conditions.

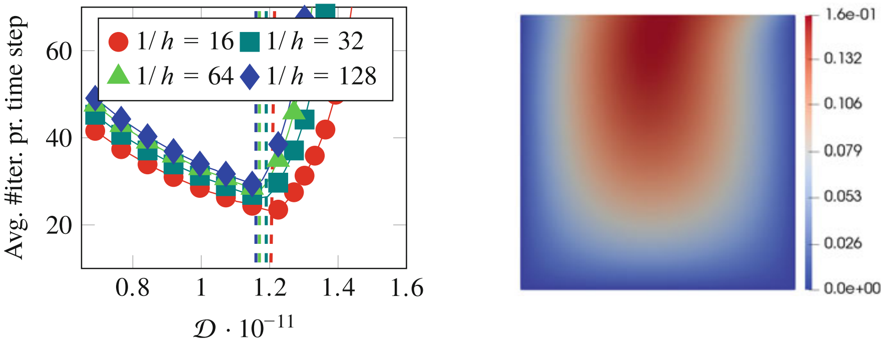
4.1 Computing the Optimal Stabilization Parameter

There are two main options now for finding the optimal stabilization parameter. One could estimate $K_{dr}^\star$ by K_{dr} and find β by some table search of inf-sup constants, but our experience is that this generally is not good enough. Therefore, we suggest to approximate the eigenvalues $\lambda_{\max}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right)$ and $\lambda_{\min}\left(\mathbf{M}^{-\frac{1}{2}}\mathbf{S}\mathbf{M}^{-\frac{1}{2}}\right)$ using the power iteration [12] or some other cheap inexact scheme for finding maximal and minimal eigenvalues. Notice that this is possible to do without explicitly computing $\mathbf{S}$. Also, the computation of $K_{dr}^\star$ is relatively cheap while the computation of β is not—it involves applying the fixed-stress splitting scheme with a non-optimal stabilization parameter. We suggest using coarse approximations of both to define an approximation of L_{opt}. This approximation can be expected to still yield relatively good performance of the fixed-stress splitting scheme.

5 Numerical Examples

We test Theorem 1 numerically including the optimality of the proposed stabilization parameter. For this, we consider a unit square test case with source terms that enforce parabolic displacement and pressure profiles and zero Dirichlet boundary conditions everywhere, but at the top boundary for the momentum balance equation where zero Neumann conditions are considered, see Fig. 1b or [4] for a more thorough explanation. We choose the parameters from Table 1. The choice of the extreme values $M = \infty$ and $\kappa = 0$ mimics the limit scenario of incompressibility and impermeability. In this regime, the coupling strength of the Biot equations as defined by Kim et al. [5] is infinite, and we consider the test a suitable stress test.

For fixed physical parameters, we test the performance of the fixed-stress splitting scheme employing different tuning parameters. For the inf-sup stable P2-P1 discretization, the average number of iterations is presented in Fig. 1a. Notice especially that the proposed optimal stabilization parameter truly is optimal.



(a)

(b)

Fig. 1 Average number of iterations per time step for different stabilization parameters, $L = \frac{\alpha^2}{\mathcal{D}}$, using parameters from Table 1. The dashed lines highlight the location of the tuning parameter, L_{opt}, computed using fine tolerances in the power iteration. (**a**) Numerical results. (**b**) Displacement ($|\boldsymbol{u}_h|$)

Table 1 Table of coefficients

Name	Symbol	Value
Lamé parameters	μ, λ	$41.667 \cdot 10^9, 27.778 \cdot 10^9$
Permeability and compressibility	$\kappa, \frac{1}{M}$	0, 0
Temporal parameters	t_0, τ, T	0, 0.1, 1
Biot-Willis coefficient	α	1
Relative error tolerance	ϵ_r	10^{-6}
Inverse of mesh size diameter	$1/h$	16, 32, 64, 128

6 Conclusions

In this work, we derived mathematically a way to *a priori* determine the optimal stabilization parameter for the fixed-stress splitting scheme using the theory for the modified Richardson iteration. This optimal parameter involves the computations of maximal eigenvalues for rather large matrices. However, they do not need to be computed to high accuracy. Through a numerical experiment, we showed that the proposed stabilization parameter is optimal for this problem.

References

1. A. Settari and F. M. Mourits. A Coupled Reservoir and Geomechanical Simulation System. *Soc Petrol Eng J*, 3:219–226, 1998.
2. A. Mikelić and M. F. Wheeler. Convergence of iterative coupling for coupled flow and geomechanics. *Computat Geosci*, 17(3):455–461, 2013.
3. J. W. Both, M. Borregales, K. Kumar, J. M. Nordbotten, and F. A. Radu. Robust fixed stress splitting for Biot's equations in heterogeneous media. *Appl Math Lett*, 68:101–108, 2017.
4. E. Storvik, J. W. Both, K. Kumar, J. M. Nordbotten, and F. A. Radu. On the optimization of the fixed-stress splitting for Biot's equations. *Int J Numer Meth Eng*, 120(2):179–194, 2019.
5. J. Kim, H. A. Tchelepi, and R. Juanes. Stability and convergence of sequential methods for coupled flow and geomechanics: Fixed-stress and fixed-strain splits. *Comput Method Appl M*, 200(13):1591–1606, 2011.
6. J. W. Both, K. Kumar, J. M. Nordbotten, and F. A. Radu. The gradient flow structures of thermo-poro-visco-elastic processes in porous media, 2019. arXiv:1907.03134 [math.NA].
7. M. Bause, F. A. Radu, and U. Köcher. Space-time finite element approximation of the Biot poroelasticity system with iterative coupling. *Comput Method Appl M*, 320:745–768, 2017.
8. F. J. Gaspar and C. Rodrigo. On the fixed-stress split scheme as smoother in multigrid methods for coupling flow and geomechanics. *Comput Method Appl M*, 326:526–540, 2017.
9. J. A. White, N. Castelletto, and H. A. Tchelepi. Block-partitioned solvers for coupled poromechanics: A unified framework. *Comput Method Appl M*, 303:55–74, 2016.
10. O. Coussy. *Poromechanics*. John Wiley & Sonş 2004.
11. J. Kim, H. A. Tchelepi, and R. Juanes. Stability and convergence of sequential methods for coupled flow and geomechanics: Drained and undrained splits. *Comput Method Appl M*, 200(23):2094–2116, 2011.
12. Y. Saad. *Iterative methods for sparse linear systems*, volume 82. SIAM, 2003.

Modeling and Simulation of Bed Dynamics in Oxyfuel Fluidized Bed Boilers

M. Beneš, P. Eichler, J. Klinkovský, M. Kolář, J. Solovský, P. Strachota, and A. Žák

Abstract This contribution focuses on CFD modeling of the dynamics of the bubbling fluidized bed under conditions specific for oxyfuel combustion. A custom OpenFOAM solver is developed based on the Multiphase Particle-In-Cell framework for handling the fluid-particle and inter-particle interactions. Features of this Euler-Lagrange approach are discussed, and some of the solver design details are given. Some simulation results of a laboratory-scale combustion device or its combustion chamber are demonstrated, showing the capabilities of the solver. The current limitations and plans for further development are also included.

1 Introduction

Oxyfuel combustion [14] is a promising technology allowing efficient CO_2 capture and storage [6], leading to environment friendly energy production. In addition, when biomass is used as the primary fuel, negative carbon footprint can be achieved. Currently, this technology is not ready for large-scale industrial application in power plants. Intense research and development is under way in the area of design, tuning, control, and mathematical modeling of both laboratory and pilot devices.

This paper presents the state of the development of a complex CFD algorithm for simulating oxyfuel combustion conditions in modern fluidized bed boilers [2]. Oxyfuel combustion employs an oxidizing atmosphere consisting of pure oxygen mixed with recirculating flue gases, so that nitrogen is largely eliminated from the gas, whereas CO_2 becomes its prominent component. In the combustion chamber, fuel particles and an additional granular material such as limestone take part in

M. Beneš · P. Eichler · J. Klinkovský · M. Kolář · J. Solovský · P. Strachota (✉) · A. Žák
Department of Mathematics, FNSPE CTU in Prague, Praha 2, Czech Republic
e-mail: michal.benes@fjfi.cvut.cz; eichlpa1@fjfi.cvut.cz; klinkjak@fjfi.cvut.cz; miroslav.kolar@fjfi.cvut.cz; jakub.solovsky@fjfi.cvut.cz; pavel.strachota@fjfi.cvut.cz; alexandr.zak@fjfi.cvut.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_91

multiphase flow, combustion, heat transfer and other processes [13] while being maintained in the fluidized state by a sufficiently strong gas flow.

An analysis of several models and software packages applied to the multiphase-flow problem without combustion has been performed in our previous paper [3]. Based on these findings, the OpenFOAM CFD toolbox was chosen as the basis for the numerical solver. OpenFOAM provides a library of numerical solvers for various coupled physical phenomena. Its object-oriented design opens wide possibilities for combining and extending its features in order to obtain a solver suited to our specific scenario.

The developed algorithm builds upon the `coalChemistryFoam` solver, which readily employs the Eulerian finite volume approach for gas flow simulation together with Lagrangian tracking of discrete parcels of particles. In addition, the framework for chemical reactions in combustion and heat transfer is already present there with the possibility to incorporate additional phenomena (devolatilization, biomass particle disintegration). The effort to simulate multiphase flow in both circulating and bubbling fluidized bed (CFB, BFB) [3] required heavy modifications. The most important of them is the transition to the Multiphase Particle-in-Cell (MP-PIC) method [1, 7, 9, 10] described in Sect. 2. The resulting solver has been named `kotelFoam`, *kotel* being the Czech term for *boiler*.

In this paper, we present `kotelFoam` with features limited to the simulation of multiphase flow of the oxygen/flue gas mixture and limestone particles. Additional phenomena available in the original `coalChemistryFoam` are disabled. Further details on the performed steps are laid out in Sect. 3 and the solver properties together with some simple results are discussed in Sect. 4. Preliminary application in the complex geometry of a laboratory-scale BFB boiler is shown in Sect. 5.

2 MP-PIC Multiphase Flow Model

In a bubbling fluidized bed, the granular material occupies a significant fraction of the lower part of the combustion chamber. In this situation, particle collisions strongly account for the dynamics of the solid phase and cannot be neglected. In addition, it must be ensured that the particle volume fraction at close packing never exceeds a maximum value known as the packing limit [1, 7]. Finally, the fact that the domain volume is partly occupied by the particles influences the gas flow. The above phenomena were not treated in `coalChemistryFoam` at all. However, it is possible to handle them in an efficient way by means of the MP-PIC method, which is described below.

The MP-PIC method treats the fluid phase as continuum within the Eulerian reference frame and the solid phase as particle parcels within the Lagrangian reference frame. The dynamics of the fluid and solid phase is modeled using a fixed Eulerian grid and Lagrangian parcels, respectively. The inter-phase interaction is arranged using interpolation operators [9]. The interaction between particles is carried out on the Eulerian grid using the averaged quantities without the need to calculate the individual collisions.

The description of the solid phase is based on determination of particle distribution function $\phi(\mathbf{x}, m_p, \mathbf{v}_p, t)$, where $\mathbf{x}$ is the position of the particle, m_p is the mass of the particle, $\mathbf{v}_p$ is the particle velocity, and t is the time coordinate. The time evolution of the particle distribution function is governed by the Liouville equation [9, 10], which is dependent on the particle acceleration given by Andrews and O'Rourke [1]

$$\frac{\mathrm{d}\mathbf{v}_p}{\mathrm{d}t} = D_p \left(\mathbf{v}_f - \mathbf{v}_p\right) - \frac{1}{\rho_p}\nabla p + \mathbf{g} - \frac{1}{\alpha_p \rho_p}\nabla\tau \,. \tag{1}$$

In the equation above, D_p is the drag coefficient given (in our case) by the Gidaspow–Ergun–Wen&Yu model [4, 5], $\mathbf{v}_f$ is the fluid velocity, ρ_p is the particle density, p is the fluid pressure, $\mathbf{g}$ is the gravitational acceleration, α_p is the particle volume fraction, and τ is the inter-particle stress. The particle volume fraction is given by integrating over particle mass and velocity as

$$\alpha_p = \int \phi \frac{m_p}{\rho_p} \, \mathrm{d}m_p \, \mathrm{d}\mathbf{v}_p \,, \tag{2}$$

and the inter-particle stress τ is given by the modified Harris and Crighton model [9]

$$\tau = \frac{P_p \alpha_p^{\beta}}{\max\left[(\alpha_{CP} - \alpha_p) \,, \; \varepsilon(1 - \alpha_p)\right]} \,, \tag{3}$$

where α_{CP} is the particle volume fraction at close packing and P_p, β, and ε are constants.

Instead of directly solving the Liouville equation, the MP-PIC method relies on tracking of particles clustered into computational parcels using Lagrangian equations of motion and applying formula (1).

The dynamics of the fluid phase is governed by the mass and momentum balance equations which are written as

$$\frac{\partial(\alpha_f \rho_f)}{\partial t} + \nabla \cdot (\alpha_f \rho_f \mathbf{v}_f) = 0, \tag{4}$$

$$\frac{\partial(\alpha_f \rho_f \mathbf{v}_f)}{\partial t} + \nabla \cdot (\alpha_f \rho_f \mathbf{v}_f \otimes \mathbf{v}_f) = -\nabla p + \nabla \cdot (\alpha_f \mathbb{T}_f) + \alpha_f \rho_f \mathbf{g} + \mathbf{F} \,, \tag{5}$$

where α_f is the fluid volume fraction, ρ_f is the fluid density, $\mathbf{v}_f$ is the fluid velocity, $\mathbb{T}_f$ is the fluid stress tensor, and $\mathbf{F}$ is the interphase momentum transfer function. The interphase momentum transfer is given by integrating over particle mass and velocity coordinates as

$$\mathbf{F} = \int \phi m_p \left[D_p \left(\mathbf{v}_f - \mathbf{v}_p \right) - \frac{1}{\rho_p} \nabla p \right] \mathrm{d}m_p \, \mathrm{d}\mathbf{v}_p \, . \tag{6}$$

The fluid is treated as a Newtonian fluid with stress tensor

$$\mathbb{T}_f = \mu_f \left(\nabla \mathbf{v}_f + \left(\nabla \mathbf{v}_f \right)^T - \frac{2}{3} \nabla \cdot \mathbf{v}_f \mathbb{I} \right) , \tag{7}$$

where μ_f is the fluid viscosity, and the ideal gas equation of state is used to relate its pressure and density.

3 Solver Details

In the following, we discuss some important details on the `kotelFoam` solver implementation and setup within the OpenFOAM framework.

- As a derivative of `coalChemistryFoam`, the solver is ready for heat transfer computations within each phase and between phases. However, as far as multiphase flow alone is concerned, it is also possible to run isothermal flow simulations only. Energy balance equations are therefore not discussed in this paper.
- Currently, MP-PIC is implemented for limestone particles by means of `basicThermoMPPICCloud`. In contrast to that, fuel particles rely on the original `coalCloud` which is derived from `kinematicCloud` and does not implement the phenomena described at the beginning of Sect. 2. However, fuel particles are sparse in comparison to limestone and their collisions are currently neglected.
- Changes in the governing equations for the gas phase were made according to (4) and (5) so that α_f is taken into account. Calculation of α_f also allows to use any of the drag models available in the Eulerian multiphase solvers (e.g. `twoPhaseEulerFoam`).
- Gas phase turbulence models have been amended to include the effect of α_f.
- Restrictions for the time step are computed separately based on both $\mathbf{v}_f$, $\mathbf{v}_p$ and the more restrictive bound is chosen. Including $\mathbf{v}_p$ in the computation turned out to be vital for the stability of the simulation.
- Sutherland's law for gas viscosity has been correctly implemented for the individual chemical components to allow for changes in the chemical composition of the gas mixture (due to combustion in oxyfuel conditions). The respective coefficients were taken from [12].

4 Solver Properties and Computational Results

The development of `kotelFoam` is a work in progress. Here we present our current knowledge and experience that testifies to both the benefits and issues of the solver.

`kotelFoam` is used for simulations in a reference domain representing a 2 m tall cylindrical combustion chamber. The inlet is located at its bottom base with a diameter of 102 mm. The chamber is widened in its upper part and the outlet at the top has a diameter of 154 mm. The same geometry was used in our previous experiments in [3] and represents a real experimental device described in [8].

The composition of gas used in the simulations is either that of air [3] or 25% of O_2 and 75% of CO_2, which closely resembles the real gas mixture during oxyfuel combustion. A constant temperature $T = 1000$ K is prescribed. The properties of solid particles are the same as in [3] so that direct comparison with the results therein is possible. Gas turbulence is modeled by the $k - \varepsilon$ RAS model.

In the first test scenario, we verify the MP-PIC approach of `kotelFoam` against `twoPhaseEulerFoam` and `coalChamistryFoam` solvers under circulating fluidization conditions where the particle volume fraction is low. Mass inflow rates of solids and air are 0.15 kg s^{-1} and 0.07275 kg s^{-1}, respectively. The resulting comparison at time $t = 5$ s is shown in Fig. 1 by means of the particle volume fraction. At this time, the flow pattern is de facto steady as the particle inflow and outflow are equal. The results indicate that for this situation, all solvers perform equally well with a particularly tight agreement between the two Lagrangian models.

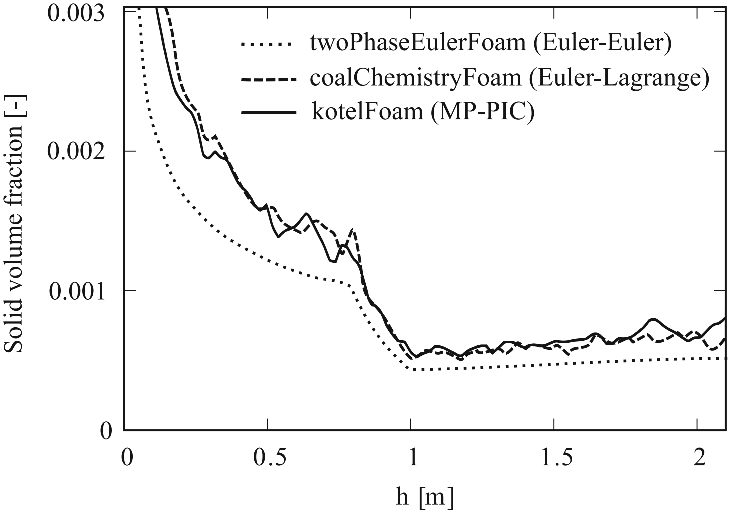


Fig. 1 Cross section-averaged particle volume fraction along the vertical axis of the combustion chamber during circulating fluidization. Comparison of results obtained by different solvers

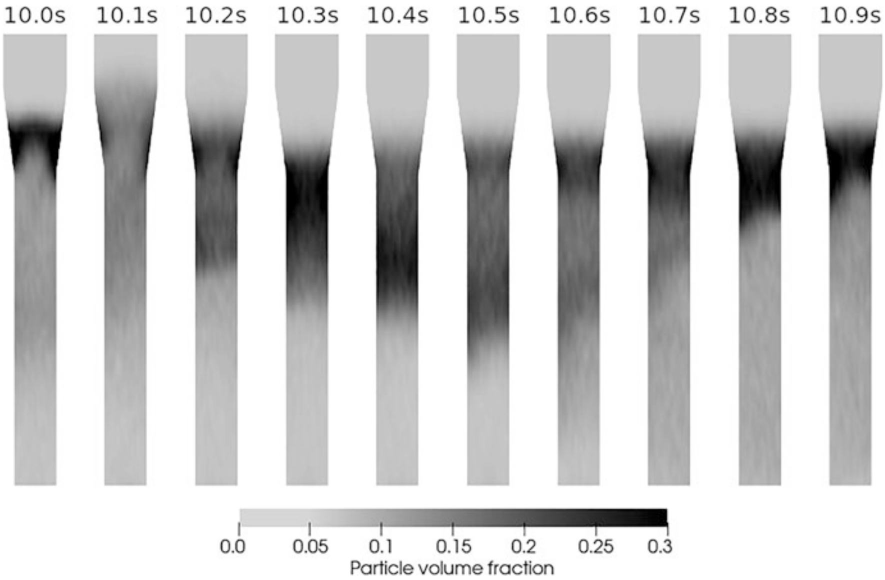


Fig. 2 Evolution of the particle volume fraction in the lower part of the combustion chamber under the bubbling regime conditions. Slices through the center of the 3D domain (details in [3]) are displayed

Next, we focus on simulation of the bubbling regime. In Fig. 2, we present such behavior in the combustion chamber. As in [3], the mass inflow rates of oxyfuel gas mixture and solid particles are 0.02912 kg s^{-1} and 0.15 kg s^{-1}, respectively. Since the fluid velocity decreases in the widening part of the chamber, the drag force fades there. Hence, the particles are no more carried up to the outlet, remain inside the chamber, and form a bed in its narrower part. After the initial 10 s of solids injection, a bed with the total mass of 1.5 kg is obtained. The result demonstrates the formation of bubbles from below and also testifies that the solver is able to work with particle volume fractions common in the bubbling regime.

Other tests in the simple domain revealed the following observations and issues:

- Lagrangian models (naturally) precisely satisfy the conservation of particle mass. On the other hand, `twoPhaseEulerFoam` with some turbulence models for solid phase (e.g. the `phasePressure` model) yield completely nonphysical results with the particle volume fraction far exceeding the packing limit and solid mass going to infinity.
- Results and stability of the solver relies on the settings of some submodels. In particular, the explicit or implicit packing models yield qualitatively different fluidization patterns and the implicit model appears to be more computationally stable.

- The parallel implementation of MP-PIC gives results dependent on domain decomposition and numerical stability issues arise in some cases with parallel computations.

5 Flow in Complex Geometry of the Experimental BFB Boiler

Figure 3 demonstrates a preliminary simulation of gas flow though the whole duct system of the experimental BFB boiler operating at the laboratory of the Czech Technical University in Prague. The main features that need to be simulated are the cyclone for solid particle separation and the recirculation system that uses forced recirculation to inject flue gas back to the combustion chamber. Inspired by Svenning [11], the model of the recirculation fan is based on the addition of external forcing term to the momentum equation in the volume where the fan rotor is located. The axial $\mathbf{f}_x$ and tangential $\mathbf{f}_\theta$ force is computed using formulae given in [11]. The

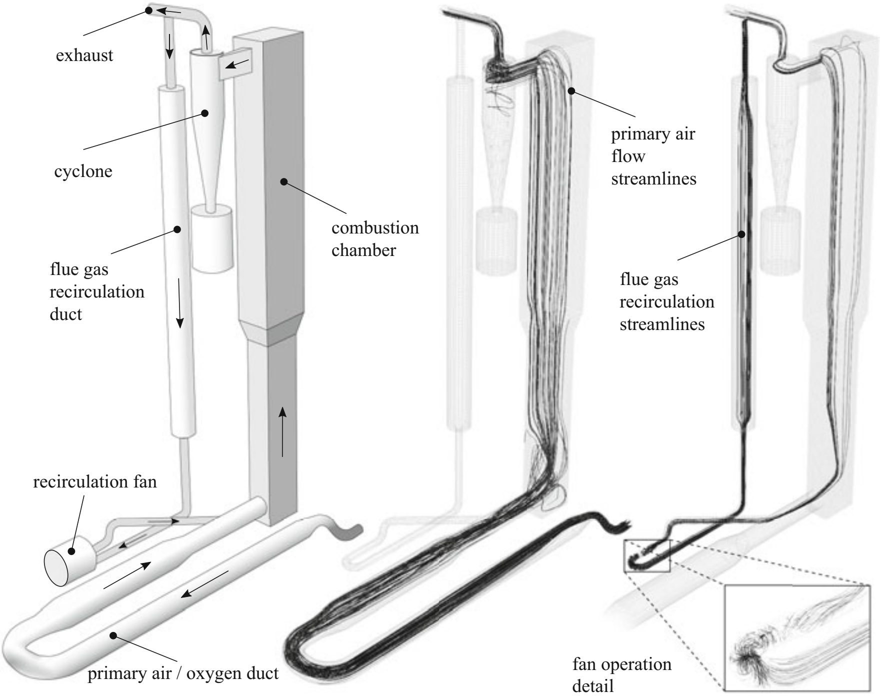

exhaust
cyclone
flue gas recirculation duct
combustion chamber
recirculation fan
primary air / oxygen duct
primary air flow streamlines
flue gas recirculation streamlines
fan operation detail

Fig. 3 Schematic drawing of the duct system of the experimental BFB boiler. Arrows indicate the direction of flow during operation. Streamlines demonstrate the results of the `kotelFoam` simulation. The direction of flow is correct

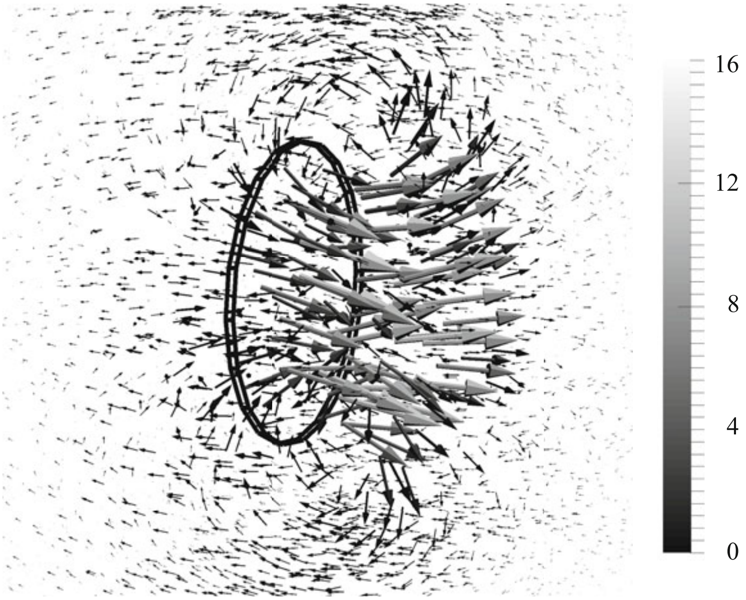


Fig. 4 Visualization of the fluid flow through the fan disk in a large surrounding domain. Gas velocity magnitude is represented by the gray scale as well as the arrow size

inputs to these formulae are the total thrust T and total torque Q. The detail of fluid flow through the fan disk in a large surrounding domain is illustrated in Fig. 4.

6 Conclusion and Further Work

The presented `kotelFoam` solver combines features from several parts of OpenFOAM together with other custom improvements to be able to simulate the bubbling fluidized bed in the combustion chamber and gas/particle flows in the accompanying piping systems of fluidized bed boilers. Modifications are aimed at both air and oxyfuel regimes of operation. In the current state, `kotelFoam` is already capable of handling situations untractable by the solvers readily available in OpenFOAM. However, further research is needed to tackle the numerical stability issues in the parallel implementation of the MP-PIC method. Then, the other important phenomena occurring in the reactor, such as disintegration of non-spherical fuel particles, devolatilization, combustion, and heat transfer are to be added, again combining the available features of `coalChemistryFoam` and custom code.

Acknowledgments This work was supported by the Ministry of Education, Youth and Sports of the Czech Republic under the OP RDE grant number CZ.02.1.01/0.0 /0.0/16_019/0000753 “Research centre for low-carbon energy technologies”.

References

1. Michael J Andrews and Peter J O'Rourke. The multiphase particle-in-cell (MP-PIC) method for dense particulate flows. *Int. J. Multiphase Flow*, 22(2):379–402, 1996.
2. Prabir Basu. *Combustion and Gasification in Fluidized Beds*. CRC Press, 2006.
3. Michal Beneš, Pavel Eichler, Jakub Klinkovský, Miroslav Kolář, Jakub Solovský, Pavel Strachota, and Alexandr Žák. Numerical simulation of fluidization for application in oxyfuel combustion. *Discrete. Cont. Dyn. S. S*, https://doi.org/10.3934/dcdss.2020232:1–15, 2019. Online First.
4. Dimitri Gidaspow. *Multiphase Flow and Fluidization: Continuum and Kinetic Theory Description*. Academic Press, 1994.
5. Yefei Liu and Olaf Hinrichsen. CFD modeling of bubbling fluidized beds using OpenFOAM® : Model validation and comparison of TVD differencing schemes. *Computers and Chemical Engineering*, 69:75–88, 2014.
6. M. Mercedes Maroto-Valer, editor. *Developments and innovation in carbon dioxide* (CO_2) *capture and storage technology*. Woodhead Publishing, 2010.
7. Peter J O'Rourke, Paul Pinghua Zhao, and Dale M Snider. A model for collisional exchange in gas/liquid/solid fluidized beds. *Chem. Eng. Sci.*, 64(8):1784–1797, 2009.
8. Farooq Sher, Miguel A. Pans, Chenggong Sun, Colin Snape, and Hao Liu. Oxy-fuel combustion study of biomass fuels in a 20 kw fluidized bed combustor. *Fuel*, 215:778–786, 2018.
9. Dale M Snider. An incompressible three-dimensional multiphase particle-in-cell model for dense particle flows. *J. Comput. Phys.*, 170(2):523–549, 2001.
10. Dale M Snider and Peter J O'Rourke. The multiphase particle-in-cell (MP-PIC) method for dense particle flow. In *Computational Gas-Solids Flows and Reacting Systems: Theory, Methods and Practice*, pages 277–314. IGI Global, 2011.
11. Erik Svenning. Implementation of an actuator disk in OpenFOAM. *Bachelor thesis (Chalmers University of Technology, Sweden, 2010)*, 2010.
12. Zhongchao Tan. *Air pollution and greenhouse gases: from basic concepts to engineering applications for air emission control*. Springer, 2014.
13. Wen-Ching Yang, editor. *Handbook of Fluidization and Fluid-Particle Systems*. Marcel Dekker, 2003.
14. Ligang Zheng, editor. *Oxy-fuel combustion for power generation and carbon dioxide* (CO_2) *capture*. Woodhead Publishing, 2011.

Monotonicity Considerations for Stabilized DG Cut Cell Schemes for the Unsteady Advection Equation

Florian Streitbürger, Christian Engwer, Sandra May, and Andreas Nüßing

Abstract For solving unsteady hyperbolic conservation laws on cut cell meshes, the so called *small cell problem* is a big issue: one would like to use a time step that is chosen with respect to the background mesh and use the same time step on the potentially arbitrarily small cut cells as well. For explicit time stepping schemes this leads to instabilities. In a recent preprint [arXiv:1906.05642], we propose penalty terms for stabilizing a DG space discretization to overcome this issue for the unsteady linear advection equation. The usage of the proposed stabilization terms results in stable schemes of first and second order in one and two space dimensions. In one dimension, for piecewise constant data in space and explicit Euler in time, the stabilized scheme can even be shown to be monotone. In this contribution, we will examine the conditions for monotonicity in more detail.

1 A Stabilized DG Cut Cell Scheme for the Unsteady Advection Equation

We consider the time dependent linear advection problem on a cut cell mesh. In [1], we propose new stabilization terms for a cut cell discontinuous Galerkin (DG) discretization in two dimensions with piecewise linear polynomials. In the following we will refer to this as *Domain-of-Dependence stabilization*, abbreviated by DoD stabilization.

F. Streitbürger (✉)
TU Dortmund University, Dortmund, Germany
e-mail: florian.streitbuerger@math.tu-dortmund.de

C. Engwer · A. Nüßing
University of Münster, Münster, Germany
e-mail: christian.engwer@uni-muenster.de; andreas.nuessing@uni-muenster.de

S. May
Technical University of Munich, Garching bei München, Germany
e-mail: sandra.may@ma.tum.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_92

While the usage of finite element schemes on embedded boundary or cut cell meshes has become increasingly popular for elliptic and parabolic problems in recent years, only very little work has been done for hyperbolic problems. The general challenge is that cut cells can have various shapes and may in particular become arbitrarily small. Special schemes have been developed to guarantee stability. Perhaps the most prominent approach for elliptic and parabolic problems is the ghost penalty stabilization [2], which regains coercivity, independent of the cut size.

For hyperbolic conservation laws the problems caused by cut cells are partially of different nature. One major challenge is that standard explicit schemes are not stable on the arbitrarily small cut cells when the time step is chosen according to the cell size of the background mesh. This is what is often called the *small cell problem*. Adapting the time step size to the cut size is infeasible, as there is no lower bound on the cut size. An additional complication is the fact that there is typically no concept of coercivity that could serve as a guideline for constructing stabilization terms.

In [1], we consider the small cell problem for the unsteady linear advection equation. We propose a stabilization of the spatial discretization, which uses a standard DG scheme with upwind flux, that makes explicit time stepping stable again. Our penalty terms are designed to restore the correct domains of dependence of the cut cells and their outflow neighbors (therefore the name DoD stabilization), similar to the idea behind the h-box scheme [4] but realized in a DG setting using penalty terms. In one dimension, we can prove L^1-stability, monotonicity, and TVD (total variation diminishing) stability for the stabilized scheme of first order using explicit Euler in time. For the second-order scheme, we can show a TVDM (TVD in the means) result if a suitable limiter is used.

In this contribution, we will focus on the monotonicity properties in one dimension for the first-order scheme and examine them in more detail. In particular, we will show that a straight-forward adaption of the ghost penalty approach [2] to the unsteady transport problem, as proposed in [3] for the steady problem, cannot ensure monotonicity. Further, we will examine the parameter that we use in our new DoD stabilization in more detail than done in [1].

2 Problem Setup for Piecewise Constant Polynomials

For the purpose of a theoretical analysis with focus on monotonicity, we will consider piecewise constant polynomials in 1D. We use the interval $I = [0, 1]$ and assume the velocity $\beta > 0$ to be constant. The time dependent linear advection equation reads

$$u_t(x,t) + \beta u_x(x,t) = 0 \text{ in } I \times (0,T), \qquad (1)$$

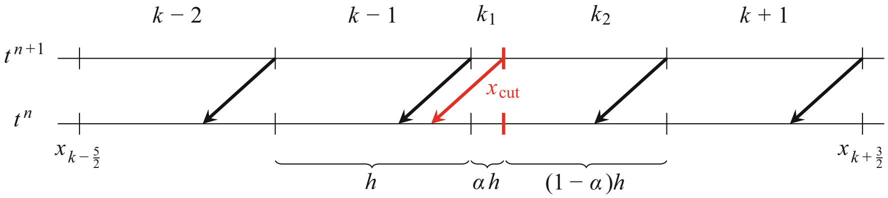


Fig. 1 Domains of dependence for the solution at time t^{n+1} for the considered model problem for a time step with length $\Delta t = \frac{\lambda}{\beta} h$ with $\lambda = \frac{1}{3}$ and $\beta = 1$

with initial data $u(x, 0) = u_0(x)$ and periodic boundary conditions. We discretize the interval I in N equidistant cells $I_j = [x_{j-\frac{1}{2}}, x_{j+\frac{1}{2}}]$ with cell length h. Then, we split one cell, the cell k, into a pair of two cut cells using the volume fraction $\alpha \in (0, \frac{1}{2}]$, see Fig. 1: The first cut cell, which we call k_1, has length αh, the second cut cell, which we call k_2, has length $(1-\alpha)h$. Therefore, cell k_1 corresponds to a small cut cell, which induces instabilities, if $\alpha \ll \frac{1}{2}$.

We define the function space

$$V_h^0(I) := \left\{ v_h \in L^2(I) \,\middle|\, v_h|_{I_j} \in \mathcal{P}^0,\ j = 1, \dots, N \right\}, \tag{2}$$

with $\mathcal{P}^0$ being the function space of constant polynomials. The semidiscrete scheme, which uses the standard DG scheme with an upwind flux in space and is not yet discretized in time, is given by: Find $u_h \in V_h^0(I)$ such that

$$\int_I d_t u_h(t)\, w_h\, dx + a_h^{\text{upw}}(u_h(t), w_h) = 0, \quad \forall w_h \in V_h^0(I), \tag{3}$$

with the bilinear form defined as

$$a_h^{\text{upw}}(u_h, w_h) = \sum_{j=1}^{N} \beta u_h(x_{j+\frac{1}{2}}^-) [\![w_h]\!]_{j+\frac{1}{2}} + \beta u_h(x_{\text{cut}}^-) [\![w_h]\!]_{\text{cut}},$$

and the jump being given by

$$[\![w_h]\!]_{j+\frac{1}{2}} = w_h(x_{j+\frac{1}{2}}^-) - w_h(x_{j+\frac{1}{2}}^+), \quad x_{j+\frac{1}{2}}^{\pm} = \lim_{\varepsilon \to 0^+} x_{j+\frac{1}{2}} \pm \varepsilon.$$

We use explicit Euler in time. Then, (3) results in the global system

$$\mathcal{M} u^{n+1} = \mathcal{B} u^n. \tag{4}$$

Here, $u^n = [u_1^n, \ldots, u_N^n]^T$ is the vector of the piecewise constant solution at time t^n and $\mathcal{M}$ is the global mass matrix. Note that $\mathcal{M}$ is a diagonal matrix with positive diagonal entries. Finally, the global system matrix $\mathcal{B}$ is given by $\mathcal{B} = \mathcal{M} - \Delta t \mathcal{A}$ with $\mathcal{A}u^n$ corresponding to the discretization of the bilinear form a_h^{upw} at time t^n.

For a standard equidistant mesh, the scheme (4) is stable for $0 < \lambda < 1$ with the CFL number λ being given by

$$\lambda = \frac{\beta \Delta t}{h}. \tag{5}$$

Our goal is to make the scheme stable for the mesh containing the cut cell pair for $0 < \lambda < \frac{1}{2}$, independent of α. The reduced CFL condition is due to the fact that we will only stabilize cut cell k_1, and not the bigger cut cell k_2.

To illustrate one interpretation of the small cell problem that we need to overcome, we refer to Fig. 1. There, we determine the exact solution at time t^{n+1}, based on piecewise constant data at time t^n, by tracing back characteristics. For standard cells j, such as $k-1$, the domain of dependence of u_j^{n+1} only includes cells j and $j-1$. For the outflow neighbor of the small cut cell k_1, the cell k_2, however, $u_{k_2}^{n+1}$ depends on u_{k-1}^n, $u_{k_1}^n$, and $u_{k_2}^n$. The issue is that standard DG schemes such as (3) only provide information from direct neighbors. We will see that the proposed stabilization that ensures monotonicity will also fix this problem. We will return to this specific interpretation of the small cell problem in Sect. 4, when discussing the proper choice of the penalty parameter in the stabilization.

3 Monotonicity Considerations for Two Different Stabilization Terms

In the following, we will examine the monotonicity properties of different stabilizations. A monotone scheme guarantees in particular that $\min_j u_j^0 \leq u^n \leq \max_j u_j^0$ for all times t^n. We will use the following definition of a monotone scheme.

Definition 1 A method $u_j^{n+1} = H(u_{j-i_L}^n, u_{j-i_L+1}^n, \ldots, u_{j+i_R}^n)$ is called *monotone*, if for all j there holds for every l with $-i_L \leq l \leq i_R$

$$\frac{\partial H}{\partial u_{j+l}}(u_{j-i_L}, \ldots, u_{j+i_R}) \geq 0. \tag{6}$$

For the linear scheme (4) this implies that all coefficients of $\mathcal{B}$ need to be non-negative. This is due to the fact that $\mathcal{M}$ is a diagonal matrix with strictly positive entries. On an equidistant mesh, the scheme (4) is monotone for $0 < \lambda < 1$.

We will compare the entries of the matrix $\mathcal{B}$ for three different cases: the unstabilized case, a stabilization in the spirit of the ghost penalty method [2], and the DoD stabilization [1] that we propose.

3.1 Unstabilized Case

In this case, the matrix $\mathcal{B}$ is given by

$$\mathcal{B} = \begin{pmatrix} h-\tau & 0 & \cdots & & & & \cdots & 0 & \tau \\ \tau & h-\tau & 0 & & & & & & 0 \\ 0 & \ddots & \ddots & & & & & & \vdots \\ \vdots & & \tau & \boxed{\alpha h - \tau} & & & & & \\ & & & \tau & \boxed{(1-\alpha)h - \tau} & & & & \vdots \\ \vdots & & & & \ddots & & \ddots & & 0 \\ 0 & \cdots & & & \cdots & & 0 & \tau & h-\tau \end{pmatrix},$$

with $\tau := \beta \Delta t > 0$. We therefore focus on the diagonal entries. On standard cells, and on cell k_2, the entries $h - \tau$ and $(1-\alpha)h - \tau$ are positive due to the CFL condition $\beta \Delta t = \lambda h$ if the reduced CFL condition $0 < \lambda < \frac{1}{2}$ is used. On the small cut cell k_1, the entry $\alpha h - \tau$ is clearly negative for $\alpha < \lambda$, which is the case that we are interested in.

3.2 Ghost Penalty Stabilization

We first consider the option of using the ghost penalty method for stabilization, an approach that is, e.g., used in [3] for stabilizing the steady advection equation. Adapting the stabilization to our model mesh (compare Fig. 1) changes the formulation of (3) to: Find $u_h \in V_h^0(I)$ such that

$$\int_I d_t u_h(t)\, w_h\, dx + a_h^{\text{upw}}(u_h(t), w_h) + J_h^{\text{GP}}(u_h, w_h) = 0, \quad \forall w_h \in V_h^0(I), \tag{7}$$

with

$$J_h^{\text{GP}} = \beta \eta_1 [\![u_h]\!]_{k-\frac{1}{2}} [\![w_h]\!]_{k-\frac{1}{2}} + \beta \eta_2 [\![u_h]\!]_{\text{cut}} [\![w_h]\!]_{\text{cut}} . \tag{8}$$

As a result, the matrix $\mathcal{B}$ in (4) is modified in the following way

$$\mathcal{B}_{\mathrm{GP}} = \begin{pmatrix} h-\tau & 0 & \cdots & & & & \cdots & 0 & \tau \\ \tau & h-\tau & 0 & & & & & & 0 \\ 0 & \ddots & \ddots \ \ddots & & & & & & \vdots \\ & & \tau & \boxed{h-\tau(1+\eta_1)} & \tau\eta_1 & & & & \\ \vdots & & 0 & \tau(1+\eta_1) & \boxed{\alpha h-\tau-\tau\eta_1-\tau\eta_2} & \tau\eta_2 & & & \\ & & & 0 & \tau(1+\eta_2) & \boxed{(1-\alpha)h-\tau(1+\eta_2)} & & & \vdots \\ \vdots & & & & & \ddots & \ddots & & 0 \\ 0 & \cdots & & & & \cdots & 0 & \tau & h-\tau \end{pmatrix}.$$

Our goal is to determine the parameters η_1 and η_2 such that every entry of $\mathcal{B}_{\mathrm{GP}}$ is non-negative. The two entries on the first superdiagonal prescribe the restriction

$$\eta_1 \geq 0 \quad \text{and} \quad \eta_2 \geq 0. \tag{9}$$

Next, we consider the entry $\mathcal{B}_{\mathrm{GP}}(k_1, k_1)$.This results in the condition

$$\alpha h - \tau - \tau\eta_1 - \tau\eta_2 \overset{!}{\geq} 0.$$

Since $\alpha h - \tau$ is negative for $\alpha < \lambda$, we need to choose η_1 or η_2 to be negative. This is a contradiction to (9). Therefore, it is *not* possible to create a monotone scheme using this setup.

3.3 Domain-of-Dependence Stabilization

We now consider the DoD stabilization, which we introduced in [1]. The resulting scheme is of the same form as (7), but instead of adding J_h^{GP} we use the term

$$J_h^{\mathrm{DoD}}(u_h, w_h) := \beta\eta \, [\![u_h]\!]_{k-\frac{1}{2}} \, [\![w_h]\!]_{\mathrm{cut}} \,. \tag{10}$$

One big difference between (8) and (10) is that the locations of the jump terms were moved. As a result, the position of the stabilization terms in the matrix $\mathcal{B}$ changed:

$$\mathcal{B}_{\text{DoD}} = \begin{pmatrix} h-\tau & 0 & \cdots & & & \cdots\ 0 & \tau \\ \tau & h-\tau & 0 & & & & 0 \\ 0 & \ddots & \ddots & & & & \vdots \\ \vdots & & \tau(1-\eta) & \boxed{\alpha h - \tau\,(1-\eta)} & 0 & & \\ & & \tau\eta & \tau(1-\eta) & \boxed{(1-\alpha)h - \tau} & & \vdots \\ \vdots & & & & \ddots & \ddots & 0 \\ 0 & \cdots & & & \cdots & 0\ \ \tau & h-\tau \end{pmatrix}.$$

In [1], we examined the monotonicity conditions of the stabilized scheme for the theta-scheme in time and a fixed value of η. Here, we will focus on using explicit Euler in time and vary η instead. Requiring that all entries become non-negative results in the following three inequalities:

$$\begin{array}{cll} \textbf{I} & \alpha h - \tau(1-\eta) & \geq 0, \\ \textbf{II} & \tau\eta & \geq 0, \\ \textbf{III} & \tau(1-\eta) & \geq 0. \end{array}$$

Short calculations show that this implies the following restrictions on η

$$\eta \overset{\textbf{II}}{\geq} 0, \quad 1 - \frac{\alpha}{\lambda} \overset{\textbf{I}}{\leq} \eta \overset{\textbf{III}}{\leq} 1, \quad \text{i.e., we need to choose} \quad \eta \in \left[1 - \frac{\alpha}{\lambda}, 1\right]$$

and should not stabilize for $\alpha > \lambda$. In other words, for $\alpha \ll \lambda < \frac{1}{2}$, the resulting scheme using explicit Euler in time is monotone for $\eta \in \left[1 - \frac{\alpha}{\lambda}, 1\right]$, despite the CFL condition on the cut cell k_1 being violated. Next, we will discuss how to best choose η within the prescribed range.

4 Choice of η in DoD Stabilization

We denote the discrete solution on cell j at time t^n by u_j^n. Resolving the system $\mathcal{M}u^{n+1} = \mathcal{B}_{\text{DoD}}u^n$ for the update on the two cut cells under the condition $\alpha < \lambda < \frac{1}{2}$, we get

$$\begin{aligned} u_{k_1}^{n+1} &= u_{k_1}^n - \frac{\lambda}{\alpha}(1-\eta)\left(u_{k_1}^n - u_{k-1}^n\right), \\ u_{k_2}^{n+1} &= u_{k_2}^n - \frac{\lambda}{1-\alpha}\left(u_{k_2}^n - u_{k_1}^n\right) - \frac{\lambda}{1-\alpha}\eta\left(u_{k_1}^n - u_{k-1}^n\right). \end{aligned}$$

For monotonicity, we need to choose $\eta \in \left[1 - \frac{\alpha}{\lambda}, 1\right]$. We will now examine the two extreme choices, $\eta = 1 - \frac{\alpha}{\lambda}$ and $\eta = 1$, in more detail.

For $\eta = 1 - \frac{\alpha}{\lambda}$, the two update formulae have the following form:

$$u_{k_1}^{n+1} = u_{k-1}^n \quad \text{and} \quad u_{k_2}^{n+1} = \left(1 - \frac{\lambda}{1-\alpha}\right) u_{k_2}^n + \frac{\alpha}{1-\alpha} u_{k_1}^n + \frac{\lambda - \alpha}{1-\alpha} u_{k-1}^n.$$

We observe, comparing with Fig. 1, that the new update formulae now use the correct domains of dependence. In particular, $u_{k_1}^{n+1}$ now coincides with u_{k-1}^n and $u_{k_2}^{n+1}$ now includes information from u_{k-1}^n, which is the neighbor of its inflow neighbor. Actually, the resulting updates correspond to exactly advecting a piecewise constant solution at time t^n to time t^{n+1} and to then averaging. Therefore, thanks to the stabilization, we have implicitly restored the correct domains of dependence. In that sense, the new stabilization has a certain similarity to the h-box method [4].

For the choice $\eta = 1$ the update formulae reduce to:

$$u_{k_1}^{n+1} = u_{k_1}^n \quad \text{and} \quad u_{k_2}^{n+1} = u_{k_2}^n - \frac{\lambda}{1-\alpha}\left(u_{k_2}^n - u_{k-1}^n\right).$$

We observe that in this case the smaller cut cell k_1 will not be updated. Instead, it just keeps its old value. In addition, the update of the solution on cell k_2 does not include information of its inflow neighbor k_1. Therefore choosing $\eta = 1$ can be interpreted as skipping the small cut cell and let the information flow directly from its inflow neighbor into its outflow neighbor.

Remark 1 In [1], we propose to use $\eta = 1 - \frac{\alpha}{2\lambda} \in \left[1 - \frac{\alpha}{\lambda}, 1\right]$. One reason is that this produces more accurate results for piecewise linear polynomials than $\eta = 1 - \frac{\alpha}{\lambda}$, as the latter one is too restrictive in terms of slope limiting.

5 Numerical Results

We will now compare the different choices of η for the DoD stabilization numerically. We consider the grid described in Fig. 1 and place cell k such that $x_{k-\frac{1}{2}} = 0.5$.

We use discontinuous initial data

$$u_0(x) = \begin{cases} 1 & 0.1 \le x \le 0.5, \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

with the discontinuity being placed right in front of the small cut cell k_1. We set $\beta = 1$, $\alpha = 0.001$, $\lambda = 0.4$, and $h = 0.1$, and use $V_h^0(I)$ as well as periodic boundary conditions. We test four different values for η: the extreme cases $\eta = 1$

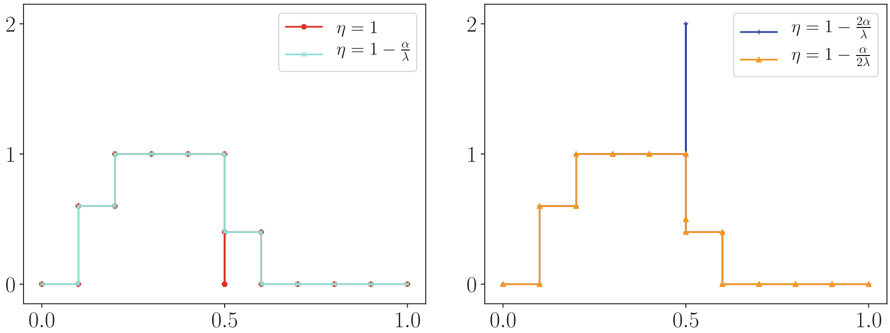


Fig. 2 Results after one time step for the DoD stabilization for different values of η

and $\eta = 1 - \frac{\alpha}{\lambda}$ as well as $\eta = 1 - \frac{\alpha}{2\lambda}$ and a value, $\eta = 1 - \frac{2\alpha}{\lambda}$, that violates the monotonicity considerations.

In Fig. 2 we show the different solutions after one time step. For $\eta = 1$ we observe that the solution on cell k_1 has not been updated, while the updates on the other cells are correct. Obviously, cell k_1 has simply been skipped. The solution for $\eta = 1 - \frac{\alpha}{\lambda}$ corresponds to exactly advecting the initial data and to then apply averaging. If we choose $\eta = 1 - \frac{\alpha}{2\lambda}$, we observe that $u^1_{k_1}$ lies between u^1_{k-1} and $u^1_{k_2}$. Finally, for $\eta = 1 - \frac{2\alpha}{\lambda}$, which is not included in the suggested interval, we observe a strong overshoot on the small cut cell. This cannot happen for a monotone scheme. Therefore, the numerical results confirm our theoretical considerations above.

References

1. C. Engwer, S. May, C. Nüßing, and F. Streitbürger, A stabilized discontinuous Galerkin cut cell method for discretizing the linear transport equation. arXiv:1906.05642, (2019)
2. E. Burman, Ghost penalty. C.R. Math., 348(21):1217–1220, (2010)
3. C. Gürkan and A. Massing, A stabilized cut discontinuous Galerkin framework: II. Hyperbolic problems. arXiv:1807.05634, (2018)
4. M. Berger, C. Helzel, and R. J. Leveque, H-box methods for the approximation of hyperbolic conservation laws on irregular grids. SIAM J. Numer. Anal., 41(3):893–918, (2003)

Global Random Walk Solutions for Flow and Transport in Porous Media

Nicolae Suciu

Abstract This article presents a new approach to solve the equations of flow in heterogeneous porous media by using random walks on regular lattices. The hydraulic head is represented by computational particles which are spread globally from the lattice sites according to random walk rules, with jump probabilities determined by the hydraulic conductivity. The latter is modeled as a realization of a random function generated as a superposition of periodic random modes. One- and two-dimensional numerical solutions are validated by comparisons with analytical manufactured solutions. Further, an ensemble of divergence-free velocity fields computed with the new approach is used to conduct Monte Carlo simulations of diffusion in random fields. The transport equation is solved by a global random walk algorithm which moves computational particles representing the concentration of the solute on the same lattice as that used to solve the flow equations. The integrated flow and transport solution is validated by a good agreement between the statistical estimations of the first two spatial moments of the solute plume and the predictions of the stochastic theory of transport in groundwater.

1 Introduction

Global random walk (GRW) algorithms, which are unconditionally stable and free of numerical diffusion, solve parabolic partial differential equations by moving computational particles on regular lattices according to random walk rules [7]. The GRW approach, intensively used in Monte Carlo simulations of diffusion in random velocity fields [4], is based on the relationship between the ensemble of trajectories

N. Suciu (✉)
Department of Mathematics, Friedrich-Alexander University Erlangen-Nuremberg, Erlangen, Germany
e-mail: suciu@math.fau.de

N. Suciu
Tiberiu Popoviciu Institute of Numerical Analysis, Romanian Academy, Cluj-Napoca, Romania

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_93

of the diffusion process governed by the Itô equation and the probability density of the process which verifies a Fokker–Planck equation. The strength of the approach is that instead of sequentially generating random walk trajectories one counts the number of particles at lattice sites, updated according to the binomial distributions which describe the number of jumps to neighboring lattice sites. Since the total number of particles is arbitrarily large, one obtains in this way highly accurate numerical approximations of the probability density, which, in case of transport simulations, is the normalized concentration of the solute that is transported by advection and diffusion.

The second order differential operator of the Fokker–Planck equation coincides with the diffusion operator in the mass balance equation with closure given by Fick's law only if the diffusion coefficient is constant. Otherwise, in order to rewrite the diffusion equation as a Fokker–Planck equation, the drift coefficients have to be augmented by the spatial derivatives of the diffusion coefficients (e.g., [5, Sect. 4.2.1]). The same procedure can be used to write the pressure equation for flow in porous media based on Darcy's law (e.g., [2]) as a Fokker Planck equation. Further, numerical solutions of the flow problem can be obtained by solving the problem formulated for the equivalent Fokker–Planck equation by GRW methods.

In case of highly variable coefficients, the particles may jump over several lattice sites, the variability of the coefficients is smoothed, and the GRW solutions are affected by overshooting errors [6]. Such errors are avoided by using biased-GRW algorithms, where the particles are only allowed to jump to neighboring sites and the advective displacements are accounted for by biased jump probabilities that are larger in the direction of the drift [5, Sect. 3.3.3]. A different remedy, appropriate for GRW solutions of the flow problem, consists of approximating directly the second order operator in diffusion form, with flux terms estimated at the middle of the lattice intervals by using staggered grids [5, Sect. 3.3.4.1].

We consider a two-dimensional domain, $(x, y) \in \Omega = [0, L_x] \times [0, L_y]$ and the pressure equation for flow in saturated porous media with constant porosity,

$$S\frac{\partial h}{\partial t} - \left[\frac{\partial}{\partial x}\left(K\frac{\partial h}{\partial x}\right) + \frac{\partial}{\partial y}\left(K\frac{\partial h}{\partial y}\right)\right] = f, \tag{1}$$

where $h(t, x, y)$ is the hydraulic head, $K(x, y)$ is an isotropic hydraulic conductivity, S is the specific storage, and $f(x, y)$ is a source/sink term. For an arbitrary initial condition (IC) and time independent boundary conditions (BC), the non-steady solution of Eq. (1) approaches a stationary hydraulic head $h(x, y)$. With $f = 0$ and

$$h(0, y) = H, \quad h(L_x, y) = 0, \quad \forall y \in [0, L_y], \tag{2}$$

$$\frac{\partial h}{\partial y}(x, 0) = \frac{\partial h}{\partial y}(x, L_y) = 0, \quad \forall x \in [0, L_x], \tag{3}$$

the gradient of the hydraulic head $h(x, y)$ determines the components of the divergence-free velocity field according to Darcy's law

$$V_x = -K\frac{\partial h}{\partial x}, \quad V_y = -K\frac{\partial h}{\partial y}. \tag{4}$$

The GRW solutions presented in the following are tested against analytical manufactured solutions [3] obtained from data files and codes given in the Git repository https://github.com/PMFlow/FlowBenchmark. The isotropic hydraulic conductivity is a space random function with mean $\langle K \rangle = 15$ m/day. The random fields $\ln K$ are normally distributed, with Gaussian correlations of fixed correlation length $\lambda = 1$ m and increasing variances $\sigma^2 = 0.1, \ 1, \ 2$, with realizations generated as sums of a finite number $N = 100$ of cosine random modes [5, Appendix C.3.1.2].

2 One-Dimensional GRW Algorithms

The three different GRW approaches discussed in Sect. 1 are tested in the following by solving the one-dimensional (1D) version of Eq. (1), with $S = 1/a = 1$ m^{-1}, written as an advection-diffusion equation,

$$\frac{1}{a}\frac{\partial h}{\partial t} + \frac{\partial}{\partial x}\left(\frac{\partial K}{\partial x}h\right) = \frac{\partial^2}{\partial x^2}(Kh) + f, \tag{5}$$

in the interval $(0, L)$, $L = 200\lambda$. The source f and the BCs $h(0)$ and $h(L)$ are determined by the manufactured solution $\tilde{h}(x) = 3 + \sin(x)$ (see Git repository, /FlowBenchmark/Manufactured_Solutions/Matlab/Gauss 1D/). The precision of the numerical solution $h(x)$ is quantified by the discrete L^2 norm $\varepsilon = \|h - \tilde{h}\|_{L^2}$.

GRW algorithms approximate the hydraulic head by the distribution $n(i, k)$ at lattice sites i and times k of a system of $\mathcal{N}$ random walkers, $h(i\Delta x, k\Delta t) \approx n(i, k)a/\mathcal{N}$. The unbiased GRW moves groups of particles on the lattice according to

$$n(j, k) = \delta n(j + v_j, j, k) + \delta n(j + v_j - d, j, k) + \delta n(j + v_j + d, j, k), \tag{6}$$

$$n(i, k + 1) = \delta n(i, i, k) + \sum_{j \neq i} \delta n(i, j, k) + \lfloor \mathcal{N} f \Delta t \rfloor, \tag{7}$$

where $\lfloor \cdot \rfloor$ is the floor function, $v_j = \lfloor V_j \Delta t/\Delta x + 0.5 \rfloor$, $V_j = a\frac{\partial K}{\partial x}(j\Delta x)$, and d is the size of the diffusive jumps. The time step Δt and the space step Δx are related by

$$K(j\Delta x) = r_j \frac{(d\Delta x)^2}{2a\Delta t}, \tag{8}$$

where r_j, $0 \leq r_j \leq 1$, is a variable jump probability. The number of particles undergoing diffusion jumps, $\delta n(j+v_j \mp d, j, k)$, and the number of particles waiting at $j + v_j$ over the k time step, $\delta n(j + v_j, j, k)$, are binomial random variables with mean values given by [5, Sect. 3.2.2]

$$\overline{\delta n(j + v_j, j, k)} = (1 - r_j)\overline{n(j, k)}, \quad \overline{\delta n(j + v_j \mp d, j, k)} = \frac{1}{2} r_j \overline{n(j, k)}.$$

In the biased GRW algorithm $d = 1$ and Eq. (6) is replaced by

$$n(j, k) = \delta n(j, j, k) + \delta n(j - 1, j, k) + \delta n(j + 1, j, k).$$

The quantities δn verify in the mean

$$\overline{\delta n(j, j, k)} = (1 - r_j)\overline{n(j, k)}, \quad \overline{\delta n(j \mp 1, j, k)} = \frac{1}{2}(r_j \mp v_j)\overline{n(j, k)},$$

where $v_j = V_j \delta t/\delta x$. The biased jump probabilities are given by $(r_j \mp v_j)$ with r_j defined by (8) and, in addition to $r_j \leq 1$, one imposes the constraint $\mid v_j \mid \leq r_j$ [5, Sect. 3.3.3].

Alternatively, a GRW solution of the 1D version of the pressure equation (1),

$$\frac{1}{a}\frac{\partial h}{\partial t} - \frac{\partial}{\partial x}\left(K \frac{\partial h}{\partial x}\right) = f,$$

can be obtained as steady-state limit of the staggered finite difference scheme

$$\frac{1}{a\Delta t}[h(i, k + 1) - h(i, k)] =$$
$$\frac{1}{\Delta x^2}\{[K(i + 1/2)(h(i + 1, k) - h(i, k))] - [K(i - 1/2)(h(i, k) - h(i - 1, k))]\} + f.$$

With jump probabilities defined by $r(i \mp 1/2) = K(i \mp 1/2)a\Delta t/\Delta x^2$, $r \leq 1/2$, the staggered scheme becomes

$$\begin{aligned} n(i, k + 1) = {} & [1 - r(i - 1/2) - r(i + 1/2)]\, n(i, k) \\ & + r(i - 1/2)n(i - 1, k) + r(i + 1/2)n(i + 1, k) + \lfloor \mathcal{N} f \Delta t \rfloor. \end{aligned} \quad (9)$$

The contributions to lattice sites i from neighboring sites summed up in (9) are obtained with the GRW algorithm which moves particles from sites j to neighboring sites $i = j \mp 1$ according to

$$n(j, k) = \delta n(j, j, k) + \delta n(j - 1, j, k) + \delta n(j + 1, j, k). \quad (10)$$

For consistency with the staggered scheme (9), the quantities δn in (11) have to satisfy in the mean [5, Sect. 3.3.4.1],

$$\overline{\delta n(j,j,k)} = [1 - r(j-1/2) - r(j+1/2)]\,\overline{n(j,k)}, \quad \overline{\delta n(j \mp 1, j, k)} = r(j \mp 1/2)\overline{n(j,k)}. \tag{11}$$

The three GRW algorithms from above approximate the binomial random variables δn by rounding of the unaveraged relations for the mean, e.g., (11), summing up the reminders of multiplication by r and of the floor function $\lfloor \mathcal{N} f \Delta t \rfloor$, and allocating one particle to the lattice site where the sum reaches the unity. By giving up the particle indivisibility, one obtains deterministic GRW algorithms which represent the solution n by real numbers and use the unaveraged relations for δn.

Since the unbiased GRW algorithm is prone to large overshooting errors, due to the high variability of the coefficients $\frac{\partial K}{\partial x}$ and K, it is not appropriate to solve the flow problem. For instance, to solve the test problem for $\sigma^2 = 0.1$ with errors $\varepsilon \sim 10^{-1}$, an extremely small step $\Delta x = 10^{-5}$ is needed and one time iteration takes about 8 min. Efficient solutions can be obtained with the biased GRW algorithm (Method 1) and the GRW on staggered grids in both the random (Method 2) and the deterministic implementation (Method 3). The stationary regime of the GRW simulations, indicated by constant total number of particles and L^2 error (see Fig. 1), is reached after $T = 2 \cdot 10^7$ time iterations. The comparison shown in Table 1 indicates that the deterministic implementation of the GRW on staggered grids is the most efficient approach to solve flow problems for heterogeneous porous media.

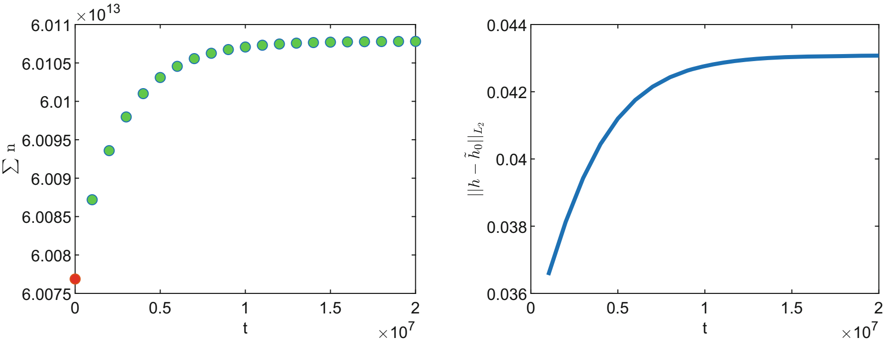


Fig. 1 Convergence of the 1D GRW solution for IC given by $h_0(x) = \tilde{h}(x)$, with f and K specified by $\langle K \rangle = 15$ m/day, $N = 100$, and $\sigma^2 = 0.1$

Table 1 Comparison of 1D GRW methods

	Method 1	Method 2	Method 3
ε	4.31e−02	1.60e−02	1.60e−02
CPU (min)	37.12	14.03	6.48

3 Two-Dimensional GRW Solutions of the Flow Problem

The 2D GRW algorithm on staggered grids is defined similarly to (9–11) as follows:

$$\begin{aligned} n(i,j,k+1) = {} & [1 - r(i-1/2,j) - r(i+1/2,j) - r(i,j-1/2) \\ & -r(i,j+1/2)]\,n(i,j,k) \\ & + r(i-1/2,j)n(i-1,j,k) + r(i+1/2,j)n(i+1,j,k) \\ & + r(i,j-1/2)n(i,j-1,k) + r(i,j+1/2)n(i,j+1,k) \\ & + \lfloor \mathcal{N} f \Delta t \rfloor, \end{aligned} \tag{12}$$

$$\begin{aligned} n(l,m,k) = {} & \delta n(l,m|l,m,k) + \delta n(l-1,m|l,m,k) + \delta n(l+1,m|l,m,k) \\ & + \delta n(l,m-1|l,m,k) + \delta n(l,m+1|l,m,k), \end{aligned} \tag{13}$$

$$\begin{aligned} & \overline{\delta n(l,m|l,m,k)} = \\ & [1 - r(l-1/2,m) - r(l+1/2,m) - r(l,m-1/2) - r(l,m+1/2)]\,\overline{n(l,m,k)} \\ & \overline{\delta n(l \mp 1,m|l,m,k)} = r(l \mp 1/2,m)\overline{n(l,m,k)} \\ & \overline{\delta n(l,m \mp 1|l,m,k)} = r(l,m \mp 1/2)\overline{n(l,m,k)}. \end{aligned} \tag{14}$$

Equation (1), with source term f, BC, and IC determined by the manufactured solution $\tilde{h}(x,y) = 1 + \sin(2x+y)$, domain dimensions $L_x = 20\lambda$, $L_y = 10\lambda$, and $\Delta x = \Delta y = 10^{-1}$, is solved by the deterministic GRW on staggered grids. The L^2 errors with respect to the manufactured solution are shown in Table 2.

The convergence of the 2D GRW solutions is investigated for the homogeneous problem ($f = 0$, and IC given by the constant slope $h_0(0,y) = 1,\ h_0(L,y) = 0$), for increasing σ^2, by successively halving the step size five times from $\Delta x = \Delta y = 10^{-1}$ to $\Delta x = \Delta y = 3.125 \cdot 10^{-3}$. Using the errors with respect to the solution on the finest grid, $\varepsilon_k = \|h^{(k)} - h^{(6)}\|_{L^2}$, the estimated order of convergence (EOC) is computed according to $EOC = \log(\varepsilon_k/\varepsilon_{k+1}) / \log(2),\ k = 1,\ldots,4$. The results presented in Table 3 indicate the convergence of order 2 for all the three values of σ^2.

Table 2 Errors of 2D GRW solutions for increasing σ^2

	$\sigma^2 = 0.1$	$\sigma^2 = 1$	$\sigma^2 = 2$
ε	3.12e−02	4.08e−02	1.35e−01
T	2e06	1e07	2e07
CPU (min)	11.26	55.49	111.54

Table 3 Computational order of convergence of the 2D GRW algorithm

σ^2	ε_1	EOC	ε_2	EOC	ε_3	EOC	ε_4	EOC	ε_5
0.1	2.78e−03	1.9979	6.96e−04	2.0167	1.72e−04	2.0687	4.10e−05	2.3219	8.20e−06
1	2.63e−03	1.9902	6.62e−04	2.0131	1.64e−04	2.0685	3.91e−05	2.3219	7.82e−06
2	2.67e−03	1.9839	6.75e−04	2.0150	1.67e−04	2.0654	3.99e−05	2.3219	7.98e−06

4 Statistical Inferences

A Monte Carlo ensemble of solutions is obtained by solving the 2D homogeneous problem (1–4) with $f = 0$ and $H = 1$, for 100 realizations of the random K-field with fixed $\sigma^2 = 0.1$. A realization of the hydraulic head is shown in Fig. 2. The components of the velocity field computed according to Darcy's law (4) are shown in Fig. 3.

The Monte Carlo estimates presented in Table 4 are close to the first-order theoretical predictions of the stochastic theories of flow and transport in groundwater [1], e.g. variances of the order $\sigma_h^2 \sim (\sigma\lambda H/L_x)^2 \sim 10^{-4}$ for the hydraulic head, $\sigma_{V_x}^2 = \frac{3}{8}\sigma^2 = 3.75 \cdot 10^{-2}$, and $\sigma_{V_y}^2 = \frac{1}{8}\sigma^2 = 1.25 \cdot 10^{-2}$, for longitudinal and transverse velocity components, respectively.

5 Validation of the Flow and Transport GRW Solutions

The numerical setup from Sect. 4 corresponds to a possible scenario of contaminant transport in groundwater systems with low heterogeneity [4]. With a typical value of the local dispersion coefficient $D_0 = 0.01\,\mathrm{m}^2/\mathrm{day}$, Monte Carlo simulations of advection-diffusion are carried out using the ensemble of 100 velocity fields, on

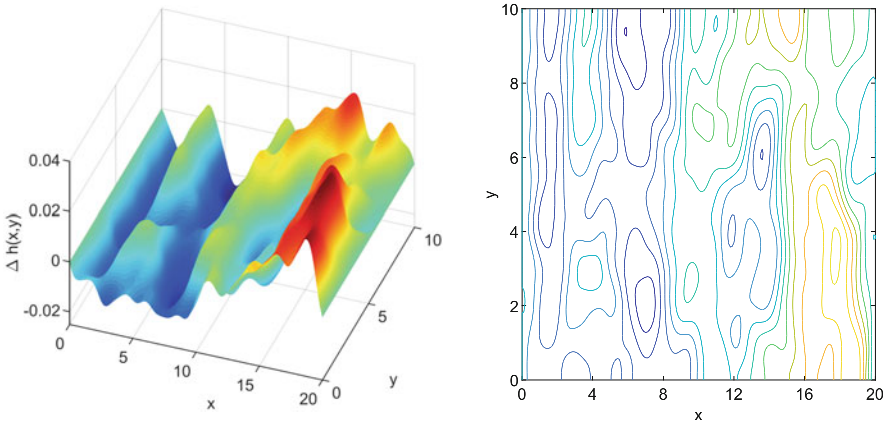


Fig. 2 Fluctuations of $h(x, y)$ about the mean (left) and the corresponding contour lines (right)

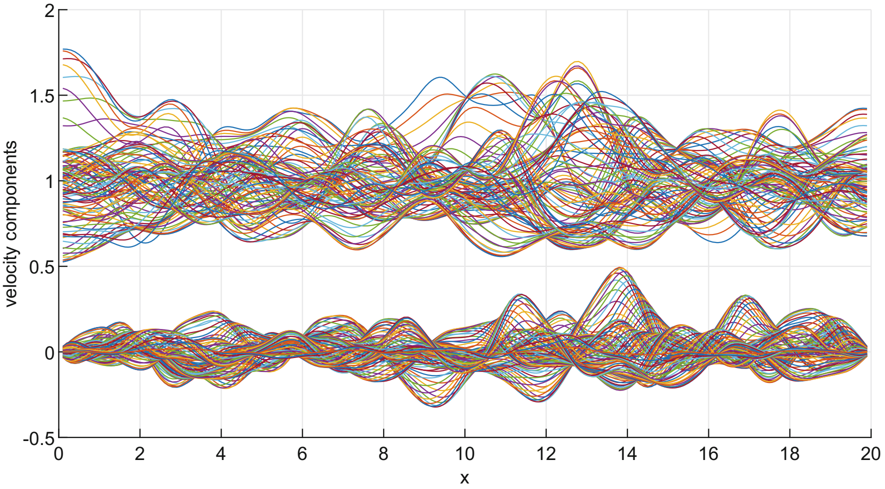


Fig. 3 Longitudinal velocity components of mean $\langle V_x \rangle = 1$ m/day and transverse components of mean $\langle V_y \rangle = 0$ m/day represented as function of x for fixed values of y

Table 4 Monte Carlo estimates of mean values and variances of the hydraulic head and of the velocity components

	h	V_x	Vy
Mean	5.00e−01± 1.74e−01	9.98e−01± 1.91e−02	−7.32e−04± 1.24e−02
Variance	3.27e−04± 6.27e−05	3.68e−02± 5.50e−03	1.30e−02± 1.90e−03

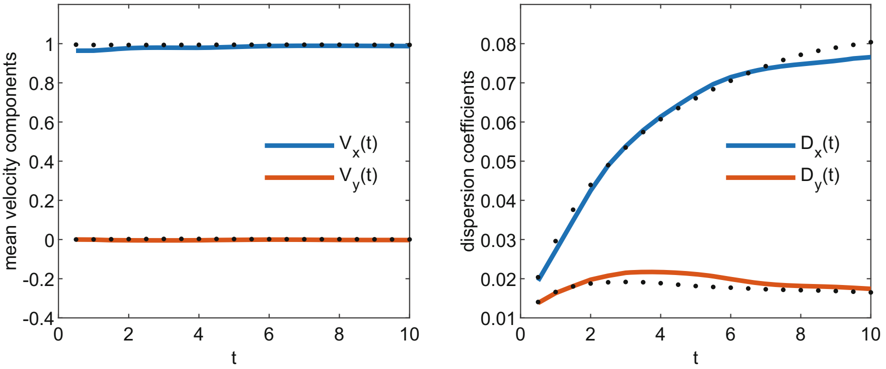


Fig. 4 Mean velocity components of the center of mass of the solute plume (left) and effective dispersion coefficients (right) compared with first-order approximations (dotted lines)

the same lattice as that used in Sect. 4 to solve the flow problem. The transport problem is solved with the 2D generalization of the unbiased GRW algorithm (6–7) [5, Appendix A.3.2]. The Monte Carlo estimates of the velocity of the plume's center of mass and of the effective dispersion coefficients presented in Fig. 4 are in good agreement with the first-order results obtained from simulations using a linear approximation of the velocity field [5, Appendix C.3.2.2].

References

1. Dagan, G.: Flow and Transport in Porous Formations. Springer, Berlin (1989) https://doi.org/10.1007/978-3-642-75015-1
2. Radu, F.A., Suciu, N., Hoffmann, J., Vogel, A., Kolditz, O., Park, C-H., Attinger, S.: Accuracy of numerical simulations of contaminant transport in heterogeneous aquifers: a comparative study. Adv. Water Resour. **34**, 47–61 (2011) http://dx.doi.org/10.1016/j.advwatres.2010.09.012
3. Roache, P.J.: Code verification by the method of manufactured solutions. J. Fluids Eng. **124** (1):4–10 (2002) http://dx.doi.org/10.1115/1.1436090
4. Suciu, N.: Diffusion in random velocity fields with applications to contaminant transport in groundwater. Adv. Water Resour. **69** (2014) 114-133. http://dx.doi.org/10.1016/j.advwatres.2014.04.002
5. Suciu, N.: Diffusion in Random Fields. Applications to Transport in Groundwater. Birkhäuser, Cham (2019) https://doi.org/10.1007/978-3-030-15081-5
6. Suciu, N., Vamoş, C.: Evaluation of overshooting errors in particle methods for diffusion by biased global random walk. Rev. Anal. Num. Th. Approx. (Romanian Academy) **35**, 119–126 (2006) https://ictp.acad.ro/jnaat/journal/article/view/2006-vol35-no1-art17
7. Vamoş, C., Suciu, N., Vereecken, H.: Generalized random walk algorithm for the numerical modeling of complex diffusion processes. J. Comput. Phys., **186**, 527–544 (2003) https://doi.org/10.1016/S0021-9991(03)00073-1.

On Finite Element Approximation of Aeroelastic Problems with Consideration of Laminar-Turbulence Transition

Petr Sváček

Abstract This paper focus on a finite element approximation of aeroelastic problems. The turbulent flow interacting with flexibly supported airfoil is considered. The flow is described by unsteady Reynolds averaged Navier–Stokes equations, where the main attention is paid to the simulation of turbulent flow with the transition. The motion of the computational domain is addressed and the coupled aeroelastic problem is discretized. Numerical results are shown.

1 Introduction

In last decades new computational methods become available for numerical simulations of fluid-structure interaction problems. However, the numerical approximation of such problems is still computationally very expensive. This is even worse for the problems where the turbulent character of the flow needs to be considered. The use of DNS or LES simulations is very limited particularly to low-Reynolds cases and consequently the turbulent flow models based on Reynolds averaged Navier–Stokes (RANS) equations are being used in technical practice in combination with a suitable turbulence model. Recently also several transition models become available, but the proper choice of the turbulence/transition model depends an additional knowledge about the addressed situation. The transition models are usually used for steady problems. Its application for unsteady flows on time dependent domain is rather rare, see e.g. [1] or [2].

This paper focus on application of a transition turbulence model in the context of aeroelastic simulations. Compared to the previous works here is the use of stabilized finite element method for approximation of so-called ALE conservative form of turbulence/transition equations. Compared to the previous works the method is

P. Sváček (✉)
Czech Technical University in Prague, Faculty of Mechanical Engineering, Department of Technical Msathematics, Praha 2, Czech Republic
e-mail: Petr.Svacek@fs.cvut.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_94

additionally applied to a different benchmark problem of flutter type of aeroelastic instability, see [3].

This paper focus on simulation of an aeroelastic benchmark problem, where Menter's $\gamma - Re_{\theta_t}$ model of transition to turbulence is used. The flow model is coupled to the motion of a flexibly supported airfoil model, where the geometrical nonlinearities (see [4]) are taken into an account. The numerical results are shown.

2 Mathematical Description

The motion of a flexibly supported airfoil immersed in a flowing fluid is described in terms of its vertical displacement h (downwards positive) and rotation by angle α (clockwise positive), see Fig. 1. The non-linear equations of motion (see [5]) are given by

$$\begin{aligned} m\ddot{h} + S_\alpha\,\ddot{\alpha}\,\cos\alpha - S_\alpha\dot{\alpha}^2\sin\alpha + k_h h &= -L(t), \\ S_\alpha\ddot{h}\cos\alpha + I_\alpha\ddot{\alpha} + k_\alpha\alpha &= M(t). \end{aligned} \tag{1}$$

where m denotes the mass of the airfoil, S_α is its static moment around the elastic axis (EA), I_α is its inertia moment around EA, k_h and k_α denotes the bending and torsional stiffness constants, respectively. The aerodynamical lift force $L(t)$ and aerodynamical torsional moment $M(t)$ are evaluated with the aid of the mean (kinematic) pressure p and the mean flow velocity $\boldsymbol{u} = (u_1, u_2)$ as

$$L = -l\int_{\Gamma_{Wt}} \sigma_{2j} n_j\, dS, \qquad M = l\int_{\Gamma_{Wt}} \sigma_{ij} n_j r_i^{\mathrm{ort}}\, dS, \tag{2}$$

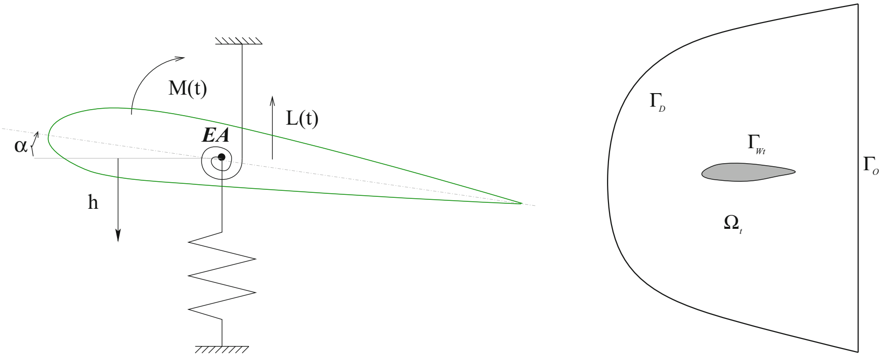


Fig. 1 The flexibly supported airfoil model and a sketch of the computational domain with boundary parts

where l denotes the depth of the airfoil section of the airfoil, and $r_1^{\text{ort}} = -(x_2 - x_2^{\text{EA}})$, $r_2^{\text{ort}} = x_1 - x_1^{\text{EA}}$, and $(x_1^{\text{EA}}, x_2^{\text{EA}})$ is the position of EA at time t. Further, σ_{ij} is the stress tensor with components $\sigma_{ij} = \rho(-p\delta_{ij} + \nu_{eff} S_{ij})$, by S_{ij} the components of the symmetric gradient of the velocity are denoted, i.e. $\boldsymbol{S}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + \nabla^T \boldsymbol{u})$. The effective viscosity ν_{eff} is given as sum of the kinematic viscosity ν and the turbulent kinematic viscosity ν_T.

The mean velocity $\boldsymbol{u}$ and the mean pressure p are modelled in Ω_t by the Reynolds Averaged Navier–Stokes equations, see [6]. In order to treat the time-dependent domain Ω_t the RANS equations are written in the conservative ALE form (see e.g. [7]) as

$$\frac{1}{\mathcal{J}} \frac{D^{\mathcal{A}_t}(\mathcal{J}\boldsymbol{u})}{Dt} + \nabla\cdot((\boldsymbol{u} - \boldsymbol{w}_D) \otimes \boldsymbol{u}) - \nabla\cdot\big(2\nu_{eff}\boldsymbol{S}(\boldsymbol{u})\big) + \nabla p = 0, \quad \nabla\cdot\boldsymbol{u} = 0, \tag{3}$$

where $\mathcal{A}_t$ is assumed to be a smooth ALE mapping of a reference configuration Ω_0^{ref} onto the current configuration Ω_t at any $t \in (0, T)$, $\mathcal{J} = \mathcal{J}(x, t)$ denotes its Jacobian, $D^{\mathcal{A}}/Dt$ is the ALE derivative (i.e. the time derivative with respect to the reference configuration Ω_0^{ref}) and $\boldsymbol{w}_D$ denotes the domain velocity (i.e. the velocity of a point of the reference configuration).

The system (3) is equipped with an initial condition and boundary conditions prescribed on the mutually disjoint parts of the boundary $\partial\Omega = \Gamma_D \cup \Gamma_O \cup \Gamma_{Wt}$:

$$\begin{aligned} &\text{(a)} \quad \boldsymbol{u} = \boldsymbol{u}_D \quad \text{on } \Gamma_D, \qquad \text{(b)} \quad \boldsymbol{u} = \boldsymbol{w}_D \quad \text{on } \Gamma_{Wt}, \\ &\text{c)} \ -\nu_{\text{eff}}(\nabla\boldsymbol{u} + \nabla^T\boldsymbol{u})\boldsymbol{n} + (p - p_{\text{ref}})\boldsymbol{n} + \tfrac{1}{2}(\boldsymbol{u}\cdot\boldsymbol{n})^-\boldsymbol{u} = 0 \qquad \text{on } \Gamma_O, \end{aligned} \tag{4}$$

where p_{ref} denotes a reference pressure and $\alpha^- = \min(0, \alpha)$ denotes the negative part of the number $\alpha \in \mathbb{R}$.

For modelling of the turbulent-laminar flow transition the Menter's SST $k - \omega$ turbulence model, see [8]. The transition from laminar to turbulence regimes is modelled with the aid of the Menter's $\gamma - \overline{Re}_{\theta t}$ model, see [9]. The turbulent viscosity ν_T is modelled by $\nu_T = \frac{k}{\omega}$, where k and ω are the turbulent kinetic energy and the turbulent specific dissipation rate, respectively. The governing system of equations (written in the ALE conservative form) for k and ω reads

$$\begin{aligned} \frac{1}{\mathcal{J}} \frac{D^{\mathcal{A}}(\mathcal{J}k)}{Dt} + \nabla\cdot((\boldsymbol{u} - \boldsymbol{w}_D)k) &= \gamma_{\text{eff}} P_k - \beta^*\omega k \overline{\gamma_{\text{eff}}} + \nabla\cdot(\varepsilon_k \nabla k), \\ \frac{1}{\mathcal{J}} \frac{D^{\mathcal{A}}(\mathcal{J}\omega)}{Dt} + \nabla\cdot((\boldsymbol{u} - \boldsymbol{w}_D)\omega) &= P_\omega - \beta\omega^2 + \nabla\cdot(\varepsilon_\omega \nabla\omega) + C_D, \end{aligned} \tag{5}$$

where the viscosity coefficients are given by $\varepsilon_k = \nu + \sigma_k \nu_T$, $\varepsilon_\omega = \nu + \sigma_\omega \nu_T$, and the source terms P_k, P_ω and C_D are defined by

$$P_k = \nu_T \mathbf{S}(\boldsymbol{u}) : \mathbf{S}(\boldsymbol{u}), \quad P_\omega = \frac{\alpha_\omega \omega}{k} P_k, \quad C_D = \frac{\sigma_D}{\omega}(\nabla k \cdot \nabla\omega)^+.$$

The closure coefficients β, β^*, σ_k, σ_ω, α_ω, σ_D are chosen, e.g., according to [10] or [11]. The production and the destruction terms are decreased using the effective intermittency $\gamma_{\text{eff}} = \max(\gamma, \gamma_{sep})$, which is taken as the maximum of the modelled intermittency γ and the intermittency γ_{sep} modelled by an algebraic model. For the destruction term of (5) the limited value of the intermittency is used, i.e. $\overline{\gamma}_{\text{eff}} = \max(\min(\gamma_{\text{eff}}, 1), 0.1)$. The intermittency γ is modelled by

$$\frac{1}{\mathcal{J}} \frac{D^{\mathcal{A}}(\mathcal{J}\gamma)}{Dt} + \nabla \cdot ((\boldsymbol{u} - \boldsymbol{w}_D)\gamma) = P_\gamma - E_\gamma + \nabla \cdot (\varepsilon_\gamma \nabla \gamma), \tag{6}$$

where $\varepsilon_\gamma = (\nu + \nu_T/\sigma_f)$, $\sigma_f = 1$. Further $P_\gamma = P_{\gamma,1} - c_{e1}\gamma P_{\gamma,1}$ and $E_\gamma = c_{e2}\gamma E_{\gamma,1} - E_{\gamma,1}$ are the transition source and destruction terms, respectively. These terms are evaluated with a transported unknown $\overline{Re}_{\theta t}$ governed by

$$\frac{1}{\mathcal{J}} \frac{D^{\mathcal{A}}(\mathcal{J}\overline{Re}_{\theta t})}{Dt} + \nabla \cdot ((\boldsymbol{u} - \boldsymbol{w}_D)\overline{Re}_{\theta t}) = P_{\theta t} + \nabla \cdot (\varepsilon_{\overline{Re}_{\theta t}} \nabla \overline{Re}_{\theta t}), \tag{7}$$

where $\varepsilon_{\overline{Re}_{\theta t}} = 2\nu_{\text{eff}}$ and the source term is given by $P_{\theta t} = c_{\theta t} \frac{\rho}{t_\infty} \left(Re_{\theta t} - \overline{Re}_{\theta t} \right) (1 - F_{\theta t})$. The production and destruction terms of Eq. (6) are given as

$$P_{\gamma,1} = c_{a1} F_{\text{length}} S \sqrt{\gamma F_{\text{onset}}}, \; E_{\gamma,1} = c_{a2} \Omega \gamma F_{\text{turb}},$$

where $c_{a1} = 2$, $c_{a2} = 0.06$, $c_{\theta t} = 0.03$, $t_\infty = 500\nu/U^2$ is the time scale, U is the local magnitude of the velocity, S and Ω are the strain rate and vorticity magnitudes. The transition onset is modelled by a function F_{onset}, function $F_{\theta t}$ is a blending function, see [12]. In order to enclose the model, the empirical correlations are used for the length of the transition F_{length}, the transition onset momentum thickness Reynolds number $Re_{\theta t}$ and for the critical transition Reynolds number $Re_{\theta c}$, see [12], see also [1].

The turbulence model (5)—(7) is equipped with the Dirichlet boundary conditions for all quantities at the inlet Γ_D, the Neumann/Newton boundary condition for all quantities at the outlet Γ_O, the Dirichlet boundary conditions for k and ω and the homogenous Neumann boundary condition for γ and $\overline{Re}_{\theta t}$ at Γ_{Wt}.

3 Numerical Approximation

The Reynolds averaged Navier–Stokes equation are numerically approximated with the aid of the fully stabilized finite element method using the Taylor-Hood finite element pair for velocity/pressure approximations in the same way as described in [1]. The turbulence model equations are weakly formulated, time discretized and the finite element method is applied. For the sake of brevity the finite element discretization procedure used for approximation of Eqs. (5)–(7) is described here for an equation of the same type for an unknown scalar variable ϕ. To this end we consider the following equation for the variable ϕ in the domain Ω_t

$$\frac{1}{\mathcal{J}} \frac{D^{\mathcal{A}}(\mathcal{J}\phi)}{Dt} + \nabla \cdot ((\boldsymbol{u} - \boldsymbol{w}_D)\phi) = P_\phi^+ \phi - P_\phi^- \phi + \nabla \cdot (\epsilon_\phi \nabla\phi), \tag{8}$$

where $\epsilon_\phi = \epsilon_\phi(\phi)$ is a diffusion coefficient and $P_\phi^+ = P_\phi^+(\phi)$ and $P_\phi^- = P_\phi^-(\phi)$ are production and destruction terms. The Eq. (8) is equipped by the boundary conditions prescribed on $\partial\Omega = \Gamma_{D'} \cup \Gamma_N$

$$\text{(a)}\ \phi = \phi_D \text{ on } \Gamma_{D'}, \qquad \text{(b)}\ \varepsilon_\phi \frac{\partial\phi}{\partial \boldsymbol{n}} = \frac{1}{2}((\boldsymbol{u} - \boldsymbol{w}_D) \cdot \boldsymbol{n})^- \phi, \text{ on } \Gamma_N, \tag{9}$$

where $\Gamma_{D'}$ either equals Γ_D for ϕ being γ and $\overline{Re}_{\theta t}$ or it is $\Gamma_D \cup \Gamma_{Wt}$ for ϕ being k or ω. Boundary condition (9b) is the homogenous Neumann boundary condition if either $\boldsymbol{u} - \boldsymbol{w}_D = 0$ (e.g. on Γ_{Wt}) or if $\boldsymbol{u} \cdot \boldsymbol{n} > 0$ (typically on Γ_O with no backward inflow).

In what follows let us consider the ALE mapping $\mathcal{A}_t$ to be already known (as well as the domain Ω_t and the domain velocity $\boldsymbol{w}_D$) and sufficiently smooth at every time instant $t \in I = (0, T)$. For the purpose of time discretization an equidistant partition $t_j = j\Delta t$ of the time interval I with the constant time step $\Delta t > 0$ is considered and we denote the approximations $\phi^j \approx \phi(\cdot, t_j)$ for $j = 0, 1, \ldots$. We assume that the approximation $\phi^j \approx \phi(, t_j)$ are already known for all $j \leq n$ as well as the approximations of the flow velocity $\boldsymbol{u}^{n+1} \approx \boldsymbol{u}(t_{n+1})$ and the discretization is described only at a fixed (but arbitrary) time instant t_{n+1}.

Let us denote by $V = H^1(\Omega_{t_{n+1}})$ the trial space (a subspace of Sobolev space) and by $X \subset V$ the space of the test functions from V being zero on Γ_D is denoted. Although the test functions $\psi \in X$ are defined at the time instant t_{n+1} on $\Omega_{t_{n+1}}$, we shall extend them to be defined for $x \in \Omega_t$ at any time t using the one-to-one ALE mapping $\mathcal{A}_t$ with the assumption, that these functions are time independent on the reference domain Ω_0^{ref}, i.e. the extension of the test function ψ_{ext} for any $x \in \Omega_t$ at any time $t \in (0, T)$ is defined by

$$\psi_{ext}(x, t) = \psi(\mathcal{A}_{t_{n+1}}(\mathcal{A}_t^{-1}(x))). \tag{10}$$

Using the extension of the test function $\psi \in X$ (formally we use the same notation $\psi = \psi_{ext}$) the weak form of the time derivative term can be written as

$$\int_{\Omega_t} \frac{1}{\mathcal{J}} \frac{D^{\mathcal{A}}(\mathcal{J}\phi)}{Dt} \psi \mathrm{dx} = \frac{d}{dt}\left(\int_{\Omega_t} \phi\psi \mathrm{dx}\right). \tag{11}$$

Problem (8) is multiplied by a test function $\psi \in X$, integrated over Ω_t, Green's theorem is applied and boundary conditions (9) are used. The time derivative—see the right hand side of equation (11)—is approximated at $t = t_{n+1}$ by the second order backward difference formula. The spaces V and X are approximated using the finite element subspaces V_h and $X_h = V_h \cap X$ constructed over an admissible triangulation $\mathcal{T}_\Delta$ of the domain Ω, respectively. In order to treat the dominating convection the SUPG stabilization method is used and the nonlinear terms are linearized. The non-linear cross-wind diffusion term is used to prevent the nonphysical negative values, see [13].

The stabilized discrete formulation reads: Find $\phi = \phi_h^{n+1} \in V_h$ such that it satisfies approximately the Dirichlet boundary condition and for all test function $\psi_h \in X_h$ holds

$$B(\phi; \phi, \psi_h) + B_S(\phi; \phi_h^{n+1}, \psi_h) = \mathcal{L}(\phi; \psi_h) + \mathcal{L}_S(\phi; \psi_h),$$

where the (Galerkin) terms $B(\cdot,\cdot)$ and $L(\cdot)$ are given by

$$B(\overline{\phi}; \phi, \psi) = \left(\overline{\varepsilon}_\phi \nabla\phi, \nabla\psi\right)_\Omega + \left(\frac{3\phi}{2\Delta t} + \overline{P_\phi^-}\phi + \frac{1}{2}(\nabla\cdot \boldsymbol{w}_D)\phi, \psi\right)_\Omega + c(\phi, \psi)$$

$$c(\phi, \psi) = \int_\Omega \left(\frac{1}{2}(\overline{\boldsymbol{w}}\cdot\nabla\phi)\psi - \frac{1}{2}(\overline{\boldsymbol{w}}\cdot\nabla\psi)\phi\right) dx + \int_{\Gamma_O} \frac{1}{2}(\overline{\boldsymbol{w}}\cdot\boldsymbol{n})^+\psi\phi.$$

$$\mathcal{L}(\overline{\phi}; \psi) = \left(\frac{4}{2\Delta t}\phi^n, \psi\right)_{\Omega_{t_n}} - \left(\frac{1}{2\Delta t}\phi^{n-1}, \psi\right)_{\Omega_{t_{n-1}}} + \left(\overline{P_\phi^+}\overline{\phi}, \psi\right)_\Omega. \qquad (12)$$

where $\overline{\boldsymbol{w}} = \boldsymbol{u} - \boldsymbol{w}_D$ denotes the convective velocity. The linearization of the nonlinear terms $\overline{\varepsilon}_\phi = \varepsilon_\phi(\overline{\phi})$ and $\overline{P_\phi^\pm} = P_\phi^\pm(\overline{\phi})$ was used. The stabilization terms based on the triangulation $\mathcal{T}_\Delta$ read

$$B^S(\overline{\phi}; \phi, \psi) = \sum_{K\in\mathcal{T}_\Delta} \delta_K \left(\frac{3\phi}{2\Delta t} + \overline{\boldsymbol{w}}\cdot\nabla\phi + \overline{P_\phi^-}\phi + \nabla\cdot\left(\overline{\varepsilon}_\phi\nabla\phi\right), \overline{\boldsymbol{w}}\cdot\nabla\psi\right)_K$$

$$L^S(\Phi) = \sum_{K\in\mathcal{T}_\Delta} \delta_K \left(\frac{4\hat{\phi}^n - \hat{\phi}^{n-1}}{2\Delta t} + \overline{P_\phi^+}\overline{\phi}, \overline{\boldsymbol{w}}\cdot\nabla\psi\right)_K$$

where the stabilizing parameter δ_K is computed from the local element length, the local viscosity coefficient, and the local magnitude of the convective velocity $\boldsymbol{u} - \boldsymbol{w}_D$, see [14] or [15].

4 Numerical Results

The described numerical method was realized within the in-house code (see [16]) was applied on solution of an aeroelastic problem. The far field velocity U_∞ was considered in the range $0 - 30$ m/s and the following data were used: the kinematic viscosity $\nu = 1.5 \times 10^{-5}$, the air density $\rho = 1.225\,\mathrm{kg\,m}^{-}3$, and the reference length was equal to the airfoil chord $c = 0.254\,\mathrm{m}$. The depth of the considered

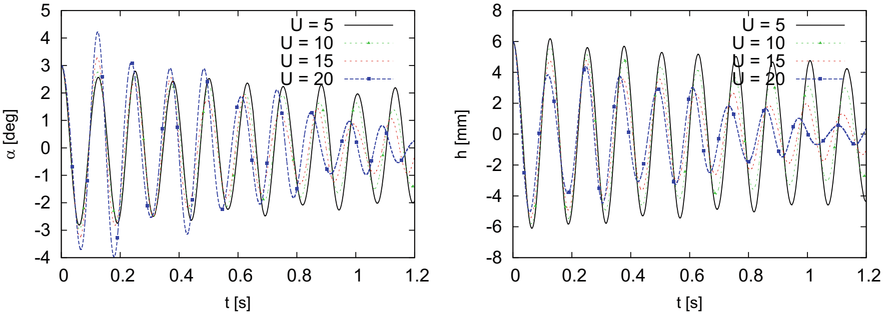


Fig. 2 The aeroelastic response for far field velocities in {5, 10, 15, 20} m/s

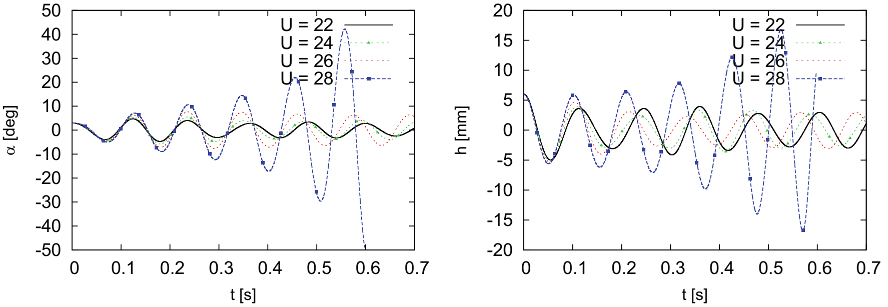


Fig. 3 The aeroelastic response for far field velocities in {22, 24, 26, 28} m/s

section was $l = 1$m. The corresponding mass was $m = 4.72\,$kg, the static moment S_α with respect to the elastic axis EA (at 40% of the airfoil from the leading edge) was $S_\alpha = 1.498 \times 10^{-1}\,$kg m and the inertia moment was $I_\alpha = 2.95 \times 10^{-2}\,$kg m^2. The bending and the torsional stiffnesses was $k_h = 14741.1$ and $k_\alpha = 121.3$N m/rad, respectively. The inlet turbulence intensity was chosen to be 2% and the inlet specific dissipation was chosen as $\omega = 10\,\text{s}^{-1}$.

The results in terms of the aeroelastic responses in terms of h, α are shown in Figs. 2 and 3. Figure 2 shows that the vibrations of the airfoil are well damped for the far field velocities up to 20 m/s. With further increase of the far field the damping becomes smaller and for 28 m/s the undamped vibrations appear. This is in a good agreement with the critical velocity U_{crit} determined by the Theodorsen analysis $U_{crit} = 27.8\,$m/s. The comparisons in terms of $V - f$ (velocity-frequency) and $V - g$ (velocity-damping) diagrams are shown in Fig. 4. Very good agreement in frequency is observed for each considered inflow velocity, whereas the damping coefficients agrees well just for lower velocities. In the near-critical range of velocities the damping coefficient determined from the simulations does not agree well with the damping coefficient determined by the Theodorsen theory. However, the determined value of the critical velocity corresponds well to the critical velocity found by the Theodorsen analysis, see Fig. 4.

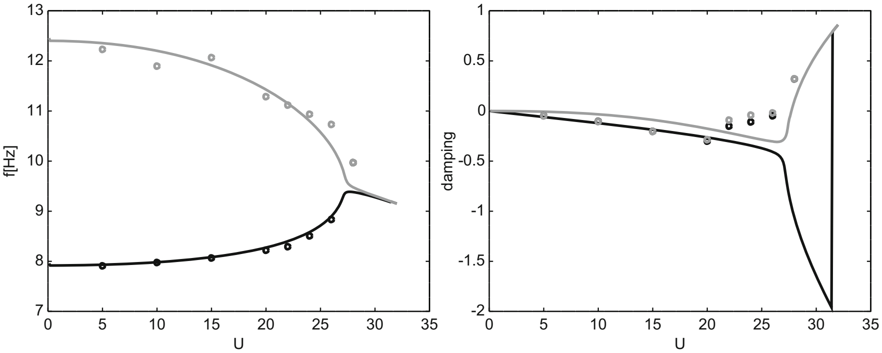


Fig. 4 The comparison of the aeroelastic response computed by the linearized approach and by the transition model

5 Conclusion

In this paper the application of a turbulence transition model for solution of aeroelastic problems was described. The described method was tested on a benchmark aeroelastic problem and it was shown that the use of the model provides reliable results even when used for unsteady aeroelastic problems and small displacements (e.g. aeroelastically stable region). The use of transition model predicts the aerodynamical quantities as drag/lift coefficients more precisely compared to standard turbulence models as $k-\omega$. On the other hand the application of the transition model for post-flutter simulations seems still questionable and the results can be aeroelastic results can be significantly influenced by possible appearance of a separation region.

Acknowledgments Authors acknowledge support from the EU Operational Programme Research, Development and Education, and from the Center of Advanced Aerospace Technology (CZ.02.1.01/0.0/0.0/16_019/0000826), Faculty of Mechanical Engineering, Czech Technical University in Prague.

References

1. P. Sváček, J. Horáček, Numerical simulation of aeroelastic response of an airfoil in flow with laminar-turbulence transition, Applied Mathematics and Computation 267 (2015) 28–41.
2. O. Winter, Sváček, On numerical simulation of flexibly supported airfoil in interaction with incompressible fluid flow using laminar-turbulence transition model, Computers & Mathematics with Applications(in press) (2020). https://doi.org/10.1016/j.camwa.2019.12.022. https://www.sciencedirect.com/science/article/pii/S0898122119305966.
3. Y. C. Fung, An Introduction to the Theory of Aeroelasticity, Courier Dover Publications, 2008.
4. M. Feistauer, J. Horáček, M. Ružička, P. Sváček, Numerical analysis of flow-induced nonlinear vibrations of an airfoil with three degrees of freedom, Computers & Fluids 49 (1) (2011) 110–127. https://doi.org/10.1016/j.compfluid.2011.05.004.

5. P. Sváček, M. Feistauer, J. Horáček, Numerical simulation of flow induced airfoil vibrations with large amplitudes, Journal of Fluids and Structures 23 (3) (2007) 391–411.
6. S. B. Pope, Turbulent Flows, Cambridge University Press, Cambridge, 2000.
7. F. Nobile, Numerical approximation of fluid-structure interaction problems with application to haemodynamics, Ph.D. thesis, Ecole Polytechnique Federale de Lausanne (2001).
8. F. R. Menter, Two-equations eddy-viscosity turbulence models for engineering applications, AIAA Journal 32 (8) (1994) 1598–1605.
9. F. Menter, R. Langtry, S. Völker, Transition modelling for general purpose CFD codes, Flow, Turbulence and Combustion 77 (1-4) (2006) 277–303.
10. J. C. Kok, Resolving the dependence on free-stream values for the k-omega turbulence model, Tech. rep., National Aerospace Laboratory NLR (1999).
11. D. C. Wilcox, Turbulence Modeling for CFD, DCW Industries, 1993.
12. R. B. Langtry, F. R. Menter, Correlation-based transition modeling for unstructured parallelized computational fluid dynamics codes, AIAA Journal 47 (12) (2009) 2894–2906.
13. R. Codina, A discontinuity capturing crosswind-dissipation for the finite element solution of the convection diffusion equation, Computational Methods in Applied Mechanical Engineering 110 (1993) 325–342.
14. G. Lube, G. Rapin, Residual-based stabilized higher-order FEM for advection-dominated problems, Computer Methods in Applied Mechanics and Engineering 195 (33-36) (2006) 4124–4138.
15. R. Codina, Stabilization of incompressibility and convection through orthogonal sub-scales in finite element methods, Computational Method in Applied Mechanical Engineering 190 (2000) 1579–1599.
16. P. Sváček, On implementation aspects of finite element method and its application, Advances in Computational Mathematics 45 (4) (2019) 2065–2081.

Second-Order Time Accuracy for Coupled Lumped and Distributed Fluid Flow Problems via Operator Splitting: A Numerical Investigation

Lucia Carichino, Giovanna Guidoboni, and Marcela Szopos

Abstract We develop a new second-order accurate operator splitting approach for the time discretization of coupled systems of partial and ordinary differential equations for fluid flows problems. The scheme is tested on a benchmark test case with an analytical solution; some of its main features, such as unconditional stability and second-order accuracy, are verified.

1 Introduction

Multiscale coupling of systems of partial and ordinary differential equations (PDEs and ODEs) is of interest when modeling fluid flow in complex hydraulic networks. To ensure physical consistency and well-posedness of the coupled problem, interface conditions enforcing continuity of mass and balance of forces are imposed [3, 6], which should also be preserved at the discrete level when solving the problem numerically. To this end, different monolithic and partitioned strategies have been proposed, where the PDE and ODE systems are solved simultaneously or in separate substeps, respectively, as reviewed in [7]. The main challenge of using splitting schemes arise from the fact that they often require sub-iterations between substeps in order to achieve convergence of the overall algorithm. However, the convergence of such sub-iterations might become an issue, especially in the case of

L. Carichino
Rochester Institute of Thechnology, School of Mathematical Sciences, Rochester, NY, USA
e-mail: lcsma1@rit.edu

G. Guidoboni
Department of Electrical Engineering and Computer Science, Department of Mathematics, University of Missouri, Columbia, MO, USA
e-mail: guidobonig@missouri.edu

M. Szopos (✉)
MAP5, UMR CNRS 8145, Université de Paris, Paris, France
e-mail: marcela.szopos@parisdescartes.fr

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_95

nonlinear problems. We developed in [1] a new technique based on operator splitting for the time discretization of PDE/ODE multiscale problems, that allows to solve separately and sequentially the Stokes problem and the ODEs without the need of sub-iterations. The scheme yielded, at most, first-order accuracy in time, since (i) the scheme included only two substeps and (ii) a first-order Backward Euler scheme was used in each substep. In the present contribution, we extend this approach to a novel second-order algorithm in time, and numerically investigate its stability and accuracy.

2 The Coupled PDE/ODE Problem Arising in Fluid Flow Modeling

To study the flow of a viscous fluid through a complex hydraulic network, we consider the case of a domain $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, where the fluid flow is described by the non-stationary Stokes equations. The domain Ω is connected *via* a resistor to a lumped circuit Υ, see Fig. 1. We assume that the boundary of Ω, denoted by $\partial\Omega$, is the union of three portions, namely $\partial\Omega = \Gamma \cup \Sigma \cup S$, where different types of boundary or interface conditions are imposed: Dirichlet boundary conditions on Γ, Neumann boundary conditions on Σ, and Stokes-circuit coupling conditions on S. For a given $T > 0$, the fully coupled problem consists in finding

- the d-dimensional velocity vector field $\boldsymbol{v}(\boldsymbol{x}, t)$ and the scalar pressure field $p(\boldsymbol{x}, t)$, for $(\boldsymbol{x}, t) \in \Omega \times (0, T)$;
- the pressure $P(t)$ at the node of the circuit sitting on S, and the flow rate $Q(t)$ through S, for $t \in (0, T)$;
- the l-dimensional vector of state variables $\boldsymbol{y}(t)$, that describes the dynamics in the lumped hydraulic circuit Υ, for $t \in (0, T)$,

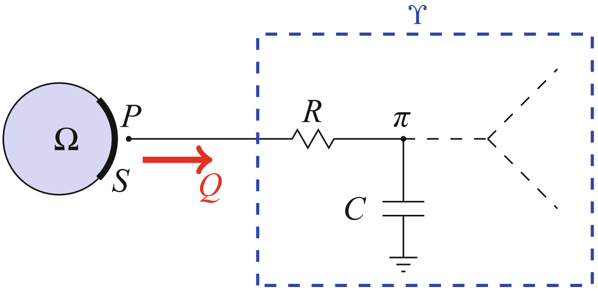


Fig. 1 Schematic representation of the coupling between a Stokes region Ω and a circuit Υ

that satisfy the following equations

$$\rho\frac{\partial \boldsymbol{v}}{\partial t} = -\nabla p + \mu\Delta\boldsymbol{v} + \rho\boldsymbol{f} \quad \text{and} \quad \nabla\cdot\boldsymbol{v} = 0 \qquad \text{in } \Omega\times(0,T), \tag{1}$$

$$\frac{d\boldsymbol{y}}{dt} = \underline{\underline{\boldsymbol{A}}}(\boldsymbol{y},t)\boldsymbol{y} + \boldsymbol{s}(\boldsymbol{y},t) + \boldsymbol{b}(Q,P) \qquad \text{in } (0,T), \tag{2}$$

where ρ and μ represent the fluid density and dynamic viscosity, respectively, $\boldsymbol{f}$ is a d-dimensional vector-valued function representing the given body forces per unit of mass, $\underline{\underline{\boldsymbol{A}}}$ is a $l\times l$ tensor (possibly non-linear) that embodies topology and physics of the connections among the circuit nodes, $\boldsymbol{s}$ and $\boldsymbol{b}$ are l-dimensional vector-valued functions, with $\boldsymbol{s}$ accounting for sources and sinks within the circuit, and $\boldsymbol{b}$ accounting for the Stokes-circuit connection. Here, we focus on lumped circuits possibly involving resistive, capacitive and inductive elements, also known as RCL circuits. As a consequence of the hydraulic analog of Kirchoff laws of currents and voltages, electrical state variables $\boldsymbol{y}$ would be pressure, pressure difference, volume, volumetric flow rate or linear momentum flux. The resistive connection between Ω and Υ, in addition to the capacitive connection to the ground at the circuit side of the resistor, allows us to adopt pressures as state variables at both ends.

System (1) and (2) is equipped with the *initial conditions*

$$\boldsymbol{v}(\boldsymbol{x},0) = \boldsymbol{v}_0(\boldsymbol{x}) \ \text{ in } \ \Omega, \quad \text{and} \quad \boldsymbol{y}(t=0) = \boldsymbol{y}_0, \tag{3}$$

and the following *boundary and interface conditions*

$$\boldsymbol{v} = \boldsymbol{0} \qquad \text{on } \Gamma\times(0,T), \tag{4}$$

$$\Big(-p\underline{\underline{\boldsymbol{I}}} + \mu\nabla\boldsymbol{v}\Big)\boldsymbol{n}_\Sigma = -\overline{p}\boldsymbol{n}_\Sigma \qquad \text{on } \Sigma\times(0,T), \tag{5}$$

$$\Big(-p\underline{\underline{\boldsymbol{I}}} + \mu\nabla\boldsymbol{v}\Big)\boldsymbol{n}_S = -P\boldsymbol{n}_S \qquad \text{on } S\times(0,T), \tag{6}$$

$$Q(t) = \int_S \boldsymbol{v}(\boldsymbol{x},t)\cdot\boldsymbol{n}_S(\boldsymbol{x})dS \qquad \text{for } t\in(0,T), \tag{7}$$

where $\underline{\underline{\boldsymbol{I}}}$ is the $d\times d$ identity tensor, $\boldsymbol{n}_\Sigma$ is the outward unit normal vector to Σ, $\overline{p} = \overline{p}(\overline{t})$ is a given function of time, and $\boldsymbol{n}_S$ is the outward unit normal vector to S.

Remark 1 According to the complexity of the hydraulic network under consideration, the geometrical architecture of the coupled problem used in this study can be expanded to account for several distributed domains coupled with several lumped circuits, with possible multiple connections among them, as described in [1].

3 A Second Order Operator Splitting Algorithm

In this section, we introduce a novel second-order algorithm for the semi-discretization in time of the coupled problem described in Sect. 2. We adopt a Strang's symmetrized splitting approach [4, Sec. 30.3] to design an algorithm that allows us to solve sequentially, in separate substeps, the PDE system associated with the Stokes region and the ODE system associated with the lumped hydraulic circuit. The main rationale for our splitting scheme is that we aim to preserve at the discrete level the physical energy balance derived at the continuous level, as in [1], for the case of resistive Stokes-circuit connections. As a result, unconditional stability with respect to the time step choice is obtained without the need of sub-iterations between PDE and ODE substeps.

Let Δt denote a fixed global time step, let $t^n = n\Delta t$ and let $\varphi^n = \varphi(t^n)$ for any general expression φ. Let $\boldsymbol{v}^0 = \boldsymbol{v}_0$ and $\boldsymbol{y}^0 = \boldsymbol{y}_0$. Then, the algorithm proceeds as follows: for any $n \geq 0$ solve

Step 1 Given the initial conditions $\boldsymbol{v}(\boldsymbol{x}, t^n) = \boldsymbol{v}^n(\boldsymbol{x})$ in Ω and $\boldsymbol{y}(t^n) = \boldsymbol{y}^n$, find $\boldsymbol{v}(\boldsymbol{x}, t)$ and $\boldsymbol{y}(t)$ such that

$$\rho\frac{\partial \boldsymbol{v}}{\partial t} = \boldsymbol{0} \qquad \text{in } \Omega \times (t^n, t^{n+\frac{1}{2}}), \tag{8}$$

$$\frac{d\boldsymbol{y}}{dt} = \underline{\underline{\boldsymbol{A}}}(\boldsymbol{y}, t)\,\boldsymbol{y} + \boldsymbol{s}(\boldsymbol{y}, t) \qquad \text{in } (t^n, t^{n+\frac{1}{2}}), \tag{9}$$

and set $\boldsymbol{v}^{n+\frac{1}{2}} = \boldsymbol{v}(\boldsymbol{x}, t^{n+\frac{1}{2}})$ and $\boldsymbol{y}^{n+\frac{1}{2}} = \boldsymbol{y}(t^{n+\frac{1}{2}})$.

Step 2 Given the initial conditions $\boldsymbol{v}(\boldsymbol{x}, 0) = \boldsymbol{v}^{n+\frac{1}{2}}(\boldsymbol{x})$ in Ω and $\boldsymbol{y}(0) = \boldsymbol{y}^{n+\frac{1}{2}}$, find $\boldsymbol{v}(\boldsymbol{x}, t)$ and $\boldsymbol{y}(t)$ such that

$$\nabla \cdot \boldsymbol{v} = 0 \qquad \text{in } \Omega \times (0, \Delta t), \tag{10}$$

$$\rho\frac{\partial \boldsymbol{v}}{\partial t} = -\nabla p + \mu\Delta\boldsymbol{v} + \rho\boldsymbol{f}(t^{n+\frac{1}{2}}) \qquad \text{in } \Omega \times (0, \Delta t), \tag{11}$$

$$\frac{d\boldsymbol{y}}{dt} = \boldsymbol{b}(Q, P) \qquad \text{in } (0, \Delta t), \tag{12}$$

$$\boldsymbol{v} = \boldsymbol{0} \qquad \text{on } \Gamma \times (0, \Delta t), \tag{13}$$

$$\Big(-p\underline{\underline{\boldsymbol{I}}} + \mu\nabla\boldsymbol{v}\Big)\boldsymbol{n}_\Sigma = -\overline{p}(t^{n+\frac{1}{2}})\boldsymbol{n}_\Sigma \qquad \text{on } \Sigma \times (0, \Delta t), \tag{14}$$

$$\Big(-p_l\underline{\underline{\boldsymbol{I}}} + \mu\nabla\boldsymbol{v}\Big)\boldsymbol{n}_S = -P\boldsymbol{n}_S \qquad \text{on } S \times (0, \Delta t), \tag{15}$$

$$\int_S \boldsymbol{v}(\boldsymbol{x}, t)\cdot\boldsymbol{n}_S(\boldsymbol{x})\,dS = Q(t) \qquad \text{in } (0, \Delta t), \tag{16}$$

and set $\hat{\boldsymbol{v}}^{n+\frac{1}{2}} = \boldsymbol{v}(\boldsymbol{x}, \Delta t)$, $p^{n+1} = p(\boldsymbol{x}, \Delta t)$ and $\hat{\boldsymbol{y}}^{n+\frac{1}{2}} = \boldsymbol{y}(\Delta t)$.

Step 3 Given the initial conditions $\boldsymbol{v}(\boldsymbol{x}, t^{n+\frac{1}{2}}) = \hat{\boldsymbol{v}}^{n+\frac{1}{2}}(\boldsymbol{x})$ in Ω and $\boldsymbol{y}(t^{n+\frac{1}{2}}) = \hat{\boldsymbol{y}}^{n+\frac{1}{2}}$, find $\boldsymbol{v}(\boldsymbol{x}, t)$ and $\boldsymbol{y}(t)$ such that

$$\rho \frac{\partial \boldsymbol{v}}{\partial t} = \boldsymbol{0} \qquad \text{in } \Omega \times (t^{n+\frac{1}{2}}, t^{n+1}), \tag{17}$$

$$\frac{d\boldsymbol{y}}{dt} = \underline{\underline{\boldsymbol{A}}}(\boldsymbol{y}, t)\, \boldsymbol{y} + \boldsymbol{s}(\boldsymbol{y}, t) \qquad \text{in } (t^{n+\frac{1}{2}}, t^{n+1}), \tag{18}$$

and set $\boldsymbol{v}^{n+1} = \boldsymbol{v}(\boldsymbol{x}, t^{n+1})$ and $\boldsymbol{y}^{n+1} = \boldsymbol{y}(t^{n+1})$.

Note that Eqs. (8) and (17) imply that the velocity vector $\boldsymbol{v}$ is actually not updated in Steps 1 and 3, and therefore $\boldsymbol{v}^{n+\frac{1}{2}} = \boldsymbol{v}^n$ and $\boldsymbol{v}^{n+1} = \hat{\boldsymbol{v}}^{n+\frac{1}{2}}$.

In the construction of the above splitting scheme, special care was taken in selecting the order in which the PDE and ODE systems are resolved. The ODE system, except the coupling term $\boldsymbol{b}$, is solved twice, see Eq. (9) in Step 1 and Eq. (18) in Step 3, whereas the PDE system implicitly coupled to a subset of the ODE system is solved only once, see Eqs. (10)–(16) in Step 2. This choice allows us to save computational time, solving the PDE system only once, and to have more freedom in the discretization of the ODE system. Note that each Step is defined on a discrete time interval, but the differential operators have yet to be fully discretized in time and space. To preserve the second-order accuracy of the overall algorithm, it is necessary to discretize each substep with (at least) second-order accurate schemes. We implemented a second-order BDF2 scheme for the time-discretization of Steps 1 and 3. The BDF2 algorithm can be written for a general initial value problem

$$\frac{d\varphi}{dt} + A(\varphi, t) = f(t), \quad \varphi(0) = \varphi_0 \tag{19}$$

under the form: given $\varphi^0 = \varphi_0$ and φ^1, then for $n \geq 1$, find φ^{n+1} by solving

$$\frac{\frac{3}{2}\varphi^{n+1} - 2\varphi^n + \frac{1}{2}\varphi^{n-1}}{\Delta t} + A(\varphi^{n+1}, t^{n+1}) = f^{n+1}. \tag{20}$$

Note that, due to the structure of the splitting and of the BDF2 algorithm, a starting step is necessary to compute φ^1 from φ^0. In order to obtain this value, we use one iteration of the following θ-scheme: given $\varphi^0 = \varphi_0$, for $n \geq 1$, find φ^{n+1} by solving

$$\frac{\varphi^{n+1} - \varphi^n}{\Delta t} + \theta A(\varphi^{n+1}, t^{n+1}) + (1-\theta)A(\varphi^n, t^n) = \theta f^{n+1} + (1-\theta) f^n, \tag{21}$$

where $\theta = 2/3$, which is "almost" second-order accurate, see [4, Chap.2]. The value of θ is chosen to preserve the matrix of the linear system solved in (20) and (21).

In Step 2, solved on the discrete time interval $(0, \Delta t)$, we evaluate the performance of two different strategies:

Strategy I: we compute one global time step Δt using the θ-scheme (21);
Strategy II: we use a time-step smaller than Δt, performing at least one iteration of the BDF2 scheme (20).

Since Step 2 consists in solving the Stokes problem implicitly coupled with a subset of the ODE system, it clearly appears that Strategy II is more computational expensive, but it is designed in order to achieve higher accuracy, as discussed later in Sect. 4.

4 Numerical Results

We illustrate the performances of the novel splitting algorithm proposed in Eqs. (8)–(18), by studying the test case represented in Fig. 2, for which detailed description and analytical solution are reported in [1]. In particular, the two-dimensional ($d = 2$) Stokes region Ω, defined as the rectangle $(0, L) \times (-H/2, H/2)$, with $H, L > 0$ given, is connected to the lumped circuit Υ, described by the vector of state variables $\boldsymbol{y} = [\pi, \omega]^T$ whose dimension is $l = 2$.

The global time step Δt is determined by the number of intervals in each time period N_τ, according to the formula $\Delta t = \tau/N_\tau$, where τ is the period of the exact solution [1]. Moreover, each substep of the algorithm is discretized with a different time step, denoted Δt_1, Δt_2, and Δt_3 respectively. In order to check that the computed numerical solution is periodic of period τ, we introduce the index k to denote the k-th period of the simulation, for $k \geq 1$. Then, t^n is in the k-th period of the simulation if $n \in \mathcal{I}_k = \{(k-1)N_\tau + 1, (k-1)N_\tau + 2, \ldots, kN_\tau\}$. We use the following criterion for $k \geq 2$:

$$\max\left\{ \max_{n\in\mathcal{I}_k} \frac{\left\|\boldsymbol{v}^n - \boldsymbol{v}^{n-N_\tau}\right\|^2_{L^2(\Omega)}}{\left\|\boldsymbol{v}^{n-N_\tau}\right\|^2_{L^2(\Omega)}},\ \max_{n\in\mathcal{I}_k} \frac{\left\|p^n - p^{n-N_\tau}\right\|^2_{L^2(\Omega)}}{\left\|p^{n-N_\tau}\right\|^2_{L^2(\Omega)}},\ \max_{n\in\mathcal{I}_k} \frac{\left\|\boldsymbol{y}^n - \boldsymbol{y}^{n-N_\tau}\right\|^2}{\left\|\boldsymbol{y}^{n-N_\tau}\right\|^2} \right\} < \varepsilon, \tag{22}$$

to identify the numerical quantities to be compared with the exact solution over one time period. The results reported have been obtained for $\varepsilon = 10^{-6}$.

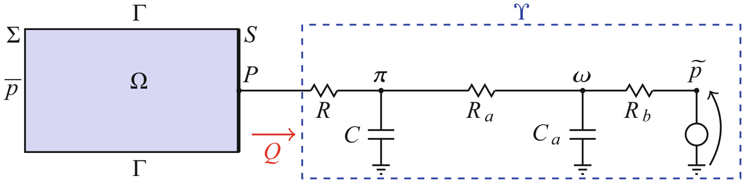


Fig. 2 The Stokes region Ω is connected to the lumped circuit Υ via a resistive element *R*

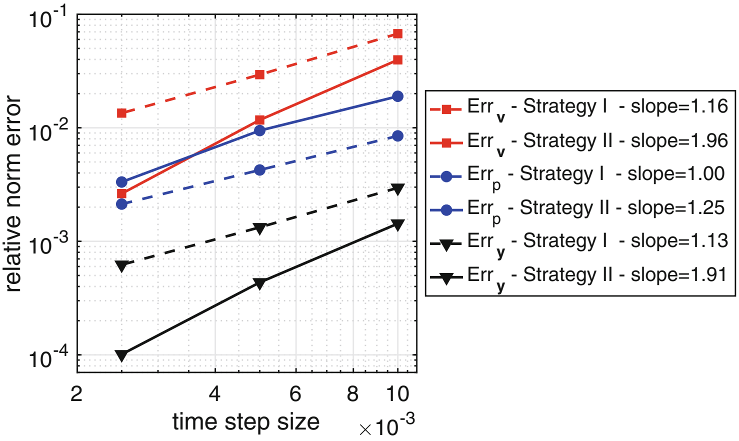


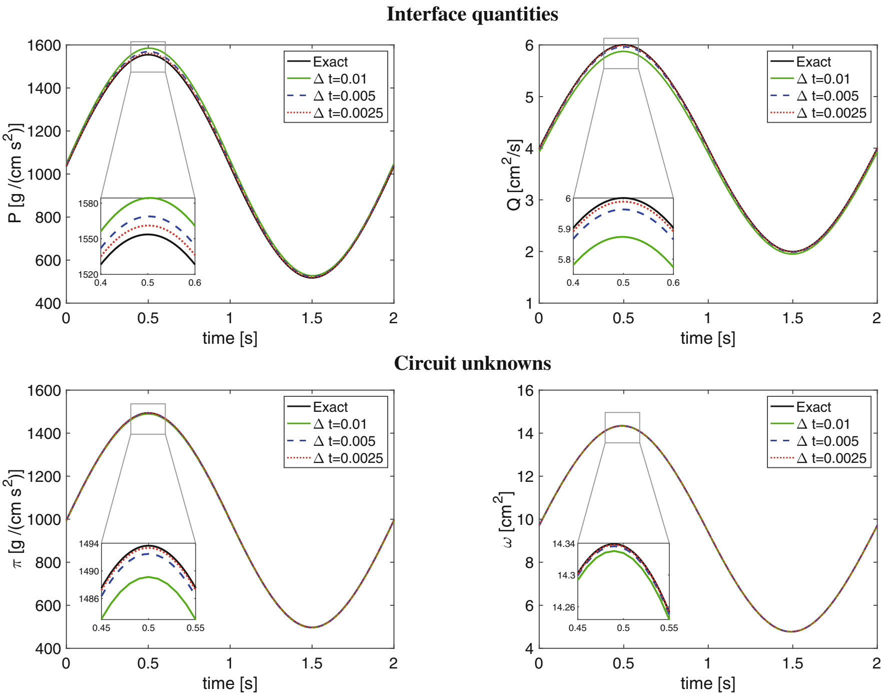
Fig. 3 Plot of the energy norm errors [1] in logarithmic scale as a function of the global time step $\Delta t = 0.01,\ 0.005,\ 0.0025$ for the numerical example considered

The spatial discretization of the Stokes problem is handled via a triangular uniform mesh of 4000 elements for Ω and an inf-sup stable finite element pair, namely (Taylor-Hood) $\mathbb{P}^2/\mathbb{P}^1$ elements. The computational framework relies on the finite element library Freefem [2]. The comparison between numerical approximations and exact solutions is performed over the first time period that satisfies Eq. (22), for three different global time steps $\Delta t = 0.01, 0.005, 0.0025$. Results are obtained with the following time discretization of Steps 1, 2 and 3.

Strategy I: $\Delta t_1 = \Delta t_3 = \Delta t/5$ and $\Delta t_2 = \Delta t$;
Strategy II: $\Delta t_1 = \Delta t_2 = \Delta t_3 = \Delta t/5$.

First, we perform a standard time refinement study, shown in Fig. 3, comparing Strategy I and II to assess if the expected second-order convergence in time is achieved. The rates predicted for velocity $\boldsymbol{v}$ and unknowns in the circuit $\boldsymbol{y}$ by the theory for a second-order operator splitting technique are obtained in Strategy II. In contrast, Strategy I, even if less computational expensive, displays an order of convergence less than $3/2$. This demonstrates the importance of using a scheme that is (at least) second-order accurate in each substep of the second-order overall splitting scheme. As expected, the approximation of the pressure does not achieve second order in both strategies, due to the lack of time derivative of p and of post-processing steps, see also [4, Sec.31.4]. Note that for a given value of Δt, the results for $\boldsymbol{v}$ and $\boldsymbol{y}$ obtained with Strategy I are less accurate than the ones of Strategy II.

Figure 4 displays a comparison between the exact solution and numerical approximations of P and Q at the Stokes-circuit interface S (upper panel), and of π and ω in the circuit (lower panel), all obtained with Strategy II. The numerical results and exact solution almost superimpose for π and ω, even for $\Delta t = 0.01$. The approximation of the interface quantities, P and Q, improves as Δt decreases, capturing periodicity and peaks. Figure 4 also shows that the numerical solution is


Interface quantities

Circuit unknowns

Fig. 4 Comparison between the exact solution and the corresponding numerical approximation for interface quantities and circuit unknowns, for the time steps Δt considered, over one time period

not affected by spurious oscillations or instabilities, even for the largest time step. These findings confirm that the choice of the time step affects the accuracy of the computed solution but not the stability of the numerical scheme, thereby supporting the unconditional stability properties of the algorithm.

5 Conclusions and Outlook

We have presented a numerical investigation of a new splitting approach for the time discretization of a coupled Stokes/ODEs system. Our results suggest that the proposed algorithm is unconditional stable without the need of sub-iterations between substeps, and it is second-order accurate in time, provided that a stable and (at least) second-order accurate time-discretization scheme is used in each substep. In particular, we compared two strategies based on the BDF2 and θ-schemes, for $\theta = 2/3$, and found that the BDF2 method should be preferred in order to obtain optimal convergence behavior in time, despite the extra computational cost. The modular structure of the method allows us to maintain some flexibility in choosing the numerical method for the solution of each sub-problem; therefore, as an alternative, a different θ-scheme, with $\theta = 1/2$, could be implemented. We are also currently exploring the possibility of using the second-order (modified) θ-scheme [5] in some of the substeps.

Acknowledgments This work has been published under the framework of the IdEx Unistra and partially benefited from a funding from the state managed by the French National Research Agency as part of the Investments for the future program.

References

1. Carichino, L., Guidoboni, G., Szopos, M. Energy-based operator splitting approach for the time discretization of coupled systems of partial and ordinary differential equations for fluid flows: the Stokes case. J. Comput. Phys., **364**, 235–256 (2018).
2. Hecht, F. New development in FreeFem++. J. Numer. Math., **20**, 251–266 (2012).
3. Heywood J. G., Rannacher R., Turek S. Artificial boundaries and flux and pressure conditions for the incompressible Navier-Stokes equations, Int. J. Numer. Methods Fluids **22**, 325–352, (1996).
4. Glowinski, R. Finite Element methods for incompressible viscous flow. In: Handbook of Numerical Analysis, Vol. IX, P.G. Ciarlet & J.L. Lions (eds.) pp. 3–1176, North-Holland, Amsterdam (2003).
5. Turek, S., Rivkind, L., Hron, J., Glowinski, R. Numerical study of a modified time-stepping θ-scheme for incompressible flow simulations. J. Sci. Comp., **28**, 533–547 (2006).
6. Quarteroni A., Veneziani A. Analysis of a geometrical multiscale model based on the coupling of ODE and PDE for blood flow simulations. SIAM J. on Multiscale Modeling & Simulation **1**, 173–195, (2003).
7. Quarteroni A., Veneziani A., Vergara C. Geometric multiscale modeling of the cardiovascular system, between theory and practice, Comp. Meth. Appl. Mech. and Engng. **302**, 193–252, (2016).

A CSCM Approximation of Steady MHD Flow and Heat Transfer Between Parallel Plates with Hydrodynamic Slip and Convective Boundary Conditions

Münevver Tezer-Sezgin and Önder Türk

Abstract The steady magnetohydrodynamic (MHD) flow and heat transfer between parallel plates is considered in which the electrically conducting fluid has temperature dependent properties such as viscosity, thermal and electrical conductivity. The fluid is driven by a constant pressure gradient, and a uniform external transverse magnetic field is applied perpendicular to the plates. The effects of viscous and Joule dissipations are considered in the energy equation, and the fluid is assumed to be slipping in the vicinity of the plates. The effects of the magnetic field, the hydrodynamic slip, and convective thermal boundary conditions on the flow and heat transfer are investigated as well as the temperature dependent parameters. The Chebyshev spectral collocation method which is easy to implement is presented for the approximation of the solutions to the governing equations. The velocity and the temperature of the fluid are obtained with a cheap computational expense.

1 Introduction

The magnetohydrodynamic flow and heat transfer of a viscous, electrically conducting, incompressible fluid between parallel plates has important industrial applications. Among them are MHD generators and accelerators, fluidization, centrifugal separation of matter from fluid, purification of crude oil, nuclear reactors, and blood flow in capillaries.

Alpher [1] considered thermally developed MHD flow between parallel plates by adopting small temperature differences and constant viscosity parameter. Numerical

M. Tezer-Sezgin (✉)
Department of Mathematics, Middle East Technical University, Ankara, Turkey
e-mail: munt@metu.edu.tr

Önder Türk
Department of Mathematics, Gebze Technical University, Kocaeli, Turkey
e-mail: onder.turk@yandex.com

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_96

solutions using finite difference method for the steady MHD heat transfer between parallel plates are given in [5] by Attia and Kotb, and in [3] by Attia considering time-dependent viscosity. Attia extended the analysis in the former to the unsteady MHD flow and heat transfer in [2] for capturing more accurate behavior of the flow and heat transfer. A hybrid solution technique (generalized integral transform) has been used by Lima et al. [8] for the MHD flow and heat transfer of a Newtonian fluid in parallel-plates channels. In their work, both the stationary plates and moving upper plate cases, and the inflow/outflow through plates are considered with the assumption of variable viscosity. The effect of variable properties on the unsteady Couette flow with heat transfer under a magnetic field is given in [4] by taking into account two components of the velocity field, but the variation is in one dimension.

The MHD flow in an insulating duct has been considered with the extension of velocity boundary condition to the hydrodynamic slip at the interface between the fluid and the solid wall in the work of Smolentsev [12]. In the works of Sweet et al. [13] and Maikap [9], the MHD flow of a viscous fluid between moving parallel plates and plates with smooth expansion are given, respectively. In these papers, the homotopy analysis and finite difference methods have been used, respectively. Ibáñez obtained analytical solutions to the equations govern the steady flow of an incompressible electrically conducting fluid through a channel with permeable plates that are accompanied by hydrodynamic slip conditions and thermal boundary conditions of the third kind [7].

In this work, we present a numerical solution of steady MHD flow and heat transfer between parallel plates using the Chebyshev spectral collocation method (CSCM). The electrically conducting fluid has temperature dependent viscosity, thermal and electrical conductivity and the hydrodynamic slip conditions are imposed on the velocity of the fluid, and convective boundary conditions are taken for the temperature. In the physical model, the Joule effect, that is, the generation of heat by the passage of electricity through a resistance, is taken into account in the energy equation. Also viscous dissipation is considered in the energy equation, thus, the viscosity of the fluid will take energy from the motion of the fluid (kinetic energy) and transform it into internal energy of the fluid, increasing the heat of the fluid. The CSCM allows one to be able to use considerably small number of clustered Chebyshev–Gauss–Lobatto points near the boundaries and to approximate higher order derivatives of the solution by using Chebyshev differentiation matrices. The velocity and temperature of the fluid are simulated for several values of Hartmann number, Prandtl number, viscosity, thermal conductivity, and electric conductivity parameters, slip lengths, and Biot numbers, for depicting the influences on the flow and temperature.

2 Basic Equations for the Unsteady MHD Flow and Heat Transfer Between Parallel Plates

The fluid is assumed to flow between two infinitely parallel plates located horizontally at $y = -1$ and $y = 1$. A constant pressure gradient is applied in the x-direction. A uniform magnetic field of intensity B_0 is applied in the y-direction. The induced magnetic field is neglected due to the assumption of small magnetic Reynolds number. The governing equations in non-dimensional form are [3]

$$P + f_1(T)\frac{\mathrm{d}^2u}{\mathrm{d}y^2} + \frac{\mathrm{d}f_1(T)}{\mathrm{d}y}\frac{\mathrm{d}u}{\mathrm{d}y} - f_3(T)Ha^2u = 0,$$

$$\frac{1}{Pr}f_2(T)\frac{\mathrm{d}^2T}{\mathrm{d}y^2} + \frac{1}{Pr}\frac{\mathrm{d}f_2(T)}{\mathrm{d}y}\frac{\mathrm{d}T}{\mathrm{d}y} + Ecf_1(T)\left(\frac{\mathrm{d}u}{\mathrm{d}y}\right)^2 + f_3(T)Ec\,Ha^2\,u^2 = 0, \tag{1}$$

where $-1 \leq y \leq 1$. In these equations, u and T denote the velocity and temperature of the fluid, respectively. $f_1(T) = e^{-aT}$ represents the exponentially varying viscosity of the fluid, a being the viscosity parameter. $f_2(T) = 1+bT$ and $f_3(T) = 1 + cT$ are the temperature dependent thermal and electric conductivity functions, respectively, where b and c are the respective thermal and electric conductivity parameters. The third and the fourth terms in the energy equation are due to the viscous and Joule dissipations, respectively. The heat is exchanged by convection between the lower and upper plates when they are kept at constant temperatures T_1 and T_2, respectively, with $T_1 < T_2$.

The following dimensionless variables are employed in the above equations

$$P = -\frac{\mathrm{d}p}{\mathrm{d}x}, \quad Pr = \frac{\mu_0 c_p}{k_0}, \quad Ha = B_0h\sqrt{\sigma_0/\mu_0}, \quad Ec = \frac{\mu_0^2}{h^2c_p\rho^2(T_2 - T_1)}. \tag{2}$$

Here, p is the pressure, ρ is the density of the fluid, and c_p is the specific heat at constant pressure. μ_0, k_0 and σ_0 are the viscosity, thermal conductivity and the electrical conductivity of the fluid at a reference temperature, e.g. at $T = T_1$, respectively. Pr, Ha, and Ec are the Prandtl, Hartmann, and Eckert numbers, respectively.

The conventional boundary condition is the no-slip condition between the fluid and a solid where the fluid velocity is taken as the same as of the boundaries. However, the boundary condition which corresponds to momentum transfer during the flow can vary from stick (no-slip) to slip, saving energy in response to physical chemical properties of the solid surface. The velocity of the fluid is completely zero on a solid boundary only if thermodynamic equilibrium is ensured near the boundary. For small scale systems, on the other hand, the collisions between the fluid and the solid surface is not high enough to have this equilibrium. Thus, tangential velocity slip is allowed. The slip velocity is assumed to be proportional to the tangential viscous stress [10]. Moreover, the thermal boundary condition of the third kind at each wall, referred as convective heat transfer condition, is considered in the present study. In this type of boundary condition, the heat flux to or from the surface is assumed to be related to the surface and fluid temperature difference.

Under these assumptions, the equations in System (1) are accompanied with the following Robin's type boundary conditions

$$\begin{aligned} &u+\alpha_1\frac{\mathrm{d}u}{\mathrm{d}y}=0 \ \text{ on } \ y=1, \quad u-\alpha_2\frac{\mathrm{d}u}{\mathrm{d}y}=0 \ \text{ on } \ y=-1, \\ &Bi_2(T-1)+\frac{\mathrm{d}T}{\mathrm{d}y}=0 \ \text{ on } \ y=1, \quad -Bi_1T+\frac{\mathrm{d}T}{\mathrm{d}y}=0 \ \text{ on } \ y=-1. \end{aligned} \tag{3}$$

In the above equations, Bi_j, $j = 1$ (lower) and $j = 2$ (upper), is the Biot number reflecting the ratio of the convective heat transfer resistance β_1, β_2 of the plates to conduction resistance of the fluid as $Bi_j = h\beta_j/k_0$. α_1 and α_2 are the (non-dimensional) slip lengths of the upper and lower plates, respectively.

3 Application of the CSCM to MHD and Heat Transfer Equations

The discretization of the governing equations (1) is mainly based on requiring the residual to be zero at the extreme points of the Chebyshev polynomials. The interpolating polynomials are differentiated analytically, and a differentiation matrix, known as Chebyshev differentiation matrix, is constructed for derivative approximations. The higher order derivatives are obtained by multiplying these differentiation matrices [6, 11, 14]. The collocation points are the Chebyshev–Gauss–Lobatto points defined as $x_j = \cos(j\pi/N)$, $j = 0, 1, \ldots, N$, N being a positive integer. The first and second order differentiation matrices $D_N^{(1)}$ and $D_N^{(2)}$, respectively, are constructed on these points, and the discretized equations are obtained by substituting the approximations u_N and T_N to u and T, respectively, to the equations (1). These equations are written in matrix-vector form as follows

$$\begin{aligned} &\left[\mathscr{D}(f_{1N})D_N^{(2)} - a\mathscr{D}(f_{1N})\mathscr{D}(D_N^{(1)}T_N)D_N^{(1)} - Ha^2\mathscr{D}(f_{3N})\right]u_N = -P_N, \\ &\left[\frac{1}{Pr}\mathscr{D}(f_{2N})D_N^{(2)} + \frac{b}{Pr}\mathscr{D}(D_N^{(1)}T_N)D_N^{(1)}\right]T_N = -Ec\mathscr{D}(\mathscr{D}(f_{1N})D_N^{(1)}u_N)D_N^{(1)}u_N \\ &-EcHa^2\mathscr{D}(f_{3N})\mathscr{D}(u_N)u_N, \end{aligned} \tag{4}$$

where $\mathscr{D}(\varphi)$ denotes the diagonal matrix with the entries of a vector φ on its diagonal. ${f_i}_N$, $i = 1, 2, 3$, denotes the vector computed as ${f_i}_N(x_j) = f_i(x_j)$, for $j = 1, \ldots, N + 1$. Similarly, P_N is the vector whose all entries are (the constant) P. Each equation given in (4) is of order $N + 1$.

System (4) is composed of two coupled and nonlinear equations, where the nonlinearity is inherited from System (1). In order to solve the resulting equations in (4), an iterative method is introduced which reduces the equations into a set of linear algebraic equations in each iteration. The algebraic equations are solved by imposing the corresponding boundary conditions. The iterative procedure starts with a given initial estimate for the temperature. This allows the solution of the first equation in (4). The second equation is solved next, with the use of newly obtained values. These steps are repeated until the convergence criteria $\|{u_N}^k - {u_N}^{k-1}\|_\infty \le \varepsilon$ and $\|{T_N}^k - {T_N}^{k-1}\|_\infty \le \varepsilon$ are met for a given tolerance ε, where the superscript k denotes the iteration level.

4 Numerical Results

The velocity and the temperature behaviors of the fluid for the MHD flow between parallel plates are presented for several values of the problem parameters as a, b, c, α_1, α_2, Bi_1, Bi_2, and Ha. In the computations, $N = 12$ is set for all values of these parameters, and the convergence tolerance is taken as $\varepsilon = 10^{-8}$.

In the first test problem, $u = T = 0$ is specified at the lower plate, whereas $u = 0$ and $T = 1$ is taken at the upper plate, and the values $Pr = 1$, $Ec = 0.2$, and $Ha = 3$ are set in the simulations as in [3] in order to validate the present computational method. The results are tabulated in terms of the center line temperature values in Table 1 for $b = 0$ and $-0.5 \le a, c \le 0.5$. An accuracy of 10^{-3} is obtained when compared with the results of [3].

The rest of this section is devoted to the illustration of the numerical results obtained from the approximation of equations (1) accompanied with the most general boundary conditions (3).

The variations of u and T with various values of c and Ha are investigated and simulated in Figs. 1, 2, and 3, respectively, for $a = -0.5$, $a = 0$, and $a = 0.5$, when $b = 0$ is taken. As the electric conductivity parameter c increases, the velocity

Table 1 Variation of the temperature at $y = 0$, $Ha = 3$, and $b = 0$

	$a = -0.5$	$a = -0.1$	$a = 0.0$	$a = 0.1$	$a = 0.5$
$c = -0.5$	0.7190	0.7520	0.7606	0.7692	0.8043
$c = -0.1$	0.6842	0.7023	0.7066	0.7107	0.7259
$c = 0.0$	0.6774	0.6934	0.6971	0.7007	0.7138
$c = 0.1$	0.6712	0.6855	0.6887	0.6919	0.7032
$c = 0.5$	0.6506	0.6602	0.6624	0.6644	0.6716

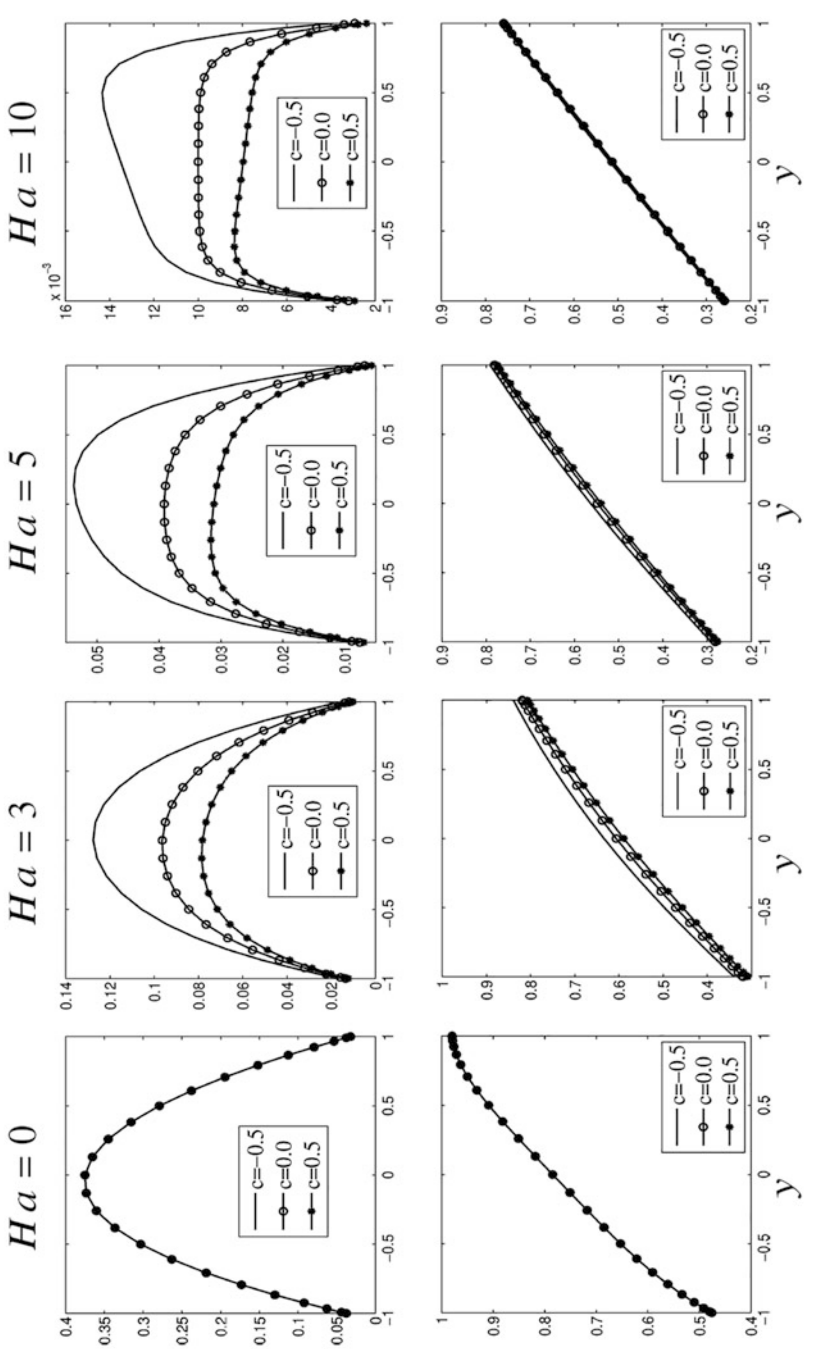


Fig. 1 Variation of u and T with various values of c and Ha, for $Pr = 1$, $Ec = 1$, $Bi_1 = 1$, $Bi_2 = 1$, $\alpha_1 = 0.05$, $\alpha_2 = 0.05$, $a = -0.5$, $b = 0$

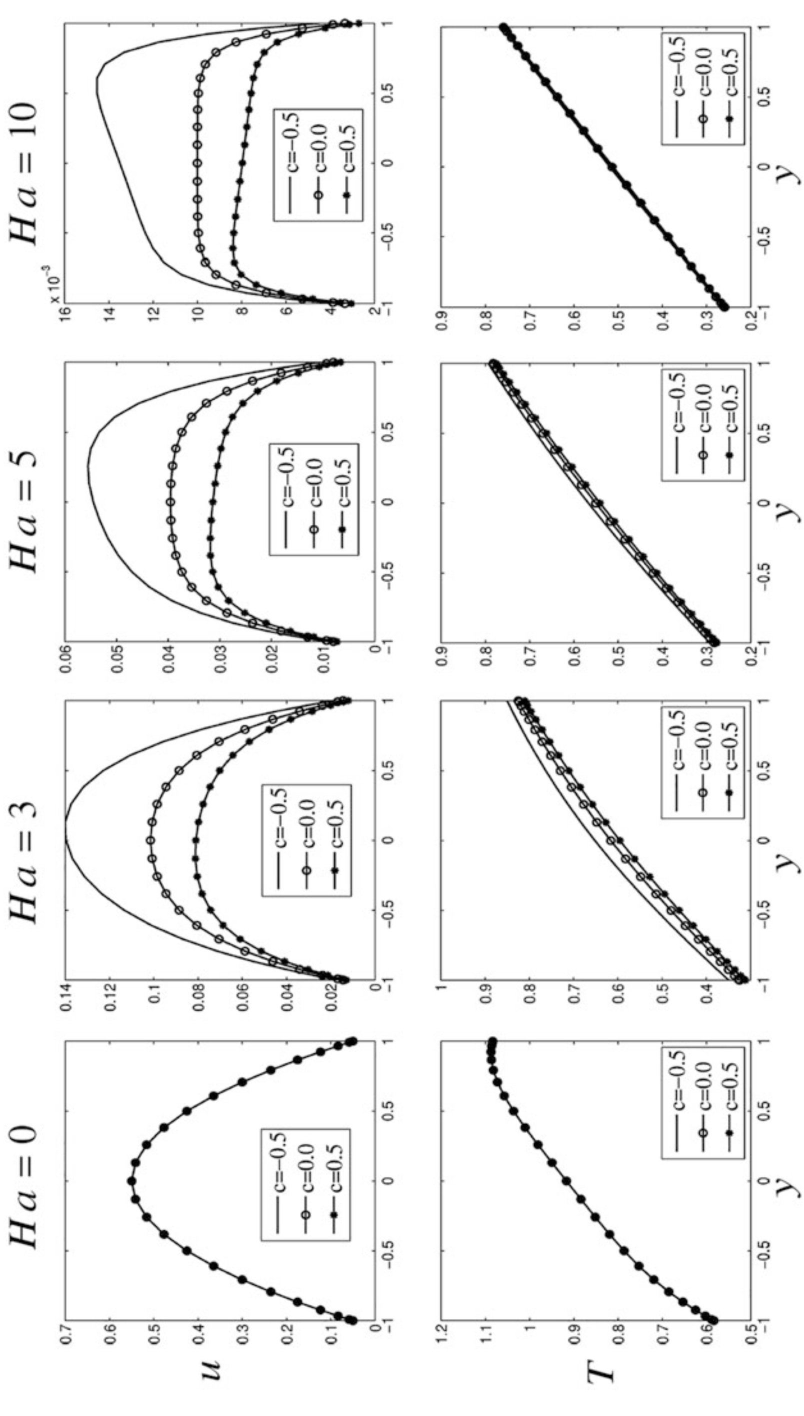


Fig. 2 Variation of u and T with various values of c and Ha, for $Pr = 1$, $Ec = 1$, $Bi_1 = 1$, $Bi_2 = 1$, $\alpha_1 = 0.05$, $\alpha_2 = 0.05$, $a = 0$, $b = 0$

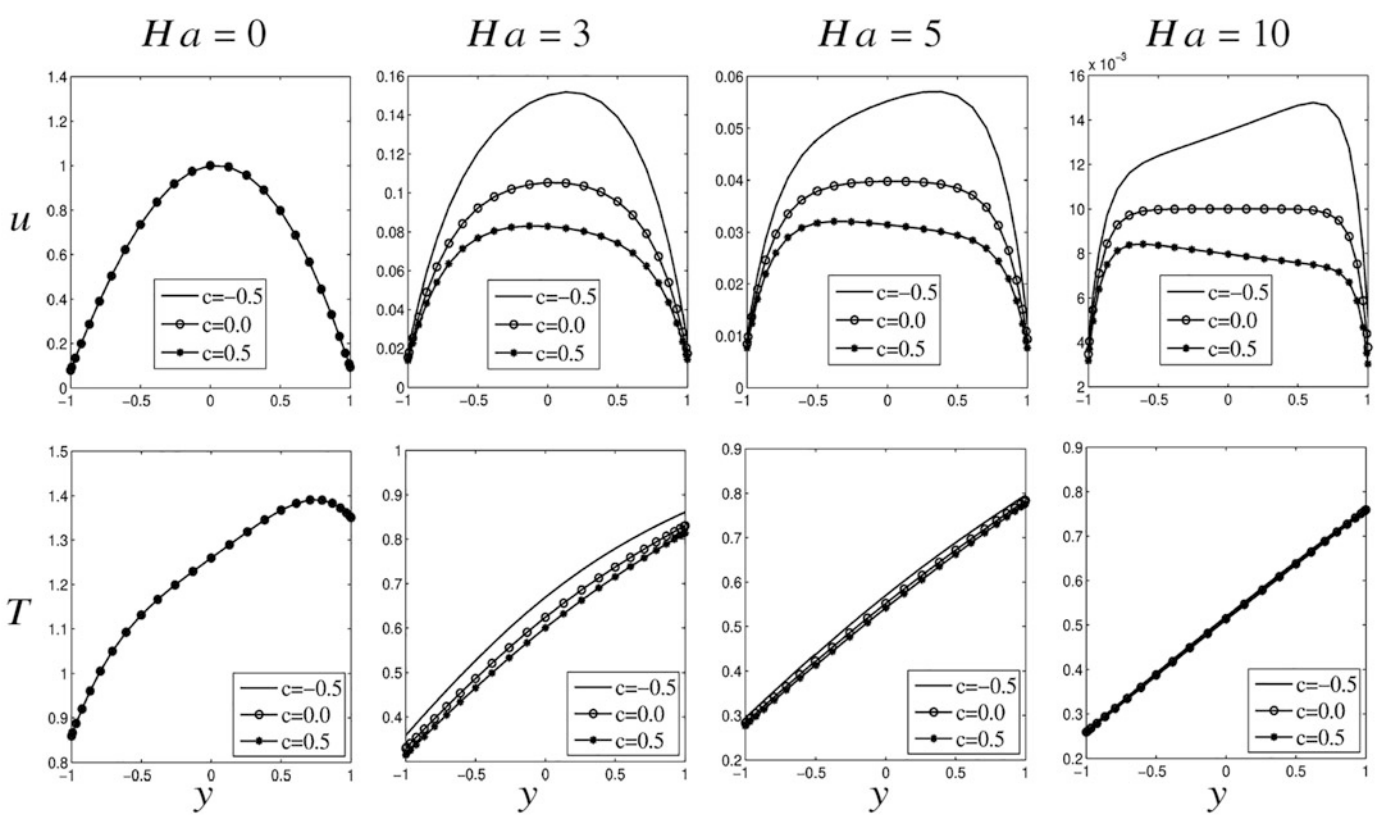


Fig. 3 Variation of u and T with various values of c and Ha, for $Pr = 1$, $Ec = 1$, $Bi_1 = 1$, $Bi_2 = 1$, $\alpha_1 = 0.05$, $\alpha_2 = 0.05$, $a = 0.5$, $b = 0$

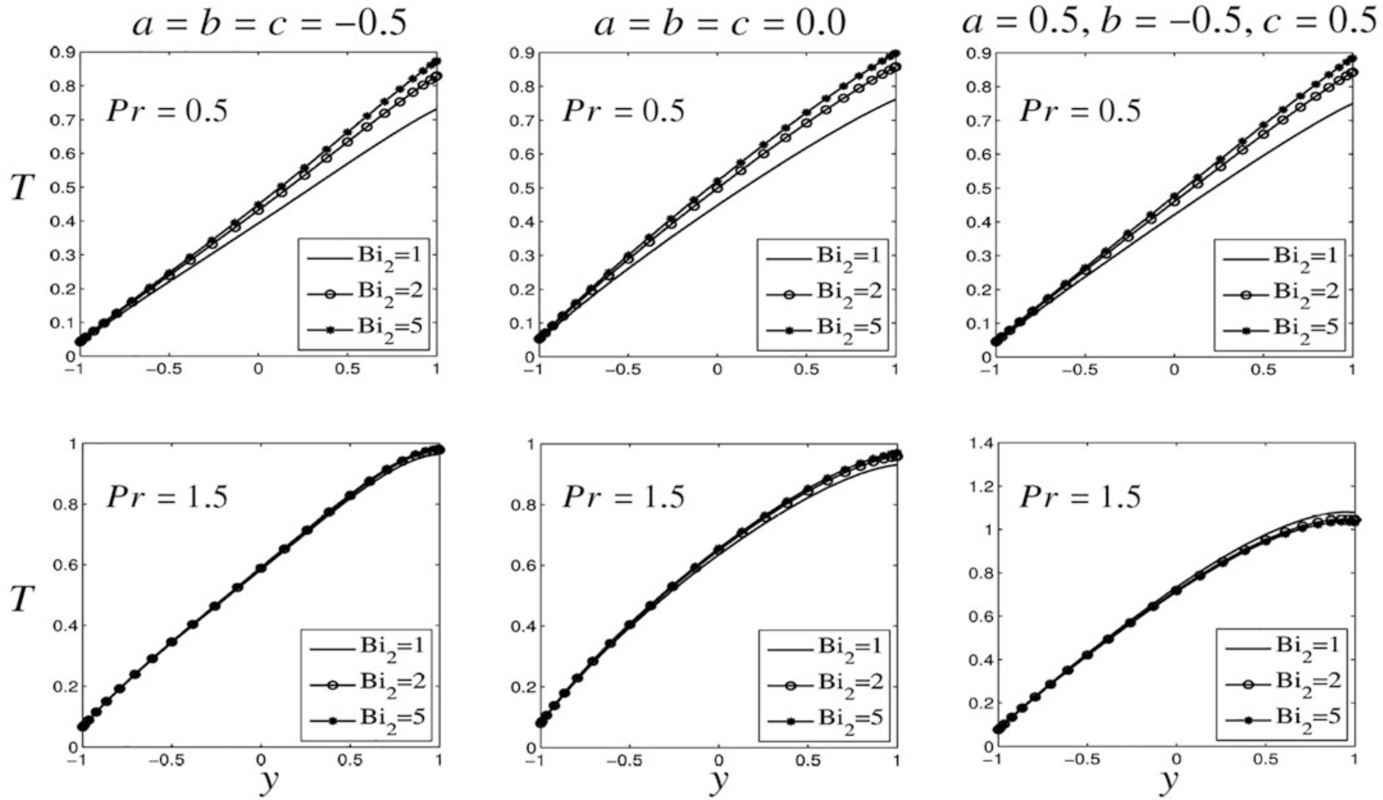


Fig. 4 Variation of T for various values of a, b, c and Pr Bi_2, when $Ha = 1$, $Ec = 1$, $Bi_1 = 10$, $\alpha_1 = \alpha_2 = 0.01$

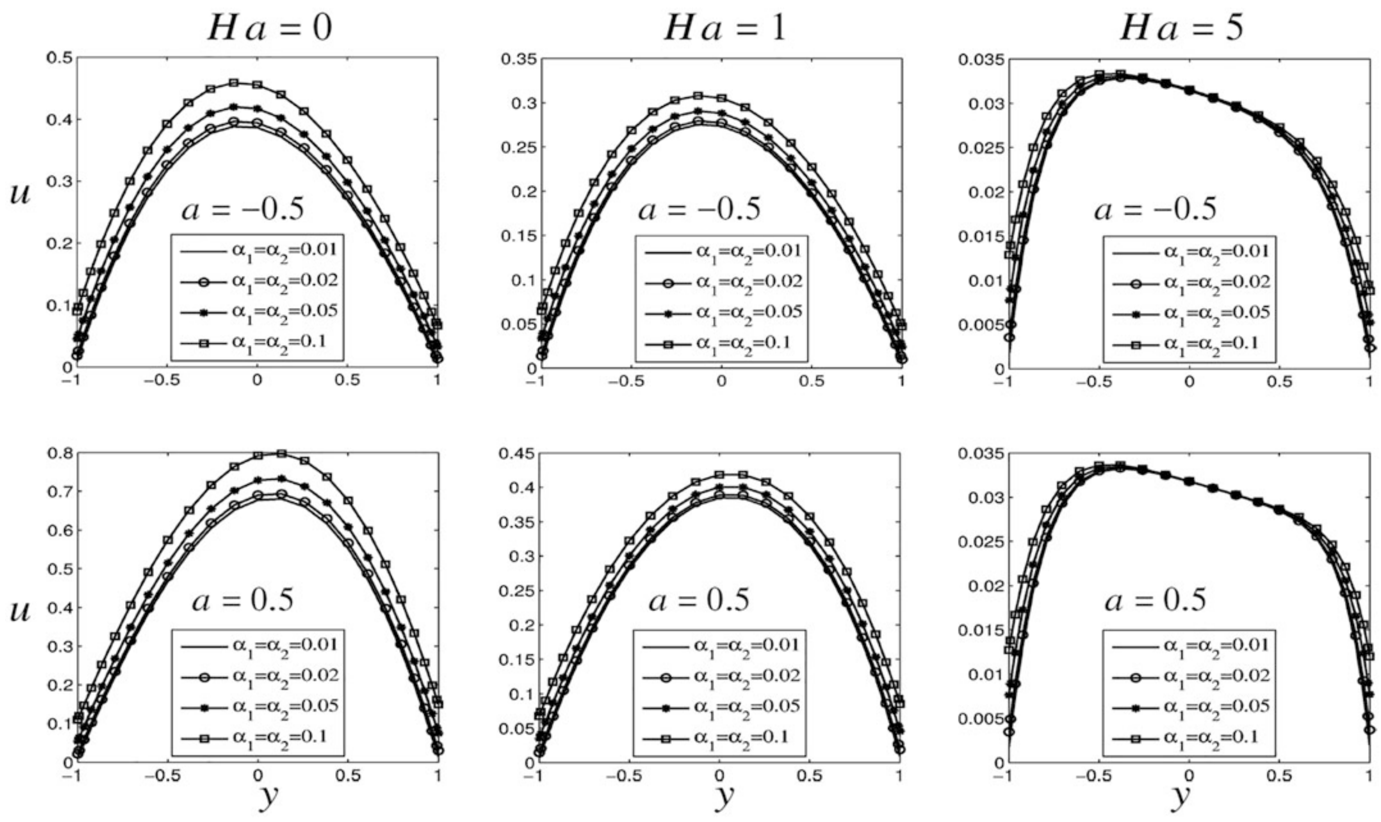


Fig. 5 Variation of u with $\alpha_1(=\alpha_2)$ and Ha, for $Ec = 1$, $Pr = 1$, $Bi_1 = Bi_2 = 100$, $b = 0$, and $c = 0.5$

magnitude decreases for all values of viscosity parameter a, and the symmetry of the velocity with respect to y is lost when Ha is increased as the magnetic damping force on u is increased. The increase in Ha, that is, when the external magnetic field is stronger, the flow is flattened (velocity drops) for all values of a and c. The drop in the fluid temperature for increasing values of c and Ha is not that much pronounced as it is seen in u. For $Ha \geq 5$, the temperature settles down taking the same profile for all c values. It is also observed that the influences of both the viscosity parameter and the electric conductivity parameter are weakened when Ha is increasing.

Figure 4 puts forward the variation of T with various values of Bi_2, a, b, c and Pr values for $Ha = 1$, $Ec = 1$, $Bi_1 = 10$, and $\alpha_1 = \alpha_2 = 0.01$. As the Biot number for the upper plate increases, the temperature of the fluid is increased for fixed a, b, c, Ha, Ec, Bi_1 values, and when $\alpha_1 = \alpha_2$. This increase is weakened when Pr is increased.

Variation of u with increasing $\alpha_1 = \alpha_2$ and Ha values, for $Ec = 1$, $Pr = 1$, $Bi_1 = Bi_2 = 100$, $b = 0$, $c = 0.5$ is illustrated in Fig. 5. The velocity magnitude increases as the slip length is increased for all values of Ha (see, e.g., [7]). However, the increase in Ha weakens the slip effect on u.

5 Conclusion

This study, exploits the efficiency of CSCM for solving the steady MHD flow and heat transfer between parallel plates with temperature dependent viscosity, thermal and electrical conductivity. The hydrodynamic slip conditions are imposed on the velocity of the fluid, and the convective boundary conditions are considered for the temperature. The numerical results revealed the fact that as the electric conductivity increases, velocity magnitude decreases for all values of the viscosity parameter. Increase in the viscosity parameter increases the temperature and velocity magnitude. The velocity magnitude also increases with an increase in the slip lengths. Moreover, an increase in Ha weakens the effect of a, c, and the slip effect. As Biot number for the upper plate is increased, the temperature rises.

References

1. Alpher, R.A.: Heat transfer in magnetohydrodynamic flow between parallel plates. International Journal of Heat and Mass Transfer **3**, 108–112 (1961)
2. Attia, H.A.: Transient MHD flow and heat transfer between two parallel plates with temperature dependent viscosity. Mechanics Research Communications **26**, 115–121 (1999)
3. Attia, H.A.: On the effectiveness of variation in the physical variables on the steady MHD flow between parallel plates with heat transfer. International Journal for Numerical Methods in Engineering **65**, 224–235 (2006)

4. Attia, H.A.: The effect of variable properties on the unsteady Couette flow with heat transfer considering the Hall effect. Communications in Nonlinear Science and Numerical Simulation **13**, 1596–1604 (2008)
5. Attia, H.A., Kotb, N.A.: MHD flow between two parallel plates with heat transfer. Acta Mechanica **117**, 215–220 (1996)
6. Boyd, J.P.: Chebyshev and Fourier Spectral Methods. Dover, New York (2000)
7. Ibáñez, G.: Entropy generation in MHD porous channel with hydrodynamic slip and convective boundary conditions. International Journal of Heat and Mass Transfer **80**, 274–280 (2015)
8. Lima, J.A., Quaresma, J.N.N., Macedo, E.N.: Integral transform analysis of MHD flow and heat transfer in parallel-plates channels. International Communications in Heat and Mass Transfer **34**, 420–431 (2007)
9. Maikap, T.K., Mahapatra, T.R., Niyogi, P., Ghosh, A.K.: Numerical study of magnetohydrodynamic laminar flow separation in a channel with smooth expansion. International Journal for Numerical Methods in Fluids **59**, 495–518 (2009)
10. Matthews, M.T., Hill, J.M.: Newtonian flow with nonlinear Navier boundary condition. Acta Mechanica **191**, 195–217 (2007)
11. Shen, J., Tang, T., Wang, L.L.: Spectral Methods. Springer Series in Computational Mathematics. Springer, Berlin, Heidelberg (2011)
12. Smolentsev, S.: MHD duct flows under hydrodynamic “slip” condition. Theoretical and Computational Fluid Dynamics **23**(6), 557–570 (2009)
13. Sweet, E., Vajravelu, K., Van Gorder, R.A., Pop, I.: Analytical solution for the unsteady MHD flow of a viscous fluid between moving parallel plates. Communications in Nonlinear Science and Numerical Simulation **16**, 266–273 (2011)
14. Trefethen, L.N.: Spectral Methods in Matlab. Siam, Philadelphia (2000)

Towards Scalable Automatic Exploration of Bifurcation Diagrams for Large-Scale Applications

Jonas Thies, Michiel Wouters, Rebekka-Sarah Hennig, and Wim Vanroose

Abstract The Trilinos library LOCA (http://www.cs.sandia.gov/LOCA/) allows computing branches of steady states of large-scale dynamical systems like (discretized) nonlinear PDEs. The core algorithms typically are (pseudo-)arclength continuation, Newton–Krylov methods and (sparse) eigenvalue solvers. While LOCA includes some basic techniques for computing bifurcation points and switching branches, the exploration of a complete bifurcation diagram still takes a lot of programming effort and manual interference.

On the other hand, recent developments in algorithms for fully automatic exploration are condensed in PyNCT (https://pypi.org/project/PyNCT/). The scope of this algorithmically versatile software is, however, limited to relatively small (e.g. 2D) problems because it relies on linear algebra from Python libraries like NumPy. Furthermore, PyNCT currently does not support problems with a non-Hermitian Jacobian matrix, which rules out interesting applications in chemistry and fluid dynamics.

In this paper we aim to combine the best of both worlds: a high-level implementation of algorithms in PyNCT with parallel models and linear algebra implemented in Trilinos. PyNCT is extended to non-symmetric systems and its complete backend is replaced by the PHIST library (https://bitbucket.org/essex/phist), which allows us to use the same underlying HPC libraries as LOCA does.

We then apply the new code to a reaction-diffusion model to demonstrate its potential of enabling fully automatic bifurcation analysis on parallel computers.

J. Thies (✉) · R.-S. Hennig
German Aerospace Center, Cologne, Germany
e-mail: Jonas.Thies@DLR.de; Rebekka-Sarah.Hennig@DLR.de

M. Wouters · W. Vanroose
University of Antwerp, Antwerp, Belgium
e-mail: Michiel.Wouters2@uantwerpen.be; wim.vanroose@uantwerpen.be

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_97

1 Introduction

Numerical bifurcation analysis is a key technology in understanding the properties of dynamical systems. The methodology comprises techniques for constructing a qualitative map from model parameters to model behavior. It has a wide range of applications, e.g. in biology [5], fluid dynamics [2], and superconductivity [13]. The basic technique investigated here is the automatic computation of landscapes of steady states under a single varying parameter. If the dynamical system is described by a (system of) partial differential equation(s) (PDEs), one typically performs a spatial discretization and then directly computes its steady state solutions rather than using time integration. A standard technique here is pseudo-arclength continuation [8], combined with a Newton–Krylov method for solving the arising nonlinear system of algebraic equations. By following connected branches of steady states, the approach achieves convergence even to linearly unstable solutions, which cannot be found using time stepping methods.

The building blocks for an efficient implementation of the approach are data structures for sparse matrices and dense (blocks of) vectors, iterative methods for solving large linear systems (typically Krylov subspace methods), preconditioning and/or deflation to accelerate their convergence, and sparse eigenvalue solvers. Eigenvalues of the Jacobian indicate whether a calculated solution is linear stable or not, and eigenvectors are useful for e.g. switching to another branch of solutions [8]. In particular in three space dimensions, the algebraic linear and eigenvalue problems that need to be solved can easily become too large to fit in the main memory (RAM) of a single computer. Moreover, computing a complete bifurcation diagram requires hundreds of systems to be solved, so that time-to-solution is key. Modern computers offer plenty of parallelism like executing a single instruction on multiple data (SIMD), symmetric multi-processing (SMP) on multi- or manycore CPUs, and accelerator hardware like graphics processing units (GPUs). As memory limitations may force us to use a cluster of multiple such nodes, the additional layer of distributed memory must be adequately addressed when implementing an algorithm.

As both HPC hardware and programming models are developing rapidly today, it becomes almost impossible for algorithm developers to provide their methods in stable, portable and maintainable software libraries, and even the large HPC software initiatives Trilinos [7] and PETSc [1] are facing difficulties to keep up with the rapid pace of the hardware development. In this paper we therefore take an existing abstraction layer for high performance sparse linear and eigenvalue computations (PHIST [11]) and develop a simple and easy-to-use Python frontend providing the functionality required to implement a continuation method. This Python layer is then used to painlessly update the PyNCT package to run on arbitrary distributed memory HPC systems.

2 Overview of Available Software

PyNCT is a Python toolbox for the automatic exploration of bifurcation diagrams based on the sparse linear algebra framework of NumPy and SciPy. It was originally developed to trace single solution branches in auxin transport models [3–5], a biology application. Recently, the PyNCT package has been extended with algorithms that allow for automatic exploration of bifurcation diagrams of Hermitian nonlinear problems [12]. Starting from an (approximate) initial solution, a landscape consisting of interconnected solution curves is generated. In [13], the efficiency of PyNCT is shown by automatically generating an interconnected solution landscape for a superconductor model (the extreme type-II Ginzburg-Landau equation). The generated landscape consists of 43 different solution curves, connected through a total of 60 branch points. Automatic exploration is handled through two main steps: searching branch points and calculating tangent directions to curves emanating from these. PyNCT includes various algorithms for these steps, including ones for constructing tangent directions in branch points of multiplicity 2 (double zero eigenvalues of the Jacobian) [13], which typically appear in nonlinear problems with a 2-dimensional symmetry. The algorithms in PyNCT also work on nonlinear problems with continuous symmetries. These symmetries induce null vectors in the system's Jacobian, when evaluated in a solution. Without proper adjustments, these symmetry-induced null vectors would lead to failures during automatic exploration, e.g. when detecting bifurcation points. Though bifurcation points and directions to new curves are calculated without direct user interference, internal parameters (e.g. tolerances) still need to be provided. The extension of the algorithms to non-Hermitian problems is also still in progress. An initial result, where PyNCT is applied to a non-Hermitian reaction-diffusion model, is discussed in Sect. 3.

LOCA is part of the open source high performance computing (HPC) framework Trilinos. It is written in C++ and in a modular way, exploiting algorithms and data structures from various other Trilinos libraries for solving the arising linear systems of equations and eigenvalue problems. LOCA provides algorithms for tracking down certain bifurcations along a branch, but there is no automatic exploration technology, and the provider of the model must take care of storing and postprocessing the solutions, as well as switching to new branches. On the other hand, the modular design of Trilinos allows using LOCA with a wide range of solvers and preconditioners, and with (hybrid) parallel backends such as Epetra and Tpetra.

3 Example: A Turing Problem

In this section we apply the PyNCT package to an example of a non-Hermitian nonlinear problem, derived from a coupled PDE. We generate a part of the example's solution landscape, demonstrating a successful extension of PyNCT to this type of problems. The example we consider is the Barrio–Varea–Aragon–Maini (BVAM) model, a Turing system that has applications in imitating the pattern formation on various fish species' skin [9]. The BVAM model is described by the following set of PDE's:

$$\begin{aligned}\frac{\partial U}{\partial t} =& D\delta\nabla^2 U + \alpha U(1 - r_1 V^2) + V(1 - r_2 U),\\ \frac{\partial V}{\partial t} =& \delta\nabla^2 V + V(\beta + \alpha r_1 U V) + U(\gamma + r_2 V).\end{aligned} \tag{1}$$

We consider a square domain $\Omega = [0, 30] \times [0, 30]$ and periodic boundary conditions. The functions $U = U(x, t)$ and $V = V(x, t)$ are two concentrations of chemicals, D, α, β, γ, δ, r_1 and r_2 are physical parameters. A detailed description of these parameters is given in [10].

The steady state equation is derived from (1) by setting the time derivatives of U and V equal to zero. We choose a uniform discretization of Ω with 32^2 points ($n = 32$), and denote $u, v \in \mathbb{R}^{n^2}$, respectively the discrete variants of U and V. Denoting A the discretization of the Laplacian ∇^2, we have a nonlinear function $\mathcal{F}$ that can be analyzed by PyNCT:

$$\mathcal{F} : \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} \to \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} : (u, v) \to \begin{pmatrix} D\delta A u + \alpha u(1 - r_1 v^2) + v(1 - r_2 u) \\ \delta A v + v(\beta + \alpha r_1 u v) + u(\gamma + r_2 v) \end{pmatrix}. \tag{2}$$

The analysis requires the Jacobian of $\mathcal{F}$, given by the linear operator

$$\begin{aligned}\mathcal{F}'(u, v) :& \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} \to \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} :\\ &(x, y) \to \begin{pmatrix} D\delta A x + (\alpha - \alpha r_1 v^2 - r_2 v)x + (-2\alpha r_1 u v + 1 - r_2 u)y \\ \delta A y + (\alpha r_1 v^2 + \gamma + r_2 v)x + (\beta + 2\alpha r_1 u v + u r_2)y \end{pmatrix}.\end{aligned} \tag{3}$$

The function (2) is invariant under both discrete and continuous (translational) symmetries. Both sorts induce challenges when performing continuation, as described in Sect. 2.

PyNCT uses a Newton–Krylov (GMRES) algorithm to find solutions of $\mathcal{F}(u, v) = 0$. Eigenpairs (used to identify, search and analyze branch points) are approximated by Ritz pairs. A preconditioner is used to speed up these processes:

$$\mathcal{P} : \mathbb{R}^{n^2} \times \mathbb{R}^{n^2} : w \to B^{-1} w, \text{ with } B = \begin{pmatrix} D\delta A + \alpha I & 0 \\ 0 & \delta A + \beta I \end{pmatrix}. \tag{4}$$

Systems with B are solved by a sparse direct solver. Note that solutions (zeros of (2)), branch and turning points are the same for the preconditioned and unpreconditioned problems. In this section, we use the NumPy/SciPy backend of PyNCT because the eigensolver requires mixing of real and complex data types, which is not supported by PHIST.

3.1 Partial Bifurcation Diagram of the BVAM Model

We perform a numerical continuation of the nonlinear function (2) with PyNCT, choosing r_2 as the continuation parameter. The other parameters are kept fixed, their values are given in Table 1.

We start the continuation from an initial solution at $r_2 = 0$, given by Fig. 1. Subsequent solutions are generated automatically by PyNCT. We restrict ourselves to a maximum of 4 solution curves and 200 points per curve. This yields the partial bifurcation diagram of Fig. 2. Representative solutions of each curve are provided in Fig. 3.

Starting from the solution on curve A at $r_2 = 0$, PyNCT locates two branch points: at $r_2 = 0.1164$ and $r_2 = 0.4302$. The point at $r_2 = 0.1164$ has multiplicity 2. Two solution curves emerge from it: curves B and C. Note that these curves contain solutions with a reduced symmetry, which is predicted by the equivariant branching lemma [6]. The branch point at $r_2 = 0.4302$ is of multiplicity 1, and leads to curve D.

Both curves B, C and D contain further branch points, at respectively $r_2 = 0.3062$, $r_2 = 0.2888$ and $r_2 = 0.4375$. These points would lead to other solution curves, which were not generated due to the chosen restrictions (maximal 4 curves). It is possible for curves A, B, C and D to have more branch points as well, again not found due to restricting ourselves to 200 points per curve.

Table 1 Values of parameters in (2) used for generating the bifurcation diagram of Fig. 2

D	α	β	γ	δ	r_1
0.516	0.899	−0.91	−0.899	2	3.5

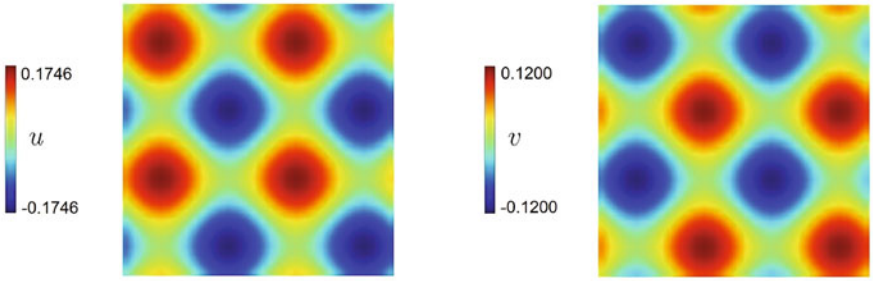


Fig. 1 Initial solution of the nonlinear function (2), used to generate the partial bifurcation diagram of Fig. 2

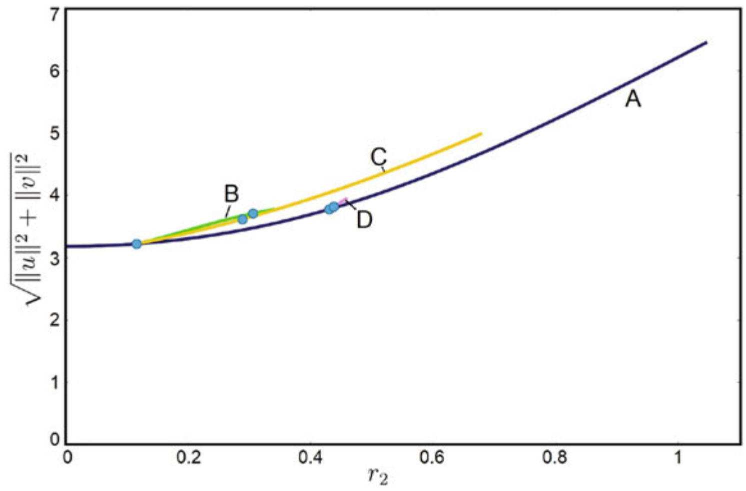


Fig. 2 Part of a connected solution landscape for the BVAM model on a square domain of length 30. Blue dots indicate branch points. Representative solutions for the curves are given in Fig. 3

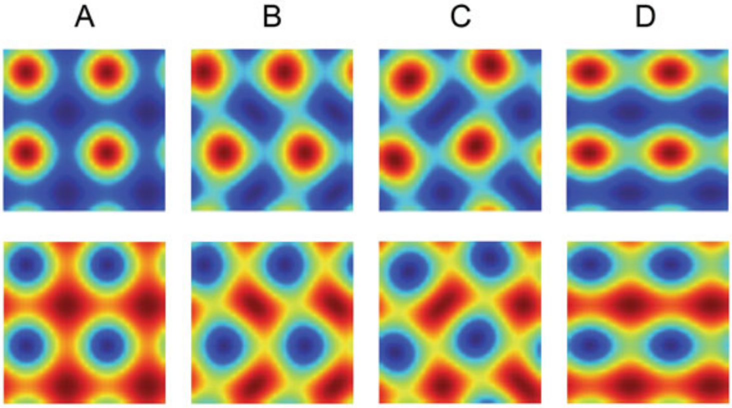


Fig. 3 Representative solutions for the different curves of Fig. 2

Though the generated bifurcation diagram is only partial, it acts as a proof of concept: branch points of a non-Hermitian nonlinear problem can be found and analysed by PyNCT, indicating the possibility of the automatic exploration of an interconnected solution landscape.

4 Comparison with LOCA

In order to assess the potential of our parallelization approach for PyNCT, we compute a single branch of steady state solutions for the 2D Turing model. This process involves Newton's method, GMRES, and a multigrid preconditioner, and can be carried out easily using either LOCA or PyNCT with the PHIST/Epetra backend. The two libraries use a slightly different set of solver parameters, and algorithmic details such as step size control and GMRES orthogonalization scheme differ. The direct comparison therefore only gives a first impression, but is nevertheless insightful. The preconditioner consists of a multigrid sweep on the matrix B in (4), implemented using the Trilinos package ML (this was also used in [10]).

Figure 4 (left) shows the total number of preconditioned GMRES iterations performed by the two packages in order to compute 10 steady states on a solution branch ranging from $r_2 = 0$ to $r_2 = 1$, on a 128^2 grid. Despite the fact that LOCA uses an adaptive tolerance for the GMRES algorithm, the number of iterations is relatively similar, so that we can assume that a rough comparison of timing results makes sense. This comparison is shown in Fig. 4 (right) for a larger problem (on a 1024^2 grid). Here we see that both implementations yield similar scalability, but also that the LOCA implementation is about twice as fast. Recall that both implementations use the same implementation for their linear algebra operations (the Trilinos package Epetra), so the difference must lie in the number and type of operations performed.

The timing and profiling features of PHIST make it relatively easy to get more insight. Figure 5 shows run time profiles of the most important basic operations for 1 and 16 processes, respectively, for the 128^2 problem. The PyNCT implementation requires about 15× more inner products, which shows that it has not been developed with parallel computing in mind (algorithmically the difference is a modified Gram-Schmidt process in the GMRES solver in PyNCT). This reduces the strong scaling performance. The substantially larger contribution of 'other' functions in the PyNCT runs is explained mostly by overhead for calling relatively small C functions via the Python ctypes module. This leads to a constant latency for

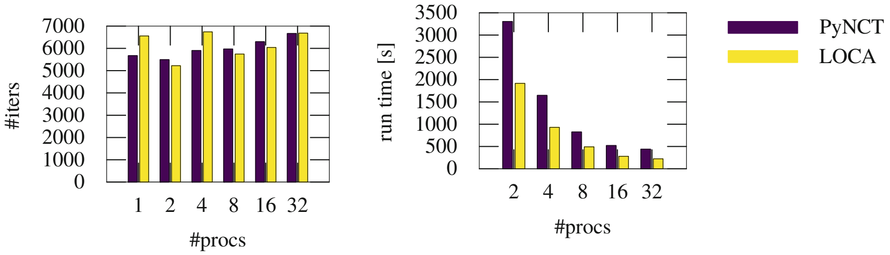


Fig. 4 Rough comparison between PyNCT and LOCA when computing a branch of 10 solutions of the BVAM model on a 1 024^2 grid. Left: total number of GMRES iterations. Right: total runtime

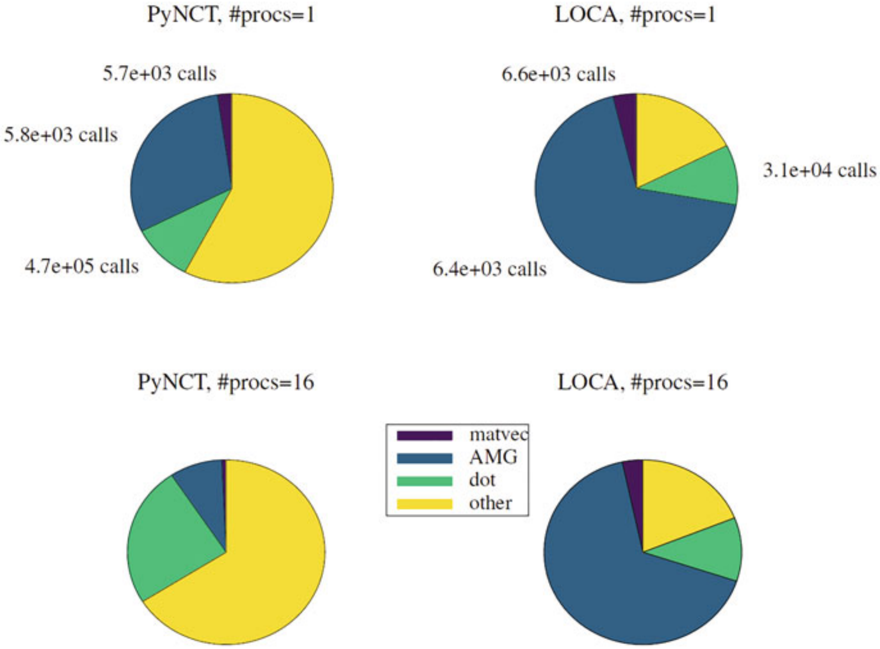


Fig. 5 Basic performance profile when computing a branch of 10 solutions on a 128^2 grid

otherwise perfectly scalable functions such as vector additions, further reducing strong scalability.

5 Conclusion

The paper proposes two improvements to the PyNCT software for automatic bifurcation analysis: extension to non-Hermitian problems, and parallelization by introducing a backend layer. The first point is mathematically non-trivial, and by no means treated exhaustively in this paper. However, the fact that the code is written in Python allowed us to relatively quickly arrive at a working 'proof of concept', demonstrated by computing branch points of a Turing-type reaction-diffusion model. A major challenge (consuming more than 90% of the time in our experiments) is the solution of sparse eigenvalue problems. Here, we hope to use the Jacobi–Davidson method available in PHIST in the future to significantly reduce this number by recycling subspaces and preconditioning, as was also done in [10].

Second, we investigated a possibility for quickly parallelizing a Python code which relies heavily on sparse linear algebra. By introducing a thin layer of wrapper objects, we can now switch between a pure Python implementation (which can e.g. be installed completely using PIP) and any HPC implementation supported by the

PHIST library. The approach leads to reasonable parallel performance, but it also becomes clear that algorithmic decisions (like the orthogonalization scheme used by GMRES) should be re-evaluated. It is possible to extend our wrapper layer with higher level functions for orthogonalization, which may e.g. call the PHIST implementation if available. By further integration of PHIST, one could make complete solvers for linear or eigenvalue problems available, which are designed for good performance with the underlying data structures.

In summary, the paper demonstrates that we can combine the advantages of the Python package PyNCT (rapid prototyping) and the HPC libraries Trilinos and PHIST (performance and scalability) to shorten the time-to-market for new algorithmic developments and enable the fully automatic exploration of bifurcation diagrams on HPC systems.

References

1. S. Balay, S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, and H. Zhang. PETSc Web page. http://www.mcs.anl.gov/petsc, 2016.
2. H. A. Dijkstra, F. W. Wubs, A. K. Cliffe, E. Doedel, I. F. Dragomirescu, B. Eckhardt, A. Y. Gelfgat, A. L. Hazel, V. Lucarini, A. G. Salinger, E. T. Phipps, J. Sanchez-Umbria, H. Schuttelaars, L. S. Tuckerman, and U. Thiele. Numerical bifurcation methods and their application to fluid dynamics: Analysis beyond simulation. *Communications in Computational Physics*, 15(01):1–45, Jan 2014.
3. D. Draelants, D. Avitabile, and W. Vanroose. Localized auxin peaks in concentration-based transport models of the shoot apical meristem. *J. R. Soc. Interface*, 12, May 2015.
4. D. Draelants, J. Broeckhove, G. Beemster, and W. Vanroose. Numerical bifurcation analysis of pattern formation in a cell based auxin transport model. *J. Math. Biol.*, 67, Sep 2012.
5. D. Draelants, P. Klosiewicz, J. Broeckhove, and W. Vanroose. Solving general auxin transport models with a numerical continuation toolbox in Python: PyNCT. *Lect. Notes Comput. Sci.*, 9271:211–225, 2015.
6. M. Golubitsky and I. Stewart. *The symmetry perspective*. Birkhäuser, 2002.
7. M. Heroux, R. Bartlett, V. H. R. Hoekstra, J. Hu, T. Kolda, R. Lehoucq, K. Long, R. Pawlowski, E. Phipps, A. Salinger, H. Thornquist, R. Tuminaro, J. Willenbring, and A. Williams. An overview of Trilinos. Technical Report SAND2003-2927, Sandia National Laboratories, 2003.
8. H. B. Keller. Numerical solution of bifurcation and nonlinear eigenvalue problems. In P. H. Rabinowitz, editor, *Applications of Bifurcation Theory*, pages 359–384. Academic Press, New York, U.S.A., 1977.
9. T. Leppänen. Computational studies of pattern formation in Turing systems, 2004-11-27.
10. W. Song, F. Wubs, J. Thies, and S. Baars. Numerical bifurcation analysis of a 3D Turing-type reaction-diffusion model. *Communications in nonlinear science and numerical simulation*, 60:145–164, 7 2018.
11. J. Thies, M. Röhrig-Zöllner, N. Overmars, A. Basermann, D. Ernst, G. Hager, and G. Wellein. PHIST: a pipelined, hybrid-parallel iterative solver toolkit. *ACM Transactions on Mathematical Software*, 46(4), 2020.
12. M. Wouters. *Automatic exploration techniques for the numerical continuation of large–scale nonlinear systems*. PhD thesis, Universiteit Antwerpen, 2019.
13. M. Wouters and W. Vanroose. Automatic exploration techniques of numerical bifurcation diagrams illustrated by the Ginzburg–Landau equation. *SIAM Journal on Applied Dynamical Systems*, 18(4):2047–2098, 2019.

Generalized Monge–Ampère Equations for Freeform Optical System Design

J. H. M. ten Thije Boonkkamp, L. B. Romijn, and W. L. IJzerman

Abstract We present the derivation of the generalized Monge–Ampère equation for two optical systems, viz. a freeform lens with parallel incident and refracted light rays, which transforms a source emittance into a desired target illuminance, and a freeform reflector converting the intensity of a point source into a far-field distribution. The derivations are based on Hamilton's characteristic functions. We outline a least-squares solution method and apply it to a test problem from laser beam shaping.

1 Introduction

The standard problem in freeform illumination optics is to design optical systems that convert a given source light distribution into a desired target distribution. Inverse methods are very useful simulation methods to compute one or two freeform optical surfaces in an optical system and are an alternative to classical ray tracing. A freeform optical surface is either a lens or reflector surface of arbitrary shape, without any symmetries, as opposed to for example a rotationally symmetric surface. The underlying mathematical model is based on the principles of geometrical optics, expressed in terms of the optical map connecting source and target domains, and the energy conservation law. Combining the optical map with the energy balance, we can derive a fully nonlinear elliptic PDE determining the shape of an optical surface.

Alternatively, the optical design problem can be cast in the framework of optimal transport, which concerns the minimization of a cost functional, i.e., the integral

J. H. M. ten Thije Boonkkamp (✉) · L. B. Romijn
Eindhoven University of Technology, Eindhoven, The Netherlands
e-mail: j.h.m.tenthijeboonkkamp@tue.nl; l.b.romijn@tue.nl

W. L. IJzerman
Signify Research, Eindhoven, The Netherlands
e-mail: wilbert.ijzerman@signify.com

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139,
https://doi.org/10.1007/978-3-030-55874-1_98

of a cost function weighted with the source distribution, under the constraint of energy conservation. For some basic systems, the cost function is quadratic and the optical map is the gradient of a potential function. The governing PDE reduces to the standard Monge–Ampère equation. However, in this contribution, we focus on systems for which the cost function is no longer quadratic, and consequently the governing PDE becomes a generalized Monge–Ampère equation.

The emphasis in this paper is on the derivation of the generalized Monge–Ampère equation for two optical systems, viz., a freeform lens with parallel incident a refracted rays, and a freeform reflector in combination with a point source. The derivations are based on Hamilton's characteristic functions and are detailed in Sect. 2. In Sect. 3 we briefly outline a least-squares method to compute the freeform surface(s). Next, in Sect. 4 we demonstrate the performance of our mathematical/numerical model for a test problem from laser beam shaping.

2 Mathematical Formulation

The derivations in this section are a concise reformulation of the mathematical models in [2] for the freeform lens and in [3] for the freeform reflector. In the following, $\mathcal{S}$ and $\mathcal{T}$ denote the source and target domain, respectively, and the map $\boldsymbol{m} : \mathcal{S} \to \mathcal{T}$ is referred to as the optical map.

Freeform Lens, Parallel in and Out Consider a planar light source, e.g., a collimated beam, located in the plane $z = 0$, a target plane $z = L$ and a lens with two freeform surfaces in between; see Fig. 1. The first lens surface ($\mathcal{L}_1$) is defined by the relation $z = u_1(\boldsymbol{x})$ with $\boldsymbol{x} \in \mathcal{S}$ and the second one ($\mathcal{L}_2$) by $L - z = u_2(\boldsymbol{y})$ for $\boldsymbol{y} \in \mathcal{T}$. The index of refraction of the lens is n. The source emits a parallel beam of light in the positive z-direction. Light rays hit $\mathcal{L}_1$, are refracted, hit $\mathcal{L}_2$, are refracted again, creating a parallel beam of light, also in the positive z-direction.

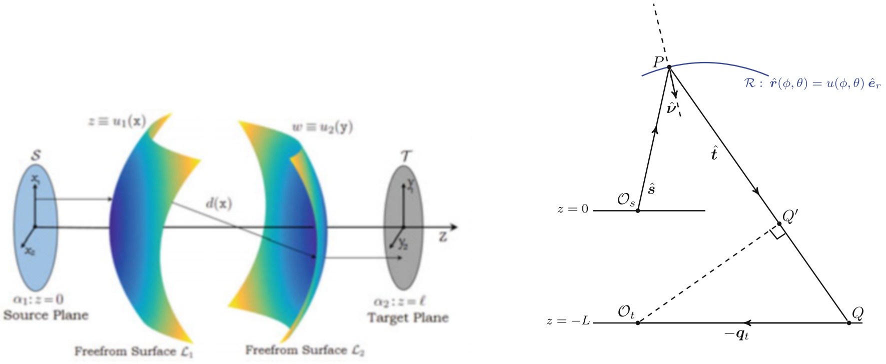


Fig. 1 Freeform lens (left) and freeform reflector (right)

To derive a relation for the optical map, we employ Hamilton's point characteristic function V [1, p. 94–100]. In the following, $\boldsymbol{q}$ and $\boldsymbol{p}$ denote the (two-dimensional) position and direction vectors, respectively, of a ray intersecting a plane. The vector $\boldsymbol{p}$ is the projection on the plane of the unit direction vector of the ray, multiplied by n. The subscripts s and t refer to source and target plane, respectively. Consider a typical ray connecting a point on $\mathcal{S}$, for which $\boldsymbol{q}_\mathrm{s} = \boldsymbol{x}$ and $\boldsymbol{p}_\mathrm{s} = \boldsymbol{0}$, with a point on the target plane, characterized by $\boldsymbol{q}_\mathrm{t} = \boldsymbol{y}$ and $\boldsymbol{p}_\mathrm{t} = \boldsymbol{0}$. The characteristic function V is the optical path length between both points and is defined by

$$V(\boldsymbol{q}_\mathrm{s}, \boldsymbol{q}_\mathrm{t}) = u_1(\boldsymbol{x}) + nd + u_2(\boldsymbol{y}), \quad d^2 = |\boldsymbol{x} - \boldsymbol{y}|^2 + (L - (u_1(\boldsymbol{x}) + u_2(\boldsymbol{y})))^2, \tag{1}$$

where d is the distance between both lens surfaces, measured along the refracted ray. Since $\boldsymbol{p}_\mathrm{s} = -\partial V/\partial \boldsymbol{q}_\mathrm{s} = \boldsymbol{0}$ and $\boldsymbol{p}_\mathrm{t} = \partial V/\partial \boldsymbol{q}_\mathrm{t} = \boldsymbol{0}$, the characteristic function $V(\boldsymbol{q}_\mathrm{s}, \boldsymbol{q}_\mathrm{t}) = V = \text{const}$. In the derivation that follows, it is convenient to introduce the variable $c = c(\boldsymbol{x}, \boldsymbol{y}) = u_1(\boldsymbol{x}) + u_2(\boldsymbol{y})$. Combining both relations in (1), we arrive at the following quadratic equation for c:

$$(V - c)^2 - n^2(L - c)^2 = n^2|\boldsymbol{x} - \boldsymbol{y}|^2. \tag{2}$$

Completing the square we find

$$\Big(c - L + \frac{\beta}{n^2 - 1}\Big)^2 = \Big(\frac{n}{n^2 - 1}\Big)^2 \big(\beta^2 - (n^2 - 1)|\boldsymbol{x} - \boldsymbol{y}|^2\big), \tag{3}$$

with $\beta = V - L$ the reduced optical path length. This equation has two real solutions since $\beta^2 - (n^2 - 1)|\boldsymbol{x} - \boldsymbol{y}|^2 = \big(n\beta - (n^2 - 1)d\big)^2 > 0$. Comparing the two possible solutions with the first relation in (1), we conclude that the we have to choose the negative root of equation (3). This way we find the relations

$$u_1(\boldsymbol{x}) + u_2(\boldsymbol{y}) = c(\boldsymbol{x}, \boldsymbol{y}), \tag{4a}$$

$$c(\boldsymbol{x}, \boldsymbol{y}) = L - \frac{\beta}{n^2 - 1} - \frac{n}{n^2 - 1}\sqrt{\beta^2 - (n^2 - 1)|\boldsymbol{x} - \boldsymbol{y}|^2}. \tag{4b}$$

We refer to $c(\boldsymbol{x}, \boldsymbol{y})$ as the cost function. From (4) we will derive shortly a relation for the optical map $\boldsymbol{y} = \boldsymbol{m}(\boldsymbol{x})$.

Single Reflector, Point Source, Far Field Out Consider a point source located in the origin O_s emitting rays radially upward and a freeform reflector $\mathcal{R}$ given by the parametrization $\boldsymbol{r}(\phi, \theta) = u(\phi, \theta)\hat{\boldsymbol{e}}_r$, with ϕ $(0 \leq \phi \leq \pi)$ and θ $(0 \leq \theta < 2\pi)$ the polar and azimuthal angle, respectively, and $\hat{\boldsymbol{e}}_r$ the radial basis vector of the spherical coordinate system; see Fig. 1. All unit vectors are denoted by a hat (ˆ). The variable $u = u(\phi, \theta)$ denotes the radial distance between source and reflector surface. An emitted ray has direction vector $\hat{\boldsymbol{s}} = \hat{\boldsymbol{e}}_r$, is intercepted by $\mathcal{R}$ and reflects off in the direction $\hat{\boldsymbol{t}}$. Since $\hat{\boldsymbol{s}} = (s_1, s_2, s_3)^\mathrm{T}, \hat{\boldsymbol{t}} = (t_1, t_2, t_3)^\mathrm{T} \in \mathrm{S}^2$, it is convenient to describe source and target domain in terms of stereographic coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$, respectively. Our choice is

$$\boldsymbol{x}(\hat{\boldsymbol{s}}) = \frac{1}{1+s_3}\begin{pmatrix} s_1 \\ s_2 \end{pmatrix}, \quad \boldsymbol{y}(\hat{\boldsymbol{s}}) = \frac{1}{1-t_3}\begin{pmatrix} t_1 \\ t_2 \end{pmatrix}. \tag{5}$$

The coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$ are the projections from the south pole $(0, 0, -1)$ and the north pole $(0, 0, 1)$, respectively, suitable to represent the upward incident and downward reflected rays.

Analogous to the derivation of (4), we use Hamilton's angular characteristic T [1, p. 104–107]. Consider a typical light ray connecting the source with a point on the target screen $z = -L(< 0)$. The angular characteristic is defined by

$$T(\boldsymbol{p}_{\mathrm{s}}, \boldsymbol{p}_{\mathrm{t}}) = V(\boldsymbol{q}_{\mathrm{s}}, \boldsymbol{q}_{\mathrm{t}}) + \boldsymbol{q}_{\mathrm{s}} \cdot \boldsymbol{p}_{\mathrm{s}} - \boldsymbol{q}_{\mathrm{t}} \cdot \boldsymbol{p}_{\mathrm{t}}, \tag{6a}$$

$$V(\boldsymbol{q}_{\mathrm{s}}, \boldsymbol{q}_{\mathrm{t}}) = u(\hat{\boldsymbol{s}}) + d, \quad d^2 = |\boldsymbol{q} - u(\hat{\boldsymbol{s}})\boldsymbol{p}_{\mathrm{s}}|^2 + (L + u(\hat{\boldsymbol{s}})s_3)^2, \tag{6b}$$

where, with a slight abuse of notation, $u(\hat{\boldsymbol{s}}) = u(\phi, \theta)$, and where d denotes the distance between P, the intersection point of the incident ray and $\mathcal{R}$, and Q, the intersection point of the reflected ray and the target screen. The following relations hold: $\boldsymbol{q}_{\mathrm{s}} = \boldsymbol{0}$, $\boldsymbol{p}_{\mathrm{s}} = (s_1, s_2)^{\mathrm{T}}$ and $\boldsymbol{p}_{\mathrm{t}} = (t_1, t_2)^{\mathrm{T}}$. For $\boldsymbol{q}_{\mathrm{t}}$ we use the short hand notation $\boldsymbol{q}_{\mathrm{t}} = \boldsymbol{q}$. Since $\boldsymbol{q}_{\mathrm{s}} = \partial T/\partial \boldsymbol{p}_{\mathrm{s}} = \boldsymbol{0}$, we have $T = T(\boldsymbol{p}_{\mathrm{t}}) = T(t_1, t_2)$. To evaluate $T(t_1, t_2)$, we need the relations $\boldsymbol{p}_{\mathrm{t}} = (\boldsymbol{q} - u(\hat{\boldsymbol{s}})\boldsymbol{p}_{\mathrm{s}})/d$ and $t_3 = -(L + u(\hat{\boldsymbol{s}})s_3)/d$. Substituting these in (6), we obtain

$$\begin{aligned} T(t_1, t_2) &= u(\hat{\boldsymbol{s}}) + d - \boldsymbol{q}_{\mathrm{t}} \cdot \boldsymbol{p}_{\mathrm{t}} \\ &= u(\hat{\boldsymbol{s}}) + d - \frac{1}{d}\boldsymbol{q} \cdot (\boldsymbol{q} - u(\hat{\boldsymbol{s}})\boldsymbol{p}_{\mathrm{s}}) \\ &= u(\hat{\boldsymbol{s}}) + \frac{1}{d}\big(|\boldsymbol{q} - u(\hat{\boldsymbol{s}})\boldsymbol{p}_{\mathrm{s}}|^2 - \boldsymbol{q} \cdot (\boldsymbol{q} - u(\hat{\boldsymbol{s}})\boldsymbol{p}_{\mathrm{s}}) + (L + u(\hat{\boldsymbol{s}})s_3)^2\big) \\ &= u(\hat{\boldsymbol{s}}) + \frac{1}{d}\big(- u(\hat{\boldsymbol{s}})\boldsymbol{p}_{\mathrm{s}} \cdot (\boldsymbol{q} - u(\hat{\boldsymbol{s}})\boldsymbol{p}_{\mathrm{s}}) - dt_3(L + u(\hat{\boldsymbol{s}})s_3)\big) \\ &= u(\hat{\boldsymbol{s}}) - u(\hat{\boldsymbol{s}})\big(\boldsymbol{p}_{\mathrm{s}} \cdot \boldsymbol{p}_{\mathrm{t}} + s_3t_3\big) - Lt_3 \\ &= (1 - \hat{\boldsymbol{s}} \cdot \hat{\boldsymbol{t}})u(\hat{\boldsymbol{s}}) - Lt_3. \end{aligned} \tag{7}$$

Subsequently, rearranging terms we find

$$T(t_1, t_2) + Lt_3 = (1 - \hat{\boldsymbol{s}} \cdot \hat{\boldsymbol{t}})u(\hat{\boldsymbol{s}}), \tag{8}$$

where the left hand side solely depends on $\hat{\boldsymbol{t}}$. Furthermore, $\partial(T(t_1, t_2) + Lt_3)/\partial L = 0$ implying that $T(t_1, t_2) + Lt_3$ is independent of L, as anticipated. Taking the logarithm of (8) we find $\tilde{u}_1(\hat{\boldsymbol{s}}) + \tilde{u}_2(\hat{\boldsymbol{t}}) = -\log(1 - \hat{\boldsymbol{s}} \cdot \hat{\boldsymbol{t}})$, where $\tilde{u}_1(\hat{\boldsymbol{s}}) = \log(u(\hat{\boldsymbol{s}}))$

and $\tilde{u}_2 = -\log(T(t_1, t_2) + Lt_3)$. Finally, transforming to stereographic coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$, defined in (5), and introducing the variables $u_1(\boldsymbol{x}) = \tilde{u}_1(\hat{\boldsymbol{s}}) - \log(1 + |\boldsymbol{x}|^2)$ and $u_2(\boldsymbol{y}) = \tilde{u}_2(\hat{\boldsymbol{t}}) - \log(\frac{1}{2}(1 + |\boldsymbol{y}|^2))$, we obtain

$$u_1(\boldsymbol{x}) + u_2(\boldsymbol{y}) = c(\boldsymbol{x}, \boldsymbol{y}), \tag{9a}$$

$$c(\boldsymbol{x}, \boldsymbol{y}) = -\log\big(1 - 2\boldsymbol{x} \cdot \boldsymbol{y} + |\boldsymbol{x}|^2|\boldsymbol{y}|^2\big), \tag{9b}$$

from which we can determine a relation for the optical map; cf. (4). Note that the variable $u_1 = u_1(\boldsymbol{x})$ defines the location of the reflector surface, whereas $u_2 = u_2(\boldsymbol{y})$ is an auxiliary variable representing $T(t_1, t_2)$.

Optical Map Equations (4a) and (9a) have many possible solutions for $u_1(\boldsymbol{x})$ and $u_2(\boldsymbol{y})$. Assuming $\mathcal{S}$ and $\mathcal{T}$ are closed and bounded sets, we can prove that one possible solution reads

$$u_1(\boldsymbol{x}) = \max_{\boldsymbol{y}\in\mathcal{T}} \big(c(\boldsymbol{x}, \boldsymbol{y}) - u_2(\boldsymbol{y})\big), \quad u_2(\boldsymbol{y}) = \max_{\boldsymbol{x}\in\mathcal{S}} \big(c(\boldsymbol{x}, \boldsymbol{y}) - u_1(\boldsymbol{x})\big), \tag{10}$$

referred to as the c-convex solution. This solution necessarily requires that $\boldsymbol{x}$ be a stationary point of $c(\boldsymbol{x}, \boldsymbol{y}) - u_1(\boldsymbol{x})$, i.e.,

$$\nabla_{\boldsymbol{x}} c(\boldsymbol{x}, \boldsymbol{y}) - \nabla u_1(\boldsymbol{x}) = \boldsymbol{0}, \tag{11}$$

where $\nabla_{\boldsymbol{x}} c$ denotes the gradient of c w.r.t. the variable $\boldsymbol{x}$. From the implicit function theorem, we conclude that this relation implicitly defines the optical map $\boldsymbol{y} = \boldsymbol{m}(\boldsymbol{x})$, provided the matrix $\boldsymbol{C} = \mathrm{D}_{\boldsymbol{xy}} c = \big(c_{x_i, y_j}\big)$ is regular for all $\boldsymbol{x}$ and $\boldsymbol{y}$, which is true for the optical systems above [2, 3]. Next, substituting $\boldsymbol{y} = \boldsymbol{m}(\boldsymbol{x})$ in (11) and subsequently differentiating w.r.t. $\boldsymbol{x}$, we obtain for the Jacobi matrix $\mathrm{D}\boldsymbol{m}$ the equation

$$\boldsymbol{C}(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x}))\mathrm{D}\boldsymbol{m}(\boldsymbol{x}) = \boldsymbol{P}(\boldsymbol{x}), \quad \boldsymbol{P}(\boldsymbol{x}) = \mathrm{D}^2 u_1(\boldsymbol{x}) - D_{\boldsymbol{xx}} c(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x})), \tag{12}$$

where $\mathrm{D}_{\boldsymbol{xx}} c$ and $\mathrm{D}^2 u_1$ denote the Hessian matrices of c (w.r.t. $\boldsymbol{x}$) and u_1, respectively. A sufficient condition for the existence of the c-convex solution (10) is that $\boldsymbol{P}(\boldsymbol{x})$ be symmetric positive definite (SPD). Alternatively, we could introduce the c-concave solution, replacing the maximum in (10) by minimum. Note that we do not explicitly compute the optical map $\boldsymbol{m}$, instead we determine $\boldsymbol{m}$ from Eq. (12) combined with energy conservation, to be discussed next.

Energy Balance For the freeform lens, assume the source has emittance $E(\boldsymbol{x})$ (lm/m^2) and the required target illuminance is $G(\boldsymbol{y})$ (lm/m^2). The energy balance then reads

$$\int_{\mathcal{A}} E(\boldsymbol{x})\,\mathrm{d}A(\boldsymbol{x}) = \int_{\boldsymbol{m}(\mathcal{A})} G(\boldsymbol{y})\,\mathrm{d}A(\boldsymbol{y}), \tag{13}$$

for arbitrary $\mathcal{A} \subset \mathcal{S}$ and image set $\boldsymbol{m}(\mathcal{A}) \subset \mathcal{T}$. Substituting the optical map $\boldsymbol{y} = \boldsymbol{m}(\boldsymbol{x})$, the energy constraint becomes

$$E(\boldsymbol{x}) = G(\boldsymbol{m}(\boldsymbol{x})) \det\big(\mathrm{D}\boldsymbol{m}(\boldsymbol{x})\big), \tag{14}$$

restricting ourselves to the case $\det\big(\mathrm{D}\boldsymbol{m}(\boldsymbol{x})\big) > 0$.

Next, for the freeform reflector, assume the source has intensity $I(\phi, \theta)$ (lm/sr) and the desired far-field intensity is $G(\psi, \chi)$ (lm/sr), with (ψ, χ) another set of angular spherical coordinates. Let $\boldsymbol{r} = u(\hat{\boldsymbol{s}})\hat{\boldsymbol{e}}_r + R\hat{\boldsymbol{t}}$ denote the position vector of a point on a reflected ray in the far field, i.e., $R \gg u(\hat{\boldsymbol{s}})$, corresponding to an intersection point Q with the target screen $z = -L$ with $L \gg u(\hat{\boldsymbol{s}})$. Let $\hat{\boldsymbol{r}} = \boldsymbol{r}/|\boldsymbol{r}|$. Applying the far-field approximation, i.e., $\hat{\boldsymbol{r}} = \hat{\boldsymbol{t}}$, the energy balance reads

$$\int_{\mathcal{A}} I(\phi, \theta)\, \mathrm{d}S(\phi, \theta) = \int_{\hat{\boldsymbol{t}}(\mathcal{A})} G(\psi, \chi)\, \mathrm{d}S(\psi, \chi), \tag{15}$$

for any set $\mathcal{A} \subset \mathrm{S}^2$ and image set $\hat{\boldsymbol{t}}(\mathcal{A}) \subset \mathrm{S}^2$. Introducing the stereographic coordinates $\boldsymbol{x}$ and $\boldsymbol{y}$ and subsequently substituting the optical map $\boldsymbol{y} = \boldsymbol{m}(\boldsymbol{x})$, assuming once more that $\det\big(\mathrm{D}\boldsymbol{m}(\boldsymbol{x})\big) > 0$, we obtain the energy constraint

$$I(\phi, \theta)\frac{4}{(1 + |\boldsymbol{x}|^2)^2} = G(\psi, \chi)\frac{4}{(1 + |\boldsymbol{m}(\boldsymbol{x})|^2)^2} \det\big(\mathrm{D}\boldsymbol{m}(\boldsymbol{x})\big), \tag{16}$$

where the angular coordinates (ϕ, θ) and (ψ, χ) still have to be converted to $\boldsymbol{x}$ and $\boldsymbol{y} = \boldsymbol{m}(\boldsymbol{x})$, respectively.

Both Eqs. (14) and (16) are of the generic form $\det\big(\mathrm{D}\boldsymbol{m}(\boldsymbol{x})\big) = F(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x}))$. Combining this equation with (12), we obtain the fully nonlinear elliptic PDE

$$\det\big(\boldsymbol{C}(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x}))\big)F(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x})) = \det\big(\mathrm{D}^2 u_1(\boldsymbol{x}) - \mathrm{D}_{\boldsymbol{xx}} c(x, \boldsymbol{m}(\boldsymbol{x}))\big), \tag{17a}$$

which is a generalized Monge–Ampère equation. The corresponding boundary condition reads

$$\boldsymbol{m}(\partial\mathcal{S}) = \partial\mathcal{T}, \tag{17b}$$

referred to as the transport boundary condition and is a consequence of the relation $\boldsymbol{m}(\mathcal{S}) = \mathcal{T}$, stating that all light from the source reaches the target.

Connection with Optimal Transport In [4] the reflector problem is related to an optimal transport problem. More specifically, for a convex reflector $\mathcal{R}$ it is shown that the ray trace map $\boldsymbol{\gamma} : \hat{\boldsymbol{s}} \mapsto \hat{\boldsymbol{t}}$ (law of reflection) minimizes the transportation cost

$$\boldsymbol{\mu} \mapsto \int_{\mathrm{S}^2} \tilde{c}(\hat{\boldsymbol{s}}, \boldsymbol{\mu}(\hat{\boldsymbol{s}}))\, I(\phi, \theta)\, \mathrm{d}S(\phi, \theta),$$

among all plans $\boldsymbol{\mu}$, i.e., measure preserving maps $\boldsymbol{\mu} : \mathrm{S}^2 \to \mathrm{S}^2$. The function $\tilde{c}(\hat{\boldsymbol{s}}, \hat{\boldsymbol{t}}) = -\log(1 - \hat{\boldsymbol{s}} \cdot \hat{\boldsymbol{t}})$ is the cost function. Therefore, we refer to $c = c(\boldsymbol{x}, \boldsymbol{y})$ as the cost function, regardless of its arguments.

3 Least-Squares Algorithm

We outline a least-squares method to compute the freeform optical surfaces; for a detailed account see, e.g. [2, 5]. Our method is inspired by the least-squares method of Caboussat el al. [6]. The method proceeds in two stages, first we compute the optical map, and subsequently, we compute the shape of the freeform surface(s).

We compute the optical map $\boldsymbol{m}$ from Eq. (12) in a least-squares sense, where the matrix $\boldsymbol{P}(\boldsymbol{x})$ is SPD and satisfies the constraint $\det(\boldsymbol{C}(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x})))F(\boldsymbol{x}, \boldsymbol{m}(\boldsymbol{x})) = \det(\boldsymbol{P}(\boldsymbol{x}))$; cf. (17a). Therefore, we minimize the functional

$$J_{\mathrm{I}}[\boldsymbol{m}, \boldsymbol{P}] = \tfrac{1}{2} \int_{\mathcal{S}} \|\boldsymbol{C}\mathrm{D}\boldsymbol{m} - \boldsymbol{P}\|^2 \,\mathrm{d}\boldsymbol{x}. \tag{18}$$

The norm used is the Frobenius norm. Moreover, to impose the transport boundary condition (17b) we minimize the functional

$$J_{\mathrm{B}}[\boldsymbol{m}, \boldsymbol{b}] = \tfrac{1}{2} \int_{\partial\mathcal{S}} \left|\boldsymbol{m} - \boldsymbol{b}\right|_2^2 \,\mathrm{d}s, \tag{19}$$

where $\boldsymbol{b} : \partial\mathcal{S} \to \partial\mathcal{T}$. To close the numerical model, we combine the functional J_{I} for the interior domain and J_{B} for the boundary in a weighted average as

$$J[\boldsymbol{m}, \boldsymbol{P}, \boldsymbol{b}] = \alpha J_{\mathrm{I}}[\boldsymbol{m}, \boldsymbol{P}] + (1 - \alpha) J_{\mathrm{B}}[\boldsymbol{m}, \boldsymbol{b}] \tag{20}$$

with $0 < \alpha < 1$. Starting from an initial guess $\boldsymbol{m}^0$ we perform the iteration

$$\boldsymbol{b}^{k+1} = \mathrm{argmin}_{\boldsymbol{b} \in \mathcal{B}} J_{\mathrm{B}}[\boldsymbol{m}^k, \boldsymbol{b}], \tag{21a}$$

$$\boldsymbol{P}^{k+1} = \mathrm{argmin}_{\boldsymbol{P} \in \mathcal{P}(\boldsymbol{m}^k)} J_{\mathrm{I}}[\boldsymbol{m}^k, \boldsymbol{P}], \tag{21b}$$

$$\boldsymbol{m}^{k+1} = \mathrm{argmin}_{\boldsymbol{m} \in \mathcal{M}} J[\boldsymbol{m}, \boldsymbol{P}^{k+1}, \boldsymbol{b}^{k+1}]. \tag{21c}$$

The separate minimization steps are over the following spaces

$$\mathcal{B} = \{\boldsymbol{b} \in C^1(\partial\mathcal{S})^2 \big| \boldsymbol{b}(\boldsymbol{x}) \in \partial\mathcal{T}\}, \tag{22a}$$

$$\mathcal{P}(\boldsymbol{m}) = \{\boldsymbol{P} \in C^1(\mathcal{S})^{2\times 2} \big| \boldsymbol{P} \text{ SPD}, \det(\boldsymbol{P}) = \det(\boldsymbol{C}(\cdot, \boldsymbol{m}))F(\cdot, \boldsymbol{m})\}, \tag{22b}$$

$$\mathcal{M} = C^2(\mathcal{S})^2. \tag{22c}$$

The minimization procedure for J_I reduces to a scalar constrained minimization problem for each grid point separately. On the other hand, applying calculus of variations, the minimization problem for J leads to a (coupled) elliptic PDE for the components of $\boldsymbol{m}$. For space discretisation we employ the finite volume method.

Upon convergence of (21), we compute u_1 from (11), also in a least-squares sense. Therefore, we minimize the functional

$$I[u_1] = \tfrac{1}{2}\int_{\mathcal{S}} \left|\nabla_{\boldsymbol{x}} c(\cdot,\boldsymbol{m}) - \nabla u_1\right|^2 \mathrm{d}\boldsymbol{x}. \tag{23}$$

Applying calculus of variations leads to a Neumann problem for u_1, for which we use central differences. Finally, we solve all linear systems using QR-decomposition.

4 Numerical Example

As an example we compute a freeform lens that generates a circular top-hat target illuminance from a Gaussian source emittance. The source and target domains are given by $\mathcal{S} = \mathcal{T} = [-1,1]\times[-1,1]$. The source has emittance $E(\boldsymbol{x}) = A\mathrm{e}^{-10|\boldsymbol{x}|^2}$ and the target plane receives the illuminance $I(\boldsymbol{y})$ given by $I(\boldsymbol{y}) = 1/\pi$ if $|\boldsymbol{y}| \le 1$, otherwise $I(\boldsymbol{y}) = 0$. The constant A is chosen to enforce global energy conservation, i.e., relation (13) should hold for the entire source domain $\mathcal{A} = \mathcal{S}$. The numerically computed lens is shown in Fig. 2. Clearly, the lens surface $z = u_1(\boldsymbol{x})$ closest to the source is convex. To validate the result we have traced 10^7 rays through the lens to compute the target irradiance; a selected ray set is shown. The

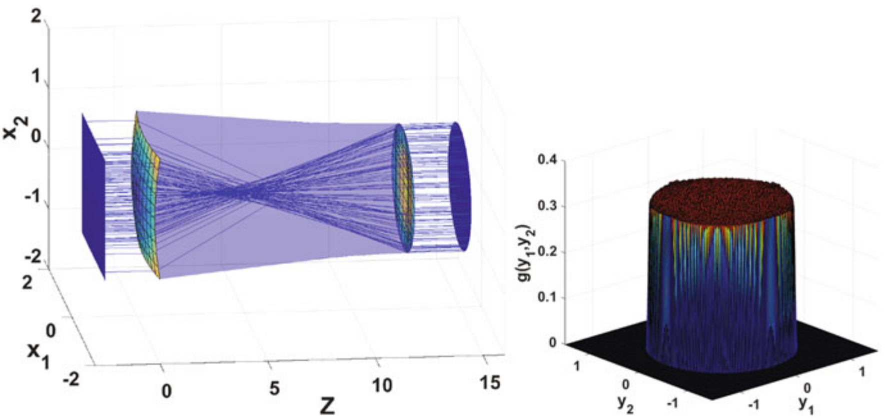


Fig. 2 Computed double freeform lens (left) and target illuminance (right). Parameter values are: $n = 1.5$, $L = 15$ and $\beta = 2\pi$

resulting illuminance is also shown in Fig. 2. We conclude that the computed target illuminance is in good approximation a circular top-hat.

References

1. Luneburg, R.K.: Mathematical Theory of Optics. University of California Press, Berkeley and Los Angeles (1966)
2. Yadav, N.K., ten Thije Boonkkamp, J.H.M., IJzerman, W.L.: A Monge-Ampère problem with non-quadratic cost function to compute freeform lens surfaces. J. Sci. Comput. **80**, 475–499 (2019)
3. Romijn, L.B., ten Thije Boonkkamp, J.H.M., IJzerman, W.L.: Inverse reflector design for a point source and far-field target. J. Comput. Phys. **408**, 109–283 (2020)
4. Glimm, T. and Oliker, V.: Optical design of single reflector systems and the Monge-Kantorovich mass transfer problem, J. Math. Sciences **117**, 4096–4108 (2003)
5. Prins, C., Beltman, R., ten Thije Boonkkamp, J., IJzerman, W. and Tukker, T.: A least-squares method for optimal transport using the Monge-Ampère equation, SIAM J. on Sci. Comput. **37**, B937 - B961 (2015)
6. Caboussat, A., Glowinski, R. and Sorensen, D.: A least-squares method for the numerical solution of the Dirichlet problem for the elliptic Monge-Ampère equation in dimension two. ESAIM: Control, Optimisation and Calculus of Variations **19**, 780–810 (2013)

A Direct Projection to Low-Order Level for p-Multigrid Methods in Isogeometric Analysis

Roel Tielen, Matthias Möller, and Kees Vuik

Abstract Isogeometric Analysis (IgA) can be considered as the natural extension of the Finite Element Method (FEM) to high-order B-spline basis functions. The development of efficient solvers for discretizations arising in IgA is a challenging task, as most (standard) iterative solvers have a detoriating performance for increasing values of the approximation order p of the basis functions. Recently, p-multigrid methods have been developed as an alternative solution strategy. With p-multigrid methods, a multigrid hierarchy is constructed based on the approximation order p instead of the mesh width h (i.e. h-multigrid). The coarse grid correction is then obtained at level $p = 1$, where B-spline basis functions coincide with standard Lagrangian P_1 basis functions, enabling the use of well known solution strategies developed for the Finite Element Method to solve the residual equation. Different projection schemes can be adopted to go from the high-order level to level $p = 1$. In this paper, we compare a direct projection to level $p = 1$ with a projection between each level $1 \leq k \leq p$ in terms of iteration numbers and CPU times. Numerical results, including a spectral analysis, show that a direct projection leads to the most efficient method for both single patch and multipatch geometries.

1 Introduction

Isogeometric Analysis (IgA) [1] can be considered as a natural extension of the Finite Element Method (FEM) to high-order B-spline basis functions. The use of these basis functions enables a highly accurate representation of the geometry. Furthermore, the higher continuity of the basis functions leads to a higher accuracy per degree of freedom compared to FEM [2]. Solving linear systems of equations for

R. Tielen (✉)
Delft University of Technology, Delft, Netherlands
e-mail: r.p.w.m.tielen@tudelft.nl

M. Möller · K. Vuik
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, Netherlands

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_99

discretizations arising in IgA remains, however, a challenging task. The condition number of the system matrices increase exponentially with the approximation order p of the basis functions [3]. Therefore, (standard) iterative methods detoriate for higher values of p which has led to the development of efficient solvers for IgA [4, 5].

Multigrid methods [6, 7] are considered among the most efficient solution techniques for elliptic problems. Within h-multigrid methods, a hierarchy is constructed based on different mesh widths h. At the coarsest level, a correction is obtained by solving the residual equation, which is used to update the fine grid solution. At each level of the multigrid hierarchy, a basic iteration scheme is applied, also known as the smoother. The combination of coarse grid correction and smoothing leads to a highly efficient iterative solver, where the CPU time needed for convergence grows linearly with the number of degrees of freedom. In the context of Isogeometric Analysis, h-multigrid methods have been developed with enhanced smoothers to obtained convergence rates independent of both the mesh width h and approximation order p [8, 9].

As an alternative solution strategy, p-multigrid methods can be adopted. In contrast to h-multigrid methods, a multigrid hierarchy is constructed based on different values of p. As a result, the residual equation is solved at level $p = 1$, where B-spline basis functions coincide with Lagrangian P_1 basis functions, allowing the use of established solution techniques for standard FEM. Equiped with a smoother that is based on an Incomplete LU factorization [10], the resulting p-multigrid method shows convergence rates independent of both h and p [11]. Compared to h-multigrid methods, the coarse grid correction is obtained at $p = 1$. As a result, the overall assembly costs are lower for higher values of p due to a significant reduction of the number of non zero entries. For a detailed comparison with h-multigrid methods, the authors refer to [11].

In recent papers by the authors, a p-multigrid hierarchy has been constructed for all levels k, where $1 \leq k \leq p$. However, the scheme could be adopted in which the residual at level p is directly projected to the coarse level ($p = 1$). In this paper, we compare both schemes in terms of spectral properties, iteration numbers and CPU times for both a single patch and multipatch geometry. This paper is organized as follows: Sect. 2 describes the considered model problem and IgA discretization. The p-multigrid method, together with the different projection schemes studied in this paper, are described in detail in Sect. 3. Numerical results for the considered benchmark problems, including a spectral analysis, iteration numbers and CPU times are presented in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Model Problem and IgA Discretization

As a model problem, we consider the convection-diffusion-reaction (CDR) equation on a connected, Lipschitz domain $\Omega \subset \mathbb{R}^2$. Defining $\mathcal{V} = H_0^1(\Omega)$ as the Sobolev space $H^1(\Omega)$ with functions that vanish on $\partial\Omega$, the variational form of the CDR-equation becomes: Find $u \in \mathcal{V}$ such that

$$a(u, v) = (f, v) \qquad \forall v \in \mathcal{V}, \tag{1}$$

where

$$a(u, v) = \int_\Omega (\mathbf{D}\nabla u) \cdot \nabla v + (\mathbf{v} \cdot \nabla u)v + Ruv \, \mathrm{d}\Omega \;\text{ and }\; (f, v) = \int_\Omega f v \, \mathrm{d}\Omega. \tag{2}$$

Here, $\mathbf{D}$ denotes the diffusion tensor, $\mathbf{v}$ a divergence-free velocity field and R a reaction term. Furthermore, we have $f \in L^2(\Omega)$ and $u = 0$ on the boundary $\partial\Omega$. The physical domain Ω is then parameterized by a geometry map

$$\mathbf{F} : \hat{\Omega} \to \Omega, \qquad \mathbf{F}(\boldsymbol{\xi}) = \mathbf{x}. \tag{3}$$

The geometry map $\mathbf{F}$ describes an invertible mapping connecting the parameter domain $\hat{\Omega} = (0, 1)^2$ with the physical domain Ω. In case Ω cannot be described by a single geometry map, the domain is divided into a collection of non-overlapping subdomains $\Omega^{(d)}$, where $1 \leq d \leq D$. A family of geometry maps $\mathbf{F}^{(d)}$ is then defined to parameterize each subdomain separately and we refer to Ω as a multipatch domain consisting of D patches.

In this paper, the tensor product of univariate B-spline functions of order p is used for the spatial discretization. Univariate B-spline basis functions are defined on the one-dimensional parameter domain $\hat{\Omega} = (0, 1)$ and are uniquely determined by a knot vector $\Xi = \{\xi_1, \xi_2, \ldots, \xi_{N+p}, \xi_{N+p+1}\}$, consisting of a sequence of non-decreasing knots $\xi_i \in \hat{\Omega}$ with, in this paper, constant knot span size or mesh width h. Here, N denotes the number of basis functions of order p defined by this knot vector. B-spline basis functions are defined recursively by the Cox de Boor formula [12]. The resulting B-spline basis functions $\phi^i_{h,p}$ are non-zero on the interval $[\xi_i, \xi_{i+p+1})$ and possess the partition of unity property. In this paper, an *open* knot vector is considered, implying that the first and last knots are repeated $p + 1$ times. As a consequence, the basis functions considered are globally C^{p-1} continuous and interpolatory only at the two end points; see also Fig. 1.

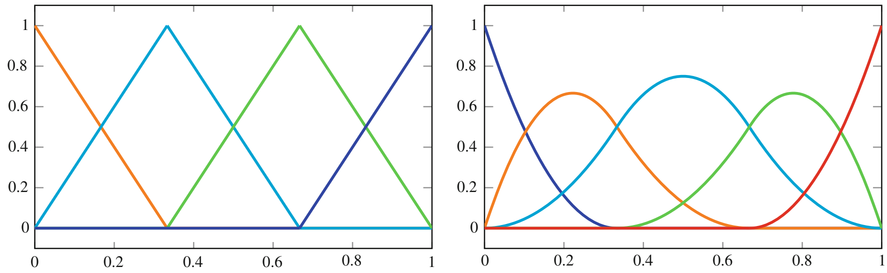


Fig. 1 Univariate linear (left) and quadratic (right) B-spline basis functions based on the knot vectors $\Xi_1 = \{0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1\}$ and $\Xi_2 = \{0, 0, 0, \frac{1}{3}, \frac{2}{3}, 1, 1, 1\}$, respectively

The solution u of Eq. (1) is then approximated by a linear combination of bivariate B-spline basis functions:

$$u(\boldsymbol{\xi}) \approx u_{h,p}(\boldsymbol{\xi}) = \sum_{i=1}^{N_{\text{dof}}} c^i \Phi^i_{h,p}(\boldsymbol{\xi}), \tag{4}$$

where $\Phi^i_{h,p}(\boldsymbol{\xi}) = \phi^{i_1}_{h,p}(\xi_1)\phi^{i_2}_{h,p}(\xi_2)$ and N_{dof} denotes the number of bivariate B-spline functions, where $N_{\text{dof}} = N^2$. Defining $\mathcal{V}_{h,p}$ as the span of all bivariate B-spline basis functions, the Galerkin formulation of (1) becomes: Find $u_{h,p} \in \mathcal{V}_{h,p}$ such that

$$a(u_{h,p}, v_{h,p}) = (f, v_{h,p}) \qquad \forall v_{h,p} \in \mathcal{V}_{h,p}. \tag{5}$$

Equation (5) can be written as a linear system resulting from this discretization with B-spline basis functions of approximation order p and mesh width h. For a more detailed description of the spatial discretization in IgA, the authors refer to [1].

3 *p*-Multigrid Method

To solve Eq. (5) efficiently, a p-multigrid method is adopted. Starting from $\mathcal{V}_{h,1}$, a sequence of spaces $\mathcal{V}_{h,1}, \ldots, \mathcal{V}_{h,p}$ is obtained by applying refinement in p. As C^{p-1} continuous basis functions are considered on all levels of the multigrid hierarchy, these spaces are not nested.

Starting from an initial guess $\mathbf{u}^{(0)}_{h,p}$, a single step of the two-grid correction scheme for the p-multigrid method consists of the following steps [13]:

$$\mathbf{u}^{(0)}_{h,p} = \mathbf{u}^{(0)}_{h,p} + \mathcal{S}_{h,p}\left(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}^{(0)}_{h,p}\right), \tag{6}$$

$$\mathbf{r}_{h,p-1} = \mathcal{I}^{p-1}_{p}\left(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}^{(0)}_{h,p}\right). \tag{7}$$

$$\mathbf{e}_{h,p-1} = \left(\mathbf{A}_{h,p-1}\right)^{-1}\mathbf{r}_{h,p-1}, \tag{8}$$

$$\mathbf{u}^{(0)}_{h,p} = \mathbf{u}^{(0)}_{h,p} + \mathcal{I}^{p}_{p-1}\left(\mathbf{e}_{h,p-1}\right), \tag{9}$$

$$\mathbf{u}^{(1)}_{h,p} = \mathbf{u}^{(0)}_{h,p} + \mathcal{S}_{h,p}\left(\mathbf{f}_{h,p} - \mathbf{A}_{h,p}\mathbf{u}^{(0)}_{h,p}\right), \tag{10}$$

Here, $\mathcal{S}_{h,p}$ denotes a single smoothing step applied to the high-order problem, while $\mathcal{I}^{p-1}_{p}$ and $\mathcal{I}^{p}_{p-1}$ denote the restriction and prolongation operator, respectively. The coarse grid operator $\mathbf{A}_{h,p-1}$ is obtained by rediscretizing equation (1).

Recursive application of this scheme on Eq. (8) until level $p = 1$ is reached, results in a V-cycle. As the coarsest problem in p-multigrid can become large

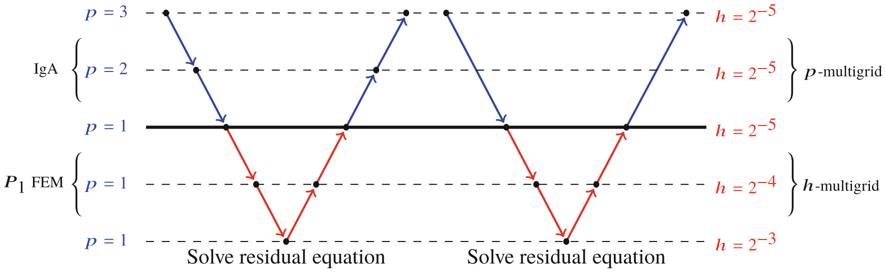


Fig. 2 Illustration of both an indirect (left) and direct (right) projection scheme within p-multigrid

for small values of h, a single V-cycle of a standard h-multigrid method (with canonical prolongation, weighted restriction and a single smoothing step) is adopted to approximately solve the coarse grid problem in our p-multigrid scheme.

In this paper, we also consider a direct projection from the high-order level to level $p = 1$. Both considered multigrid schemes, referred to as an indirect and direct projection scheme, are shown in Fig. 2.

The operators to project between different p-levels are based on an L_2 projection and have been used extensively in the literature [14–16]. The prolongation and restriction operator are defined, respectively, as follows:

$$\mathcal{I}_{p-1}^{p}(\mathbf{v}_{p-1}) = (\mathbf{M}_p)^{-1}\mathbf{P}_{p-1}^{p}\ \mathbf{v}_{p-1} \qquad \mathcal{I}_{p}^{p-1}(\mathbf{v}_p) = (\mathbf{M}_{p-1})^{-1}\mathbf{P}_{p}^{p-1}\ \mathbf{v}_p, \quad (11)$$

with the mass matrix $\mathbf{M}_p$ and transfer matrix $\mathbf{P}_{p-1}^{p}$ given by:

$$(\mathbf{M}_p)_{(i,j)} := \int_\Omega \Phi_{h,p}^{i}\Phi_{h,p}^{j}\ \mathrm{d}\Omega, \qquad (\mathbf{P}_{p-1}^{p})_{(i,j)} := \int_\Omega \Phi_{h,p}^{i}\Phi_{h,p-1}^{j}\ \mathrm{d}\Omega. \quad (12)$$

The choice of the prolongation and restriction operator leads to a non-symmetric multigrid method. Choosing the prolongation and restriction operator as the transpose of eachother would restore symmetry. Numerical experiments, not presented in this paper, show, however, that this leads to a less robust p-multigrid method. To prevent the explicit solution of a linear system of equations for each projection step, the consistent mass matrix $\mathbf{M}_p$ in both transfer operators is replaced by its lumped counterpart $\mathbf{M}_p^L$ by applying row-sum lumping, i.e. $(\mathbf{M}_p^L)_{(i,i)} = \sum_{j=1}^{N_{\text{dof}}}(\mathbf{M}_p)_{(i,j)}$. Note that in IgA the mass matrix can easily be lumped due to the non-negativity of the B-spline basis functions. It was shown in [11] that the use of a lumped mass matrix in Eq. (13) hardly influences the convergence behaviour or accuracy of the resulting p-multigrid methods. Note that, alternatively, the mass matrix could be inverted efficiently by exploiting the tensor product structure, see [18].

Since the use of standard smoothers (i.e. Gauss–Seidel) within p-multigrid leads to convergence rates which detoriate for higher values of p [13], we adopt a smoother based on an ILUT factorization. This factorization is determined completely by a tolerance τ and fillfactor m, which are chosen such that the number of nonzeros is approximately the same as for the orignal operator $\mathbf{A}_{h,p}$. We applied this smoother successfully within p-multigrid methods to solve linear systems arising in IgA [11].

4 Numerical Results

To assess the quality of both projection schemes, two benchmarks are considered. For the first benchmark, the model problem (1) is considered with coefficients:

$$\mathbf{D} = \begin{bmatrix} 1.2 & -0.7 \\ -0.4 & 0.9 \end{bmatrix}, \qquad \mathbf{v} = \begin{bmatrix} 0.4 \\ -0.2 \end{bmatrix}, \qquad R = 0.3. \tag{13}$$

Here, Ω is chosen to be the unit square, i.e. $\Omega = [0, 1]^2$, described by a single patch. The second benchmark is Poisson's equation ($\mathbf{D}$ is the identity matrix) on an L-shaped domain ($\Omega = \{[-1, 1] \times [-1, 1]\} \backslash \{[0, 1] \times [0, 1]\}$), consisting of 4 patches. The resulting linear systems are then solved with the proposed p-multigrid methods. At level $p = 1$, coarsening in h is applied until $h = 2^{-3}$, corresponding to 81 degrees of freedom.

To investigate the interplay between smoothing and the coarse grid correction, the error reduction factors when applying a single smoothing step (only on the finest level) or coarse grid correction (without smoothing) have been determined for both projection schemes. This analysis has been performed before in literature, in the context of h-multigrid methods [17]. Figure 3 (left) denotes the error reduction factors of the generalized eigenvectors $\mathbf{v}_j$ ($j = 1, \ldots N_{\mathrm{dof}}$) of the operator $\mathbf{A}_{h,p}$ for $p = 4$ and $h = 2^{-5}$. For both a direct and indirect projection, the smoother and coarse grid correction are complementary to eachother, where the smoother is effective for the high-frequency components and the coarse grid correction for the low frequency components. Remarkably, the coarse grid correction with a direct projection is not only more efficient in terms of less computational work, but also leads to lower reduction factors. Note that, no smoothing is applied here on the coarser levels.

To further analyze the performance of both projection schemes, the asymptotic convergence rate of the resulting p-multigrid methods has been determined. For any multigrid method, the asymptotic convergence rate is given by the spectral radius ρ of the iteration matrix describing the effect of a single V-cycle. The spectra of the iteration matrices for both projection schemes are shown in Fig. 3 (right). For comparison, a circle with radius 0.025 has been added to the plot. Visually, both spectra are almost identical, which is also confirmed by the obtained spectral radia: $\rho_1 = 0.02032$ and $\rho_2 = 0.02035$ for a direct and indirect projection, respectively, implying an equally efficient p-multigrid method for both configurations.

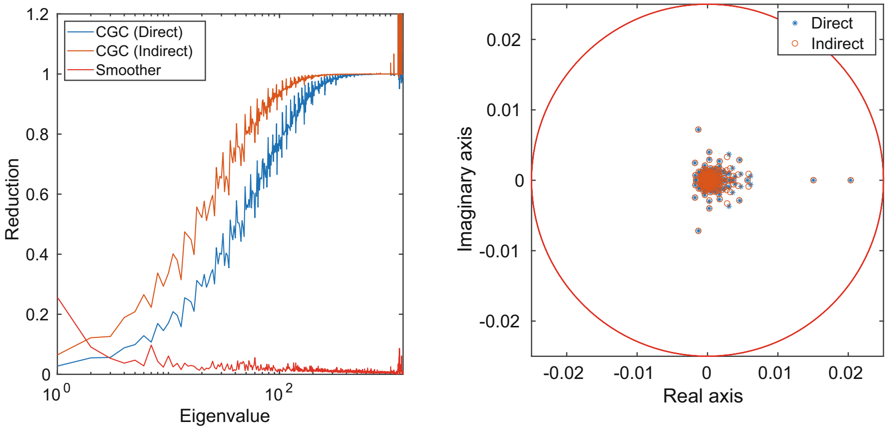


Fig. 3 Error reduction in $\mathbf{v}_j$ (left) and the spectrum of the iteration matrix (right) for the first benchmark obtained with both projection schemes $\left(p = 4, h = 2^{-5}\right)$

Table 1 Number of iterations needed to achieve convergence for both benchmarks when applying a direct or indirect projection for different values of h and p

	$p = 2$		$p = 3$		$p = 4$		$p = 5$	
	Direct	Indirect	Direct	Indirect	Direct	Indirect	Direct	Indirect
(a) CDR-equation on the unit square								
$h = 2^{-5}$	5	5	4	4	3	3	3	3
$h = 2^{-6}$	5	5	4	4	4	3	4	4
$h = 2^{-7}$	5	5	4	4	4	4	4	4
$h = 2^{-8}$	5	5	4	4	4	4	4	4
(b) Poisson's equation on an L-shaped domain								
$h = 2^{-5}$	6	6	6	5	5	5	5	4
$h = 2^{-6}$	7	7	6	5	5	5	5	4
$h = 2^{-7}$	7	7	6	6	6	5	6	4
$h = 2^{-8}$	8	8	6	6	7	6	6	5

Table 1 shows the number of iterations needed to achieve convergence for both benchmarks, respectively. For all numerical experiments, the initial guess $\mathbf{u}_{h,p}^{(0)}$ is chosen randomly, where each entry is sampled from a uniform distribution on the interval $[-1, 1]$. The p-multigrid iteration is considered converged when the initial residual has decreased with a factor of 10^8. Note that for both projection schemes and benchmarks, the number of iterations is robust in both the mesh width h and the approximation order p and similar for all configurations. For the first benchmark, with $p = 4$ and $h = 2^{-5}$, the same number of iterations is needed, as expected from our spectral analysis. Note that for the multipatch geometry, more iterations are required to achieve convergence. This behaviour for p-multigrid methods has been observed and analyzed in literature by the authors, see [19].

Table 2 CPU timings (secs) for the first benchmark for different values of h and p

	$p = 2$		$p = 3$		$p = 4$		$p = 5$	
	Direct	Indirect	Direct	Indirect	Direct	Indirect	Direct	Indirect
(a) Set-up times								
$h = 2^{-5}$	0.2	0.2	0.3	0.4	0.5	0.9	0.9	1.8
$h = 2^{-6}$	0.6	0.6	1.1	1.6	2.1	3.6	3.7	7.5
$h = 2^{-7}$	2.5	2.5	4.6	6.4	8.5	14.9	16.7	35.2
$h = 2^{-8}$	10.0	9.9	18.7	26.2	36.1	65.7	66.4	142.9
(b) Solving times								
$h = 2^{-5}$	0.004	0.004	0.004	0.005	0.004	0.007	0.005	0.01
$h = 2^{-6}$	0.01	0.01	0.01	0.02	0.02	0.03	0.02	0.05
$h = 2^{-7}$	0.04	0.04	0.05	0.07	0.07	0.1	0.1	0.2
$h = 2^{-8}$	0.2	0.2	0.2	0.3	0.3	0.5	0.4	0.8

To compare the computational costs of both approaches, CPU timings have been determined for the first benchmark. A serial implementation in the C++ library G+Smo is considered on an Intel(R) Core(TM) i7-8650 CPU (1.90GHz). Table 2 shows the measured set-up and solver times (in seconds). Although for both projection schemes, the set-up and solver time scales linearly with the number of degrees of freedom, the CPU times obtained with a direct projection scheme are significantly lower. Furthermore, the relative difference increases for higher values of p, as the number of levels in the p-multigrid hierarchy grows when adopting an indirect projection scheme: for $p = 5$ a reduction of the set-up and solving times of around 50% is achieved.

5 Conclusions

Recently, the use of p-multigrid methods has become more popular in solving linear systems of equations arising in Isogeometric Analysis. In this paper, various schemes to set up the p-multigrid hierarchy have been compared. In particular, a direct projection to level $p = 1$ has been compared with constructing a hierarchy for each order $1 \leq k \leq p$. Numerical results, presented for the CDR-equation on the unit square and Poisson's equation on an L-shaped multipatch domain, show that in terms of iteration numbers both projection schemes lead to (almost) identical results. This is also confirmed by the performed spectral analysis. However, CPU timings show that a direct projection scheme leads to the most efficient solution strategy, reducing the set-up and solving times up to a factor of 2 for higher values of p.

References

1. T.J.R. Hughes, J.A. Cottrell and Y. Bazilevs. Isogeometric analysis: CAD, finite elements, NURBS, exact geometry and mesh refinement. Computer Methods in Applied Mechanics and Engineering, **194**, pp. 4135–4195, 2005
2. T.J.R. Hughes, A. Reali and G. Sangalli. Duality and unified analysis of discrete approximations in structural dynamics and wave propagation: Comparison of p-method finite elements with k-method NURBS. Computer Methods in Applied Mechanics and Engineering, **197**(49–50), pp. 4104–4124, 2008
3. K.P.S. Gahalaut, J.K. Kraus and S.K. Tomar. Multigrid methods for isogeometric discretizations. Computer Methods in Applied Mechanics and Engineering, **253**, pp. 413–425, 2013
4. L. Beirao da Veiga, D. Cho, L.F. Pavarino and S. Scacchi. Overlapping Schwarz methods for isogeometric analysis. SIAM Journal on Numerical Analysis, **50**(3), pp. 1394–1416, 2012
5. M. Donatelli, C. Garoni, C. Manni, S. Capizzano and H. Speleers. Symbol-based multigrid methods for Galerkin B-spline isogeometric analysis. SIAM Journal on Numerical Analysis, **55**(1), pp. 31–62, 2017
6. A. Brandt. Multi-level adaptive solutions to boundary-value problems. Mathematics of Computation, **31**(138): pp. 333–390, 1977
7. W. Hackbusch. Multi-grid methods and applications. Springer, Berlin, 1985
8. C. Hofreither, S. Takacs and W. Zulehner. A robust multigrid method for Isogeometric Analysis in two dimensions using boundary correction. Computer Methods in Applied Mechanics and Engineering, **316**, pp. 22–42, 2017
9. A. de la Riva, C. Rodrigo and F. Gaspar. A robust multigrid solver for Isogeometric Analysis based on multiplicative Schwarz smoothers. SIAM Journal of Scientific Computing, **41**(5), pp. 321–345, 2019
10. Y. Saad. ILUT: A dual threshold incomplete LU factorization. Numerical Linear Algebra with Applications, **1**(4), pp. 387–402, 1994
11. R. Tielen, M. Möller, D. Göddeke and C. Vuik. p-multigrid methods and their comparison to h-multigrid methods within Isogeometric Analysis. Computer Methods in Applied Mechanics and Engineering, **372**, 2020
12. C. De Boor. A practical guide to splines. 1st edition. Springer-Verlag, New York, 1978
13. R. Tielen, M. Möller and C. Vuik. Efficient multigrid based solvers for Isogeometric Analysis. Proceedings of the 6th European Conference on Computational Mechanics and the 7th European Conference on Computational Fluid Dynamics, Glasgow, UK, 2018.
14. S.C. Brenner and L.R. Scott. The mathematical theory of finite element methods, Texts in Applied Mathematics, Springer, New York, 1994
15. R.S. Sampath and G. Biros. A parallel geometric multigrid method for finite elements on octree meshes, SIAM Journal on Scientific Computing, **32**(3): pp. 1361–1392, 2010
16. W.L. Briggs, V. E. Henson and S.F. McCormick. A Multigrid Tutorial 2nd edition, SIAM, Philadelphia, 2000.
17. C. Hofreither and W. Zulehner. Spectral Analysis of Geometric Multigrid Methods for Isogeometric Analysis. Numerical Methods and Applications, **8962**, pp. 123–129, 2015
18. L. Gao and V. Calo, Fast isogeometric solvers for explicit dynamics. Computer Methods in Applied Mechanics and Engineering, 274, pp. 19–41, 2014
19. R. Tielen, M. Möller and C. Vuik. Efficient p-Multigrid based solvers for multipatch geometries in Isogeometric Analysis. Proceedings of the 3rd conference on Isogeometric Analysis and Applications (IgAA 2018), Delft, the Netherlands, 2018

The Concept of Prehandling as Direct Preconditioning for Poisson-Like Problems

Dustin Ruda, Stefan Turek, Peter Zajac, and Dirk Ribbrock

Abstract To benefit from current trends in HPC hardware, such as increasing availability of low precision hardware, we present the concept of prehandling as a direct way of preconditioning and the hierarchical finite element method which is exceptionally well-suited to apply prehandling to Poisson-like problems, at least in 1D and 2D. Such problems are known to cause ill-conditioned stiffness matrices and therefore high computational errors due to round-off. We show by means of numerical results that by prehandling via the hierarchical finite element method the condition number can be significantly reduced (while advantageous properties are preserved) which enables us to obtain sufficiently accurate solutions to Poisson-like problems even if lower computing precision (i.e. single or half precision format) is used.

1 Motivation

When PDEs are solved numerically by the finite element method, the resulting error $u - \tilde{u}_h$ can be subdivided into two different types of errors by means of the identity $u - \tilde{u}_h = (u - u_h) + (u_h - \tilde{u}_h)$, whereby u, u_h and $\tilde{u}_h$ denote the exact solution, the exact solution to the discrete problem and the actual numerical solution respectively. On the one hand, one obtains the discretization error $u - u_h$ depending on the smoothness of the exact solution u and the choice of the finite element space. If (bi)linear shape functions are used (P1 or Q1), the discretization error satisfies $\|u - u_h\| = \mathcal{O}\left(h^2\right)$ with respect to the L^2-norm. On the other hand, roundoff errors cause the computational error $u_h - \tilde{u}_h$ depending on the data error, that is at least equal to the machine accuracy (TOL), amplified by the condition number of the stiffness matrix $\kappa(A)$. To be more precise, it follows from perturbation theory of

D. Ruda · S. Turek (✉) · P. Zajac · D. Ribbrock
TU Dortmund, Chair of Applied Mathematics and Numerics (LS3), Dortmund, Germany
e-mail: dustin.ruda@math.tu-dortmund.de; stefan.turek@math.tu-dortmund.de; peter.zajac@math.tu-dortmund.de; dirk.ribbrock@math.tu-dortmund.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_100

linear systems that we have $\|u_h - \tilde{u}_h\| \approx \kappa(A) \cdot \mathrm{TOL}$ which holds true sharply. It is known that in the case of Poisson's equation the condition number of the related stiffness matrix satisfies $\kappa(A) = \mathcal{O}\left(h^{-2}\right)$.

Consequently, we face a dilemma: The finer the grid the lower the discretization error the higher the computational error. Indeed, if the grid width falls below a certain value, the total error increases because the computational error becomes dominant as shown in Fig. 1. This value is roughly reached at the intersection of the discretization and the computational error at $h \approx \sqrt{\kappa(A) \cdot \mathrm{TOL}}$, yielding $h \approx \sqrt[4]{\mathrm{TOL}}$ if we substitute the condition number by its approximate value h^{-2}. Thus, in order to make sensible use of lower, i.e. single or even half precision (accelerator) hardware, it is indispensable to utilise sophisticated methods to decrease the condition number. Our approach, the method of prehandling, is presented in the following sections.

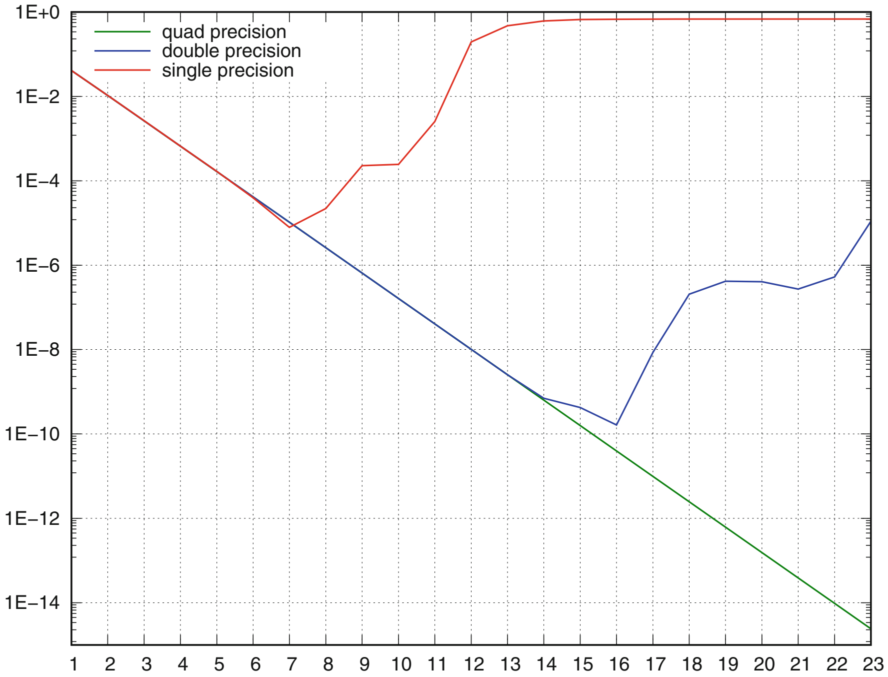


Fig. 1 L^2-error when solving Poisson's equation with FEM as a function of the refinement level, i.e. $h = 2^{-\text{level}}$, in 1D using single-, double- and quad precision

2 The Concept of Prehandling

By ‘prehandling’ we denote a method of directly manipulating linear systems of equations inspired by but different from conventional preconditioning. The central idea consists in transforming the original linear system, given as $Ax = b$, into an equivalent form $\tilde{A}\tilde{x} = \tilde{b}, B\tilde{x} = x$ with more advantageous properties. The difference between preconditioning and prehandling can be easily shown by the example of the Richardson iteration. The preconditioned version is

$$x^{I+1} = x^I - C^{-1}\left(Ax^I - b\right) , \tag{1}$$

whereas by applying prehandling one obtains

$$x^{I+1} = x^I - \left(C^{-1}Ax^I - C^{-1}b\right) = x^I - \left(\tilde{A}x^I - \tilde{b}\right) . \tag{2}$$

Thus, prehandling can be seen as an explicit form of preconditioning. Assuming exact arithmetic and using the exact application of C^{-1}, both methods yield the same iteration vectors x^I for all I. However, the methods (1) and (2) can yield significantly different results when finite precision arithmetic is applied, especially if the matrix A is ill-conditioned.

As mentioned in Sect. 1, finite element stiffness matrices arising from Poisson’s equation are highly ill-conditioned. Yet, an advantageous property is their sparse structure. Via prehandling we intend to reduce the condition number while preserving the sparsity of the matrix. A lower condition number is desirable because it enables us to obtain relevant solutions using fast lower precision hardware and reduces the number of iterations when solving the linear system. To sum up, the three central requirements for the prehandled system are:

1. Strong decrease of the condition number, $\kappa(\tilde{A}) \ll \kappa(A)$.
2. The matrix $\tilde{A}$ is only moderately less sparse than A.
3. There is an efficient transformation (in $\mathcal{O}(n \log n)$ operations for n unknowns) to $\tilde{A}, \tilde{b}$ and the solution to the original system x (via $x = B\tilde{x}$).

A common method is approximating the inverse of A by a matrix $C \approx A^{-1}$ and computing $\tilde{A} = CA$. For this purpose, one can use e.g. matrix splitting ($C = D^{-1}, (D + L)^{-1}$), incomplete LU-Decomposition (ILU) or the Sparse Approximate Inverse (SPAI) but the requirements 1 and 2 are not satisfied by any of these methods.

A promising technique to meet the demands (at least in 1D and 2D) is the hierarchical finite element method presented in the following section.

3 The Hierarchical Finite Element Method

The hierarchical finite element method (also referred to as hierarchical basis multigrid method, abbr.: HFEM) has been known since the 1980s and was developed and analysed by H. Yserentant et al. in [1], amongst others. The main idea and aspects of the realisation as well as the properties that make this method a proper choice for prehandling are shortly outlined.

3.1 Idea and Realisation

In order to apply this method, a nested sequence of refinements of an initial triangulation is required. The general idea is the usage of a hierarchical instead of a nodal basis. This means that basis functions of coarser grids are reused in the course of the refinement. Figure 2 shows nodal compared to hierarchical bases in one dimension. This concept can be straightforwardly applied to higher dimensions, too.

It seems more complicated to assemble the stiffness matrix with respect to a hierarchical basis since many basis functions have a greater support, but it is in fact not necessary to compute the matrix itself in the first place if the stiffness matrix with respect to a nodal basis is known. Instead, we can transform the nodal basis representation to a hierarchical basis representation via a matrix S. It is computed as the product $S = S_j S_{j-1} \cdots S_1$, whereby each factor is associated with one step of refinement. More precisely, multiplying by S_k yields the values of level $k - 1$ basis functions at the new nodes of level k if the values on the coarser grid are known. In other words, the matrix S_k corresponds to the prolongation regarding multigrid methods. The matrices S_k are concretely computed as follows: They are identity matrices with additional entries in the rows that belong to the newly added nodes of level k. When using a triangular mesh with linear basis functions (P1) and

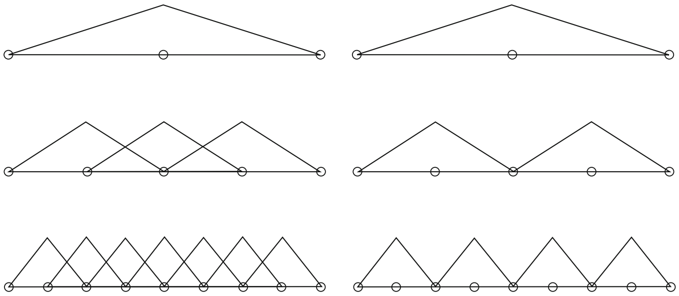


Fig. 2 Nodal bases (left) and hierarchical bases (right) in the one-dimensional case assuming homogeneous Dirichlet boundary conditions. Note that for the hierarchical bases only the newly added basis functions of the respective meshes are depicted. Source: [2]

uniform refinement (subdividing each triangle into four congruent triangles), each newly added node of level k with index i has two neighbouring nodes on level $k-1$, which are denoted by $n_1(i), n_2(i)$, and the ith row of S_k is adjusted according to

$$S_k(i, n_1(i)) = S_k(i, n_2(i)) = \frac{1}{2}\,. \tag{3}$$

When using a rectangular 2D mesh with bilinear basis functions (Q1), however, uniform refinement generates two different types of new nodes, namely midpoints of edges with two and midpoints of faces with four neighbouring nodes $n_j(i)$. In this case one needs to apply

$$S_k(i, n_j(i)) = \begin{cases} \frac{1}{2}\,, j = 1, 2\,, & \text{if } x_i \text{ is midpoint of edge} \\ \frac{1}{4}\,, j = 1, \ldots, 4\,, & \text{if } x_i \text{ is midpoint of face} \end{cases} \tag{4}$$

to achieve a correct interpolation.

Due to this construction, S is a very sparse block unit lower triangular matrix. If by $\hat{A}\hat{u} = \hat{b}$ we denote the system with respect to the nodal basis, we obtain the representation with respect to the hierarchical basis $Au = b$ by means of the transformation

$$A = S^{\mathsf{T}} \hat{A} S\,,\ b = S^{\mathsf{T}} \hat{b}\,,\ \hat{u} = Su\,. \tag{5}$$

3.2 Properties

The remarkable property about the hierarchical finite element method is that the condition number of the emerging matrix is significantly lower in comparison to standard finite elements. It was shown in [1] that the spectral condition number of the matrix with respect to the hierarchical basis satisfies

$$\kappa(A) = O\left(\left(\log \frac{1}{h}\right)^2\right) \tag{6}$$

in the one- and two-dimensional case which is a strong improvement compared to $O\left(h^{-2}\right)$. Furthermore, the matrix A is obviously denser than $\hat{A}$, but the number of nonzero entries per row is low enough to be referred to as sparse and due to the sparse structure of the matrix S, the transformation can be realised efficiently. Numerical results that show the reduction of the condition number and the sparsity of the transformed matrix (approx. 16 nonzero entries per row—if an orthogonal mesh is used—compared to 9 without prehandling) are given in Sect. 4, Table 1.

In conclusion, the hierarchical finite element method, if used according to the concept of prehandling, satisfies the required properties, at least in one and two dimensions. In the three-dimensional case, though, the same improvement of the condition number cannot be achieved. Instead, it is shown in [3] that $\kappa(A) = \mathcal{O}\left(\frac{1}{h}\log\frac{1}{h}\right)$ respectively $\mathcal{O}\left(\frac{1}{h}\right)$ if further prehandling is used.

3.3 Additional Prehandling via Partial Cholesky Decomposition

If Poisson's equation with a discontinuous coefficient ϱ

$$-\nabla \cdot (\varrho \nabla u) = f \ \text{ in } \Omega\ , \tag{7}$$

whereby $\varrho(x, y) = 1$ in a subdomain $\Omega_1 \subset \Omega$ and $\varrho(x, y) = \varrho \gg 1$ in $\Omega \setminus \Omega_1$, is considered, the condition number of the stiffness matrix with respect to a nodal basis additionally depends on the ratio $\frac{\max(\varrho)}{\min(\varrho)}$ (denoted by $\Delta\varrho$) in the form of $\kappa(\hat{A}) = \mathcal{O}\left(\Delta\varrho \cdot h^{-2}\right)$. Especially in this case but also in the case of standard Poisson's equation ($\varrho = 1$) there is a powerful way for further prehandling using a partial Cholesky decomposition presented in [1, 2].

By A_0 we denote the part of the matrix A (which is assumed to be represented with respect to a hierarchical basis) that corresponds to the coarse grid and the rest of it by $\tilde{A}$ and compute the following Cholesky decomposition

$$\begin{pmatrix} A_0 & 0 \\ 0 & \operatorname{diag}(\tilde{A}) \end{pmatrix} = LL^{\mathsf{T}}\ . \tag{8}$$

Consequently, we get the additionally prehandled matrix and right hand side as $L^{-1}AL^{-\mathsf{T}}$ respectively $L^{-1}b$. Note that the solution needs to be transformed to the nodal basis representation by multiplication with $SL^{-\mathsf{T}}$.

It shows in our numerical test on an orthogonal 2D mesh (see Sect. 4, Table 2) that the condition number of $L^{-1}AL^{-\mathsf{T}}$ is now virtually independent of $\Delta\varrho$. The only condition is that the coefficient ϱ is constant within the interior of the elements of the coarse grid.

Since the matrix A_0 is very small compared to A, the computational cost of the Cholesky decomposition (8) is low. On the other hand, the further prehandling only works to the disadvantage of the sparsity, but if the coarse grid is not chosen too fine, this effect is not excessive (approx. 16–25 nonzero entries per row if the coarse grid width is greater or equal to $h_0 = 1/8$) as one can see in Sect. 4, Tables 1 and 2.

4 Numerical Results

In order to validate the presented methods, they were practically applied to the Poisson-like equation (7) in the unit square $\overline{\Omega} = [0, 1]^2$ with $f = 1$ and

$$\varrho(x, y) = \begin{cases} \varrho\ , & \text{if } (x, y) \in \left[\frac{1}{4}, \frac{3}{4}\right]^2 \\ 1\ , & \text{else} \end{cases} . \tag{9}$$

Two different widths h of the fine grid and three widths h_0 of the coarse grid in each case were chosen and the density measured as the average number of nonzero entries per row (NNZ/Row) and the spectral condition number of the respective matrix (cond) as well as the number of CG-iterations (NOI) necessary to reach a relative residual less than 10^{-6} were determined for the hierarchical method with and without the additional Cholesky prehandling and the plain finite element method for comparison. Q1 finite elements on a grid consisting of squares were used. Table 1 shows the results in the case $\varrho = 1$ (which yields the standard Poisson's equation) and Table 2 in the case $\varrho = 10^6$.

One can observe a vast decrease of the condition number and thus the number of iterations when the matrix is transformed to a hierarchical basis representation when Poisson's equation is considered. The method of further prehandling by a partial Cholesky decomposition turns out to be very robust with respect to ϱ in sharp contrast the other listed methods as Table 2 shows.

Table 1 Results for Poisson's equation ($\varrho = 1$)

		HFEM			HFEM + Chol.			FEM		
h	h_0	$\frac{\text{NNZ}}{\text{Row}}$	Cond	NOI	$\frac{\text{NNZ}}{\text{Row}}$	Cond	NOI	$\frac{\text{NNZ}}{\text{Row}}$	Cond	NOI
1/64	1/4	15.61	23.39	27	16.20	20.92	23	8.81	829.86	63
	1/8	15.38	27.51	27	25.25	15.21	20			
	1/16	14.51	82.96	32	100.12	9.62	14			
1/128	1/4	16.68	31.57	32	16.99	28.76	27	8.91	3319.93	127
	1/8	16.56	33.30	33	21.79	22.08	24			
	1/16	16.11	92.21	37	65.40	15.44	19			

Table 2 Results for the Poisson-like equation ($\varrho = 10^6$)

		HFEM			HFEM + Chol.			FEM		
h	h_0	$\frac{\text{NNZ}}{\text{Row}}$	Cond	NOI	$\frac{\text{NNZ}}{\text{Row}}$	Cond	NOI	$\frac{\text{NNZ}}{\text{Row}}$	Cond	NOI
1/64	1/4	16.03	$1.57 \cdot 10^7$	623	16.20	22.48	21	8.81	$7.05 \cdot 10^8$	2331
	1/8	15.61	$2.01 \cdot 10^7$	693	25.26	15.52	21			
	1/16	14.61	$6.86 \cdot 10^7$	783	100.66	9.67	17			
1/128	1/4	17.31	$2.23 \cdot 10^7$	892	16.99	30.64	25	8.91	$2.82 \cdot 10^9$	8431
	1/8	16.98	$2.44 \cdot 10^7$	1042	21.77	22.48	25			
	1/16	16.34	$7.61 \cdot 10^7$	1110	65.53	15.52	21			

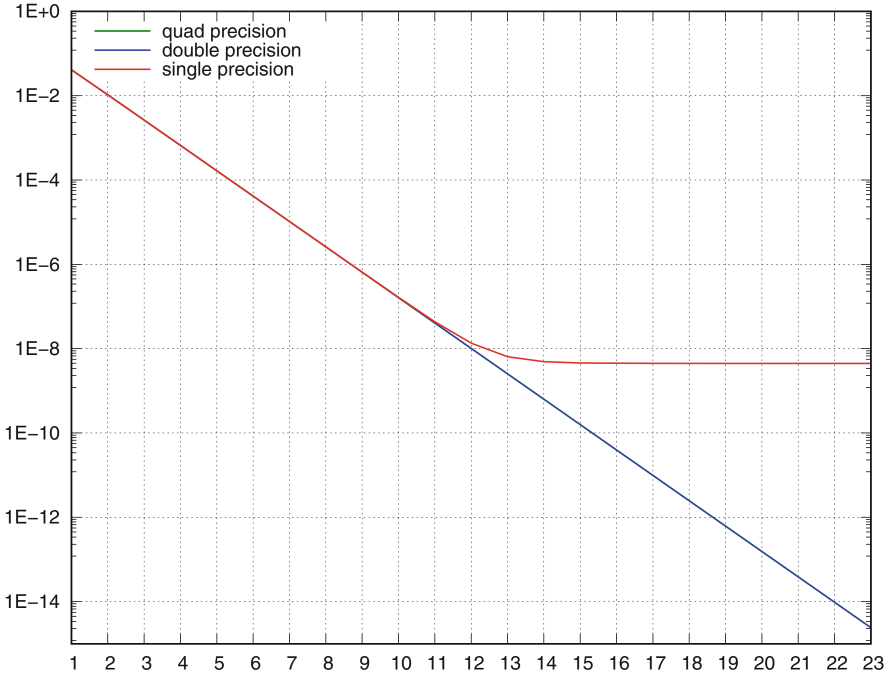


Fig. 3 L^2-error when solving Poisson's equation with prehandling by HFEM as a function of the refinement level, i.e. $h = 2^{-\text{level}}$, in 1D using single, double and quad precision. Compare to Fig. 1

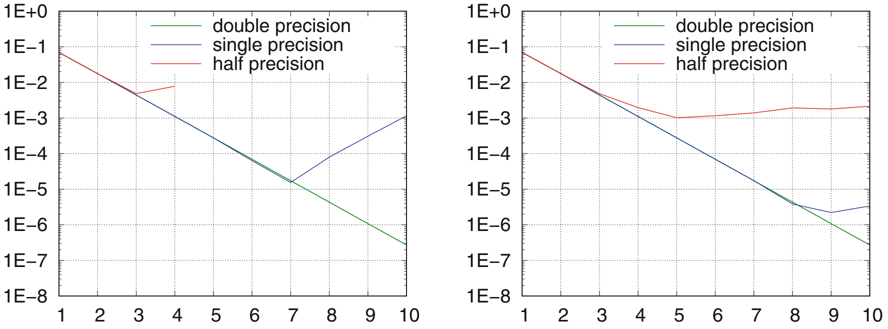


Fig. 4 L^2-error with FEM (left) and prehandling by HFEM (right) as a function of the refinement level in 2D using half, single, and double precision

Furthermore, it was tested if the application of the hierarchical method actually enables us to lower the error when using single or even half precision floating-point format. The course of the L^2-error depicted in the Figs. 3 (1D) and 4 (2D) show that by prehandling via the hierarchical method one achieves more accurate approximations in single and double precision in the 1D case and in half and single precision in the 2D case compared to the finite element method. For more detailed and further numerical results, such as P1 finite elements and rectangular domains (where the hierarchical method has basically identical effects), we refer to [4].

5 Summary and Conclusion

By the concept of prehandling together with the hierarchical finite element method the requirements of a lower condition number, preservation of sparsity and an efficient transformation are successfully met in the case of Poisson's equation in 1D and 2D as numerical results confirm. This allows us to use low precision hardware without losing too much accuracy. In the context of technical applications where an error of 1% is often satisfactory, even the usage of half precision might be appropriate. Additionally on the plus side, the expense of the iterative solution to the linear system is reduced if the hierarchical method is applied. By implementing further prehandling via a partial Cholesky decomposition, we can also deal with the case of the Poisson-like equation with a jumping coefficient. A central task for future research is to extend this method to apply prehandling in 3D, too.

Acknowledgments The results presented in Figs. 1, 3, and 4 have been created using the FEM software package FEAT3.[1]

References

1. Yserentant, H.: On the Multi-Level Splitting of Finite Element Spaces. In: Numer. Math. 49, pp. 379–412. Springer (1986)
2. Deuflhard, P., Leinen, P., Yserentant, H.: Concepts of an Adaptive Hierarchical Finite Element Code. In: IMPACT of Computing in Science and Engineering 1, pp. 3–35. Academic Press, Inc. (1989)
3. Ong, M. E. G.: Hierarchical Basis Preconditioners in Three Dimensions. In: SIAM J. Sci. Comput. 18 (1997), pp. 479–498.
4. Ruda, D.: Numerische Studien zur "Vorbehandlung" (prehandling) von Poisson-artigen Problemen durch die hierarchische Finite-Elemente-Methode. Master thesis, TU Dortmund (2020)

[1] see http://www.featflow.de/en/software/feat3.html.

Modal Analysis of Elastic Vibrations of Incompressible Materials Based on a Variational Multiscale Finite Element Method

Ramon Codina and Önder Türk

Abstract In this study, we extend the standard modal analysis technique that is used to approximate vibration problems of elastic materials to incompressible elasticity. The second order time derivative of the displacements in the inertia term is utilized, and the problem is transformed into an eigenvalue problem in which the eigenfunctions are precisely the amplitudes, and the eigenvalues are the squares of the frequencies. The finite element formulation that is based on the variational multiscale concept preserves the linearity of the eigenproblem, and accommodates arbitrary interpolations. Several eigenvalues and eigenfunctions are computed, and then the time approximation to the continuous solution is obtained taking a few modes of the whole set, those with higher energy. We present an example of the vibration of a linear incompressible elastic material showing how our approach is able to approximate the problem. It is shown how the energy of the modes associated to higher frequencies rapidly decreases, allowing one to get good approximate solution with only a few modes.

1 Introduction

Modal analysis is a widely used technique to approximate vibration problems of elastic materials. Starting from the transient equations of elasticity, with a second order time derivative of the displacements in the inertia term, the key idea is to assume a harmonic behavior for the displacement, each mode being of the form $\mathbf{u}(\mathbf{x})e^{i\omega t}$, where $\mathbf{u}(\mathbf{x})$ is the vector field of displacement amplitudes associated to the frequency ω. Substituting this expression into the equilibrium equations (without forcing terms if free vibrations are considered) leads to an eigenvalue

R. Codina
Universitat Politècnica de Catalunya, Barcelona, Spain
e-mail: ramon.codina@upc.edu

Ö. Türk (✉)
Gebze Technical University, Kocaeli, Turkey

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_101

problem (EVP) in which the eigenfunctions are precisely the amplitudes $\mathbf{u}(\mathbf{x})$, and the eigenvalues are the squares of the frequencies, ω^2. The elasticity operator is symmetric and positive definite, thus, a complete set of eigenfunctions and associated positive eigenvalues exist, and the exact solution can be expressed as a series of modes.

While this approach offers no difficulty when applied to compressible materials, and has been applied to different structural models, incompressible media pose the difficulty associated to the need of introducing the pressure (or mean stress) as a variable and to interpolate it in an adequate manner. In particular, when the problem is approximated using a finite element method (FEM), the standard Galerkin formulation requires the use of interpolations for the displacement and the pressure that satisfy the classical inf-sup condition, see, e.g., [1, 2, 4, 5]. This is true for both the classical boundary value problem found in stationary incompressible elasticity, the so called Stokes problem, and for the EVP encountered in modal analysis, as described above.

The alternative to use interpolations satisfying the inf-sup condition is to resort to stabilized finite element formulations. However, particular care is needed when dealing with the EVP. The well known Galerkin least squares approach, for example, a widely used stabilization technique, yields a quadratic EVP even if the continuous one is linear. We have proposed a FEM for the Stokes EVP that preserves the linearity of the continuous problem in [6]. It is based on the variational multiscale (VMS) concept, which assumes that the unknown can be split into a finite element component and a subgrid scale that needs to be modeled. The key point is to consider that this subgrid scale is orthogonal, in the L^2-sense, to the finite element component. This yields an EVP that is linear, and that can be solved using arbitrary interpolations for the velocity and the pressure.

In this work, we extend this procedure to the modal analysis of incompressible elastic materials, using displacements and pressures as variables. We show that each mode of the modal analysis (amplitude and frequency) can be obtained from an EVP that can be split into the finite element scale and the subgrid scale. The latter needs to be approximated, and we show that this approximation should depend on the frequency of the mode being considered. Since this frequency is unknown, an iterative procedure must be devised. The result is a problem for the finite element component of the displacement amplitude and the pressure which allows for any spatial interpolation.

Several eigenvalues and eigenfunctions of the Stokes EVP need to be computed to perform the modal analysis. The time approximation to the continuous solution is obtained taking a few modes of the whole set, those with higher energy. We present an example of the vibration of a linear incompressible elastic material showing how our approach is able to approximate the solution to the problem. We show that the energy of the modes associated to higher frequencies rapidly decreases, and thus a reasonable approximate solution can be obtained with only a few modes.

2 The Problem Definition

Let us consider the problem in a domain Ω with boundary $\partial\Omega$, and time $t > 0$ as follows

$$\rho\partial_{tt}^2\mathbf{u} - \mu\Delta\mathbf{u} + \nabla p = \mathbf{0} \quad \text{in } \Omega,\ t > 0, \tag{1}$$

$$\nabla\cdot\mathbf{u} = 0 \quad \text{in } \Omega,\ t > 0, \tag{2}$$

$$\mathbf{u} = \mathbf{0} \quad \text{on } \partial\Omega,\ t > 0, \tag{3}$$

$$\mathbf{u} = \mathbf{u}_0 \quad \text{in } \Omega,\ t = 0, \tag{4}$$

$$\partial_t\mathbf{u} = \mathbf{v}_0 \quad \text{in } \Omega,\ t = 0. \tag{5}$$

In these equations, Δ and ∇ are the Laplacian and gradient operators, respectively, $\mathbf{u}(\mathbf{x}, t)$ is the displacement, $p(\mathbf{x}, t)$ is the pressure, ρ is the mass density, μ is the shear modulus (the second Lamé constant), $\mathbf{u}_0(\mathbf{x})$ is the initial displacement, and $\mathbf{v}_0(\mathbf{x})$ is the initial velocity.

If we assume that $\mathbf{u}$ and p can be decomposed in modes of the form

$$\mathbf{u}(\mathbf{x}, t) = e^{\mathrm{i}\omega t}\boldsymbol{\phi}(\mathbf{x}), \quad p(\mathbf{x}, t) = e^{\mathrm{i}\omega t}\psi(\mathbf{x}),$$

these will be the solutions of the EVP

$$-\mu\Delta\boldsymbol{\phi} + \nabla\psi = \rho\,\omega^2\boldsymbol{\phi}, \tag{6}$$

$$\nabla\cdot\boldsymbol{\phi} = 0, \tag{7}$$

accompanied with the boundary condition $\boldsymbol{\phi}|_{\partial\Omega} = 0$.

It is known that there is a complete set of eigenfunctions and corresponding eigenvalues

$$\{\boldsymbol{\phi}_1(\mathbf{x}), \ldots, \boldsymbol{\phi}_n(\mathbf{x}), \ldots\}, \{\psi_1(\mathbf{x}), \ldots, \psi_n(\mathbf{x}), \ldots\}, 0 < \omega_1^2 \leq \omega_2^2 \leq \ldots\omega_n^2 \leq \ldots,$$

which are eigenpairs (non-trivial solutions) of (6)–(7). The displacement eigenfunctions can be taken to be $L^2(\Omega)$- and $H^1(\Omega)$-orthogonal such that

$$(\boldsymbol{\phi}_i, \boldsymbol{\phi}_j) = \delta_{ij}, \quad \mu(\nabla\boldsymbol{\phi}_i, \nabla\boldsymbol{\phi}_j) = \rho\,\omega_i^2\delta_{ij}, \tag{8}$$

where δ_{ij} is the Kronecker delta function.

3 Stabilized Finite Element Approximation of the Eigenproblem

If a VMS finite element method is used to approximate problem (1)–(5) *with orthogonal subscales*, the resulting matrix form of this problem is given as

$$\rho M\ddot{U} + \mu K U + G P = 0, \quad t > 0, \tag{9}$$

$$G^T U - S P = 0, \quad t > 0, \tag{10}$$

$$U = U_0 \quad \text{at } t = 0, \tag{11}$$

$$\dot{U} = V_0 \quad \text{at } t = 0, \tag{12}$$

where $U : \mathbb{R} \longrightarrow \mathbb{R}^{N_u}$ and $P : \mathbb{R} \longrightarrow \mathbb{R}^{N_p}$ are the arrays of nodal unknowns of displacement and pressure, respectively, N_u and N_p being the respective total number of degrees of freedom for the displacement and pressure. M, K, G and S are matrices of the appropriate size, the latter signifying the stabilization (the definitions of these matrices, and further details can be found in [6]).

Then, the discrete form of (6)–(7) can be written as

$$\mu K\Phi + G\Psi = \rho\,\hat{\omega}^2 M\Phi, \tag{13}$$

$$G^T\Phi - S\Psi = 0. \tag{14}$$

Since $\operatorname{rank}(M) = N_u$, the discrete eigenvectors and eigenvalues can be written in the form

$$\{\Phi_1, \dots, \Phi_{N_u}\}, \quad \{\Psi_1, \dots, \Psi_{N_u}\}, \quad 0 \le \hat{\omega}_1^2 \le \cdots \le \hat{\omega}_{N_u}^2.$$

Let us note here that the pressure eigenfunctions are associated to the displacement ones. A generalized EVP with a positive definite matrix in the right-hand side multiplying both displacements and pressures would have $N_u + N_p$ eigenvalues [3].

4 Modal Analysis of the Approximate Problem

We consider an approximate solution to the stabilized formulation of the problem (9)–(12) of the form

$$\begin{bmatrix} U_a(t) \\ P_a(t) \end{bmatrix} = \sum_{j=1}^{N} Z_j(t) \begin{bmatrix} \Phi_j \\ \Psi_j \end{bmatrix}, \tag{15}$$

with $N \leq N_u$ being the number of modes to be included in the expansion. Let

$$\Xi = \begin{bmatrix} \Phi_1 & \cdots & \Phi_N \\ \Psi_1 & \cdots & \Psi_N \end{bmatrix} \in \mathbb{R}^{(N_u+N_p)\times N}, \quad Z(t) = \begin{bmatrix} Z_1(t) \\ \vdots \\ Z_N(t) \end{bmatrix} \in \mathbb{R}^N, \tag{16}$$

which allows us to write

$$\begin{bmatrix} U_a(t) \\ P_a(t) \end{bmatrix} = \Xi Z(t). \tag{17}$$

Substituting this into (9)–(10), and projecting on the space generated by Ξ we get

$$\Xi^T \begin{bmatrix} \rho M & 0 \\ 0 & 0 \end{bmatrix} \Xi \ddot{Z}(t) + \Xi^T \begin{bmatrix} \mu K & G \\ G^T & -S \end{bmatrix} \Xi Z(t) = 0. \tag{18}$$

In expanded form, we have

$$\Phi_i^T \sum_{j=1}^{N} \rho M \Phi_j \ddot{Z}_j + \Phi_i^T \sum_{j=1}^{N} \left(\mu K \Phi_j + G \Psi_j\right) Z_j = 0, \quad i = 1, \ldots, N, \tag{19}$$

$$\Psi_i^T \sum_{j=1}^{N} \left(G^T \Phi_j - S \Psi_j\right) Z_j = 0, \quad i = 1, \ldots, N. \tag{20}$$

The discrete counterpart of the L^2-orthogonality in (8) is

$$\Phi_i^T M \Phi_j = m_i \delta_{ij}, \tag{21}$$

therefore, from (13) we get

$$\Phi_i^T \left(\mu K \Phi_j + G \Psi_j\right) = \rho \hat{\omega}_i^2 m_i \delta_{ij}. \tag{22}$$

From (14) we have that (20) is automatically satisfied, whereas form (19) we obtain

$$\ddot{Z}_i + \hat{\omega}_i^2 Z_i = 0, \quad i = 1, \ldots, N. \tag{23}$$

It is important to remark that the equations obtained are the same as for the non-incompressible case and as the Galerkin approximation to the incompressible case. For the Galerkin case ($S = 0$), we would have $\Phi_i^T G \Psi_j = 0$, but in fact, what we need is (22), which also holds in the stabilized case. Also note that the form (15) has to be assumed in the method we have employed, however, one could use different expressions of Z for the displacement and pressure in the Galerkin case (see (20)).

Now, inserting the solution of (23) into (15) we have

$$\begin{bmatrix} U_a(t) \\ P_a(t) \end{bmatrix} = \sum_{j=1}^{N} (A_j e^{\mathrm{i}\hat{\omega}_j t} + B_j e^{-\mathrm{i}\hat{\omega}_j t}) \begin{bmatrix} \Phi_j \\ \Psi_j \end{bmatrix}, \tag{24}$$

where A_j, B_j, $j = 1, \ldots, N$, are the coefficients to be determined from the initial conditions projected onto the subspace generated by the modes.

5 A Numerical Test

We consider a classical incompressible linear elasticity problem, namely, a simply supported rectangular cantilever beam. A clamped beam is undergoing a sudden deflection caused by an initial load, and then allowed to vibrate harmonically by removing the constraint that provokes this deformation. The width and length of the beam are discretized using 5 and 50 triangular elements, respectively, with quadratic interpolations for all the unknowns. ρ and μ are taken as unity in the simulations. The initial configuration of the model is depicted in Fig. 1.

The displacement field plots of the first eight eigenfunctions are presented in Fig. 2. The figure also includes the ordinal number of the eigenvalue associated to the eigenmode, and the L^2-norm of the corresponding term (referred as energy for brevity) in the expansion (24), at the top of each graph. Since this is a pure bending problem, the bending modes are expected to have high energy that is decreasing with an increasing cardinality. This can readily be observed from Fig. 2; the first few bending modes have higher energies compared to the rest of them. All the compression modes (the second, fourth, and seventh) presented have zero (to the machine's precision) energies, in other words, their contribution is insignificant compared to the other terms. It is interesting, from a structural mechanics point of view, that all modes are either bending or compression, and there exist no coupled modes.

In order to measure the contribution of the modes, the variation of the total energy loss of the first i modes defined as $TEL_i = (1 - \sum_{j=1}^{i} E_j / \sum_{j=1}^{N} E_j)100\%$, where E_j refers to the energy of the j-th term in the modal expansion, with respect to

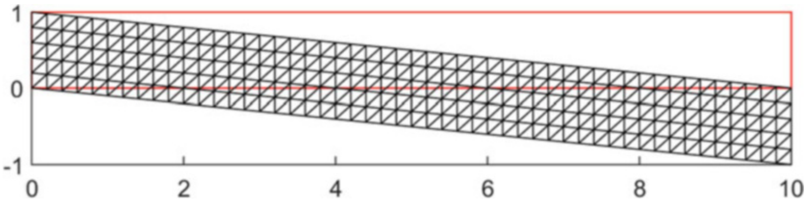


Fig. 1 The initial setting where the displacements are magnified 100 times

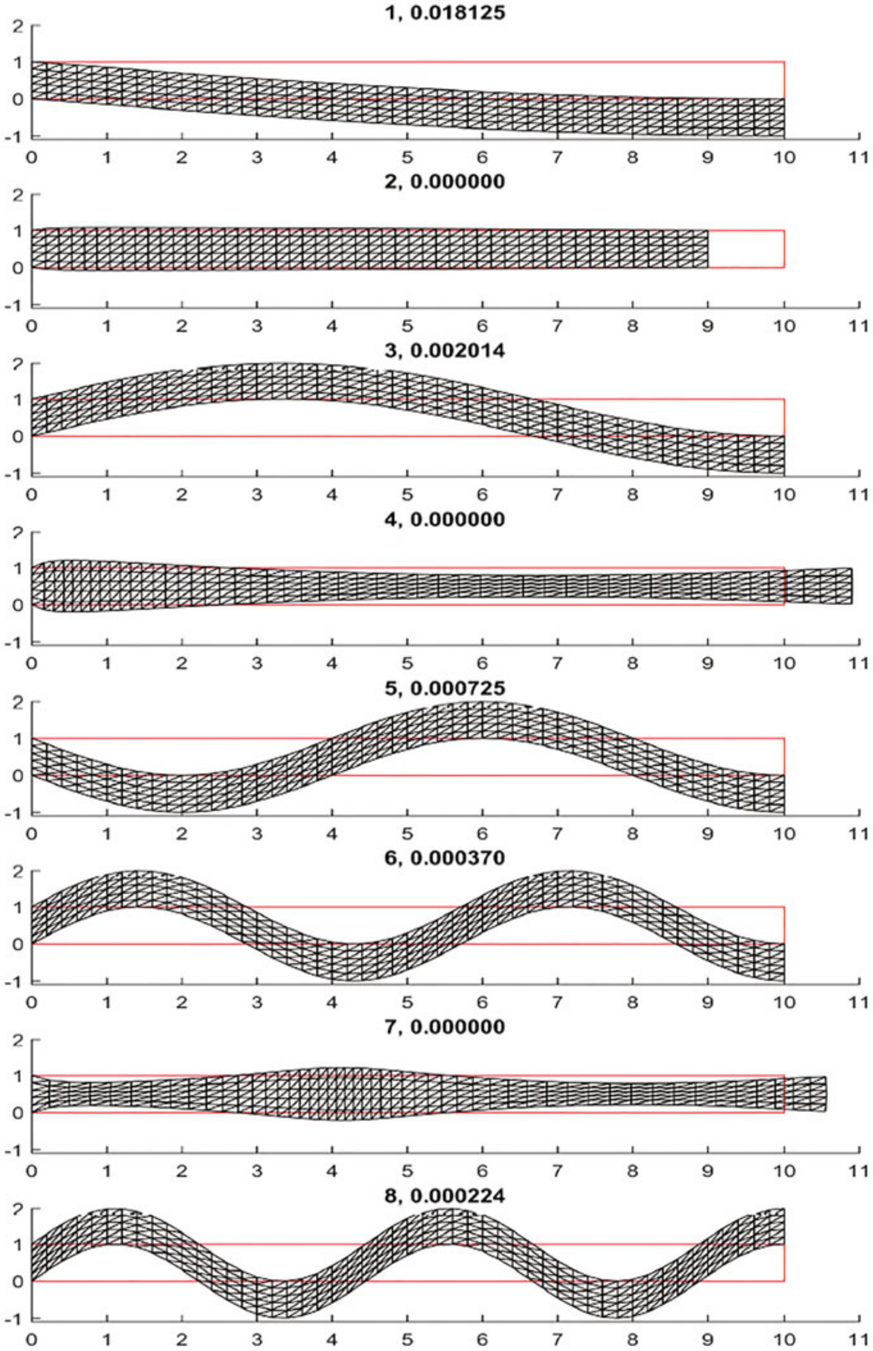


Fig. 2 Plot of the first eight eigenfunctions

the first 50 modes is depicted in Fig. 3. In coherence with the previous results, the compression modes do not contribute to the energy loss and the first three, and the first six modes account for more than, respectively, 90% and 95% of the total energy.

Figure 4 shows the time variations of the displacement at the lower right tip of the beam, evaluated using three different schemes: a direct solution using a central finite difference (FD) time integration scheme, the modal solution using 6 modes, and a reference solution obtained by including 100 modes. All three solutions agree reasonably well over cycles of period observed to be 40.

Finally, a quantification of the L^2-norm of the error, normalized using the reference solution mentioned above, at the lower right tip of the beam calculated over a periodic cycle with respect to the number of modes is provided in Fig. 5.

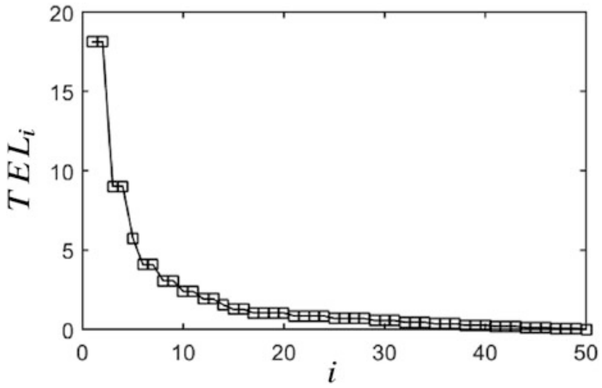


Fig. 3 The total energy loss of the first i modes

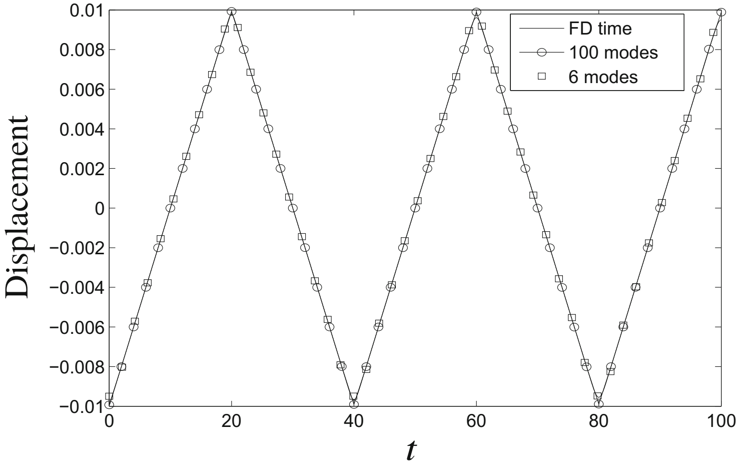


Fig. 4 Transient behavior of the displacement at the lower right tip of the beam

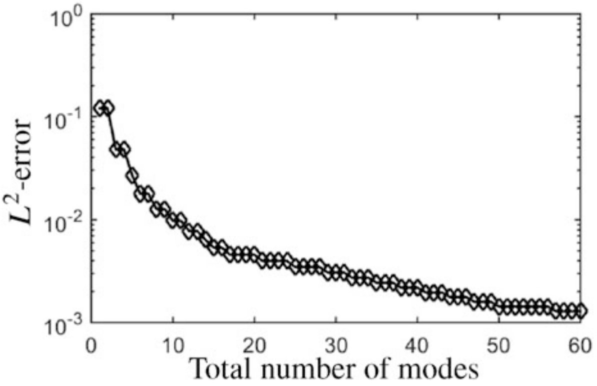


Fig. 5 The L^2-norm of the error at the lower right tip over the time interval $[0, 40]$

6 Conclusion

A stabilized FEM based on orthogonal subgrid scales that allows any spatial interpolation is extended to the modal analysis of incompressible elastic materials. It is shown that each mode of the modal analysis can be obtained from an EVP that can be split into the finite element scale and the subgrid scale. The time approximation to the continuous solution can be obtained taking a few modes of the whole set, those with higher energy. The energy of the modes associated to higher frequencies rapidly decreases, allowing one to get good approximate solution with only a few modes.

References

1. Barbone, P.E., Nazari, N., Harari, I.: Stabilized finite elements for time-harmonic waves in incompressible and nearly incompressible elastic solids. International Journal for Numerical Methods in Engineering **120**(8), 1027–1046 (2019)
2. Chiumenti, M., Valverde, Q., de Saracibar, C.A., Cervera, M.: A stabilized formulation for incompressible elasticity using linear displacement and pressure interpolations. Computer Methods in Applied Mechanics and Engineering **191**(46), 5253–5264 (2002)
3. Gao, X.B., Golub, G.H., Liao, L.Z.: Continuous methods for symmetric generalized eigenvalue problems. Linear Algebra and its Applications **428**(2), 676–696 (2008)
4. Hansbo, P., Larson, M.G.: Discontinuous Galerkin methods for incompressible and nearly incompressible elasticity by Nitsche's method. Computer Methods in Applied Mechanics and Engineering **191**(17), 1895–1908 (2002)
5. Lamichhane, B.P.: A mixed finite element method for nearly incompressible elasticity and Stokes equations using primal and dual meshes with quadrilateral and hexahedral grids. Journal of Computational and Applied Mathematics **260**, 356–363 (2014)
6. Türk, Ö., Boffi, D., Codina, R.: A stabilized finite element method for the two-field and three-field Stokes eigenvalue problems. Computer Methods in Applied Mechanics and Engineering **310**, 886–905 (2016)

A Learning-Based Formulation of Parametric Curve Fitting for Bioimage Analysis

Soham Mandal and Virginie Uhlmann

Abstract Parametric curve models are convenient to describe and quantitatively characterize the contour of objects in bioimages. Unfortunately, designing algorithms to fit smoothly such models onto image data classically requires significant domain expertise. Here, we propose a convolutional neural network-based approach to predict a continuous parametric representation of the outline of biological objects. We successfully apply our method on the Kaggle 2018 Data Science Bowl dataset composed of a varied collection of images of cell nuclei. This work is a first step towards user-friendly bioimage analysis tools that extract continuously-defined representations of objects.

1 Introduction

Parametric curve models have been extensively used in the past in the context of active contours for image segmentation [1]. In its classical formulation, a parametric active contour algorithm requires the definition of a curve model, which is then initialized in the image and evolves to capture the boundaries of an object of interest by minimizing a handcrafted cost functional referred to as the energy [2]. In practice, designing an energy implies formalizing visual intuition in mathematical terms and requires expert domain knowledge [3], limiting the usability of parametric active contour methods. The energy most often consists in a combination of multiple terms, which must be weighted properly and are difficult to robustly optimize at once. In the context of bioimage segmentation, more efficient, robust, and generalizable learning-based methods such as convolutional neural networks [4] and random forests [5] are nowadays preferred over parametric active contours. Being able to model the outline of objects in a (possibly already segmented) image with

S. Mandal · V. Uhlmann (✉)
European Bioinformatics Institute (EMBL-EBI), European Molecular Biology Laboratory, Cambridge, UK
e-mail: smandal@ebi.ac.uk; uhlmann@ebi.ac.uk

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_102

parametric curves however remains of interest. The crux of bioimage analysis indeed consists of extracting quantitative measurements to describe, model and understand living phenomena [6]. Continuously-defined parametric curve models offer a convenient representation to extract morphological descriptors without discretization artefacts. Discrete segmentation masks are in fact often continuously interpolated prior to quantitative analysis [7]. Devising the best set of (discrete) contour points to interpolate is, however, not trivial, as mask boundaries are likely to be noisy. It thus appears that parametric curve fitting, as historically considered in active contour methods, deserves to be considered and modernized as a tool to retrieve optimal continuously-defined representations from discrete sets of connected pixels. In this work, we present a first attempt in this direction: we rely on a convolutional neural network (CNN) to predict the best parametric curve fit onto an object contour. In that way, we trade the energy design step, which requires domain-expert knowledge, for a data-driven approach. Our network is trained with a loss designed to penalise the discrepancy between a discrete ground truth mask and a sampled version of the predicted continuous contour. From this, it learns to generate the continuous representation that most accurately approximates the discrete contour of an object.

Using neural networks to predict a set of interpolation points and retrieve a continuous model of the contour of objects in images has recently attracted attention. Most relevant to our work are those of [8] and [9]. In [8], a joint structure composed of a CNN and an autoencoder is proposed to smoothly model the surface between vertebral bodies and posterior elements in the vertebra. The surface model is a thin-plate spline, whose control points are predicted using a shape model of vertebral bodies. While yielding very promising results, this approach heavily relies on the structural specificity of the vertebra and therefore has limited application in bioimages featuring other objects. The more general problem of predicting the locations of the set of control point of a parametric spline curve that best represents an object's contour is explored in [9]. The task is formulated as deducing the values of a variable length sequence of coefficients. The proposed solution consists of the combination of a CNN and a recurrent neural network. The loss is defined from a set of ideal, ground truth control point locations. Such a construction assumes that a unique set of control points yields the best parametric curve representation of an object contour. Since the parametrization of a continuously-defined contour is not unique, several control point sets can generate the same curve. The considered loss thus imposes artificial restrictions on the contour representation. Here, we alleviate this by adopting a curve-based (instead of control-points-based) loss.

In Sect. 2, we recall the formulation of a parametric curve model built using spline interpolation, present the architecture of our network, and discuss the design of the loss, which is central to our work. Section 3 is devoted to experiments: we tested our network on the Kaggle 2018 Data Science Bowl dataset [10], which contains images of cell nuclei acquired under a variety of conditions, and that vary in the cell type, magnification, and imaging modality. Finally, in Sect. 4, we conclude with a brief discussion and explore future directions.

2 Description of the Method

2.1 *Parametric Curve Model*

Our parametric curve model is constructed relying on uniform spline interpolation as in [2]. In the 2D plane, a closed parametric spline curve $\mathbf{r}(t)$, $t \in \mathbb{R}$ is expressed by two coordinate functions $x_1(t)$ and $x_2(t)$ as

$$\mathbf{r}(t) = \begin{bmatrix} x_1(t) \\ x_2(t) \end{bmatrix} = \sum_{k=0}^{M-1} \mathbf{c}[k]\varphi_{\text{per}}(t-k). \tag{1}$$

The coefficients $\{\mathbf{c}[k]\}_{k=0,\ldots,M-1} \in \mathbb{R}^2$ are the control points and, in our use-case, correspond to coordinates in the image space. The number of control points, M relate to the flexibility of the curve, as smaller number of control points yield more rigid contours. In our case, the function $\varphi_{\text{per}}(t) = \sum_{n=-\infty}^{\infty} \beta^3(t - nM)$ is the M-periodized version of the cubic B-spline generator [11]. We rely on cubic B-splines for their good approximation properties of smooth curves [12]. Our method would however equivalently work with other interpolation kernels $\varphi : \mathbb{R} \rightarrow \mathbb{R}$, such as exponential splines [13].

2.2 *Neural Network Architecture*

Using a neural network, we aim at robustly determining how control points should be distributed along the boundary of objects in a variety of bioimages. CNN are the tools of choice when it comes to understanding spatial information from image content as they explicitly connect neurons that are spatially near in consecutive layers [14]. In particular, CNN already achieve state-of-the-art performance in several image analysis tasks such as image classification, image segmentation, and bounding box detection.

We adopt the architecture depicted in Fig. 1. This construction, consisting of blocks of convolutional layers followed by pooling and ending with a dense fully connected layer, is typical of a classical CNN [15]. For a two-dimensional parametric spline curve composed of M control points, the final layer of our architecture is composed of $2M$ nodes, where each subsequent pair of nodes corresponds to the image coordinates of a control point. The network architecture, as well as hyperparameters such as learning rate and batch size, could be adapted and optimized for specific biological objects.

Our training pipeline is illustrated in Fig. 2. The input image goes through the CNN, which predicts the locations of the M control points $\{\mathbf{c}[k]\}_{k=0,\ldots,M-1}$ of a spline model of the form (1), where the value of M and the spline generator φ have been chosen prior to training. The loss is then calculated between the discrete ground

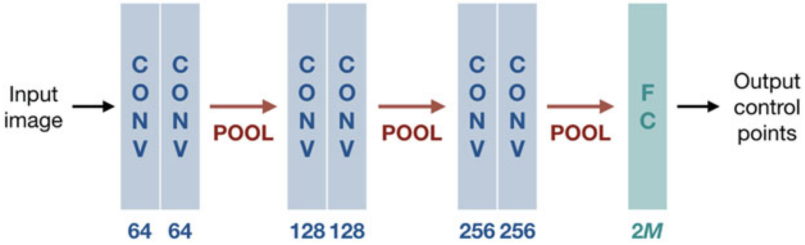


Fig. 1 CNN Architecture. CONV: 3 × 3 leaky ReLU convolutional layer; POOL: 2 × 2 max-pooling layer; FC: fully connected layer; M: number of control points. The number of feature maps is indicated under each block

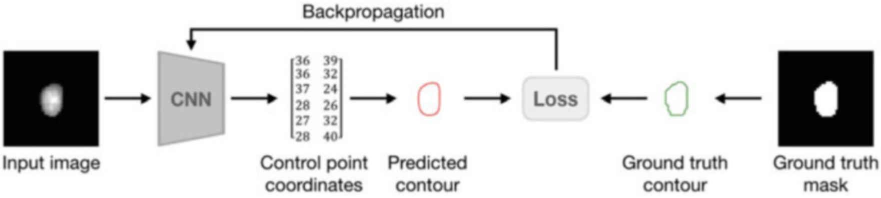


Fig. 2 Training Pipeline. The input image goes through a CNN, which predicts the M control points $\{\mathbf{c}[k]\}_{k=0,\ldots,M-1}$ of the parametric curve model (1). The loss reflects the discrepancy between the ground truth (pixel-based) contour and the discretization of the predicted (continuous) one

truth contour, extracted from the ground truth segmentation mask, and a sampled version of the predicted continuously-defined parametric curve. The network is updated accordingly for a given number of epochs using the ADAM optimizer [16].

2.3 Loss Function

Objects in a segmented image are usually represented by a segmentation mask. To train our network, we extract the collections of pixels composing the contour of a ground truth segmentation mask using Satoshi and Abe's algorithm [17] as implemented in OpenCV 3.4.2 [18]. The continuous curve predicted by the network must faithfully capture the ground truth contour. When sampling the predicted continuous curve, one should then retrieve the points located on the object contour (that is, the ground truth contour). We therefore define our loss as a Wasserstein (or Earth mover's) distance [19] over ordered point sets.

We consider A, the set of N connected pixels composing the ground truth contour. By uniformly sampling N values on (1), we then retrieve B, an equally-sized set of points describing the predicted contour. Inspired by the work of [20] on image annotation, we train our network with the loss

$$\mathcal{L}(A, B) = \min_{j \in [0,\dots,N-1]} \sum_{i=0}^{N-1} \left\| a_i - b_{(j+i) \bmod N} \right\|, \tag{2}$$

which corresponds to the minimum distance between A and all circular permutation of the elements in B. Classically, the Wasserstein distance is defined over unordered point clouds. Its calculation thus results in high computational complexity, and is in practice achieved relying on approximations. In our case, we can make use of the fact that successive points on the contour are inherently ordered, which dramatically reduces the complexity of the problem. Moreover, we can always ensure that A and B in (2) are equally-sized thanks to the continuously-defined nature of the predicted contour.

3 Experimental Results

In order to test the validity of our method, we consider the Kaggle 2018 Data Science Bowl dataset (also referred to as BBBC038v1) available from the Broad Bioimage Benchmark Collection [10]. It is composed of a diverse collection of images of cell nuclei, which aims at reflecting the type of images collected by research biologists at universities, biotechs, and hospitals. Nuclei in images vary in four different ways: the organism they are derived from, including but not limited to humans, mice, and flies; the way they have been treated and imaged, in terms of types of staining, magnification, and illumination; the context in which they appear, including cultured mono-layers, tissues, and embryos; and their physiological state, such as cell division, genotoxic stress, and differentiation. The dataset faithfully reflects the variability of object appearance and image types in bioimages and is designed to challenge the generalization capabilities of a method across these variations. We relied on the pre-defined stage1_train set, composed of 670 images, to train our network and perform cross-validation. We saved the stage1_test set, which consists of 65 images, to assess the performance of the network after training. Images are generally composed of more than one nucleus, but ground truth binary masks of each individual nucleus instance are provided for both sets. As we focus on continuously modelling the contour of a single object at once, we tile images according to bounding boxes around each individual nucleus. This step can be performed relying on a separate neural network trained for object detection, such as [21]. Ultimately, our final cross-validation and test sets are composed of respectively 29,461 and 4152 tiles of isolated nuclei.

We divide the cross-validation set into 10 random partitions and follow a tenfold cross-validation strategy. In each fold, we pick 9 partitions of the cross-validation set for training and use the remaining partition as validation set to monitor performance. We ensure that each sample in the cross-validation set is used for training in 9 independent folds and for validation in the remaining one. We train the network for 100 epochs, keeping the learning rate at 0.0001. We record the evolution of the loss in training and validation to ensure convergence and monitor for overfitting. For each fold, we compute the median Dice score over the whole test set. The Dice score [22] is defined as $\text{Dice}(A, B) = \frac{2|A \cap B|}{|A|+|B|}$, where A is the original ground truth mask and B the mask derived from the predicted contour. We obtain a median Dice score of 0.9562 ± 0.0014 over the ten folds, and provide visual illustration of high, intermediate, and low Dice score results in Fig. 3.

Finally, in order to challenge the learning abilities of our rather shallow network architecture, we carry out a data ablation experiment. We keep the stage1_test set untouched and randomly ablate a fraction of the stage1_train set before training and carrying out tenfold cross-validation. The depth of a network affects its ability to learn an appropriate data representation, but also dictates the amount of free parameters (weights) to be set. As we designed ours to offer a good trade-off between simplicity and performance, we here investigate the effect of training set size on prediction quality. In boxplots shown in Fig. 4, we report the distribution of median Dice scores across folds on the full BBBC038v1 stage1_train set, as well as on ablated versions of it. No statistically significant decrease in performance is observed when ablating up to 90% of the stage1_train set. Prediction accuracy is clearly affected when training only on 10% of the data, but the median Dice nevertheless remains around 0.948. We provide source code to reproduce these experiments at gitlab.ebi.ac.uk/smandal/cpnet.

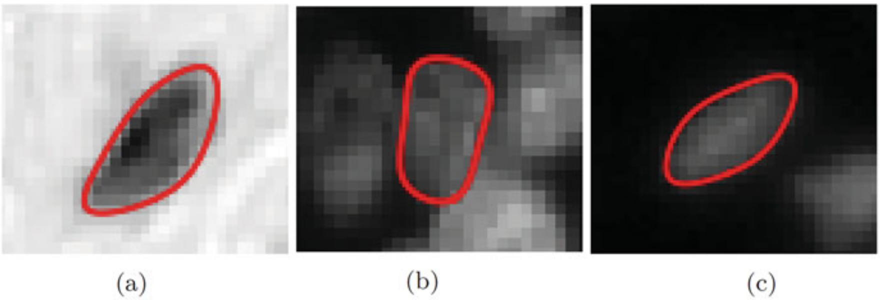


Fig. 3 Predicted continuously-defined contours. We report examples of (**a**) low quality (0.93), (**b**) intermediate quality (0.95), and (**c**) high quality (0.97) results from the BBBC038v1 stage1_test set after training on the BBBC038v1 stage1_train set

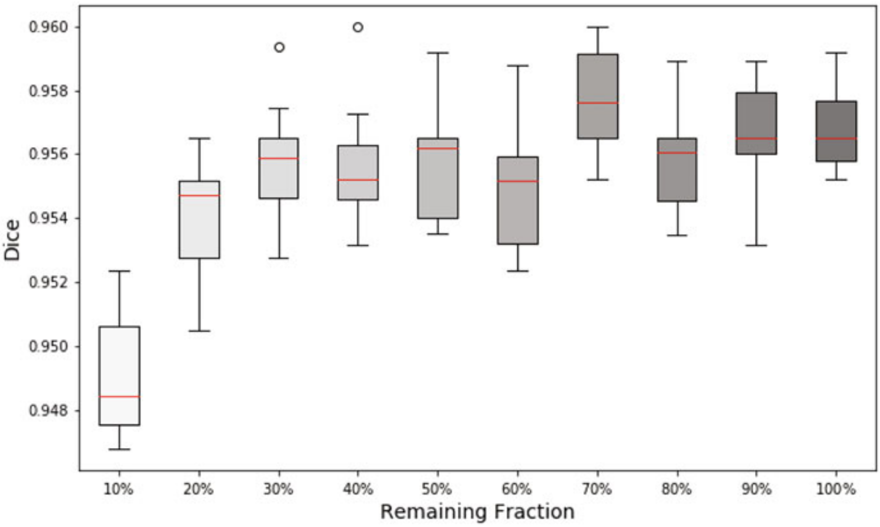


Fig. 4 Data ablation study. Distribution of median Dice score across folds on the stage1_test set, when training on truncated versions of the stage1_train set

4 Discussion

In this work, we propose a CNN-based pipeline to predict a continuous contour representation for objects in bioimages. The output of the network is a collection of discrete points that fully determine a continuously-defined parametric spline curve. Our method successfully learns and predicts continuous models of nuclei contours in images from the Kaggle 2018 Data Science Bowl dataset, which realistically reflects variations in the visual appearance of objects in bioimages. Here, the number of control points M and the basis function φ in (1) were determined prior to training and kept fixed. A natural future direction for this work aims at integrating these variables in the network and learning them from the nature of the objects to be represented. An additional worthy avenue would be to explore the generalization capabilities of the network on datasets featuring more complex biological object shapes.

Acknowledgments The authors are grateful to Dr. James Klatzow and Dr. Julien Fageot for valuable discussions and helpful comments on the manuscript. This work is supported by EMBL core funding.

References

1. R. Delgado-Gonzalo et al. Snakes on a plane: A perfect snap for bioimage analysis. *IEEE Signal Process Mag*, 32(1):41–48, 2015.
2. P. Brigger, J. Hoeg, and M. Unser. B-Spline snakes: A flexible tool for parametric contour detection. *IEEE Trans Image Process*, 9(9):1484–1496, 2000.
3. M. Jacob, T. Blu, and M. Unser. Efficient energies and algorithms for parametric snakes. *IEEE Trans Image Process*, 13(9):1231–1244, 2004.
4. T. Falk et al. U-Net: deep learning for cell counting, detection, and morphometry. *Nat Methods*, 16(1):67–70, 2018.
5. S. Berg et al. ilastik: interactive machine learning for (bio) image analysis. *Nat Methods*, pages 1–7, 2019.
6. G. Myers. Why bioimage informatics matters. *Nat Methods*, 9(7):659, 2012.
7. L. Qu and H. Peng. A principal skeleton algorithm for standardizing confocal images of fruit fly nervous systems. *Bioinformatics*, 26(8):1091–1097, 2010.
8. N. Lessmann et al. Vertebra partitioning with thin-plate spline surfaces steered by a convolutional neural network. In *Proc of MIDL'19*, London, UK, July 8–10, 2019.
9. J. Gao et al. Deepspline: Data-driven reconstruction of parametric curves and surfaces. *arXiv:1901.03781*, 2019.
10. V. Ljosa, K.L. Sokolnicki, and A.E. Carpenter. Annotated high-throughput microscopy image sets for validation. *Nat Methods*, 9(7):637–637, 2012.
11. M. Unser. Splines: A perfect fit for signal and image processing. *IEEE Signal Processing Mag*, 16(6):22–38, 1999.
12. T. Blu and M. Unser. Quantitative Fourier analysis of approximation techniques: Part I—Interpolators and projectors. *IEEE Trans Signal Process*, 47(10):2783–2795, 1999.
13. R. Delgado-Gonzalo, P. Thévenaz, and M. Unser. Exponential splines and minimal-support bases for curve representation. *Comput Aided Geom Des*, 29(2):109–128, 2012.
14. A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *Proc of NIPS'12*, pages 1097–1105, Lake Tahoe, NV, USA, December 3–8, 2012.
15. Simonyan K. and Zisserman A. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014.
16. D. Kingma and J. Ba. ADAM: A method for stochastic optimization. In *Proc of ICLR'15*, San Diego, CA, USA, May 7–9, 2015.
17. S. Satoshi and K. Abe. Topological structural analysis of digitized binary images by border following. *Comput Gr Image Process*, 30(1):32–46, 1985.
18. G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.
19. C. Villani. *Optimal transport: old and new*. Springer, 2008.
20. H. Ling et al. Fast interactive object annotation with curve-GCN. In *Proc of CVPR'19*, Long Beach, CA, USA, June 16–20, 2019.
21. R. Girshick. Fast R-CNN. In *Proc of ICCV'15*, pages 1440–1448, Santiago, Chile, December 13–16, 2015.
22. L.R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

Adaptive Time Stepping Methods Within a Data Assimilation Framework Applied to Non-isothermal Flow Dynamics

Ferdinand Evert Uilhoorn

Abstract This contribution discusses the performance of time stepping schemes within a data assimilation framework, applied to the method of lines solutions of the non-isothermal compressible gas flow equations. We consider important classes of schemes, namely an embedded explicit Runge–Kutta (ERK) scheme, a diagonally implicit Runge–Kutta (DIRK) scheme, a fully implicit Runge–Kutta (IRK) scheme and a Rosenbrock–Krylov (ROK) scheme. For the numerical illustration, we estimated the flow transients in a subsea pipeline system. Errors from numerical discretization, missing and variability of physical parameters and inaccuracy of initial and boundary conditions are assumed non-Gaussian. Efficiency, robustness and estimation accuracy were evaluated. Results showed that the DIRK scheme is a good compromise between efficiency and robustness. Spurious oscillations were filtered out by the sequential Monte–Carlo algorithm.

1 Introduction

Numerical modeling of flow transients plays an important role in real-time control of pipeline systems. Data assimilation combines measurements with numerical models, i.e.,

$$\pi_t = f(\pi_{t-1}) + v_{t-1}, \tag{1}$$

$$\zeta_t = h(\pi_t) + n_t, \tag{2}$$

F. E. Uilhoorn (✉)
Gas Engineering Group, Warsaw University of Technology, Warsaw, Poland
e-mail: ferdinand.uilhoorn@pw.edu.pl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_103

where $\pi_t \in \mathbb{R}^{n_\pi}$ is the state at time t, $\zeta_t \in \mathbb{R}^{n_\zeta}$ is the measurement, $v_t \in \mathbb{R}^{n_v}$ is a random forcing that represents uncertainties in model parameters, missing physics and discretization errors and $n_t \in \mathbb{R}^{n_n}$ is the measurement noise. The mappings $f : \mathbb{R}^{n_\pi} \times \mathbb{R}^{n_v} \mapsto \mathbb{R}^{n_\pi}$ and $h : \mathbb{R}^{n_\zeta} \times \mathbb{R}^{n_n} \mapsto \mathbb{R}^{n_\zeta}$ represent the discretized flow and measurement model, respectively. In general data assimilation aims better prediction and understanding of the fundamental physics.

Unlike the Kalman filter and its variants, sequential Monte–Carlo (SMC) methods or particle filters (PFs) are not constrained by linearity or Gaussianity and characterized by attractive convergence properties. From the law of large numbers [1]

$$I(f) \approx \frac{1}{N_p} \sum_{i=1}^{N_p} f_t \left(\pi_{0:t}^{(i)} \right) \xrightarrow[N \to +\infty]{\text{a.s.}} \int f_t \left(\pi_{0:t} \right) P \left(\mathrm{d}\, \pi_{0:t} | \zeta_{1:t} \right), \tag{3}$$

where a.s. refers to almost sure convergence and f_t is a function of interest. This property makes the use of ensemble-based data assimilation methods attractive. It enables us to approximate the posterior distribution $p(\pi_{0:t}|\zeta_{1:t})$ and the filtering density $p(\pi_t|\zeta_{1:t})$ with $\zeta_{1:t} = \{\zeta_1, \zeta_2, \ldots, \zeta_t\}$. It starts with sampling N_p random draws, also called ensemble members, or particles π_0^i from the initial model probability density function (pdf) $p(\pi_0)$. Next, we solve the numerical model from $t-1$ to t and sample all model states, i.e., $\pi_t^{(i)} \sim p(\pi_t|\pi_{t-1}^{(i)})\ \forall i$. For the gas flow equations, the method of lines paradigm is used. Thus, for each model state, we solve the ordinary differential equations added with a stochastic term that represents the unknown external and internal terms. This requires an efficient and robust time integration scheme, which is the subject of investigation in this work. Finally, when measurements become available, the weights $\omega_t^{(i)} = p(\zeta_t|\pi_t^{(i)}) / \sum_{i=1}^{N_p} p(\zeta_t|\pi_t^{(i)})$ are computed for each particle where $p(\zeta_t|\pi_t^{(i)})$ is the pdf of the measurements. Subsequently, a resampling step selects particles with high weights that are kept for the posterior pdf.

For the time stepping, we concentrate on important classes of schemes, that is, an embedded explicit Runge–Kutta (ERK) scheme [2] a diagonally Runge–Kutta scheme (DIRK) [3], a fully implicit Runge–Kutta scheme (IRK) [4] and a Rosenbrock–Krylov (ROK) method [5]. We impose non-Gaussian noise because it can be argued that stochastic forcing, discretization errors and missing physics follow a Gaussian distribution. The statistics describing these errors are in general predefined, as here, but in practise most often a priori unknown [6]. For the test problem, we consider a real subsea pipeline system.

2 Unsteady Gas Flow Equations

The governing equations describing the unsteady gas flow dynamics are derived from the conservation principles of mass, momentum and energy and read

$$\frac{\partial p}{\partial t} = \frac{a_s^2}{c_p T}\left(1 + \frac{T}{z}\left(\frac{\partial z}{\partial T}\right)_p\right)\left(\frac{q}{A} + \frac{\dot{m}zRT}{pA^2}w\right) - \left[\frac{\dot{m}zRT}{pA} - \frac{a_s^2\dot{m}}{pA}\right.$$
$$\left.\cdot\left(1 - \frac{p}{z}\left(\frac{\partial z}{\partial p}\right)_T\right)\right]\frac{\partial p}{\partial x} - \frac{a_s^2\dot{m}}{TA}\left(1 + \frac{T}{z}\left(\frac{\partial z}{\partial T}\right)_p\right)\frac{\partial T}{\partial x} - \frac{a_s^2}{A}\frac{\partial \dot{m}}{\partial x}, \tag{4}$$

$$\frac{\partial T}{\partial t} = \frac{a_s^2}{c_p p}\left(1 - \frac{p}{z}\left(\frac{\partial z}{\partial p}\right)_T\right)\left(\frac{q}{A} + w\frac{\dot{m}zRT}{pA^2}\right) - \frac{\dot{m}zRT}{pA}\frac{\partial T}{\partial x}$$
$$- \frac{a_s^2}{c_p}\left(1 + \frac{T}{z}\left(\frac{\partial z}{\partial T}\right)_p\right)\left[\frac{\dot{m}zR}{pA}\left(1 + \frac{T}{z}\left(\frac{\partial z}{\partial T}\right)_p\right)\frac{\partial T}{\partial x} - \frac{\dot{m}TRz}{p^2A}\right.$$
$$\left.\cdot\left(1 - \frac{p}{z}\left(\frac{\partial z}{\partial p}\right)_T\right)\frac{\partial p}{\partial x} + \frac{zTR}{pA}\frac{\partial \dot{m}}{\partial x}\right], \tag{5}$$

$$\frac{\partial \dot{m}}{\partial t} = -\frac{\dot{m}}{T}\left(1 + \frac{T}{z}\left(\frac{\partial z}{\partial T}\right)_p\right)\frac{\partial T}{\partial t} + \frac{\dot{m}}{p}\left(1 - \frac{p}{z}\left(\frac{\partial z}{\partial p}\right)_T\right)\frac{\partial p}{\partial t} - \frac{\dot{m}^2zR}{pA}$$
$$\cdot\left(1 + \frac{T}{z}\left(\frac{\partial z}{\partial T}\right)_p\right)\frac{\partial T}{\partial x} + \left(\frac{\dot{m}^2TRz}{p^2A}\left(1 - \frac{p}{z}\left(\frac{\partial z}{\partial p}\right)_T\right) - A\right)\frac{\partial p}{\partial x}$$
$$- \frac{\dot{m}zTR}{pA}\frac{\partial \dot{m}}{\partial x} - w - \frac{pAg\sin(\theta)}{zTR}, \tag{6}$$

within the domain $\{(x,t) : 0 \leqslant x \leqslant L, 0 \leqslant t \leqslant t_f\}$ with length L and final time t_f. Other variables and parameters are pressure p, temperature T, mass flux $\dot{m}$, frictional force w, cross-sectional area A, gravitational acceleration g, angle of inclination θ, heat transfer q and compressibility factor z. The isentropic wave speed $a_s = (\partial p/\partial \rho)_s^{1/2}$ is defined as

$$\left(\frac{\partial p}{\partial \rho}\right)_s = \left[\frac{\rho}{p}\left(1 - \frac{p}{z}\left(\frac{\partial z}{\partial p}\right)_T - \frac{p}{\rho c_p T}\left(1 + \frac{T}{z}\left(\frac{\partial z}{\partial T}\right)_p\right)^2\right)\right]^{-1}, \tag{7}$$

with density ρ. The frictional force is computed from $w = \frac{1}{8} f \rho v \, |v| \, \pi d$. Here the friction factor f is defined as

$$\frac{1}{\sqrt{f}} = -2 \log \left(\frac{\varepsilon}{3.7d} + \frac{2.51}{Re\sqrt{f}} \right), \tag{8}$$

with roughness ϵ, diameter d and Reynolds number Re. The heat transfer between the gas and it surroundings is calculated from $q = -\pi d U \, (T - T_s)$ with U as the total heat transfer coefficient and T_s as the surrounding temperature. Equations (4)–(6) are completed with initial conditions $p(x, 0) = p_0(x)$ and $T(x, 0) = T_0(x)$ and boundary conditions $p(0, t) = p_0$, $T(0, t) = T_0$ and $\dot{m}(L, t) = \phi(t)$. The function $\phi(t)$ represents the end-use gas demand measured at the outlet node whereas p_0 and T_0 are known at the inlet node of the system.

2.1 Numerical Discretization

The method of lines is used to solve the system of PDEs (4)–(6). These set of equations can be rewritten in the compact form $u_t + F u_x + S = 0$ with matrix F and source term S. The spatial domain $[0, L]$ was uniformly discretized by n_x points $x_i = i\Delta x$, $i = 1, \ldots, n_x$. The system of ordinary differential equations yields

$$\frac{\mathrm{d}u(t)}{\mathrm{d}t} = -F(u)\mathcal{B}u(t) - S(u(t)) = f(u(t)), \quad u(t_0) = u_0, \quad t \in [t_0, t_f], \tag{9}$$

with state vector $u(t) = \left[p_1(t), \cdots, p_{n_x}(t), T_1(t), \cdots, T_{n_x}(t), \dot{m}_1(t), \cdots, \dot{m}_{n_x}(t) \right]^\top$ and $\mathcal{B} = \sum_{j=1}^{n} I_j \otimes \mathcal{B}^{(j)}$ defining the computational stencil. In the analysis, we use a central difference scheme of order two (CD2) and four (CD4) whereas the weighting coefficients in sub-matrix $\mathcal{B}^{(j)}$ are defined as [7]

$$\frac{1}{2\Delta x} \begin{bmatrix} -3 & 4 & -1 \\ -1 & 0 & 1 \\ 1 & -4 & 3 \end{bmatrix}, \quad \frac{1}{24\Delta x} \begin{bmatrix} -50 & 96 & -72 & 32 & -6 \\ -6 & -20 & 36 & -12 & 2 \\ 2 & -16 & 0 & 16 & -2 \\ -2 & 12 & -36 & 20 & 6 \\ 6 & -32 & 72 & -96 & 50 \end{bmatrix}, \tag{10}$$

respectively. The structure of the Jacobian $J = \partial_u f(t, u)$ is defined by the discretization scheme. If we considering for example 101 and 1001 spatial nodes only 4.9% and 0.5% of the entries, respectively, are nonzeros. Thus, we are handling sparse Jacobians. For our test problem, the eigenvalues λ of J are complex with negative real parts. Using the CD4 scheme, the ratio $\max |\mathrm{Re}(\lambda_j)| / \min |\mathrm{Re}(\lambda_j)|$ for a grid size of 101 and 1001 are $8.4 \cdot 10^{-3}$ and 0.8, respectively. If we compute max $|\mathrm{Re}(\lambda_j)| \cdot (t_f - t_0)$, the corresponding values are 2.6 and $7.8 \cdot 10^3$ for the interval [0, 24 h]. These values suggest that a nonstiff solver is preferred. On the other hand, stiffness based on eigenvalues can be a too liberal condition [8]. As mentioned in [4], more pragmatic is to consider the system as stiff in case implicit schemes are more efficient than explicit schemes. Therefore, we examine both type of solvers.

Runge–Kutta Schemes

The general s-stage Runge–Kutta scheme is of the form

$$k_i = f(t_n + c_i \Delta t, u_n + \Delta t \sum_{j=1}^{s} a_{ij} k_j), \quad i = 1, \ldots, s, \tag{11}$$

$$u_{n+1} = u_n + \Delta t \sum_{i=1}^{s} b_i k_i, \tag{12}$$

and characterized by the coefficients a_{ij}, b_i and c_i [2]. For explicit Runge–Kutta methods, $a_{ij} = 0$ for $i \leqslant j$ and require only a right-hand-side function evaluation per stage but the Courant number sets Δt and as a results lowers the efficiency. Embedded Runge–Kutta methods provide an efficient way to estimate the local error for step size control. A successful nonstiff scheme is the RK method of order (5)4 [9]. If $a_{ij} = 0$ for $i < j$ and at least one $a_{ii} \neq 0$ we obtain a DIRK method. If on the other hand, $a_{ii} = \gamma$ for $i = 1, \ldots, s$ we obtain a single diagonally implicit Runge–Kutta schemes (SDIRK) scheme. A system of nonlinear equations at each stage is solved for k_i. It increases the computation time but improves stability and allows for larger time steps.

For stiff problems, reliable and robust solvers are the fully implicit L stable 3-stage RK method of fifth order [4] and the second-order DIRK method [3]. The latter scheme contains an explicit stage. To integrate Eq. (9) from $t = t_n$ to $t_{n+1} = t_n + \Delta t_n$, it starts with the trapezial rule from t_n to $t_{n+\gamma} = t_n + \gamma \Delta t_n$, $u_{n+\gamma} = u_n + \gamma \Delta t_n / 2 \big(f(t, u_n) + f(t, u_{n+\gamma}) \big)$, followed by the second-order backward-differentiation formula from $t_{n+\gamma}$ to t_{n+1} to advance the solution in the following manner:

$$u_{n+1} = \frac{1}{\gamma(2-\gamma)} u_{n+\gamma} - \frac{(1-\gamma)^2}{\gamma(2-\gamma)} u_n + \frac{1-\gamma}{2-\gamma} \Delta t_n f(t, u_{n+1}). \tag{13}$$

It is L-stable and First-Same-As-Last.

Rosenbrock Methods

To accelerate the linear algebra, Krylov techniques have shown to be very efficient for large stiff ODE systems. An s-stage Rosenbrock method for system reads

$$(I - \Delta t_n \gamma J)k_i = f\left(u_n + t_n \sum_{j=1}^{i-1} \alpha_{ij} k_j\right) + t_n J \sum_{j=1}^{i-1} \gamma_{ij} k_j, \quad i = 1, \ldots, s. \tag{14}$$

For the Jacobian a low rank approximation $QQ^\top J$ is used. Herein Q is an orthogonal basis of the Krylov-subspace, i.e., $\mathcal{K} = \{r_i, Jr_i, \ldots, K^\kappa r_i\}$ where r_i is the residual. The matrix Q can be computed using Arnoldi's algorithm. The routine we use is the linearly implicit Runge–Kutta method of order four described in [5].

3 Numerical Experiments

The test problem represents the subsea pipeline from the UK to Belgium [10] with $L = 235$ km and $d = 1.016$ m. The physical properties were computed from the Helmholtz free energy equation of state [11]. At the inlet node the pressure is 125 bar and the temperature is 17 °C. The demand function $\phi(t)$ is based on real data with a time span of 24 h. We take a sampling interval of 10 min. The estimation accuracy is calculated in terms of the root mean square error (rmse). The integration schemes were evaluated with an absolute tolerance atol $= 10^{-6}$ and relative tolerance rtol $= 10^{-3}$. The number of grid points $n_x = 101$ and the sample size $N_p = 20$. The modeling errors are represented by the product of random variables that follow a Gaussian and Gamma distribution, whereas the measurement noise is assumed Gaussian. We assume that the noise statistics are known. In each simulation while using a different time stepping solver, we used the same random seeds. Inverse crime [12] is avoided by generating the synthetic data from the fine grid reference solutions.

We noticed that if the forward problem is solved without random perturbations, the ERK scheme showed for both CD2 and CD4 spurious oscillations in the solution domain for the mass flux (see Fig. 1). Not illustrated here but minor oscillations were detected for the ROK method. For the DIRK and IRK schemes such oscillations were not observed. If we set the absolute tolerance to 10^{-6} this issue was resolved. The rmse values are shown in Table 1 and clearly indicate a significant improvement if we tighten the tolerance.

In the next step, we examine the integration schemes within the data assimilation framework. From Table 2 we conclude that the difference in estimation accuracy between the integration schemes is small. It seems that the spurious oscillations that were observed before were filtered out. The posterior pdf and estimates for the mass flux are shown in Fig. 2. The accuracy improves if we use the finite difference scheme of order four but with an average increase of approximately 30%

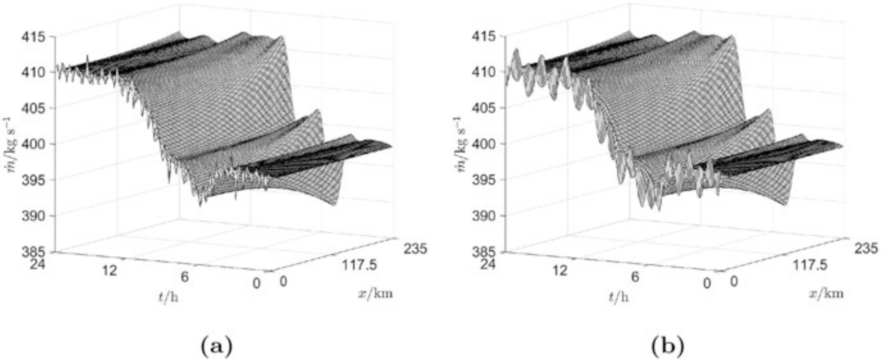


Fig. 1 Spurious oscillations for a single noise-free integration using ERK with CD2 (**a**) and CD4 (**b**). atol $= 10^{-6}$, rtol $= 10^{-3}$ and $n_x = 101$

Table 1 Results for the noise-free forward problem using ERK scheme

		atol $= 10^{-6}$, rtol $= 10^{-3}$	atol $= 10^{-6}$, rtol $= 10^{-6}$
CD2	rmse$_p$ (Pa)	34.2	4.75
	rmse$_T$ (K)	$1.95 \cdot 10^{-4}$	$1.78 \cdot 10^{-4}$
	rmse$_{\dot{m}}$ (kg s^{-1})	0.123	$2.98 \cdot 10^{-4}$
CD4	rmse$_p$ (Pa)	44.6	$6.25 \cdot 10^{-2}$
	rmse$_T$ (K)	$1.11 \cdot 10^{-4}$	$2.04 \cdot 10^{-6}$
	rmse$_{\dot{m}}$ (kg s^{-1})	0.255	$2.46 \cdot 10^{-4}$

Table 2 Results within PF framework

		t_{elaps}/s[a]	rmse$_p$	rmse$_T$	rmse$_{\dot{m}}$
ERK	CD2	14	$8.09 \cdot 10^3$	0.372	1.13
	CD4	23	$8.02 \cdot 10^3$	0.297	1.17
DIRK	CD2	18	$8.08 \cdot 10^3$	0.370	1.16
	CD4	27	$8.01 \cdot 10^3$	0.297	1.05
IRK	CD2	125	$8.09 \cdot 10^3$	0.370	1.16
	CD4	157	$8.05 \cdot 10^3$	0.297	1.06
ROK	CD2	17	$8.16 \cdot 10^3$	0.367	1.15
	CD4	26	$8.03 \cdot 10^3$	0.297	1.04

[a] Mean elapsed time for one model realization and time span

in computation time. In terms of efficiency, the ERK scheme is superior, followed by the DIRK and ROK methods. If we set rtol $= 10^{-6}$ and repeat the experiments, we did not observe significant improvements in accuracy. In fact, we are oversolving the problem, which only leads to higher computation times. Keeping in mind the oscillating behaviour of the ERK and ROK schemes, the DIRK method seems to be a good compromise between efficiency and robustness.

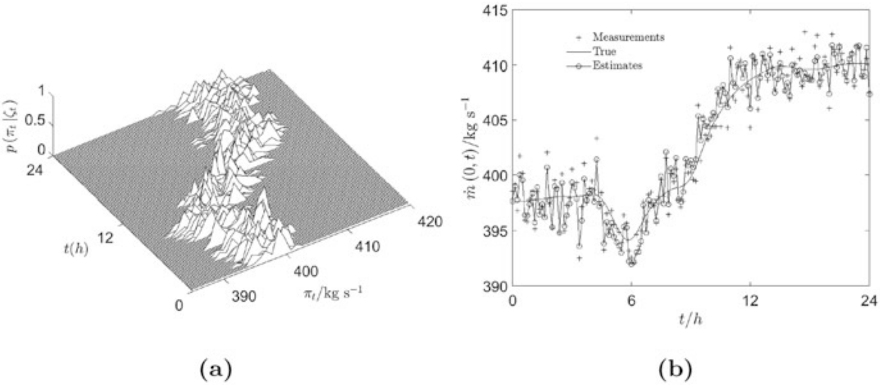


Fig. 2 Evolution of posterior pdf $p(\pi_t|\zeta_t)$ (**a**) and estimates (**b**) using ERK and CD4 with atol $= 10^{-6}$, rtol $= 10^{-3}$

4 Conclusion

In this work, we discussed the performance of different classes of integration schemes within the framework of data assimilation. For the numerical experiments we used an offshore pipeline. Results indicate that the ERK is most efficient, but suffered from spurious oscillations in case the tolerance was set too loose. On the other hand, for the parameter set we used these oscillations were filtered out by algorithm. The DIRK scheme showed to be a good compromise between efficiency and robustness. The benefits from tightening the tolerance is small and oversolves the problem at higher computation cost.

References

1. A. Doucet, N. de Freitas, N. Gordon, *An Introduction to Sequential Monte Carlo Methods*. Sequential Monte Carlo Methods in Practice. Statistics for Engineering and Information Science (Springer, New York, NY, 2001)
2. E. Hairer, S.P. Norsett, G. Wanner, *Solving Ordinary Differential Equations I. Nonstiff Problems*, vol. 8, 2nd edn. (Springer-Verlag: Berlin, Germany, 1993)
3. L. Shampine, M. Hosea, Applied Numerical Mathematics **20**, 21 (1996)
4. E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II*, 2nd edn. (Springer-Verlag Berlin Heidelberg, 1996)
5. R. Weiner, B. Schmitt, H. Podhaisky, Applied Numerical Mathematics **25**, 303 (1997)
6. F.E. Uilhoorn, Optimal Control Applications and Methods **40**(4) (2019)
7. W.E. Schiesser, *Partial Differential Equation Analysis in Biomedical Engineering: Case Studies with MATLAB* (Cambridge University Press, 2013)
8. D.J. Higham, L.N. Trefethen, BIT Numerical Mathematics **33**(2), 285 (1993)
9. J.R. Dormand, P.J. Prince, Journal of Computational and Applied Mathematics **6**, 19 (1980)
10. Interconnector, https://www.interconnector.com/.

11. O. Kunz, R. Klimeck, W. Wagner, M. Jaeschke, The GERG-2004 wide-range equation of state for natural gases and other mixtures. GERG TM15 2007. Tech. rep., Fortschr.-Ber. VDI, Reihe 6, Nr. 557, VDI Verlag, Düsseldorf; also available as GERG Technical Monograph 15 (2007)
12. J. Kaipio, E. Somersalo, *Statistical and Computational Inverse Problems*. Applied Mathematical Sciences (Springer-Verlag New York, 2005)

Matrix Oriented Reduction of Space-Time Petrov-Galerkin Variational Problems

Julian Henning, Davide Palitta, Valeria Simoncini, and Karsten Urban

Abstract Variational formulations of time-dependent PDEs in space and time yield $(d + 1)$-dimensional problems to be solved numerically. This increases the number of unknowns as well as the storage amount. On the other hand, this approach enables adaptivity in space and time as well as model reduction w.r.t. both type of variables. In this paper, we show that matrix oriented techniques can significantly reduce the computational timings for solving the arising linear systems outperforming both time-stepping schemes and other solvers.

1 Introduction

Time-stepping schemes based upon variational semi-discretizations are the standard approach for the numerical solution of time-dependent partial differential equations (PDEs). Using a variational formulation in space and a subsequent discretization e.g. in terms of finite elements, one is left with an evolution problem in time. Standard finite difference techniques then yield a time-marching scheme, where a spatial problem needs to be solved in each time step.

Even though theoretical investigations on space-time variational formulations of PDEs have been around for a long time, [11], it was seen prohibitive to treat the

J. Henning · K. Urban (✉)
Institute for Numerical Mathematics, Ulm University, Ulm, Germany
e-mail: julian.henning@uni-ulm.de; karsten.urban@uni-ulm.de

D. Palitta
Research Group Computational Methods in Systems and Control Theory (CSC), Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany
e-mail: palitta@mpi-magdeburg.mpg.de

V. Simoncini
Dipartimento di Matematica, Università di Bologna, Centro AM^2, Bologna, Italy
IMATI-CNR, Pavia, Italy
e-mail: valeria.simoncini@unibo.it

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_104

time as an additional variable for numerical simulations. In fact, if $\Omega \subset \mathbb{R}^d$ denotes the spatial domain, adding the time $t \in (0, T) =: I$ as an additional unknown results in a PDE on $\Omega_I := I \times \Omega$ in dimension $d+1$, which is costly both w.r.t. the amount of storage and the required computation time.

Also due to the increasing computing power the point of view has changed over the past years. In fact, being able to simulate problems for d up to three until about 20 years ago, adding another dimension seemed to be impossible. Nowadays, where we face high-dimensional problems (e.g. from quantum physics or finance) with $d \gg 100$, adding another dimension seems almost negligible. Another aspect to use space-time variational problems arose from model reduction of parameterized time-dependent PDEs. In fact treating both time and space as variables allows one to perform model reduction for space *and* time, [16]. The time-stepping model reduction approach yields a time-marching scheme for a reduced spatial dimension but with the same number of time steps, [8].

In this paper, we address the question of how to efficiently solve the linear systems arising from a (full) Petrov-Galerkin discretization of space-time variational formulations of time-dependent PDEs. It turns out that the involved coefficient matrices, combining space and time discretizations, have a tensorproduct structure, which allows us to use more efficient matrix equations solvers than what can be done with the usual vector representation.

This paper is organized as follows: In Sect. 2 we review space-time variational formulations of some PDEs and describe corresponding Petrov-Galerkin discretizations as well as the arising linear systems in Sect. 3; Sect. 4 is devoted to the description of the numerical schemes and Sect. 5 to numerical experiments, in particular the comparison with time-stepping schemes.

2 Space-Time Variational Formulation of PDEs

The Heat Equation Let $A : X \to X'$ be an elliptic operator on $X := H_0^1(\Omega)$ associated to a coercive bilinear form $a : X \times X \to \mathbb{R}$, and $f \in L_2(I; X')$. We look for $u \in U := H^1_{(0)}(I; X') \cap L_2(I; X)$ such that[1] $u_t + Au = f$, $u(0) = 0$, where homogeneous initial conditions are chosen only for convenience. The variational formulation then reads

$$\text{find } u \in U : \quad b(u, v) = \langle f, v \rangle \quad \text{for all } v \in V, \tag{1}$$

where $V := L_2(I; X)$, $b(u, v) := \int_0^T \int_\Omega u_t(t, x)\, v(t, x)\, dx\, dt + \int_0^T a(u(t), v(t))\, dt$ and $\langle f, v \rangle := \int_0^T \int_\Omega f(t, x)\, v(t, x)\, dx\, dt$. The well-posedness is ensured by Nečas' conditions, namely boundedness, injectivity and inf-sup condition of $b(\cdot, \cdot)$, [6].

[1] $H^1_{(0)}(I; X') := \{w : I \to X' : w \in H^1(I; X'), w(0) = 0\}$, recall that $H^1(I; X') \hookrightarrow C(\bar{I}; X')$.

The Wave Equation Next, we consider an equation of wave type. Here, for $H := L_2(\Omega)$, we view the operator A as a mapping $A : \mathrm{Dom}(A) := \{\phi \in H : A\phi \in H\} \to H$, or $A : H \to \mathrm{Dom}(A)'$. For $f \in L_2(I; H)$, we seek $u \in L_2(I; H)$ such that $u_{tt} + Au = f$, $u(0) = 0$, $u_t(0) = 0$, where we choose homogeneous initial conditions again only for convenience. In this case, it is not so obvious how to setup a well-posed variational form. It turns out that a very-weak setting is appropriate. We formulate the problem as in (1) by using $U := L_2(I; H)$ as trial and $V := \{v \in L_2(I; H) : v_{tt} + Av \in L_2(I; H),\ v(T) = v_t(T) = 0\}$ as test space. Then, one can show that (1) is well-posed for $b(u, v) := (u, v_{tt} + Av)_{L_2(I;H)}$ and $\langle f, v\rangle := (f, v)_{L_2(I;H)}$ for fixed $T < \infty$.

3 Petrov-Galerkin Discretizations

In order to determine a numerical approximation to the solution of a variational problem (1), one chooses finite-dimensional trial and test spaces, $U_\delta \subset U$, $V_\delta \subset V$, respectively. For convenience, we assume that their dimension is equal, i.e., $N_\delta := \dim U_\delta = \dim V_\delta$. The Petrov-Galerkin method then reads

$$\text{find } u_\delta \in U_\delta : \quad b(u_\delta, v_\delta) = \langle f, v_\delta\rangle \quad \text{for all } v_\delta \in V_\delta. \tag{2}$$

As opposed to the coercive case, the well-posedness of (2) is not inherited from that of (1). In fact, the spaces U_δ and V_δ need to be appropriately chosen in the sense that the discrete inf-sup (or LBB—Ladyshenskaja-Babuška-Brezzi) condition holds, i.e., there exists an $\beta > 0$ such that $\beta_\delta := \inf_{u_\delta \in U_\delta} \sup_{v_\delta \in V_\delta} \frac{b(u_\delta, v_\delta)}{\|u_\delta\|_U \|v_\delta\|_V} \geq \beta > 0$, where the crucial point is that $\beta \neq \beta_\delta$. The size of β is also crucial for the error analysis, since it holds that $\|u - u_\delta\|_U \leq \frac{1}{\beta} \inf_{w_\delta \in U_\delta} \|u - w_\delta\|_U$, [17].

The Heat Equation Starting with the temporal discretization, choose some integer $N_t > 1$ and set $\Delta t := T/N_t$ resulting in a temporal triangulation $\mathcal{T}_{\Delta t}^{\mathrm{time}} \equiv \{t^{k-1} \equiv (k-1)\Delta t < t \leq k\,\Delta t \equiv t^k, 1 \leq k \leq N_t\}$ in time. Denote by $S_{\Delta t} = \mathrm{span}\{\sigma^1, \dots, \sigma^{N_t}\}$ piecewise linear finite elements on I, where σ^k is the (interpolatory) hat-function with the nodes t^{k-1}, t^k and t^{k+1} (resp. truncated for $k \in \{0, N_t\}$) and $Q_{\Delta t} = \mathrm{span}\{\tau^1, \dots, \tau^{N_t}\}$ piecewise constant finite elements, where $\tau^k := \chi_{I^k}$, the characteristic function on the temporal element $I^k := (t^{k-1}, t^k)$. For the spatial discretization, we choose any conformal $X_h = \mathrm{span}\{\phi_1, \dots, \phi_{N_h}\} \subset X$, e.g. piecewise linear finite elements. Then, we set $U_\delta := S_{\Delta t} \otimes X_h$, $V_\delta = Q_{\Delta t} \otimes X_h$, $\delta = (\Delta t, h)$. It can be shown that this yields LBB. Moreover, for $A = -\Delta$ and choosing the energy norm on X as well as a slightly modified norm on U, one can even prove that $\beta = 1$, [16]. Finally, we remark that this specific discretization coincides with the Crank–Nicolson (CN) scheme if a trapezoidal approximation of the right-hand side temporal integration is used. Hence, we can later compare space-time Petrov-Galerkin numerical schemes with a CN time-stepping scheme.

Finally, we detail the linear system of equations $B_\delta^T u_\delta = f_\delta$, where

$$[B_\delta]_{(k,i),(\ell,j)} = (\dot{\sigma}^k, \tau^\ell)_{L_2(I)}\,(\phi_i, \phi_j)_{L_2(\Omega)} + (\sigma^k, \tau^\ell)_{L_2(I)}\,a(\phi_i, \phi_j), \tag{3}$$

$$[f_\delta]_{(\ell,j)} = (f, \tau^\ell \otimes \phi_j)_{L_2(I;H)}, \tag{4}$$

which means that we get a tensorproduct structure for the stiffness matrix $B_\delta = D_{\Delta t} \otimes M_h + C_{\Delta t} \otimes A_h$, where the matrices are defined in an obvious manner. The right-hand side is not yet in a tensorproduct structure. However, we can achieve that by determining an approximation

$$f(t,x) \approx \sum_{p=1}^{P} \vartheta_p(t) f_p(x) =: f^P(t,x), \tag{5}$$

e.g. by the *Empirical Interpolation Method* (EIM), [2]. By choosing P sufficiently large, we can achieve any desired accuracy. Then, we get $[f_\delta^P]_{(\ell,j)} = \sum_{p=1}^{P} (\vartheta_p, \tau^\ell)_{L_2(I)}\,(f_p, \phi_j)_{L_2(\Omega)}$, i.e., $f_\delta^P = \sum_{p=1}^{P} h_p \otimes g_p$.

The Wave Equation Constructing a stable pair of trial and test spaces for the wave equation is again a nontrivial task. Following an idea from [4], we first define the test space and construct the trial space in a second step in order to guarantee LBB, which, however, deteriorates with increasing T. Doing so, we set $R_{\Delta t} := \text{span}\{\varrho^1, \dots, \varrho^{N_t}\} \subset H_T^2(I) := \{\rho \in H^2(I) : \rho(T) = \dot{\rho}(T) = 0\}$, e.g. piecewise quadratic splines on $\mathcal{T}_{\Delta t}^{\text{time}}$. For space, we choose any conformal $Z_h = \text{span}\{\psi_1, \dots, \psi_{N_h}\} \subset H^2(\Omega) \cap H_0^1(\Omega)$, e.g. piecewise quadratic finite elements. Then, we define $V_\delta := R_{\Delta t} \otimes Z_h$, a tensor product space. The trial space U_δ is constructed by applying the adjoint PDE operator to each test basis function, i.e. $v_{k,i} := \frac{d^2}{dt^2}\varrho^k(t)\psi_i(x) + A(\varrho^k(t)\psi_i(x)) = \ddot{\varrho}^k(t)\psi_i(x) + \varrho^k(t)A\psi_i(x)$. We detail the arising linear system of equations starting with the stiffness matrix

$$\begin{aligned}
[B_\delta]_{(k,i),(\ell,j)} &= b(v_{k,i}, \varrho^\ell \otimes \psi_j) = (\ddot{\varrho}^k \otimes \psi_i + \varrho^k \otimes A\psi_i, \ddot{\varrho}^\ell \otimes \psi_j + \varrho^\ell \otimes A\psi_j)_{L_2(I;H)} \\
&= (\ddot{\varrho}^k, \ddot{\varrho}^\ell)_{L_2(I)}\,(\psi_i, \psi_j)_{L_2(\Omega)} + (\ddot{\varrho}^k, \varrho^\ell)_{L_2(I)}\,(\psi_i, A\psi_j)_{L_2(\Omega)} \\
&\quad + (\varrho^k, \ddot{\varrho}^\ell)_{L_2(I)}\,(A\psi_i, \psi_j)_{L_2(\Omega)} + (\varrho^k, \varrho^\ell)_{L_2(I)}\,(A\psi_i, A\psi_j)_{L_2(\Omega)},
\end{aligned}$$

so that $B_\delta = Q_{\Delta t} \otimes M_h + (D_{\Delta t} + D_{\Delta t}^T) \otimes A_h + M_{\Delta t} \otimes Q_h$, again with obvious definitions of the matrices. For the right-hand side, we perform again an EIM-type approximation $f^P(t,x)$. Then, $[f_\delta^P]_{(\ell,j)} = \sum_{p=1}^{P} (\vartheta_p \otimes f_p, \varrho^\ell \otimes \psi_j)_{L_2(I;H)} = \sum_{p=1}^{P} (\vartheta_p, \varrho^\ell)_{L_2(I)}\,(f_p, \psi_j)_{L_2(\Omega)}$, so that the right-hand side has the same structure as in the first example. Due to the asymptotic behavior of the inf-sup-constant, we expect stability problems as $\Delta t \to 0$, i.e., $N_t \to \infty$.

4 Efficient Numerical Methods for Tensorproduct Systems

In both cases described above (and in fact also in space-time variational formulations of other PDEs), we obtain a (regular) linear system of the form

$$B_\delta u_\delta = f_\delta \quad \text{with} \quad B_\delta = \sum_{p=1}^{P_B} D_p \otimes A_p, \quad f_\delta = \sum_{\ell=1}^{P_f} h_\ell \otimes q_\ell, \tag{6}$$

where all involved matrices are sparse and (at least some of) the A_q are s.p.d. Recall that $(D_p \otimes A_p)x = \text{vec}(A_p X D_p^T)$, where vec stacks the columns of a given matrix one after the other, and $x = \text{vec}(X)$. We can thus rewrite the system $B_\delta u_\delta = f_\delta$ in (6) as the linear *matrix* equation $\sum_{p=1}^{P_B} A_p U_\delta D_p^T = \sum_{\ell=1}^{P_f} q_\ell h_\ell^T$, with $u_\delta = \text{vec}(U_\delta)$. Matrix equations are receiving significant attention in the PDE context, due to the possibility of maintaining the structural properties of the discretized problem, while limiting memory consumptions; see [14]. Under certain hypotheses, a large variety of discretization methodologies such as finite differences, isogeometric analysis, spectral (element) methods, certain finite element methods as well as various parametric numerical schemes rely on tensor product spaces; see, e.g., [1, 5, 9, 10]. More recently, all-at-once time discretizations have shown an additional setting where tensor product approximations naturally arise; see, e.g., [12] and references therein. Among the various computational strategies discussed in the literature [14], here we focus on projection methods that reduce the original equation to a similar one, but of much smaller dimension.

Discretized Heat Equation The problem $B_\delta u_\delta = f_\delta$ stemming from (3,4) yields the following generalized Sylvester equation

$$M_h U_\delta D_{\Delta t} + A_h U_\delta C_{\Delta t} = F_\delta, \qquad \text{with} \quad F_\delta := [g_1, \ldots, g_P][h_1, \ldots, h_P]^T. \tag{7}$$

The spatial stiffness and mass matrices A_h and M_h typically have significantly larger dimensions N_h than the time discretization matrices $D_{\Delta t}$, $N_{\Delta t}$, i.e., $N_t \ll N_h$. We therefore use a reduction method only for the space variables by projecting the problem onto an appropriate space. A matrix Galerkin orthogonality condition is then applied to obtain the solution: given $V_m \in \mathbb{R}^{N_h \times k_m}$, $k_m \ll N_h$, with orthonormal columns, we consider the approximation space range(V_m) and seek $Y_m \in \mathbb{R}^{k_m \times N_t}$ such that $U_{\delta,m} := V_m Y_m \approx U_\delta$ and the residual $R_m := F_\delta - (M_h U_{\delta,m} D_{\Delta t} + A_h U_{\delta,m} C_{\Delta t})$ satisfies the Galerkin condition $R_m \perp \text{range}(V_m)$. Imposing this orthogonality yields that $V_m^T R_m = 0$ is equivalent to $V_m^T F_\delta V_m - (V_m^T M_h V_m) Y_m D_{\Delta t} - (V_m^T A_h V_m) Y_m C_{\Delta t} = 0$. The resulting problem is again a generalized Sylvester equation, but of much smaller size, therefore Schur-decomposition oriented methods can cheaply be used, [14, sec.4.2], see [14] for a discussion on projection methods as well as their matrix and convergence properties.

For selecting V_m, let $F = F_1 F_2^T$ with F_1 having full column rank. Given the properties of A_h, M_h, we propose to employ the rational Krylov subspace $\mathcal{RK}_m := \text{range}([F_1, (A_h - \sigma_2 M_h)^{-1} M_h F_1, (A_h - \sigma_3 M_h)^{-1} M_h F_1, \ldots, (A_h - \sigma_m M_h)^{-1} M_h F_1])$, where the shifts σ_s can be determined adaptively while the space is being generated; see [14] for a description and references. The obtained spaces are nested, $\mathcal{RK}_m \subseteq \mathcal{RK}_{m+1}$, therefore the space can be expanded if the approximation is not sufficiently good. To include a residual-based stopping criterion, the residual norm can be computed in a cheap manner, see, e.g., [7, 12] for the technical details.

Discretized Wave Problem The problem $B_\delta u_\delta = f_\delta$ now takes the matrix form

$$M_h U_\delta Q_{\Delta t}^T + A_h U_\delta (D_{\Delta t} + D_{\Delta t}^T) + Q_h U_\delta M_{\Delta t} = F_\delta. \tag{8}$$

This three-term equation cannot be solved directly as before, therefore we opt for using preconditioned GMRES on the vectorized equation. The preconditioner is given by the functional $\mathcal{P} : U \to M_h U Q_{\Delta t}^T + Q_h U M_{\Delta t}$, corresponding to the discretized forth order operators, and exploits the matrix structure. Hence, at the kth GMRES iteration we solve the generalized Sylvester equation $M_h W Q_{\Delta t}^T + Q_h W M_{\Delta t} = V_k$ where V_k is such that $v_k = \text{vec}(V_k)$ is the previous basis vector. Since in this one-dimensional problem dimensions are limited, this matrix equation is solved by explicitly diagonalizing the pairs (Q_h, M_h) and $(Q_{\Delta t}, M_{\Delta t})$ [14].

5 Numerical Experiments

In this section we show that the numerical solution of the linear system $B_\delta u_\delta = f_\delta$ can largely benefit from the exploitation of its Kronecker sum structure (6). The performance of the all-at-once methods is compared in terms of both computational time and memory requirements. For the heat equation, we also document the performances of CN in terms of computational time. We are not aware of any variant of CN that is able to exploit the low-rank structure of the underlying problem and we thus employ the classical CN scheme. Such implementation leads to running times that significantly increase with N_t and a storage demand that is always equal to $N_t \cdot N_h$ as the full U_δ is allocated.

The tolerance of the final relative residual norm is set to 10^{-8} and in the following tables we also report the number of iterations needed to achieve such accuracy and the numerical rank of the computed solution. All results were obtained by running Matlab R2017b on a standard node of the Linux cluster Mechthild hosted at the MPI in Magdeburg, Germany.[2]

Example 5.1 (The Heat Equation) We consider the equation on the cube $\Omega = (-1, 1)^3$ with homogeneous Dirichlet boundary conditions and the time interval

[2] See https://www.mpi-magdeburg.mpg.de/cluster/mechthild for further details.

Table 1 Results for Example 5.1: different values of N_h and N_t. Memory allocations for RKSM and LR-FGMRES+RKSM are given by $\mu_{\text{mem}} \cdot (N_h + N_t)$. For CN we report only the computational timings

		RKSM				LR-FGMRES+RKSM				CN	
N_h	N_t	Its	μ_{mem}	rank(U_δ)	Time (s)	Its	μ_{mem}	rank(U_δ)	Time (s)	Direct	Iterative
41,300	300	13	14	9	25.96	4	74	10	82.89	123.43	59.10
	500	13	14	9	30.46	4	75	11	83.93	143.71	78.01
	700	13	14	9	28.17	4	86	11	89.99	153.38	93.03
347,361	300	14	15	9	820.17	4	78	9	2319.67	14,705.10	792.42
	500	14	15	9	828.34	4	80	9	2384.39	15,215.47	1041.47
	700	14	15	7	826.93	4	97	9	2327.76	15,917.52	1212.57

$I := (0, 10)$ with initial conditions $u(0, x, y, z) \equiv 0$. The right-hand side is $f(t, x, y, z) := 10\sin(t)t\cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)\cos(\frac{\pi}{2}z)$ and its discretized version is thus low rank. For discretization in space, linear finite elements were chosen, leading to the discretized generalized Sylvester equation in (7). We compare the performance of the Galerkin method based upon rational Krylov spaces described in Sect. 4 (denoted RKSM) with that of a low-rank version of preconditioned GMRES (denoted LR-FGMRES-RKSM). See, e.g., [15] for further insights about low-rank Krylov routines applied to linear matrix equations. The LR-FGMRES-RKSM preconditioner is chosen as a fixed (five) number of iterations of the rational Krylov Galerkin method.[3] The results are displayed in Table 1. Due to the 3D nature (in space) of the problem, the CN method with a direct linear solver[4] leads to an excessive workload compared with the all-at-once approaches for all considered values of N_h and N_t, with the computational time growing with the number of time steps N_t. The performance of the other methods is independent of the time discretization, and it only depends on the spatial component of the overall discrete operator. In fact, spatial mesh independence seems to also be achieved. The CN method is more competitive in terms of computational time when equipped with an iterative linear solver.[5]

Example 5.2 (The Wave Equation) We consider the wave problem with $A = -\Delta$ on $\Omega = (0, 1)$ with homogeneous Dirichlet boundary conditions and $I := (0, 1)$. Setting $f(t, x) := \sin(2\pi x) + 4\pi^2 t^2 \sin(2\pi x)$ yields the analytical solution $u(t, x) = t^2 \sin(2\pi x)$. We choose cubic B-Splines for the discretization in space and time. The discretized problem thus leads to the matrix equation in (8). In Fig. 1 we report some our preliminary results. Note, that the discretization above does not yield an equivalent time-stepping scheme with which we could do comparisons.

[3] Since the preconditioner is a non-linear operator, a flexible variant of GMRES is used.

[4] The LU factors of the CN coefficient matrix are computed once and for all at the beginning of the procedure.

[5] We employ GMRES preconditioned with ILU (zero fill-in). The same solver is used for the RKSM basis construction.

		GMRES+LYAP			backslash
N_h	N_t	Its.	rank(U_δ)	Time (s)	Time (s)
256	256	16	13	0.21	1.17
	512	36	35	1.51	2.39
	1024	81	74	20.97	12.61
512	256	26	31	0.61	2.30
	512	40	43	2.64	5.09
	1024	81	74	20.97	12.61
1024	256	50	59	3.55	4.82
	512	68	72	10.12	11.13
	1024	102	92	54.15	24.28

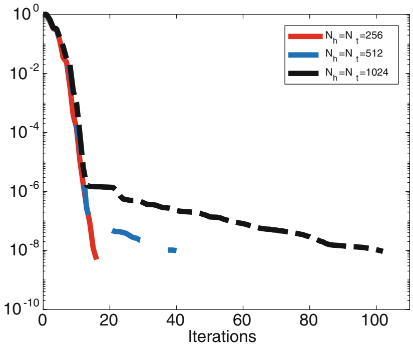


Fig. 1 Example 5.2. Left: Results for different values of N_h and N_t. Right: Relative residual norm history for some values of N_h and N_t

The table on the left shows that the performances of our preconditioned scheme are quite good for small values of N_h and N_t. Indeed, in this case, the preconditioner manages to drastically reduce the number of iterations needed to converge so that GMRES+LYAP[6] turns out to be faster than the Matlab solver `backslash` applied to the solution of the linear system $B_\delta u_\delta = f_\delta$, in spite of the 1D nature (in space) of the problem. However, the effectiveness of the adopted preconditioner worsens by increasing the number of degrees of freedom. This is due to a dramatic increment in the condition number of the coefficient matrices (see the discussion at the end of Sect. 3) that causes an abrupt very slow decrement (almost stagnation) in the GMRES residual at the level that seems to be related to the conditioning of the involved matrices, see Fig. 1 (right). As it is, the problem associated with handling this ill-conditioning in the algebraic equation is crucial for the overall solver performance, and will be the topic of future works. Alternatively, one may try to directly address the solution of the multiterm matrix equation (8) as it is done in [13] for certain stochastic PDEs.

References

1. M. Bachmayr, A. Cohen, and W. Dahmen. Parametric PDEs: sparse or low-rank approximations? *IMA J. Numer. Anal.*, 38:1661–1708, 2018.
2. M. Barrault, Y. Maday, N. C. Nguyen, and A. T. Patera. An ‘empirical interpolation’ method: application to efficient reduced-basis discretization of partial differential equations. *C. R. Math. Acad. Sci. Paris*, 339(9):667–672, 2004.
3. R. H. Bartels and G. W. Stewart. Solution of the matrix equation $AX+XB = C$ [F4]. *Commun. ACM*, 15(9):820–826, 1972.

[6] Generalized Lyapunov (i.e., Sylvester) equation solver, [3].

4. J. Brunken, K. Smetana, and K. Urban. (Parametrized) First Order Transport Equations: Realization of Optimally Stable Petrov-Galerkin Methods. *SIAM J. Sci. Comput.*, 41(1):A592–A621, 2019.
5. W. Dahmen, R. DeVore, L. Grasedyck, and E. Süli. Tensor-sparsity of solutions to high-dimensional elliptic partial differential equations. *Found. Comput. Math.*, 16:813–874, 2016.
6. R. Dautray and J.-L. Lions. *Mathematical analysis and numerical methods for science and technology. Vol. 5.* Springer-Verlag, Berlin, 1992. Evolution problems. I.
7. V. Druskin and V. Simoncini. Adaptive rational Krylov subspaces for large-scale dynamical systems. *Systems and Control Letters*, 60:546–560, 2011.
8. S. Glas, A. Mayerhofer, and K. Urban. Two ways to treat time in reduced basis methods. In *Model reduction of parametrized systems*, volume 17 of *MS&A. Model. Simul. Appl.*, pages 1–16. Springer, Cham, 2017.
9. M. Griebel and H. Harbrecht. On the construction of sparse tensor product spaces. *Math. Comp.*, 82(282):975–994, 2013.
10. B. N. Khoromskij and C. Schwab. Tensor-structured Galerkin approximation of parametric and stochastic elliptic PDEs. *SIAM J. Scient. Comput.*, 33(1):364–385, 2011.
11. J.-L. Lions and E. Magenes. *Problèmes aux limites non homogènes et applications. Vol. 2.* Travaux et Recherches Mathématiques, No. 18. Dunod, Paris, 1968.
12. D. Palitta. Matrix equation techniques for certain evolutionary partial differential equations. arXiv math.NA, no. 1908.11851, 2019.
13. C. E. Powell, D. Silvester, and V. Simoncini. An efficient reduced basis solver for stochastic Galerkin matrix equations. *SIAM J. Sci. Comput.*, 39(1):A141–A163, 2017.
14. V. Simoncini. Computational methods for linear matrix equations. *SIAM Review*, 58(3):377–441, 2016.
15. M. Stoll and T. Breiten. A low-rank in time approach to PDE-constrained optimization. *SIAM J. Sci. Comput.*, 37(1):B1–B29, 2015.
16. K. Urban and A. T. Patera. An improved error bound for reduced basis approximation of linear parabolic problems. *Math. Comp.*, 83(288):1599–1615, 2014.
17. J. Xu and L. Zikatanov. Some observations on Babuška and Brezzi theories. *Numer. Math.*, 94(1):195–202, 2003.

A Variational Formulation for LTI-Systems and Model Reduction

Moritz Feuerle and Karsten Urban

Abstract We consider a variational formulation of Linear Time-Invariant (LTI)-systems and derive a model reduction in dimension and time inspired by space-time variational reduced basis (RB) methods for parabolic problems. A residual-type RB error estimator is derived whose effectivity is investigated numerically.

1 Introduction

Model order reduction (MOR) of (linear) systems is a huge field of research with an enormous amount of literature. On the other hand, the reduced basis (RB) method has become a widely spread technique for reducing parameterized partial differential equations. We refer e.g. to [2], where both model reduction techniques are reviewed. In this paper, we consider a variational formulation of Linear Time-Invariant (LTI) systems that allows us to introduce an RB-type residual error estimator inspired by space-time RB methods for parabolic problems, [6, 7]. This, in turn, yields a reduction not only of the dimension of the LTI system but also w.r.t. the temporal discretization, i.e., the number of time steps.

The paper is organized as follows: In Sect. 2, we introduce a variational formulation of LTI systems and show its well-posedness, Sect. 3 is devoted to Petrov-Galerkin discretizations which are used as a detailed solution for the Reduced Basis Method (RBM) in Sect. 4. We present some numerical results in Sect. 5 and end by conclusions as well as an outlook in Sect. 6.

M. Feuerle · K. Urban (✉)
Institute for Numerical Mathematics, Ulm University, Ulm, Germany
e-mail: moritz.feuerle@uni-ulm.de; karsten.urban@uni-ulm.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_105

2 Variational Formulation for LTI-Systems

We consider LTI systems on some time interval $I := (0, T)$, $T > 0$. Given integers $m, n, p \in \mathbb{N}$, matrices $A \in \mathbb{R}^{n\times n}$ (which is assumed to be s.p.d. for simplicity), $B \in \mathbb{R}^{n\times p}$, $C \in \mathbb{R}^{m\times n}$, $D \in \mathbb{R}^{m\times p}$, a control $u : I \to \mathbb{R}^p$ and an initial state $x_0 \in \mathbb{R}^n$, determine the state $x : I \to \mathbb{R}^n$ and output $y : I \to \mathbb{R}^m$ s.t.

$$\dot{x}(t) + Ax(t) = Bu(t), \quad y(t) = Cx(t) + Du(t), \quad t \in I, \qquad x(0) = x_0. \tag{1}$$

W.l.o.g. we restrict ourselves to the homogeneous case, i.e., $x_0 = 0$, but note, that the inhomogeneous case can easily be incorporated.

A Variational Formulation We multiply the first equation in (1) with a test function $z : I \to \mathbb{R}^n$ and integrate over I, i.e.,

$$\int_0^T (\dot{x}(t), z(t))\, dt + \int_0^T (Ax(t), z(t))\, dt = \int_0^T (Bu(t), z(t))\, dt, \tag{2}$$

where $(\cdot, \cdot)$ denotes the Euclidean scalar product with induced norm $\|\cdot\|$ in $\mathbb{R}^d$, $d \in \{m, n, p\}$. Obviously, (2) makes sense for $z \in Z := L_2(I, \mathbb{R}^n) \equiv L_2(I)^n$, $\|z\|_Z := \|z\|_{L_2(I)^n}$. The desired state function $x : I \to \mathbb{R}^n$ is then sought in the Sobolev-Bochner Hilbert space $X := H^1_{(0)}(I)^n := \{x \in H^1(I)^n : x(0) = 0\}$. As in [6, 7] we consider a slightly stronger norm than the usual graph norm, namely

$$|||x|||^2_{X,\text{Std}} := \|\dot{x}\|^2_{L_2(I)^n} + \|x\|^2_{L_2(I)^n} + \|x(T)\|^2, \tag{3}$$

with the corresponding inner product $(x, v)_{X,\text{Std}} := (\dot{x}, \dot{v})_{L_2(I)^n} + (x, v)_{L_2(I)^n} + (x(T), v(T))$ for $x, v \in X$, which is well-defined recalling that $X \hookrightarrow C([0, T], \mathbb{R}^n)$. Then, setting $U := L_2(I)^p$ as parameter space, we obtain the following variational formulation of (1):

$$\text{for } u \in U \text{ find } x = x(u) \in X : \quad b(x, z) = f(z; u) := (Bu, z)_{L_2(I)^n} \quad \forall z \in Z, \tag{4}$$

where the parameter-independent bilinear form reads $b(x, z) := (\dot{x} + Ax, z)_{L_2(I)^n}$. We stress the fact that $f(\cdot; u)$ is *linear* in u (for $x_0 \neq 0$ affine-linear).

Well-Posedness In order to prove well-posedness of (4), we need to satisfy Nečas' conditions, namely boundedness, injectivity and inf-sup condition of $b(\cdot, \cdot)$. Since the verification is very similar to space-time variational formulation of parabolic initial value problems, we refer to it, [5–7]. In particular, the inf-sup constant can be detailed in similar way, see [6, Prop. 1] and [5, Thm. 5.1].

Proposition 2.1 *Let $A \in \mathbb{R}^{n\times n}$ be s.p.d. with constants $\alpha_A > 0$ and $\gamma_A < \infty$, such that $\alpha_A \|\phi\| \le \|A\phi\| \le \gamma_A \|\phi\|$ for all $\phi \in \mathbb{R}^n$. Then,*

$$\inf_{x\in X} \sup_{z\in Z} \frac{b(x,z)}{|||x|||_{X,Std}\, \|z\|_Z} \ge \beta^{Std} := \frac{\min\{1, \alpha_A \min\{1, \gamma_A^{-2}\}\}}{\sqrt{2}\max\{1, (\alpha_A)^{-1}\}} > 0. \tag{5}$$

In order to (quantitatively) improve the inf-sup-bound in (5), we consider an energy norm, namely $(\phi, \psi)_{\mathcal{A}} := (\phi, \mathcal{A}\psi)$, $\|\phi\|^2_{\mathcal{A}} := (\phi,\phi)_{\mathcal{A}}$ for an s.p.d. matrix $\mathcal{A} \in \mathbb{R}^{n\times n}$, $\phi, \psi \in \mathbb{R}^n$ and (with a slight double use of notation) $(z, w)_{\mathcal{A}} := \int_0^T (z(t), w(t))_{\mathcal{A}}\, dt$ as well as $\|z\|^2_{\mathcal{A}} := (z,z)_{\mathcal{A}}$ for $w, z \in L_2(I)^n$. Then, we set

$$|||x|||^2_X := \|\dot{x}\|^2_{A^{-1}} + \|x\|^2_A + \|x(T)\|^2, \qquad |||z|||_Z := \|z\|_A,$$

and following the reasoning in [7, Prop. 2.6], we can easily show that

$$\inf_{x\in X} \sup_{z\in Z} \frac{b(x,z)}{|||z|||_Z\, |||x|||_X} = \sup_{x\in X} \sup_{z\in Z} \frac{b(x,z)}{|||z|||_Z\, |||x|||_X} = 1 \equiv \beta^{\mathrm{En}}. \tag{6}$$

3 Petrov-Galerkin (Detailed) Discretizations

In order to compute an approximation to the solution of (4), we use a standard Petrov-Galerkin approach. To this end, one constructs finite-dimensional trial and test spaces $X^{\mathcal{N}} \subset X$,$Z^{\mathcal{N}} \subset Z$ with $\dim(X^{\mathcal{N}}) = \dim(Z^{\mathcal{N}}) = \mathcal{N}$. For stability, these spaces need to satisfy a discrete inf-sup (LBB) condition, i.e.,

$$\beta^{\mathcal{N}} := \inf_{x^{\mathcal{N}}\in X^{\mathcal{N}}} \sup_{z^{\mathcal{N}}\in Z^{\mathcal{N}}} \frac{b(x^{\mathcal{N}}, z^{\mathcal{N}})}{|||x^{\mathcal{N}}|||_X\, |||z^{\mathcal{N}}|||_Z} \ge \beta^{\mathrm{En}}_{\mathrm{LB}} > 0, \tag{7}$$

where the lower-bound $\beta^{\mathrm{En}}_{\mathrm{LB}}$ for the inf-sup-constant is independent of $\mathcal{N}$ as $\mathcal{N} \to \infty$. Then, the discrete version of (4) is a Petrov-Galerkin scheme of the form

$$\text{for } u \in U \text{ find } x^{\mathcal{N}}(u) \in X^{\mathcal{N}} :\ b(x^{\mathcal{N}}(u), z^{\mathcal{N}}) = (Bu, z^{\mathcal{N}})_{L_2(I)^n} \quad \forall z^{\mathcal{N}} \in Z^{\mathcal{N}}, \tag{8}$$

where $u \in U$ is possibly suitably discretized (see below). As usual, we define the *primal residual* $r^{\mathrm{pr}}(\cdot; u) \in Z'$ as

$$r^{\mathrm{pr}}(z; u) := f(z; u)_{L_2(I)^n} - b(x^{\mathcal{N}}(u), z) = b(x(u) - x^{\mathcal{N}}(u), z), \quad z \in Z, \tag{9}$$

and its norm by $R^{\mathrm{pr}}(u) := |||r^{\mathrm{pr}}(\cdot; u)|||_{Z'}$. Since $Z = L_2(I)^n$ is a Hilbert (pivot) space, we can identify $Z = Z'$, which significantly reduces the complexity in computing this dual norm (we do not need to determine Riesz representations).

Then, the following error-residual relation is straightforward and well-known (recall $\beta^{\text{En}} \equiv 1$)

$$|||x(u) - x^{\mathcal{N}}(u)|||_X \le R^{\text{pr}}(u) = \|Bu - \dot{x}^{\mathcal{N}}(u) - Ax^{\mathcal{N}}(u)\|_{A^{-1}} =: \Delta^{\text{pr}}(u). \qquad (10)$$

A Time-Marching Discretization We start by introducing a Petrov-Galerkin discretization arising from a (finite element) discretization in time, which leads to a Crank-Nicolson (CN) time-marching scheme. To this end, we choose some integer $K > 1$ and set $\Delta t := T/K$ resulting in a temporal triangulation $\mathcal{T}_{\Delta t}^{\text{time}} \equiv \{t^{k-1} \equiv (k-1)\Delta t < t \le k\,\Delta t \equiv t^k, 1 \le k \le K\}$ in time. Denote by $S_{\Delta t} = \text{span}\{\sigma^1, \dots, \sigma^K\}$ piecewise linear finite elements on I, where σ^k is the (interpolatory) hat-function with the nodes t^{k-1}, t^k and t^{k+1} (resp. truncated for $k \in \{0, K\}$) and $Q_{\Delta t} = \text{span}\{\tau^1, \dots, \tau^K\}$ piecewise constant finite elements, where $\tau^k := \chi_{I^k}$, the characteristic function on the temporal element $I^k := (t^{k-1}, t^k)$. Then, we set $X_{\text{CN}}^{\mathcal{N}} := S_{\Delta t} \otimes \mathbb{R}^n$, $Z_{\text{CN}}^{\mathcal{N}} := Q_{\Delta t} \otimes \mathbb{R}^n$, i.e., the detailed dimension is $\mathcal{N} := Kn$. Within that framework, the detailed approximation amounts computing $x_{\text{CN}}^{\mathcal{N}} \in X_{\text{CN}}^{\mathcal{N}}$ represented as[1] $x_{\text{CN}}^{\mathcal{N}}(t; u) \equiv x_{\text{CN}}^{\mathcal{N}}(t; u)(t) = \sum_{k=1}^{K} \boldsymbol{x}_{\text{CN}}^k \sigma^k(t)$, $x(t^k) \approx \boldsymbol{x}_{\text{CN}}^k \in \mathbb{R}^n$, $k = 1, \dots, K$, $t \in I$, and $\boldsymbol{x}_{\text{CN}}^{\mathcal{N}} := (\boldsymbol{x}_{\text{CN}}^k)_{k=1,\dots,K} \in \mathbb{R}^{K\times n} \cong \mathbb{R}^{Kn} = \mathbb{R}^{\mathcal{N}}$. Setting $\Pi_{\Delta t} := ([\Pi_{\Delta t}]_{k,\ell})_{k,\ell=0,\dots,K}$, $\check{\Pi}_{\Delta t} := ([\Pi_{\Delta t}]_{k,\ell})_{k=0,\dots,K-1,\ell=1,\dots,K}$, $\hat{\Pi}_{\Delta t} := ([\Pi_{\Delta t}]_{k,\ell})_{k=1,\dots,K,\ell=0,\dots,K}$ and $\bar{\Pi}_{\Delta t} := ([\Pi_{\Delta t}]_{k,\ell})_{k,\ell=1,\dots,K}$ for $\Pi \in \{K, L, M, N, O\}$,

$$\begin{aligned} &[K_{\Delta t}]_{k,\ell} := (\dot{\sigma}^k, \dot{\sigma}^\ell)_{L_2(I)}, \quad [L_{\Delta t}]_{k,\ell} := (\sigma^k, \sigma^\ell)_{L_2(I)}, \quad [M_{\Delta t}]_{k,\ell} := (\sigma^k, \tau^\ell)_{L_2(I)} \\ &[N_{\Delta t}]_{k,\ell} := (\dot{\sigma}^k, \tau^\ell)_{L_2(I)} \quad [O_{\Delta t}]_{k,\ell} := (\dot{\sigma}^k, \sigma^\ell)_{L_2(I)}, \end{aligned} \qquad (11)$$

and recalling $[M_{\Delta t}]_{k,\ell} = \frac{\Delta t}{2}(\delta_{k,\ell} + \delta_{k+1,\ell})$ and $[N_{\Delta t}]_{k,\ell} = \delta_{k,\ell} - \delta_{k+1,\ell}$, we obtain $b(x_{\text{CN}}^{\mathcal{N}}, \tau^\ell e_\mu)_{L_2(I)^n} = \big[[Id + \frac{\Delta t}{2}A]\boldsymbol{x}_{\text{CN}}^\ell - [Id - \frac{\Delta t}{2}A]\boldsymbol{x}_{\text{CN}}^{\ell-1}\big]_\mu$.

Discretization of the Control Without any discretization, we can in general not evaluate the term $(Bu, z^{\mathcal{N}})_{L_2(I)^n}$ exactly. As a first attempt, it seems reasonable (as done in the literature of LTIs) to use the same temporal discretization, i.e., $U^{\mathcal{N}} := S_{\Delta t} \otimes \mathbb{R}^p$ and interpolate the control onto the temporal nodes $\mathcal{T}_{\Delta t}^{\text{time}}$, i.e., $u^{\mathcal{N}}(t) := \sum_{k=0}^{K} \boldsymbol{u}^k \, \sigma^k(t)$, $\boldsymbol{u}_{\Delta t} = (\boldsymbol{u}^k)_{k=0,\dots,K} \in \mathbb{R}^{(K+1)\times p}$, where we note that the initial value $u(0)$ does not need to vanish, which is the reason, why the above sum starts from $k = 0$.

Crank-Nicolson Scheme We finally obtain the following iteration: $\boldsymbol{x}^0 := x_0$ and

$$\big[Id - \tfrac{\Delta t}{2} A\big]\boldsymbol{x}_{\text{CN}}^\ell = \big[Id + \tfrac{\Delta t}{2} A\big]\boldsymbol{x}_{\text{CN}}^{\ell-1} + \tfrac{\Delta t}{2}B(\boldsymbol{u}^\ell + \boldsymbol{u}^{\ell-1}), \quad \ell = 1, 2, \dots, K. \qquad (12)$$

[1] We often omit the dependency on the control for simplicity.

In particular, the reduction to homogeneous initial conditions has no effect to the temporal iteration. These considerations also show the well-posedness of the discrete problem (8). Note, that (12) yields an iteration so that one does not need to solve the potentially large linear system as for the second discretization in (15) below. Of course, (12) can also be written as a linear system $(\mathcal{B}^{\mathcal{N}}_{\text{CN}})^T \boldsymbol{x}^{\mathcal{N}}_{\text{CN}}(u^{\mathcal{N}}) = \boldsymbol{f}^{\mathcal{N}}_{\text{CN}}(u^{\mathcal{N}})$, where $[\mathcal{B}^{\mathcal{N}}_{\text{CN}}]_{(k,\nu),(\ell,\mu)} = [\check{N}_{\Delta t}]_{k,\ell}\,[Id]_{\nu,\mu} + [\check{M}_{\Delta t}]_{k,\ell}\,[A]_{\nu,\mu}$, which means that $\mathcal{B}^{\mathcal{N}}_{\text{CN}} = \check{N}_{\Delta t} \otimes Id + \check{M}_{\Delta t} \otimes A$, which is non-symmetric.

Standard Error Estimate An error estimate is derived by using well-known techniques from studying iterations. Denoting by $x(t; u^{\mathcal{N}})$ the solution of (1), we have

$$\|x(t^\ell; u^{\mathcal{N}}) - x^{\mathcal{N}}_{\text{CN}}(t^\ell; u^{\mathcal{N}})\| \leq 2\Delta t \sum_{k=0}^{\ell-1} \frac{\gamma_{\text{E}}^k}{\alpha_{\text{I}}^{k+1}} \|r^{\text{pr}}(t^{\ell-k}; x^{\mathcal{N}}_{\text{CN}}, u^{\mathcal{N}})\| =: \Delta^{\text{Std}}(u^{\mathcal{N}}), \tag{13}$$

where $\alpha_I := 1 + \frac{\Delta t}{2}\alpha_A$, $\gamma_E := 1 + \frac{\Delta t}{2}\gamma_A$ with α_A, γ_A given in Proposition 2.1 and the residual $r^{\text{pr}}(t; x^{\mathcal{N}}, u^{\mathcal{N}}) := Bu^{\mathcal{N}}(t) - \dot{x}^{\mathcal{N}}(t) - Ax^{\mathcal{N}}(t)$.

Supremizers and a Linear System Alternatively, given some choice for $X^{\mathcal{N}}$, we choose the test space in such a way that the inf-sup-constant $\beta^{\mathcal{N}}$ in (7) is maximized. This is typically done by using so called *supremizers*, [4], which reads here

$$z_{x^{\mathcal{N}}} = A^{-1}\dot{x}^{\mathcal{N}} + x^{\mathcal{N}}. \tag{14}$$

Let $\Xi^{\mathcal{N}} := \{\xi_1^{\mathcal{N}}, \ldots, \xi_{\mathcal{N}}^{\mathcal{N}}\}$, $X^{\mathcal{N}} = \text{span}(\Xi^{\mathcal{N}})$, then we set $\Theta^{\mathcal{N}} := \{\theta_1^{\mathcal{N}}, \ldots, \theta_{\mathcal{N}}^{\mathcal{N}}\}$, $\theta_i^{\mathcal{N}} := z_{\xi_i}$ and $Z^{\mathcal{N}}_{\text{sup}} := \text{span}(\Theta^{\mathcal{N}})$. We obtain a linear system for (8)

$$\mathcal{B}^{\mathcal{N}}_{\text{sup}} \boldsymbol{x}^{\mathcal{N}}_{\text{sup}}(u) = \boldsymbol{f}^{\mathcal{N}}_{\text{sup}}(u), \tag{15}$$

where the (symmetric) stiffness matrix has the entries $[\mathcal{B}^{\mathcal{N}}_{\text{sup}}]_{i,j} = b(\xi_i^{\mathcal{N}}, \theta_j^{\mathcal{N}}) = (\dot{\xi}_i^{\mathcal{N}} + A\xi_i^{\mathcal{N}}, A^{-1}\dot{\xi}_j^{\mathcal{N}} + \xi_j^{\mathcal{N}})_{L_2(I)^n} = (A\theta_i^{\mathcal{N}}, \theta_j^{\mathcal{N}})_{L_2(I)^n}$, $i, j = 1, \ldots, \mathcal{N}$, and the right-hand side reads $(\boldsymbol{f}^{\mathcal{N}}_{\text{sup}}(u))_i := (Bu, A^{-1}\dot{\xi}_i^{\mathcal{N}} + \xi_i^{\mathcal{N}})_{L_2(I)^n}$, $i, j = 1, \ldots, \mathcal{N}$. For the specific choice of the CN-trial functions $\xi_i^{\mathcal{N}} = \sigma^k \otimes e_\nu$, $i = (k, \nu)$, $k = 1, \ldots, K$, $\nu = 1, \ldots, n$, $\mathcal{N} = Kn$, we obtain $\mathcal{B}^{\mathcal{N}}_{\text{sup}} = (\bar{K}_{\Delta t} \otimes A^{-1}) + (\bar{L}_{\Delta t} \otimes A) + ((\bar{O}_{\Delta t} + \bar{O}^T_{\Delta t}) \otimes Id)$.

RB-Type Residual Error Estimate This Pertov-Galerkin formulation allows us to use a result in [7, Prop. 2.9] to derive an 'RB-type residual' error estimator to be described now. For the trial space $X^{\mathcal{N}}$ we will consider as in [7] a discrete norm $|||\cdot|||_{X,\Delta t}$. To define it, we set $\bar{\boldsymbol{x}}_k^{\mathcal{N}} := \frac{1}{\Delta t}\int_{I^k} x^{\mathcal{N}}(s)\,ds$ and $\bar{x}^{\mathcal{N}}(t) := \sum_{k=1}^{K} \bar{\boldsymbol{x}}_k^{\mathcal{N}} \tau^k(t)$, $t \in I$. Then, we set $|||x^{\mathcal{N}}|||^2_{X,\Delta t} := \|\dot{x}^{\mathcal{N}}\|^2_{A^{-1}} + \|\bar{x}^{\mathcal{N}}\|^2_{A} + \|x^{\mathcal{N}}(T)\|^2$. With these settings, it was proven in [7, Prop. 2.9] that

$$\beta^{\mathcal{N}}_{\sup} := \inf_{x^{\mathcal{N}} \in X^{\mathcal{N}}} \sup_{z^{\mathcal{N}} \in Z^{\mathcal{N}}_{\sup}} \frac{b(x^{\mathcal{N}}, z^{\mathcal{N}})}{|||z^{\mathcal{N}}|||_Z \, |||x^{\mathcal{N}}|||_{X,\Delta t}} = 1.$$

Let us stress that $\beta^{\mathcal{N}}_{\sup}$ is *independent of the control (parameter)* u and of T, Δt. Thus, for any approximation $x_N(u) \in X^{\mathcal{N}}$ (e.g., the RB approximation below), we get

$$|||x^{\mathcal{N}}_{\sup}(u) - x_N(u)|||_{X,\Delta t} \le \Delta^{\text{pr}}_N(u) := \|Bu - \dot{x}_N(u) - Ax_N(u)\|_{A^{-1}}. \tag{16}$$

We may use a discretized control $u^{\mathcal{N}}$ or any u allowing to compute $\boldsymbol{f}^{\mathcal{N}}_{\sup}(u)$, e.g. $\boldsymbol{f}^{\mathcal{N}}_{\sup}(u^{\mathcal{N}}) = [(\hat{O}_{\Delta t} \otimes (A^{-1}B)) + (\hat{L}_{\Delta t} \otimes B)]\boldsymbol{u}_{\Delta t}$ for $\xi^{\mathcal{N}}_i = \sigma^k \otimes e_\nu$ as above.

4 Reduced Basis Method (RBM)

Now, we employ the RBM to the above introduced variational formulation of an LTI. As mentioned already earlier, we view the control u as a parameter, i.e., (1) is seen as a parametric linear system. Doing so, we can reduce both the dimension n of the LTI system and the number K of time steps by reducing $\mathcal{N} := Kn$ to some $N \ll \mathcal{N}$.

RBM for Petrov-Galerkin Problems The starting point is the detailed discretization (8) of (4). Within a multi-query context, one would need to solve (8) for many different controls $u \in U$ and in a realtime scenario, a good approximation to $x^{\mathcal{N}}(u)$ would be needed extremely fast. This is precisely the situation one is facing within parameterized partial differential equations, where the RBM has proven to be a very useful tool for model reduction (at least in the elliptic and parabolic case).

We thus interpret (4) as a semi-discretized parabolic problem and follow [6, 7] to construct a RBM for the arising non-symmetric space-time-like problem. In order to do so, one looks for subspaces $X_N \subset X^{\mathcal{N}}$ and $Z_N \subset Z^{\mathcal{N}}$ of dimension $\dim(X_N) = \dim(Z_N) = N \ll \mathcal{N} = Kn$ and some $B_N \in \mathbb{R}^{N \times p}$ such that

$$\text{find } x_N \equiv x_N(u) \in X_N : \; b(x_N, z_N) = f_N(z_N; u) := (B_N u, z_N)_{L_2(I)^n} \quad \forall z_N \in Z_N \tag{17}$$

and in such a way that x_N can be computed *online efficient*, i.e., with a complexity independent of $\mathcal{N}$. Let us assume that we have (possibly orthonormal) bases $\{\xi^{(i)} : i = 1, \dots, N\}$ and $\{z^{(j)} : j = 1, \dots, N\}$ for X_N and Z_N, respectively, at hand. Then, (17) amounts solving a linear system $\mathcal{B}^T_N \boldsymbol{x}_N(u) = \boldsymbol{f}_N(u)$ of dimension N, where $\boldsymbol{f}_N(u)$ (and hence the coefficient vector $\boldsymbol{x}_N(u)$) depend on the control u and we obtain a parameter-dependent solution $\boldsymbol{x}_N(u)$. Moreover,

$[\mathcal{B}_N]_{i,j} = b(\xi^{(i)}, z^{(j)})$ and $[\boldsymbol{f}_N(u)]_j = (B_N u, z^{(j)})_{L_2(I)^n}$. Of course, the reduced system depends on the choice of the detailed Petrov-Galerkin detailed discretization. Let $P_N : X^{\mathcal{N}} \to X_N$ and $Q_N : Z^{\mathcal{N}} \to Z_N$ denote projections onto the reduced spaces and let $\boldsymbol{P}_N, \boldsymbol{Q}_N : \mathbb{R}^{\mathcal{N}} \to \mathbb{R}^N$ denote the matrix representations w.r.t. the above bases, we get $\mathcal{B}^T_{N,\text{disc}} = \boldsymbol{Q}_N (\mathcal{B}^{\mathcal{N}}_{\text{disc}})^T \boldsymbol{P}^T_N$ and $\boldsymbol{f}_{N,\text{disc}}(u) = \boldsymbol{Q}_N \boldsymbol{f}^{\mathcal{N}}_{\text{disc}}(u)$ for disc $\in$ {sup, CN}. Given some RB basis functions $\xi^{(1)}, \dots, \xi^{(N)}$ in $X^{\mathcal{N}}_{\text{CN}}$ determined as $\xi^{(i)} := x^{\mathcal{N}}_{\text{CN}}(u^{(i)})$ by (12) (the selection of the 'snapshots' $u^{(i)}$ will be detailed below) and the supremizers $z^{(1)}, \dots, z^{(N)}$ by (14), the system matrix of the reduced problem reads $\mathcal{B}_{N,\text{sup}} = (\Xi^{\mathcal{N}})^T \mathcal{B}^{\mathcal{N}}_{\text{sup}} \Xi^{\mathcal{N}}$ (recall (11)), where $\Xi^{\mathcal{N}} := (\boldsymbol{\xi}^{(i)}_{\Delta t})_{i=1,\dots,N}$. Note, that $\mathcal{B}_{N,\text{sup}}$ is symmetric and independent of the parameter, i.e., the control. We can thus pre-compute and store a LU- or QR-decomposition, which reduces the online amount of work to solve the linear system to $\mathcal{O}(N^2)$. The right-hand side is parameter-dependent and reads $\boldsymbol{f}_{N,\text{sup}}(u^{\mathcal{N}}) = (\Xi^{\mathcal{N}})^T \boldsymbol{f}^{\mathcal{N}}_{\text{sup}}(u^{\mathcal{N}})$ for some $u^{\mathcal{N}} \in U^{\mathcal{N}}$.

Reduced Basis Generation We use a greedy procedure to compute a Reduced Basis, indicated in Algorithm 1 and which is based upon some error estimator Δ^{pr}_N. After execution of this scheme, we obtain a reduced space $X_N \equiv X^{\mathcal{N}}_N := \text{span}\{x^{\mathcal{N}}_{\text{CN}}(u^{(1)}), \dots, x^{\mathcal{N}}_{\text{CN}}(u^{(N)})\}$ as well as a reduced test space $Z_N \equiv Z^{\mathcal{N}}_{N,\text{sup}} := \text{span}\{z^{\mathcal{N}}(u^{(1)}), \dots, z^{\mathcal{N}}(u^{(N)})\}$ and also a reduced control space U_N. The general procedure is indicated by Algorithm 1, which is based upon the choice of a training parameter space $U_{\text{train}} \subset U^{\mathcal{N}}$. Note, that the state snapshots are computed by using the CN-time marching scheme (12) and the reduced system is then generated by the supremizers in (14), see line 2 in Algorithm 1.

Computation of the RB Error Bound We can further detail the residual-based error estimate from (16) applied to the reduced problem, i.e.,

$$|||x^{\mathcal{N}}_{\text{sup}}(u) - x_{N,\text{sup}}(u)|||_{X,\Delta t} \leq \Delta^{\text{pr}}_N(u) := \|Bu - \dot{x}_{N,\text{sup}}(u) - Ax_{N,\text{sup}}(u)\|_{A^{-1}}. \tag{18}$$

First, we have $\Delta^{\text{pr}}(u)^2 = \|Bu\|^2_{A^{-1}} - 2\boldsymbol{f}_{N,\text{sup}}(u)^T \boldsymbol{x}_N(u) + \boldsymbol{x}_N(u)^T \mathcal{B}_{N,\text{sup}} \boldsymbol{x}_N(u)$ for $\boldsymbol{x}_N \equiv \boldsymbol{x}_{N,\text{sup}}$. Obviously, the last two terms can easily and efficiently be evaluated.

Algorithm 1 (Primal) Greedy algorithm with CN-snapshots and RB-supremizers

```
1: Choose $U_{\text{train}} \subset U^{\mathcal{N}}$, tol, $\eta^{(1)} := u^{(1)}$; set $N := 1$
2: Compute $\xi^{(N)} := x^{\mathcal{N}}_{\text{CN}}(\eta^{(N)})$, $z^{(N)} := z^{\mathcal{N}}(\eta^{(N)})$  ▷ detailed solution (12) and supremizer (14)
3: set $X_N := \text{span}\{\xi^{(1)}, \dots, \xi^{(N)}\}$, $Z_N := \text{span}\{z^{(1)}, \dots, z^{(N)}\}$, orthonormalize bases
4: set $U_N := \text{span}\{\eta^{(1)}, \dots, \eta^{(N)}\}$, orthonormalize
5: for $u \in U_{\text{train}}$ do
6:     Compute $x_N(u) \in X_N$                                ▷ RB approximation with $N$ d.o.f.
7:     Compute $\Delta^{\text{pr}}_N(u)$                            ▷ primal error estimator, e.g., (16)
8: end for
9: Set $\eta^{(N+1)} := \arg\max_{u \in U_{\text{train}}} \Delta^{\text{pr}}_N(u)$                 ▷ worst parameter
10: if $\Delta^{\text{pr}}_N(\eta^{(N+1)}) >$ tol set $N := N + 1$, goto 2 else break end if
```

Hence, we consider the first part, namely $\|Bu\|_{A^{-1}} = \|A^{-1/2}Bu\|_{L_2(I)^n}$. At this point, it is now crucial how a reduced discretization of the control u is or can be chosen:

- If the control comes from temporal measurements, it will most likely be in form of a *detailed* control, i.e., $u^{\mathcal{N}}$. Then, $\|Bu\|^2_{A^{-1}} = \boldsymbol{u}^T_{\Delta t}\big(L_{\Delta t} \otimes B^T A^{-1} B\big)\boldsymbol{u}_{\Delta t}$, which is *not fully* online efficient since the computational amount depends on K.
- If the control can be reduced a priori, e.g., in a multi-query context (think of optimal control), then one would have some u_N with N degrees of freedom so that $\|Bu_N\|_{A^{-1}}$ can be computed in $\mathcal{O}(N^2)$ operations independent of $\mathcal{N} = nK$.

5 Numerical Experiments

We report on some results of our numerical experiments for a standard example, where A arises from a Finite Element discretization of a 1d heat equation with Neumann boundary conditions on the left end and homogeneous Dirichlet boundary conditions on the right end as well as homogeneous initial conditions. The control matrix is $B := n\kappa(-1, 0, \dots, 0)^T \in \mathbb{R}^{n\times 1}$, $m = 1$ and $\kappa > 0$ is the conductivity. On the left-hand side of Fig. 1, we see the Greedy error sequence, i.e., the decay of Δ_N^{pr} over a training set of controls as $N \to \infty$. We observe a rate of about $10^{-0.1N}$. On the right-hand side, we increase the number K of time steps and observe that we can basically reach any desired accuracy. Moreover, we compare the exact error with the error estimator Δ_N^{pr} and obtain decreasing effectivities for increasing K. We stress that we measure the error in a quite strong norm $|\!|\!|\cdot|\!|\!|_{X,\Delta t}$, which is much stronger than what is usually used in model order reduction, namely $\|\cdot\|_{L_2(I)^n}$.

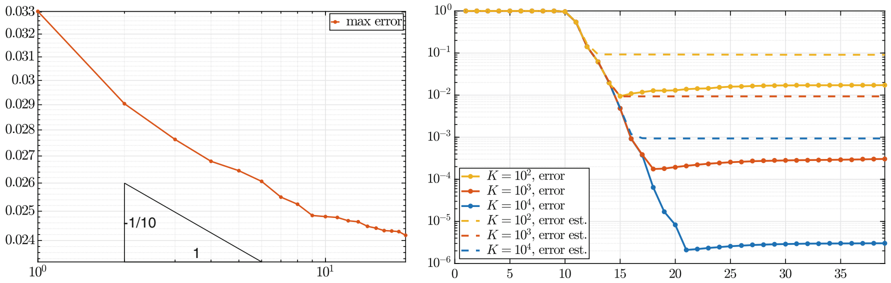


Fig. 1 Greedy error sequence (left), test error and error estimator for increasing K (right): relative error vs. N

6 Summary and Outlook

We have introduced a space-time-type RB model reduction for LTI systems which allows to reduce both the state dimension n and the number of time steps K. We obtain exponential decay w.r.t. the reduced dimension N and reasonable effectivities, in a quite strong norm, however. The next step is to extend this framework to the output using adjoint techniques. At that stage, quantitative comparisons with well-established techniques like balanced truncation, will be performed. This should result in a clear picture together with other comparisons of model order reduction and POD-Greedy [1] as well as POD-Greedy versus space-time RBM, see [3].

References

1. U. Baur, P. Benner, B. Haasdonk, C. Himpe, I. Martini, and M. Ohlberger. Comparison of methods for parametric model order reduction of time-dependent problems. In *Model reduction and approximation*, volume 15 of *Comput. Sci. Eng.*, pages 377–407. SIAM, 2017.
2. P. Benner, M. Ohlberger, A. Patera, G. Rozza, and K. Urban. *Model Reduction of Parametrized Systems*. Springer International Publishing, Cham, 2017.
3. S. Glas, A. Mayerhofer, and K. Urban. Two ways to treat time in reduced basis methods. In *Model reduction of parametrized systems*, volume 17 of *MS&A. Model. Simul. Appl.*, pages 1–16. Springer, Cham, 2017.
4. G. Rozza. Reduced basis methods for stokes equations in domains with non-affine parameter dependence. *Computing and Visualization in Science*, 12(1):23–35, Jan 2009.
5. C. Schwab and R. Stevenson. Space-time adaptive wavelet methods for parabolic evolution problems. *Mathematics of Computation*, 78(267):1293–1318, 2009.
6. K. Urban and A. T. Patera. A new error bound for reduced basis approximation of parabolic partial differential equations. *C. R. Math. Acad. Sci. Paris*, 350(3–4):203–207, 2012.
7. K. Urban and A. T. Patera. An improved error bound for reduced basis approximation of linear parabolic problems. *Mathematics of Computation*, 83(288):1599–1615, 2014.

Numerical Solution of Traffic Flow Models

Lukáš Vacek and Václav Kučera

Abstract We describe the simulation of traffic flows on networks. On individual roads we use standard macroscopic traffic models. The discontinuous Galerkin method in space and a multistep method in time is used for the numerical solution. We introduce limiters to keep the density in an admissible interval as well as prevent spurious oscillations in the numerical solution. To simulate traffic on networks, one should construct suitable numerical fluxes at junctions.

1 Macroscopic Traffic Flow Models

We consider traffic flow on networks, described by macroscopic models, cf. [1, 2]. Here the traffic flow is described by three fundamental quantities—*traffic flow* $Q(x, t)$ which determines the number of cars per second at the position x at time t; *traffic density* $\rho(x, t)$ determines the number of cars per meter at x and t; and the *mean traffic flow velocity* $V = Q/\rho$.

Greenshields described a relation between traffic density and traffic flow in [3]. He realised that traffic flow is a function depending only on traffic density in homogeneous traffic (traffic with no changes in time and space). This implies that even mean traffic flow velocity depends only on traffic density. The relationship between the traffic density and the mean traffic flow velocity or traffic flow is described by the *fundamental diagram*, cf. [3].

Since the number of cars is conserved, the basic governing equation is a first order hyperbolic partial differential equation, cf. [2]:

$$\frac{\partial}{\partial t}\rho(x,t) + \frac{\partial}{\partial x}\left(\rho(x,t)V(x,t)\right) = 0. \tag{1}$$

L. Vacek (✉) · V. Kučera
Charles University, Prague, Czech Republic
e-mail: lvacek@karlin.mff.cuni.cz; kucera@karlin.mff.cuni.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_106

Equation (1) must be supplemented by the initial condition

$$\rho(x,0)=\rho_0(x) \text{ and } V(x,0)=V_0(x), \qquad x\in\mathbb{R}$$

and the inflow boundary condition. We have only one equation for two unknowns. Thus, we need an equation for $V(x,t)$. One possibility is the *Lighthill-Whitham-Richards model* (abbreviated LWR) where we use the *equilibrium velocity* $V_e(\rho)$. There are many different proposals for the equilibrium velocity derived from the real traffic data, e.g. Greenshields model takes $V_e(\rho) = v_{\max}\left(1-\frac{\rho}{\rho_{\max}}\right)$, where $v_{\max}$ is the maximal velocity and $\rho_{\max}$ is the maximal density. Thus we get the following nonlinear first order hyperbolic equation for ρ:

$$\rho_t + (\rho V_e(\rho))_x = 0, \qquad x\in\mathbb{R},\ t>0. \tag{2}$$

Junctions

Following [4], we study a complex *network* represented by a directed graph. The graph is a finite collection of directed edges, connected together at vertices. Each vertex has a finite set of incoming and outgoing edges. It is sufficient to study our problem only at one vertex and on its adjacent edges.

On each road (edge) we consider the LWR model, while at junctions (vertices) we consider a *Riemann solver*. At each vertex J, there is a *traffic-distribution matrix* A describing the distribution of traffic among outgoing roads. Let J be a fixed vertex with n incoming and m outgoing edges. Then

$$A=\begin{bmatrix}\alpha_{n+1,1} & \cdots & \alpha_{n+1,n}\\ \vdots & \vdots & \vdots\\ \alpha_{n+m,1} & \cdots & \alpha_{n+m,n}\end{bmatrix}, \tag{3}$$

where for all $i\in\{1,\dots,n\}$, $j\in\{n+1,\dots,n+m\}$: $\alpha_{j,i}\in[0,1]$ and for all $i\in\{1,\dots,n\}$: $\sum_{j=n+1}^{n+m}\alpha_{j,i}=1$. The ith column of A describes how traffic from an incoming road I_i distributes to outgoing roads at the junction J. We denote the endpoints of road I_i as a_i, b_i, one of which coincides with J.

Let $\rho=(\rho_1,\dots,\rho_{n+m})^T$ be a *weak solution at the junction* J such that each $x\to\rho_i(x,t)$ has bounded variation. Then ρ satisfies the *Rankine-Hugoniot condition*, which represents the conservation of cars at the junction:

$$\sum_{i=1}^{n} Q_e(\rho_i(b_{i-},t)) = \sum_{j=n+1}^{n+m} Q_e(\rho_j(a_{j+},t))$$

for almost every $t>0$ at the junction J, where $\rho_j(a_{j+},t) := \lim_{(x\to a_j+)}\rho_j(x,t)$ and $\rho_i(b_{i-},t) := \lim_{(x\to b_i-)}\rho_i(x,t)$, cf. [4, Lemma 5.1.9, p. 98].

Finally, $\rho = (\rho_1, \dots, \rho_{n+m})^T$ is called an *admissible weak solution of (2)* related to the matrix A at the junction J if the following properties hold:

1. ρ is a weak solution at the junction J such that $\rho_i(\cdot, t)$ is of bounded variation for every $t \geq 0$, i.e. the Rankine-Hugoniot condition holds.
2. $Q_e(\rho_j(a_{j+}, \cdot)) = \sum_{i=1}^{n} \alpha_{j,i} Q_e(\rho_i(b_{i-}, \cdot)),\ \forall j = n+1, \dots, n+m.$
3. $\sum_{i=1}^{n} Q_e(\rho_i(b_{i-}, \cdot))$ is a maximum subject to (1) and (2).

2 Discontinuous Galerkin Method

As an appropriate method for the numerical solution of (2), we choose the *discontinuous Galerkin* (DG) method, which is essentially a combination of finite volume and finite element techniques, cf. [5]. We consider a 1D domain $\Omega = (a, b)$. Let $\mathcal{T}_h$ be a partition of $\overline{\Omega}$ into a finite number of closed intervals (elements) $[a_K,\ b_K]$. We denote the set of all boundary points of all elements by $\mathcal{F}_h$. Let $p \geq 0$ be an integer. We seek the numerical solution in the space of discontinuous piecewise polynomial functions

$$S_h = \{v;\ v|_K \in P^p(K),\ \forall K \in \mathcal{T}_h\},$$

where $P^p(K)$ denotes the space of all polynomials on K of degree at most p.

For each inner point $x \in \mathcal{F}_h$ there exist two neighbours $K_x^{(L)},\ K_x^{(R)} \in \mathcal{T}_h$ such that $x = K_x^{(L)} \cap K_x^{(R)}$. For a function $v \in S_h$ we use the notation: $v^{(L)}(x) = \lim_{y \to x_-} v(y)$, $v^{(R)}(x) = \lim_{y \to x_+} v(y)$ and $[v]_x = v^{(L)}(x) - v^{(R)}(x)$.

We formulate the DG method for the general first order hyperbolic problem

$$\begin{aligned} u_t + f(u)_x &= g, \qquad x \in \Omega,\ t \in (0, T), \\ u &= u_D, \qquad x \in \mathcal{F}_h^D,\ t \in (0, T), \\ u(x, 0) &= u_0(x), \qquad x \in \Omega, \end{aligned}$$

where g, u_D and u_0 are given functions and u is our unknown. The Dirichlet boundary condition is prescribed only on the inlet $\mathcal{F}_h^D \subseteq \{a, b\}$, respecting the direction of information propagation (characteristics).

The DG formulation then reads, cf. [5]: Find $u_h : [0, T] \to S_h$ such that

$$\int_\Omega (u_h)_t \varphi \,\mathrm{d}x - \sum_{K \in \mathcal{T}_h} \int_K f(u_h)\varphi' \,\mathrm{d}x + \sum_{x \in \mathcal{F}_h} H(u_h^{(L)}, u_h^{(R)})\, [\varphi]_x = \int_\Omega g\varphi \,\mathrm{d}x,$$

for all $\varphi \in S_h$. In boundary terms on $\mathcal{F}_h$ we use the approximation $f(u_h) \approx H(u_h^{(L)}, u_h^{(R)})$, where H is a *numerical flux*. We use the *Lax-Friedrichs flux*, cf. [5]: We define $\alpha = \max_{u \in (u_h^{(L)}, u_h^{(R)})} \left|f'(u)\right|$. In practice, we approximate the maximum by calculating $\left|f'(u)\right|$ at the points $u_h^{(L)}$, $u_h^{(R)}$ and $\frac{1}{2}(u_h^{(L)} + u_h^{(R)})$ and we take the maximal value. Then we calculate the numerical flux as

$$H(u_h^{(L)}, u_h^{(R)}) = \tfrac{1}{2}(f(u_h^{(L)}) + f(u_h^{(R)}) - \alpha(u_h^{(R)} - u_h^{(L)})).$$

3 Implementation

For time discretization of the DG method we use *Adams–Bashforth methods*, which are explicit linear multistep methods for ODEs. As a basis for S_h, we use *Legendre polynomials* and we use *Gauss–Legendre quadrature* to evaluate integrals over elements. The implementation is in the C++ language.

Because we calculate physical quantities (density and velocity), we know that the result must be in some interval, e.g. $[0, \rho_{\max}]$. Thus, we use *limiters* in each time step to obtain the solution in the admissible interval. It is important to not change the total number of cars. For a piecewise linear approximation of ρ in LWR models, we find each element K for which there exists $x \in [a_K, b_K]$ such that $\rho(x) \notin [\rho_{\min}, \rho_{\max}]$. If the average density on element K is in admissible interval, we decrease the slope of our solution so that the modified density lies in $[\rho_{\min}, \rho_{\max}]$. If the average density on element K is not in the admissible interval $[\rho_{\min}, \rho_{\max}]$ we decrease the time step. Following [6], we also apply limiting to treat spurious oscillations near discontinuities and sharp gradients in the numerical solution.

Numerical Fluxes at Junctions

Since we wish to model traffic on networks, the numerical fluxes at junctions must be specified. The basic requirement is that the number of cars at the junctions must be conserved. Moreover, we wish to prescribe the traffic distribution according to the traffic-distribution matrix (3). The number of cars which inflow or outflow through the junction is given by the traffic flow Q_e. More precisely, the traffic flow from incoming road I_i, $i = 1, \ldots, n$, at time t is given by $Q_e\left(\rho_i(b_{i-}, t)\right)$. Due to the traffic-distribution matrix, we know the ratio of the traffic flow distribution between the outgoing roads. Thus, the traffic flow to the outgoing road I_j, $j = n + 1, \ldots, n + m$, at time t is given by $Q_e\left(\rho_j(a_{j+}, t)\right) = \sum_{i=1}^{n} \alpha_{j,i} Q_e\left(\rho_i(b_{i-}, t)\right)$. Since the traffic flow at the boundary of an element is represented by the numerical flux, we take the numerical flux $H_j(t)$ at the left point of the outgoing road I_j, i.e. point at the junction, at time t as

$$H_j(t) := \sum_{i=1}^{n} \alpha_{j,i} H(\rho_{hi}(b_{i-}, t), \rho_{hj}(a_{j+}, t)),$$

for $j = n + 1, \ldots, n + m$, where ρ_{hi} is the DG solution on the ith road. The numerical flux $H_j(t)$ approximates the traffic flow $Q_e\left(\rho_j(a_{j+}, t)\right)$. Similarly, we take the numerical flux $H_i(t)$ at time t at the right point of the incoming road I_i, i.e. at the junction point, as

$$H_i(t) := \sum_{j=n+1}^{n+m} \alpha_{j,i} H(\rho_{hi}(b_{i-}, t), \rho_{hj}(a_{j+}, t)),$$

where $i = 1, \ldots, n$. Then $H_i(t)$ approximates the traffic flow $Q_e\,(\rho_i(b_{i-}, t))$.

It can be shown, that our choice of numerical fluxes conserves the number of cars at junctions. However, this choice does not distribute the traffic according to the traffic-distribution matrix (3) exactly, only approximately. We interpret this phenomenon as follows and compare to the boundary conditions from [4, 7].

A method how to obtain an admissible solution satisfying properties (1)–(3) is described in [4] or [7]. As an example, we take a junction with one incoming and two outgoing roads. In [4, 7], maximum possible fluxes are used. If there is a traffic jam in one of the outgoing roads, the maximum possible flow through the junction is 0. On the other hand, the cars in our approach can still go into the second outgoing road according to the traffic-distribution coefficients. So our choice of numerical fluxes corresponds to modelling turning lanes, which allow the cars to separate before the junction according to their preferred turning direction. In our case the junction is not blocked due to a traffic jam on one of the outgoing roads. Since the macroscopic models are aimed for long (multi-lane) roads with huge number of cars, our model makes sense in this situation. The original approach from [4, 7] is aimed for one-lane roads, where splitting of the traffic according to preference is not possible.

Another difference is that we can use all varieties of traffic lights. The model of [4, 7] can use only the full green lights. Our approach gives us an opportunity to change the lights for each direction separately.

An artefact of our model is that we do not satisfy the traffic-distribution coefficients exactly. This corresponds to the real situation where some cars decide to use another road instead of staying in the traffic jam. The problem is when there is no traffic jam. Since we do not control the traffic-distribution exactly, we do not satisfy it exactly. For this reason we interpret the matrix A as a *traffic-probability matrix*. Now the element $\alpha_{j,i}$ is the probability that the cars want to go from the incoming road I_i to outgoing road I_j.

4 Numerical Results

In this section we present our program and numerical results. We show the result of calculation on a bottleneck and on a simple network. As we mention above, we use the combination of Adams–Bashforth and DG methods. This compares to the approach in [7] where the authors use the *Runge-Kutta method* for time discretization. Piecewise linear approximations of solutions with two Gaussian quadrature points in each element were used.

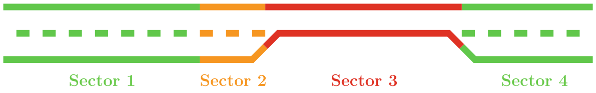


Fig. 1 Test road with a bottleneck in Sector 3 (red)

Bottleneck

First we demonstrate results for a single road with a bottleneck, cf. Fig. 1. In sector 1 and 4 we have maximal velocity $v_{\max,1} = 1.3$ and maximal density $\rho_{\max,1} = 2$, which corresponds to two lanes. The length of the first sector is $L_1 = 2$ and the length of the fourth sector is $L_4 = 1$. Sector 2 is a short sector with length $L_2 = 0.5$ and with decreased maximal velocity $v_{\max,2} = 1$ and maximal density $\rho_{\max,2} = 2$. Sector 3 is the bottleneck, where the maximal density is $\rho_{\max,3} = 1$, which corresponds to one lane. The maximal velocity is $v_{\max,3} = 0.8$ and the length of this sector is $L_3 = 2$.

The cars go from left to right. The boundary condition on the left is $\rho(0,t) = \frac{1}{20}\sin\left(\frac{2\pi t}{7} - \frac{\pi}{2}\right) + 0.18$ to simulate time-varying traffic. The initial condition is an empty road. We use Greenshields model. The time-step size is $\tau = 10^{-4}$ in the Euler method and the length of each element is $h = \frac{1}{150}$.

In Fig. 2 we can observe the emergence of a traffic jam between Sector 2 and Sector 3. The traffic jam spreads backwards to Sector 1 and becomes longer or shorter depending on the boundary influx. Because $\rho(x,t) < \rho_{\max,i}$ for all x, t and all sectors, the cars in the traffic jams are still moving.

Simple Network

Now we demonstrate how our program computes traffic on networks. Thus, we define the simple network from Fig. 3. This network is closed, so we can show the conservation of the number of cars. We have three roads and two junctions. The length of all roads is 1. At the first junction we have one incoming road and two outgoing roads. At the second junction we have the opposite situation. We use a different distribution of cars at the first junction: $\frac{3}{4}$ go from the first road to the second and $\frac{1}{4}$ from the first road to the third. This corresponds to the distribution matrices $A_1 = [0.75, 0.25]^T$ and $A_1 = [1, 1]$. The initial conditions on each road is depicted in Fig. 4a. On the first road there is a piecewise linear 'bump' in density, while the second and third roads have a constant density of 0.4. The total number of cars in the whole network is 1. We use Greenshields model on all roads. We use the Euler method with the step size $\tau = 10^{-4}$ and the number of elements is $N = 100$ on each road.

We can see the results in Fig. 4. Road 1 distributes the traffic density between the other roads. We have too many cars at the second junction, where we have two incoming roads. Thus, we create a traffic jam on Road 2 and Road 3. We can observe the transporting and the distribution of the jump from the first road through the

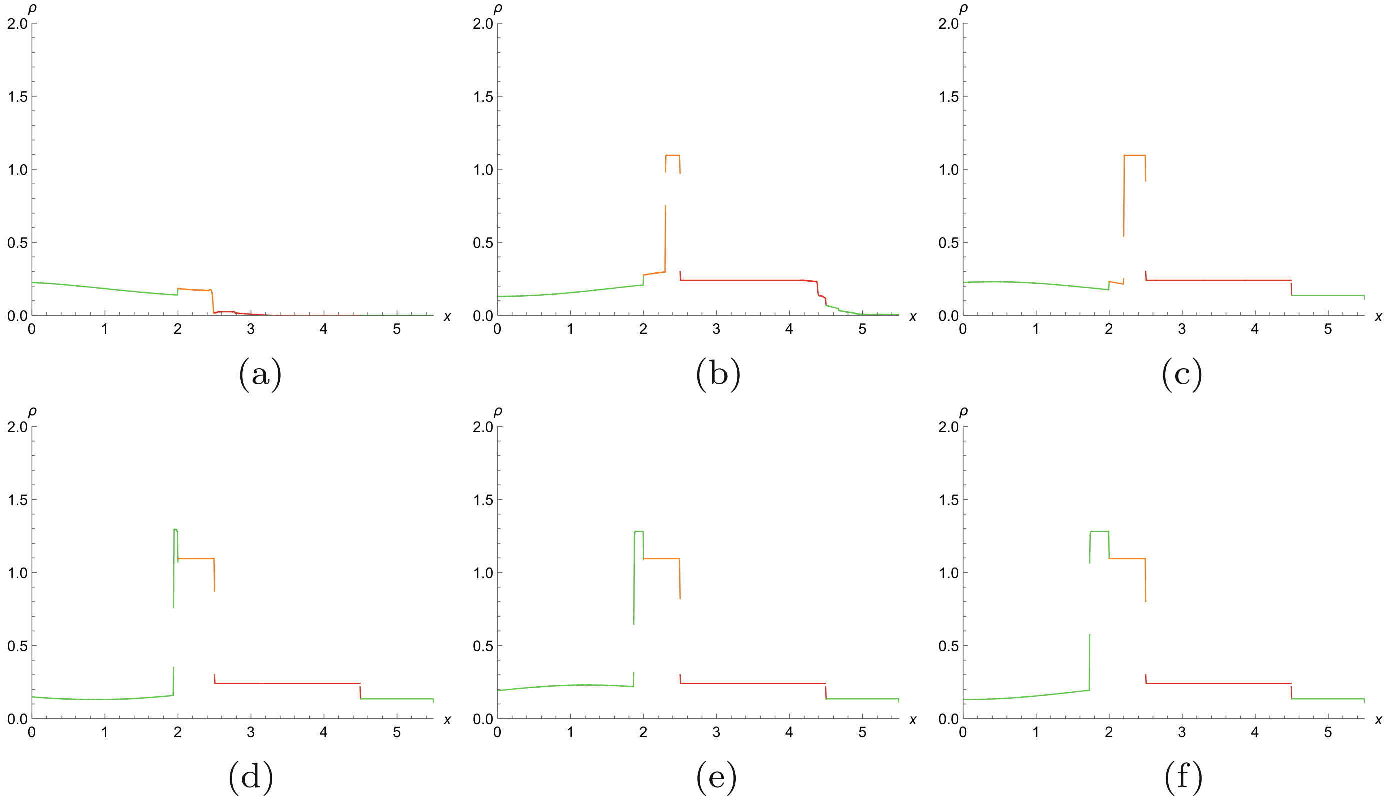


Fig. 2 Bottleneck—density on Sector 1 (green), Sector 2 (orange), Sector 3 (red) and Sector 4 (green). (**a**) $t = 3$, (**b**) $t = 7$, (**c**) $t = 11$, (**d**) $t = 15$, (**e**) $t = 19$, (**f**) $t = 21$

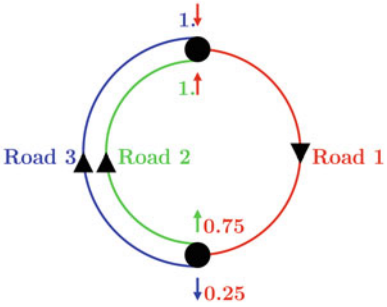


Fig. 3 Test network with Road 1 (red), Road 2 (green) and Road 3 (blue)

junction on Fig. 4d and e. The result converges to the stationary solution. The traffic density in Fig. 4f is close to a stationary solution. The amount of cars is conserved.

Our program can compute traffic on bigger networks and we are not limited by the number of incoming or outgoing roads at junctions. We can have time-dependent traffic lights at junctions. However, this contribution is too short for demonstrating these results.

5 Conclusion

We have demonstrated the numerical solution of macroscopic traffic flow models using the discontinuous Galerkin method. For the approximation in time we choose explicit multistep methods. For traffic networks, we construct special numerical fluxes at the junctions. The use of DG methods on networks is not standard. We compare our approach with the paper [7] by Čanić, Piccoli, Qiu and Ren, where Runge-Kutta methods are used along with a different choice of numerical fluxes at junctions.

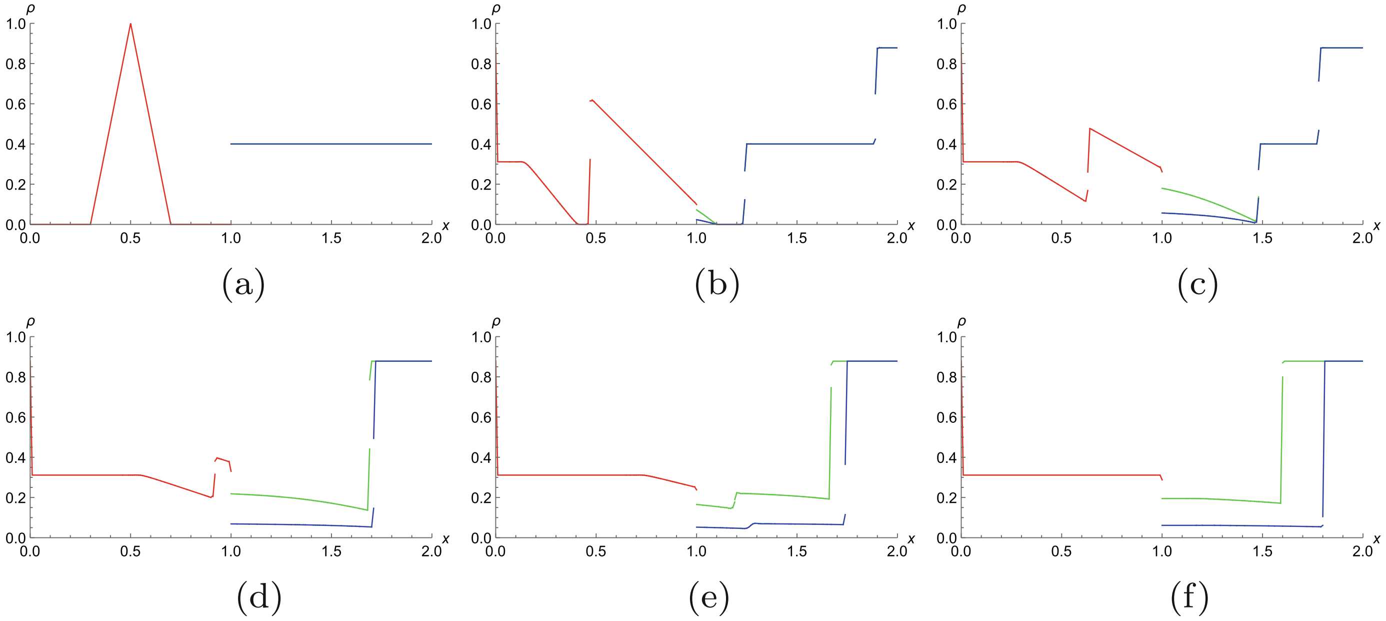


Fig. 4 Traffic density on network from Fig. 3—Road 1 (red), Road 2 (green) and Road 3 (blue). (**a**) $t = 0$, (**b**) $t = 0.4$, (**c**) $t = 0.8$, (**d**) $t = 1.5$, (**e**) $t = 2$, (**f**) $t = 3$

Acknowledgments The research of L. Vacek is supported by the Charles University, project GA UK No. 20-01074S. The work of V. Kučera was supported by the research project No. 20-01747S of the Czech Science Foundation.

References

1. F. van Wageningen-Kessels, H. van Lint, K. Vuik, S. Hoogendoorn: Genealogy of traffic flow models. EURO Journal on Transportation and Logistics, **4**, 445–473 (2015)
2. P. Kachroo, S. Sastry: Traffic Flow Theory: Mathematical Framework. In: University of California Berkeley https://www.scribd.com/doc/316334815/Traffic-Flow-Theory Cited 2 Dec 2019
3. B. D. Greenshields: A Study of Traffic Capacity. Highway Research Board, **14**, 448–477 (1935)
4. M. Garavello, B. Piccoli: Traffic flow on networks. AIMS Series on Applied Mathematics, **1**, 1–243 (2006)
5. V. Dolejší, M. Feistauer: Discontinuous Galerkin Method - Analysis and Applications to Compressible Flow. Analysis and Applications to Compressible Flow, **48** (2015)
6. C.-W. Shu: Discontinuous Galerkin methods: general approach and stability. Numerical solutions of partial differential equations, **201** (2009)
7. S. Čanić, B. Piccoli, J. Qiu, T. Ren: Runge-Kutta Discontinuous Galerkin Method for Traffic Flow Model on Networks. Journal of Scientific Computing, **63** (2014)

Numerical Approximation of Fluid-Structure Interaction Problem in a Closing Channel Near the Stability Boundary

Jan Valášek, Petr Sváček, and Jaromír Horáček

Abstract This contribution deals with the numerical simulation of a fluid-structure interaction problem. The elastic body is modelled with the aid of a linear elasticity model. The fluid flow is described by the incompressible Navier-Stokes equations in the arbitrary Lagrangian-Eulerian formulation. The coupling conditions are specified and the coupled problem is formulated. The fluid-structure interaction problem is discretized by the finite element method solver applied both to the elastic part as well as to the fluid flow approximation. For the fluid flow approximation the residual based stabilization is used. Special attention is paid to the penalization boundary condition used at the inlet. It allows to relax an exact value of the inlet velocity on the boundary during channel closing phase nearly to complete channel closure. Numerical results for flow-induced vibrations near the stability boundary are presented and the critical velocity of flutter instability is determined.

1 Introduction

The fluid-structure interaction (FSI) problem appears in many technical applications like airfoil stability or biomechanical applications as hemodynamics or vocal folds vibrations, see e.g. [2, 4]. In technical applications the main interest is usually paid to investigation of aeroelastic/hydrodynamic stability of the system. The stability of FSI system is lost usually if the inlet flow velocity exceeds a critical value which is called the flutter velocity, see [2]. The appearance of instability is usually an undesired phenomenon endangering the structure integrity, however it can be also

J. Valášek (✉) · P. Sváček
Department of Technical Mathematics, Center of Advanced Aerospace Technology, Faculty of Mechanical Engineering, Czech Technical University in Prague, Praha, Czech Republic
e-mail: Jan.Valasek1@fs.cvut.cz; Petr.Svacek@fs.cvut.cz

J. Horáček
Institute of Thermomechanics, Czech Academy of Sciences, Praha, Czech Republic
e-mail: Jaromirh@it.cas.cz

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_107

a desired process as in the case of vocal folds vibrations, see [8]. For investigation of aeroelastic stability region usually a simplified linearized theory is used, see [2]. On the other hand such a simplification can not be used for large displacement as gust response or vibrations of control valve. In such a case the solution of full FSI problem should be used.

Such a full FSI problem can be mathematically modelled by a linear elasticity model for the motion of the structure and the fluid flow problem can be modelled by the incompressible Navier-Stokes equations. The arbitrary Lagrangian-Eulerian (ALE) method utilized for the purpose of incorporating the influence of the fluid domain changes due to structure motion. Here, the special attention is paid to the boundary conditions (BC) of fluid problem. At the outlet so called a directional do-nothing BC is used due to stability reasons, see [1], while at the inlet the recently proposed penalization BC is prescribed, see [9]. In this case the inlet velocity is weakly enforced with the help of the penalization parameter. The behaviour of FSI system with this condition is very similar to the behaviour for classical Dirichlet BC in terms of aeroelastic stability, but it differs significantly during the channel closing phase. In that case the unphysically high velocities (the case with Dirichlet BC) is relaxed to reasonable values in dependence of chosen penalization parameter, see e.g. [10]. The aim of this contribution is to study the dependence of flutter velocity on the penalization parameter.

The numerical approximation of both FSI subproblems is based on the finite element method (FEM) with advanced stabilization technique employed, and the strongly coupled partitioned scheme is implemented, see e.g. [4]. Finally the numerical results of flow-induced vibrations are presented and the flutter velocity is determined.

2 Mathematical Model

For the sake of simplicity let us consider the two-dimensional FSI problem, composed of the elastic body and fluid flow represented by domains Ω^s and Ω^f, respectively, see Fig. 1. For the description of elastic body motion the Lagrangian coordinates with notation of coordinates X_i are used, i.e. the computational domain Ω^s does not depend on time and it holds $\Omega^s := \Omega^s_{\text{ref}}$. On the other hand, we distinguish the reference fluid domain Ω^f_{ref} (the domain occupied by fluid at the time instant $t = 0$) with the reference interface $\Gamma_{W_{\text{ref}}} = \Gamma_{W_0}$ and the deformed fluid domain Ω^f_t with the interface Γ_{W_t} at any time instant t. In order to describe the fluid flow on the time dependent domain the ALE method is used, see e.g. [6]. In ALE framework the coordinates are denoted as x_i.

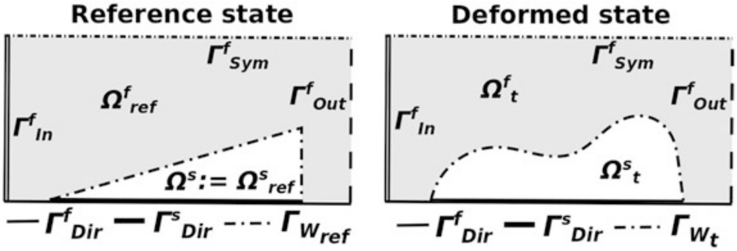

Reference state
Deformed state
Γ^f_{Sym}
Ω^f_{ref}
Γ^f_{Out}
Γ^f_{In}
$\Omega^s := \Omega^s_{ref}$
Ω^f_t
Ω^s_t
Γ^f_{Dir} Γ^s_{Dir} $\Gamma_{W_{ref}}$
Γ^f_{Dir} Γ^s_{Dir} Γ_{W_t}

Fig. 1 Scheme of FSI problem configuration. The domain Ω^s denoted the elastic body and Ω^f is the fluid domain. The FSI domain is shown in reference state on the left and in arbitrary time t undergoing a deformation on the right. The following boundaries are considered: inlet Γ^f_{In}, outlet Γ^f_{Out}, walls Γ^f_{Dir}, Γ^s_{Dir}, boundary of symmetry Γ^f_{Sym} and interface $\Gamma_{\text{W}_\text{t}}$

2.1 Elastic Body

The motion of the elastic body is described by the partial differential equations

$$\rho^s \frac{\partial^2 u_i}{\partial t^2} - \frac{\partial \tau^s_{ij}}{\partial X_j} = f^s_i, \qquad \text{in } \Omega^s \times (0, \text{T}) \tag{1}$$

where ρ^s is the structure density, $\mathbf{u}(X, t) = (u_1, u_2)$ denotes the unknown displacement, the vector of volume forces is $\mathbf{f}^s = (f_1, f_2)$ and τ^s_{ij} are the components of the Cauchy stress tensor. These components are expressed with the aid of the Hooke's law further assuming isotropic body and small displacements as

$$\tau^s_{ij} = \lambda^s \frac{\partial u_k}{\partial X_k}\delta_{ij} + 2\mu^s e^s_{ij}(\mathbf{u}), \tag{2}$$

where δ_{ij} denotes the Kronecker's delta, $e^s_{ij}(\mathbf{u}) = \frac{1}{2}\left(\frac{\partial u_j}{\partial X_i} + \frac{\partial u_i}{\partial X_j}\right)$ is the small strain tensor and λ^s, μ^s are the Lame's constants, see e.g. [7]. The elastic problem (1) is equipped with the zero initial conditions and the following boundary conditions are considered

$$\begin{aligned} &\text{a)} \qquad \mathbf{u}(X, t) = \mathbf{u}_{\text{Dir}}(X, t) \ \text{ for } X \in \Gamma^s_{\text{Dir}}, \\ &\text{b)} \qquad \tau^s_{ij}(X, t)\, n^s_j(X) = q^s_i(X, t), \quad \text{for } X \in \Gamma^s_{\text{W}_{\text{ref}}}, \end{aligned} \tag{3}$$

where the boundaries $\Gamma_{\text{W}_{\text{ref}}}$ and Γ^s_{Dir} are disjoint parts of the boundary $\partial\Omega^s$ and $\mathbf{n}^s(X) = (n^s_j)$ is the outward unit normal to $\partial\Omega^s$.

2.2 Fluid Flow

The motion of the viscous incompressible fluid in a time dependent domain Ω_t^f is modelled by the Navier-Stokes equations written in the ALE form, see [4],

$$\frac{D^A \mathbf{v}}{Dt} + ((\mathbf{v} - \mathbf{w}_D) \cdot \nabla)\mathbf{v} - \nu^f \Delta \mathbf{v} + \nabla p = \mathbf{0}, \quad \text{div}\, \mathbf{v} = 0 \quad \text{in } \Omega_t^f, \tag{4}$$

where $\mathbf{v}(x, t)$ denotes the fluid velocity, p is the kinematic pressure and ν^f is the kinematic fluid viscosity.

To the set of Eqs. (4), a zero initial condition and the following boundary conditions are added

$$\begin{aligned}
&\text{a)} && \mathbf{v}(x,t) = \mathbf{w}_D(x,t) && \text{for } x \in \Gamma^f_{\text{Dir}} \cup \Gamma_{\text{W}_\text{t}},\\
&\text{b)} && \frac{\partial v_1}{\partial x_2}(x,t) = 0, \quad v_2(x,t) = 0 && \text{for } x \in \Gamma^f_{\text{Sym}},\\
&\text{c)} && p(x,t)\mathbf{n}^f - \nu^f \frac{\partial \mathbf{v}}{\partial \mathbf{n}^f}(x,t) = -\frac{1}{2}\mathbf{v}(\mathbf{v} \cdot \mathbf{n}^f)^- && \text{for } x \in \Gamma^f_{\text{Out}},\\
&\text{d)} && p(x,t)\mathbf{n}^f - \nu^f \frac{\partial \mathbf{v}}{\partial \mathbf{n}^f}(x,t) = -\frac{1}{2}\mathbf{v}(\mathbf{v} \cdot \mathbf{n}^f)^- + \frac{1}{\epsilon}(\mathbf{v} - \mathbf{v}_{\text{Dir}}) && \text{for } x \in \Gamma^f_{\text{In}},
\end{aligned} \tag{5}$$

where $\mathbf{n}^f$ denotes the outward unit normal $\mathbf{n}^f = (n_j^f)$ to the boundary $\partial\Omega^f$, further α^+, α^- denote the positive and the negative part of real number $\alpha \in \mathbb{R}$ and $\epsilon > 0$ is a penalization coefficient. Condition (5b) prescribes symmetry of flow along boundary Γ^f_{Sym} (y = const., see Fig. 1), condition (5c) is the directional do-nothing boundary condition, which increases the stability of the scheme, particularly in the case of strong vortices passing the outlet, see [1]. Finally, condition (5d) is the penalization boundary condition, prescribing inlet velocity with the help of suitable chosen penalization coefficient ϵ, see [9]. Contrary to the classical Dirichlet boundary condition, the inlet velocity during simulation varies, especially it decreases during the channel closing phase, see [10].

2.3 Coupling Conditions

The fluid and structure models are coupled by conditions prescribed at the common interface. The elastic subproblem is closed by Neumann boundary condition prescribing the action of aerodynamic forces $\mathbf{q}^s$ in the form of Eq. (3b), where the vector $\mathbf{q}^s$ is given as

$$q_i^s = \sum_{j=1}^{2} \rho^f \left(p\delta_{ij} - \nu^f \left(\frac{\partial v_i}{\partial x_j} + \frac{\partial v_j}{\partial x_i} \right) \right) n_j^f(x). \tag{6}$$

The Dirichlet boundary condition in the form of Eq. (5a) is provided to the fluid flow subproblem.

3 Numerical Method

For the numerical approximation of subproblems (1) and (4) the FEM is used. The time discretization is performed by the finite difference method with the uniform time step Δt, $\frac{T}{N}$ of a given time interval [0, T]. The n-th time step is denoted as $t_n = n\Delta t$.

3.1 Elastic Structure

In order to get the weak formulation of Eq. (1) is multiplied by a test function $\boldsymbol{\psi} \in \mathbf{V}$, where $\mathbf{V} = \{\mathbf{f} \in \mathbf{H}^1(\Omega^s) | \mathbf{f} = \mathbf{0} \text{ on } \Gamma^s_{\text{Dir}}\}$ and $\mathbf{H}^1(\Omega^s)$ being the vector Sobolev space. The integration over the whole domain Ω^s and the application of the Green theorem together with boundary conditions (3b) provides us

$$\left(\rho^s \frac{\partial^2 \mathbf{u}}{\partial t^2}, \boldsymbol{\psi} \right)_{\Omega^s} + \left(\lambda^s (\operatorname{div} \mathbf{u}) \mathbb{I} + 2\mu^s \mathbf{e}^s(\mathbf{u}), \mathbf{e}^s(\boldsymbol{\psi}) \right)_{\Omega^s} = \left(\mathbf{f}^s, \boldsymbol{\psi} \right)_{\Omega^s} + \left(\mathbf{q}^s, \boldsymbol{\psi} \right)_{\Gamma_{\mathrm{W_{ref}}}}, \tag{7}$$

where the brackets $(\cdot, \cdot)_{\mathcal{D}}$ denotes the dot product in the Lebesque spaces $L^2(\mathcal{D})$ or $\mathbf{L}^2(\mathcal{D})$ and $\mathbb{I}$ denotes the identity matrix. The weak solution of Eq. (1) at any time $t \in (0, T)$ is a function $\mathbf{u} \in \mathbf{H}^1(\Omega^s)$ that satisfies boundary condition (3a) and Eq. (7) for all $\psi \in \mathbf{V}$.

Let us seek an approximate solution $\mathbf{u}_h$ in a finite element space $\mathbf{V}_h \subset \mathbf{V}$ with the finite dimension N_h. Then function $\mathbf{u}_h$ can be expressed as a linear combination of basis functions $\boldsymbol{\psi}_1, \ldots, \boldsymbol{\psi}_{N_h}$ of space $\mathbf{V}_h$ and time dependent coefficients $\alpha_j(t)$, i.e. $\mathbf{u}_h(x, t) = \sum_{j=1}^{N_h} \alpha_j(t)\, \boldsymbol{\psi}_j(x)$. Using this formula and replacing $\boldsymbol{\psi}$ successively by $\boldsymbol{\psi}_i,\ i \in \{1, \ldots, N_h\}$ in Eq. (7) we get the system of ordinary differential equations (ODEs) of second order for the unknown coefficients $\alpha_j(t)$

$$\mathbb{M}\ddot{\boldsymbol{\alpha}} + \mathbb{C}\dot{\boldsymbol{\alpha}} + \mathbb{K}\boldsymbol{\alpha} = \mathbf{b}(t), \tag{8}$$

where $\mathbb{M}$, $\mathbb{K}$ and $\mathbb{C}$ denote the mass matrix, the stiffness matrix and the additionally added damping matrix, respectively. Vector $\mathbf{b}(t)$ denotes the load vector. The model of Rayleigh proportional damping is applied, i.e. the matrix $\mathbb{C}$ is chosen as $\mathbb{C} = \epsilon_1\mathbb{M} + \epsilon_2\mathbb{K}$, where ϵ_1, ϵ_2 are suitably chosen parameters. The solution of the system of ODEs (8) is approximated over time by the use of the Newmark method, see [3].

3.2 Fluid Flow

In the case of fluid flow the time discretization is performed first. The ALE derivative at t_{n+1} is approximated by the backward difference formula of second order (BDF2)

$$\frac{D^A\mathbf{v}}{Dt}(t_{n+1}) \approx \frac{3\mathbf{v}^{n+1} - 4\overline{\mathbf{v}}^n + \overline{\mathbf{v}}^{n-1}}{2\Delta t}, \tag{9}$$

where $\mathbf{v}^i \approx \mathbf{v}(\cdot, t_i)$ and for a fixed time instant t_{n+1} we denote $\overline{\mathbf{v}}^i(x) = \mathbf{v}^i(A_{t_i}(A^{-1}_{t_{n+1}}(x))$ for $i \in \{n-1, n\}$ and $x \in \Omega^f_{t_{n+1}}$. In further text we will omit top time index ${}^{n+1}$ everywhere possible, e.g. we lay $\Omega^f := \Omega^f_{t_{n+1}}$.

In order to formulate problem (4) weakly we start with the definition of function spaces involved. The function space for velocity test functions $\mathbf{X} = X \times X_2$ is defined as follows $X = \{f \in H^1(\Omega^f) |\, f = 0 \text{ on } \Gamma^f_{\text{Dir}} \cup \Gamma^f_{\text{W}_{\text{t}_{n+1}}}\}$ and $X_2 = \{f \in X |\, f = 0 \text{ on } \Gamma^f_{\text{Sym}}\}$ and $M = L^2(\Omega^f)$. The weak form of fluid flow problem reads: Find a couple $(\mathbf{v}, p) \in \mathbf{H}^1(\Omega^f) \times M$ such that $\mathbf{v}$ approximately satisfies boundary condition (5a) and

$$a(\mathbf{v}, p; \boldsymbol{\varphi}, q) + c(\mathbf{v}, \mathbf{v}; \boldsymbol{\varphi}) + \frac{1}{\epsilon}(\mathbf{v}, \boldsymbol{\varphi})_{\Gamma^f_{\text{In}}} = \frac{1}{2\Delta t}\left(4\overline{\mathbf{v}}^n - \overline{\mathbf{v}}^{n-1}, \boldsymbol{\varphi}\right)_{\Omega^f} + \frac{1}{\epsilon}(\mathbf{v}_{\text{Dir}}, \boldsymbol{\varphi})_{\Gamma^f_{\text{In}}} \tag{10}$$

is fulfilled for any test functions $(\boldsymbol{\varphi}, q) \in \mathbf{X} \times M$, where the forms are defined for any $(\mathbf{v}, p) \in \mathbf{H}^1(\Omega^f) \times M$ and $(\boldsymbol{\varphi}, q) \in \mathbf{X} \times M$ by

$$\begin{aligned} a(\mathbf{v}, p; \boldsymbol{\varphi}, q) &= \left(\frac{3\mathbf{v}}{2\Delta t}, \boldsymbol{\varphi}\right)_{\Omega^f} + \nu^f(\nabla\mathbf{v}, \nabla\boldsymbol{\varphi})_{\Omega^f} - (p, \operatorname{div}\boldsymbol{\varphi})_{\Omega^f} + (q, \operatorname{div}\mathbf{v})_{\Omega^f}, \\ c(\mathbf{v}^*, \mathbf{v}; \boldsymbol{\varphi}) &= \frac{1}{2}(((\mathbf{v}^* - 2\mathbf{w}_D^{n+1}) \cdot \nabla)\mathbf{v}, \boldsymbol{\varphi})_{\Omega^f} - \frac{1}{2}((\mathbf{v}^* \cdot \nabla)\boldsymbol{\varphi}, \mathbf{v})_{\Omega^f} \\ &\quad + \frac{1}{2}((\mathbf{v}^* \cdot \mathbf{n})^+\mathbf{v}, \boldsymbol{\varphi})_{\Gamma^f_{\text{In}} \cup \Gamma^f_{\text{Out}}}. \end{aligned} \tag{11}$$

The form $a(\cdot, \cdot; \cdot, \cdot)$ is a part of the standard weak formulation of Stokes problem. The trilinear form $c(\cdot, \cdot; \cdot)$ represents the convection in the skew-symmetric form with the directional do-nothing boundary condition taken into account, see [1]. The penalization boundary condition (5d) introduces additional terms $\frac{1}{\epsilon}(\mathbf{v}, \boldsymbol{\varphi})_{\Gamma^f_{\text{In}}}$ and $\frac{1}{\epsilon}(\mathbf{v}_{\text{Dir}}, \boldsymbol{\varphi})_{\Gamma^f_{\text{In}}}$, see [10].

Finite Element Approximation

The previously derived weak formulation (11) is discretized by the FEM. In order to avoid possible numerical instabilities of the FE solution connected with high (local) Reynolds numbers or due to the possible incompatibility of the velocity and the pressure FE spaces the stabilizations—streamline-upwind/Petrov-Galerkin (SUPG), pressure-stabilization/Petrov-Galerkin (PSPG) and 'div-div' stabilization, are applied, see [4]. This approach provides a robust and accurate numerical method, which is consistent with the original problem.

The nonlinear system of Eqs. (11) is solved using a Picard iteration method. The inf-sup stable minielement $\text{P}_1^{\text{bub}}/\text{P}_1$ is implemented, see [5]. The FSI strong coupling procedure is realized, i.e. in the inner iteration cycle in every time step the convergence of aerodynamic forces is checked, see [10].

4 Numerical Results

The FSI model geometry and also material parameters have been taken from our previously published study [10]. The time step is kept constantly equal to $\Delta t = 4 \cdot 10^{-5}$ s in order to well capture the motion related to the lowest eigenfrequencies of the structure. Further, the sensitivity of the flutter instability boundary in the dependence of penalization parameter ϵ is studied. The parameter ϵ is changed in range $[10^{-6}, 10^{-2}]$ and for each fixed value of ϵ the inlet velocity $\mathbf{v}_{\text{Dir}}$ is gradually increased until the unstable vibrations appears.

Figure 2 presents the flow-induced vibrations of structure monitored in point S from the top of the elastic body. The three cases of inlet velocities $v_{\text{A}} = 1.90$ m/s (case A), $v_{\text{B}} = 1.85$ m/s (case B) and $v_{\text{C}} = 1.80$ m/s (case C) enforced by penalization parameter $\epsilon = \frac{1}{2000}$ are considered. For inlet velocity v_{A} an exponential increase of vibration amplitude is observed. In the case B with inlet velocity 1.85 m/s the vibrations remain damped, while for smaller inlet velocity equal 1.80 m/s the vibrations are damped practically up to zero. The case B indicates that the velocity v_{B} is very close to the flutter velocity, i.e. the critical velocity is set approximately 1.87 m/s.

In the end the dependence of critical flutter velocity on the penalization parameter is summarized in Fig. 3 over the whole investigated ϵ range. The influence of penalization parameter is proven to be negligible for values $\epsilon < \frac{1}{10^4}$, while for $\epsilon > \frac{1}{10^4}$ the influence is quickly increasing. For values $\epsilon > \frac{1}{500}$ the flutter velocity

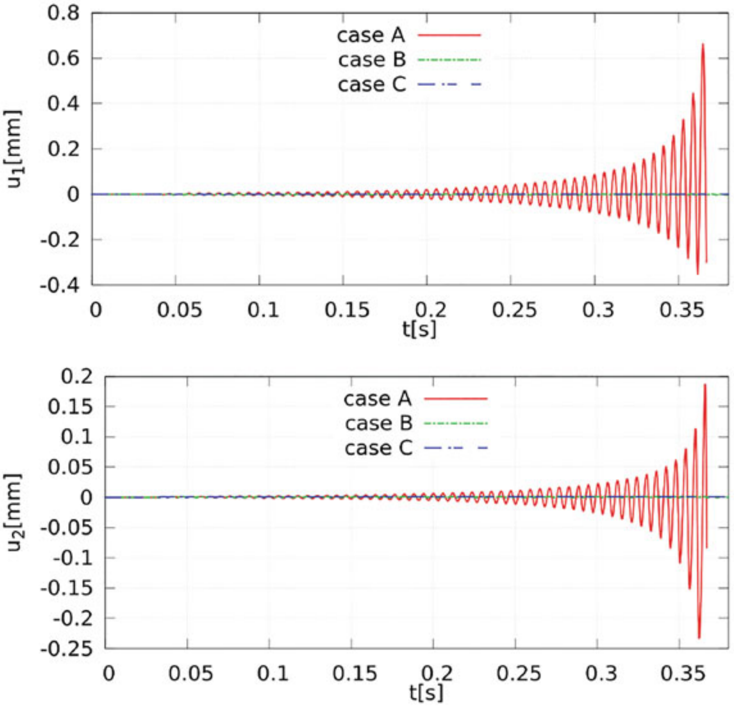


Fig. 2 The time development of u_1 (top) and u_2 (bottom) for point S is shown for three different inlet velocities: $v_{\mathrm{A}}, v_{\mathrm{B}}$ and v_{C} considering the penalization approach given by Eq. (5d) with parameter $\epsilon = \frac{1}{2000}$

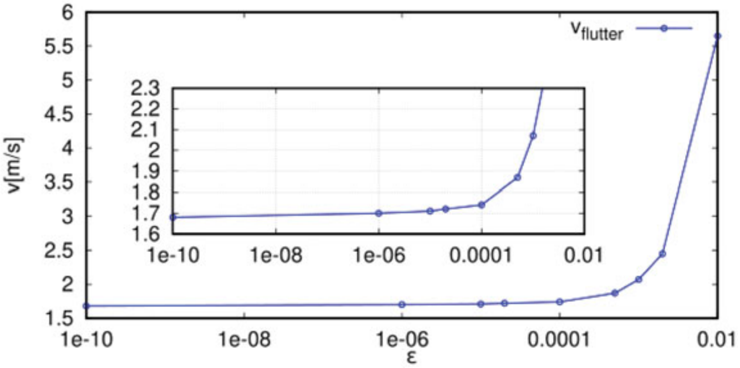


Fig. 3 Dependence of critical flutter velocity for half-channel configuration on the penalization parameter ϵ. The simulation with classical Dirichlet boundary condition is displayed as $\epsilon = 10^{-10}$

is raised by more than 50%. This can be explained by substantially decreasing the average flow rate connected with an increased penalization parameter, see [10].

5 Conclusion

The mathematical model of the FSI problem including the penalization boundary condition is described and it is numerically approximated by the (stabilized) FEM. The penalization boundary condition prescribed at the channel inlet is interesting due to possibility to mitigate unphysically high velocities at channel constriction during channel closing phase when an incompressible fluid model is used. Nevertheless the additional (penalization) parameter needs to be tuned.

This paper studied the influence of the penalization parameter ϵ on the critical velocity of flutter instability. The approximate critical flutter airflow velocity for wide range of ϵ was determined showing negligible influence for $\epsilon < \frac{1}{10^4}$. For higher values of ϵ the flutter velocity grows steeply.

Acknowledgments Authors acknowledge support from the ESIF, EU Operational Programme Research, Development and Education, and from the Center of Advanced Aerospace Technology (CZ.02.1.01/0.0/0.0/16_019/0000826), Faculty of Mechanical Engineering, CTU in Prague and by the *Grant No. SGS19/154/OHK2/3T/12* of the Grant Agency of the CTU in Prague.

References

1. M. Braack and P. B. Mucha, *Directional do-nothing condition for the Navier-Stokes equations*, Journal of Computational Mathematics **32** (2014), 507–521.
2. R. Clark and E. H. Dowell, *A modern course in aeroelasticity*, Springer, 2004.
3. A. Curnier, *Computational methods in solid mechanics*, Springer, 1994.
4. M. Feistauer, P. Sváček, and J. Horáček, *Numerical simulation of fluid-structure interaction problems with applications to flow in vocal folds*, Fluid-structure Interaction and Biomedical Applications (T. Bodnár, G. Galdi, & S. Nečasová, eds.), Birkhauser, 2014, pp. 312–393.
5. V. Girault and P. A. Raviart, *Finite element methods for Navier-Stokes equations*, Springer-Verlag, 1986.
6. T. J. Hughes, W. K. Liu, and T. K. Zimmermann, *Lagrangian-Eulerian finite element formulation for incompressible viscous flows*, Computer methods in applied mechanics and engineering **29**:3 (1981), 329–349.
7. W. S. Slaughter, *Linearized elasticity problems*, Springer, 2002.
8. P. Sváček and J. Horáček, *Numerical simulation of glottal flow in interaction with self oscillating vocal folds: comparison of finite element approximation with a simplified model*, Communications in Computational Physics **12** (2012), 789–806.
9. ———, *Finite element approximation of flow induced vibrations of human vocal folds model: Effects of inflow boundary conditions and the length of subglottal and supraglottal channel on phonation onset*, Applied Mathematics and Computation **319** (2018), 178–194.
10. J. Valášek, P. Sváček, and J. Horáček, *On suitable inlet boundary conditions for fluid-structure interaction problems in a channel*, Appl. of Mathematics **64**:2 (2019), 225–251.

Approximation Properties of Discrete Boundary Value Problems for Elliptic Pseudo-Differential Equations

Oksana Tarasova and Vladimir Vasilyev

Abstract We study some discrete boundary value problems which are treated as digital approximation for starting boundary value problem for elliptic pseudo-differential equation. Starting from existence and uniqueness theorem we give a comparison between discrete and continuous solutions for certain boundary value problems.

1 Introduction

We study a discrete variant of the following boundary value problem

$$\begin{cases} (Au)(x) = f(x), \quad x \in D, \\ (Bu)|_{\partial D} = g \end{cases} \tag{1}$$

where A, B are simplest elliptic pseudo-differential operators [1–3] with symbols $A(\xi)$, $B(\xi)$, acting in Sobolev–Slobodetskii spaces $H^s(D)$, $D \subset \mathbf{R}^m$ is a certain bounded domain, f, g are given functions.

Discrete variants of similar problems for differential operators were studied earlier (see, for example [4] with difference schemes, or [5] with difference potentials), but we would like to develop an approach for more general pseudo-differential operators and related equations. This approach is based on a concept of periodic factorization for an elliptic symbols and it is a discrete analogue of corresponding continuous methods [1].

Some first studies in this direction were done in [6–15].

O. Tarasova · V. Vasilyev (✉)
Belgrod State National Research University, Belgorod, Russia
e-mail: tarasova_o@bsu.edu.ru

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_108

2 Digital Operators and Discrete Boundary Value Problems

Here we will describe our approach to studying discrete equations and boundary value problems.

Given function u_d of a discrete variable $\tilde{x} \in h\mathbf{Z}^m$, $h > 0$, we define its discrete Fourier transform by the series

$$(F_d u_d)(\xi) \equiv \widetilde{u}_d(\xi) = \sum_{\tilde{x} \in \mathbf{Z}^m} e^{i\tilde{x}\cdot\xi} u_d(\tilde{x}), \quad \xi \in \hbar\mathbf{T}^m,$$

where $\mathbf{T}^m = [-\pi, \pi]^m$, $\hbar = h^{-1}$, and partial sums are taken over cubes

$$Q_N = \{\tilde{x} \in h\mathbf{Z}^m : \tilde{x} = (\tilde{x}_1, \cdots, \tilde{x}_m), \max_{1 \le k \le m} |\tilde{x}_k| \le N\}.$$

We will remind here some definitions of functional spaces [12] and will consider discrete analogue $S(h\mathbf{Z}^m)$ of the Schwartz space $S(\mathbf{R}^m)$. Let us denote $\zeta^2 = h^{-2} \sum\limits_{k=1}^{m} (e^{-ih\cdot\xi_k} - 1)^2$.

The space $H^s(h\mathbf{Z}^m)$ is a closure of the space $S(h\mathbf{Z}^m)$ with respect to the norm

$$||u_d||_s = \left(\int\limits_{\hbar\mathbf{T}^m} (1 + |\zeta^2|)^s |\tilde{u}_d(\xi)|^2 d\xi \right)^{1/2}. \tag{2}$$

Fourier image of the space $H^s(h\mathbf{Z}^m)$ will be denoted by $\widetilde{H}^s(\hbar\mathbf{T}^m)$.

One can define some discrete operators for such functions u_d.

If $\widetilde{A}_d(\xi)$ is a periodic function in $\mathbf{R}^m$ with the basic cube of periods $\hbar\mathbf{T}^m$ then we consider it as a symbol. We will introduce a digital pseudo-differential operator in the following way.

Definition 1 A digital pseudo-differential operator A_d in a discrete domain D_d is called the operator [12]

$$(A_d u_d)(\tilde{x}) = \sum_{\tilde{y} \in h\mathbf{Z}^m} \int\limits_{\hbar\mathbf{T}^m} \widetilde{A}_d(\xi) e^{i(\tilde{x}-\tilde{y})\cdot\xi} \tilde{u}_d(\xi) d\xi, \quad \tilde{x} \in D_d,$$

We consider a class of symbols [12] satisfying the following condition

$$c_1(1 + |\zeta^2|)^{\alpha/2} \le |A_d(\xi)| \le c_2(1 + |\zeta^2|)^{\alpha/2}, \quad \alpha \in \mathbf{R}, \tag{3}$$

and universal positive constants c_1, c_2.

Let $D \subset \mathbf{R}^m$ be a domain. We will study the equation

$$(A_d u_d)(\tilde{x}) = v_d(\tilde{x}), \quad \tilde{x} \in D_d, \tag{4}$$

in the discrete domain $D_d \equiv D \cap h\mathbf{Z}^m$ and will seek a solution $u_d \in H^s(D_d)$, $v_d \in H_0^{s-\alpha}(D_d)$ [12, 15].

In this paper we will discuss the case $D \equiv \mathbf{R}_+^m$.

Let $\tilde{A}_d(\xi)$ be a periodic symbol. Let us denote $\Pi_\pm$ half-strips in the complex plane $\mathbf{C}$

$$\Pi_\pm = \{z \in \mathbf{C} : z = s + i\tau, s \in [-\pi, \pi], \pm\tau > 0\}.$$

Definition 2 Periodic factorization of an elliptic symbol $A_d(\xi) \in E_\alpha$ is called its representation in the form

$$A_d(\xi) = A_{d,+}(\xi) A_{d,-}(\xi),$$

where the factors $A_{d,\pm}(\xi)$ admit an analytical continuation into half-strips $\hbar\Pi_\pm$ on the last variable ξ_m for almost all fixed $\xi' \in \hbar\mathbf{T}^{m-1}$ and satisfy the estimates

$$|A_{d,+}^{\pm 1}(\xi)| \leq c_1(1 + |\hat{\zeta}^2|)^{\pm\frac{æ}{2}}, \quad |A_{d,-}^{\pm 1}(\xi)| \leq c_2(1 + |\hat{\zeta}^2|)^{\pm\frac{\alpha - æ}{2}},$$

with constants c_1, c_2 non-depending on h,

$$\hat{\zeta}^2 \equiv \hbar^2 \left(\sum_{k=1}^{m-1} (e^{-ih\xi_k} - 1)^2 + (e^{-ih(\xi_m + i\tau)} - 1)^2 \right), \quad \xi_m + i\tau \in \hbar\Pi_\pm.$$

The number $æ \in \mathbf{R}$ is called an index of periodic factorization.

We consider the following discrete boundary value problem

$$\begin{cases} (A_d u_d)(\tilde{x}) = v_d(\tilde{x}), & \tilde{x} \in \mathbf{R}_+^m \\ (B_d u_d)_{|\tilde{x}_m = 0} = g_d(\tilde{x}'), & \tilde{x}' \in \mathbf{R}^{m-1}, \end{cases} \tag{5}$$

such that the discrete boundary value problem (5) will have good approximation properties for initial boundary value problem.

3 Solvability and Comparison

This section is devoted to the following questions:

1. to establish solvability for our discrete boundary value problem;
2. to give a comparison between discrete and continuous solutions.

3.1 Solvability

To describe solvability for the boundary value problem (5) we introduce the following notations.

$$(H_{\xi'}^{per}\tilde{u}_d)(\xi',\xi_m)=\frac{h}{2\pi i}\ p.v.\int\limits_{-\hbar\pi}^{\hbar\pi}\cot\frac{h(\xi_m-\eta_m)}{2}\tilde{u}_d(\xi',\eta_m)d\eta_m,$$

where

$$p.v.\int\limits_{-\hbar\pi}^{\hbar\pi}\cot\frac{h(\xi_m-\eta_m)}{2}\tilde{u}_d(\xi',\eta_m)d\eta_m$$

$$=\lim_{\varepsilon\to 0+}\left(\int\limits_{-\hbar\pi}^{\xi_m-\varepsilon}+\int\limits_{\xi_m+\varepsilon}^{\hbar\pi}\right)\cot\frac{h(\xi_m-\eta_m)}{2}\tilde{u}_d(\xi',\eta_m)d\eta_m$$

This operator generates two projectors

$$P_{\xi'}^{per}=\frac{1}{2}(I+H_{\xi'}^{per}),\quad Q_{\xi'}^{per}=\frac{1}{2}(I-H_{\xi'}^{per}),$$

which permit to formulate and solve the following problem.

The following theorem was proved in the paper [7].

Theorem 1 *Let* $æ-s=n+\delta, n\in\boldsymbol{N}, |\delta|<1/2$. *Then a general solution of Eq. (4) in Fourier images has the following form*

$$\tilde{u}_d(\xi)=\tilde{A}_{d,+}^{-1}(\xi)X_n(\xi)P_{\xi'}^{per}(X_n^{-1}(\xi)\tilde{A}_{d,-}^{-1}(\xi)\widetilde{\ell v_d}(\xi))+\tilde{A}_{d,+}^{-1}(\xi)\sum_{k=0}^{n-1}\tilde{c}_k(\xi')\hat{\zeta}_m^k,$$

where $X_n(\xi)$ *is an arbitrary polynomial of order* n *of variables* $\hat{\zeta}_k=\hbar(e^{-ih\xi_k}-1), k=1,\cdots,m$, *satisfying the condition (2),* $c_k(\xi'), j=0,1,\cdots,n-1$, *are arbitrary functions from* $H^{s_k}(h\mathbf{T}^{m-1}), s_k=s-æ+k-1/2$, ℓv_d *is an arbitrary continuation of* v_d. *from* $H^{s-\alpha}(D_d)$ *into* $H^{s-\alpha}(h\mathbf{Z}^m)$

The a priori estimate

$$||u_d||_s\leq a(||f||_{s-\alpha}^{+}+\sum_{k=0}^{n-1}[c_k]_{s_k})$$

holds, where $[\cdot]_{s_k}$ *denotes a norm in the space* $H^{s_k}(h\mathbf{T}^{m-1})$*, and the constant* a *does not depend on* h*.*

We will apply Theorem 1 for the simple case $n = 1$, because we consider only one boundary condition. Then we have

$$\tilde{u}_d(\xi) = \tilde{h}_d(\xi) + \tilde{A}^{-1}_{d,+}(\xi)\tilde{c}_0(\xi'), \tag{6}$$

where we denote

$$\tilde{h}_d(\xi) = \tilde{A}^{-1}_{d,+}(\xi)X_1(\xi)P^{per}_{\xi'}(X_1^{-1}(\xi)\tilde{A}^{-1}_{d,-}(\xi)\widetilde{\ell v_d}(\xi)) \tag{7}$$

The construction of a general solution for starting boundary value problem is very similar and exact, it was obtained in [1]. For our case it has the following form

$$\tilde{u}(\xi) = \tilde{h}(\xi) + \tilde{A}^{-1}_{+}(\xi)\tilde{C}_0(\xi'), \tag{8}$$

$$\tilde{h}(\xi) = \tilde{A}^{-1}_{+}(\xi)Y_1(\xi)P_{\xi'}(Y_1^{-1}(\xi)\tilde{A}^{-1}_{-}(\xi)\widetilde{\ell f}(\xi)), \tag{9}$$

where $P_{\xi'} = 1/2(I + H_{\xi'})$, and $H_{\xi'}$ is the classical Hilbert transform on the last variable ξ_m

$$(H_{\xi'}u(\xi', \xi_m) = \frac{1}{\pi i}\ p.v. \int\limits_{-\infty}^{+\infty} \frac{u(\xi', \tau)d\tau}{\xi_m - \tau},$$

$Y_1(\xi)$ is an arbitrary polynomial of variables $\xi_1, \cdots, \xi_m$ satisfying the condition $|Y_1(\xi)| \sim 1 + |\xi|$, $\tilde{A}_{\pm}(\xi)$ are factors of factorization for the symbol $\tilde{A}(\xi)$.

The formulas (8), (9) are valid under assumptions that the symbols $A(\xi)$ satisfies the condition

$$c_1(1 + |\xi|)^{\alpha} \leq |\tilde{A}(\xi)| \leq c_2(1 + |\xi|)^{\alpha}, \tag{10}$$

and index factorization of the symbol $A(\xi)$ equals æ.

There are arbitrary functions $\tilde{c}_0, \tilde{C}_0$ in the formulas (6), (8). To determine, for example, the function $\tilde{c}_0$ we use the boundary condition from (5). We act by the operator B on the solution u_d and then we take the restriction on the discrete half-plane $\tilde{\xi}_m = 0$. According to properties of the discrete Fourier transform we have

$$\int\limits_{-\hbar\pi}^{+\hbar\pi} \tilde{B}_d(\xi', \xi_m)\tilde{u}_d(\xi', \xi_m)d\xi_m = \int\limits_{-\hbar\pi}^{+\hbar\pi} \tilde{B}_d(\xi', \xi_m)\tilde{h}_d(\xi', \xi_m)d\xi_m + \tilde{c}_0(\xi')b_d(\xi'),$$

where

$$b_d(\xi') = \int\limits_{-\hbar\pi}^{+\hbar\pi} \tilde{B}_d(\xi', \xi_m) A^{-1}_{d,+}(\xi', \xi_m) d\xi_m$$

Here we use the condition $\inf\limits_{\xi' \in \hbar\mathbf{T}^{m-1}} |b_d(\xi'| > 0$; it is a discrete analogue of Shapiro–Lopatinskii condition [1]. Since the left hand side is $\tilde{g}_d(\xi')$ we have the following relation

$$\tilde{c}_0(\xi') = b_d^{-1}(\xi') \left(\tilde{g}_d(\xi') - \tilde{t}_d(\xi') \right), \tag{11}$$

where

$$\tilde{t}_d(\xi') = \int\limits_{-\hbar\pi}^{+\hbar\pi} \tilde{B}_d(\xi', \xi_m) \tilde{h}_d(\xi', \xi_m) d\xi_m.$$

By substitution of (11) into (6), we obtain a unique solution for the discrete boundary value problem (5):

$$\tilde{u}_d(\xi) = \tilde{h}_d(\xi) + \tilde{A}^{-1}_{d,+}(\xi) b_d^{-1}(\xi') \left(\tilde{g}_d(\xi') - \tilde{t}_d(\xi') \right), \tag{12}$$

3.2 A Comparison

According to Vishik–Eskin theory [1] we have a continuous analogue of the formula (12), namely

$$\tilde{u}(\xi) = \tilde{h}(\xi) + \tilde{A}^{-1}_{+}(\xi) b^{-1}(\xi') \left(\tilde{g}(\xi') - \tilde{t}(\xi') \right) \tag{13}$$

under the condition $\inf\limits_{\xi' \in \mathbf{R}^{m-1}} |b(\xi'| > 0$. Now we would like to compare two formulas (12) and (13). To simplify our considerations we put $f \equiv 0$. Then the functions h, h_d, t, t_d are zero.

To obtain a good approximation we choose certain elements for the discrete solution in a particular way.

First, let us denote by q_h the following operator of restriction and periodization; this operator acts in Fourier images. Given function $\tilde{u}$ the notation $q_h\tilde{u}$ means that we take a restriction of $\tilde{u}$ on $\hbar\mathbf{T}^m$ and periodically continue it into whole $\mathbf{R}^m$. The symbol $\tilde{A}_d(\xi)$ of the discrete operator A_d is the following. We take the factorization

$$\tilde{A}(\xi) = \tilde{A}_+(\xi) \cdot \tilde{A}_-(\xi)$$

and introduce the periodic symbol by the formula

$$\tilde{A}_d(\xi) \equiv (q_h \tilde{A}_+)(\xi) \cdot (q_h \widetilde{A}_-)(\xi),$$

so we have immediately the needed periodic factorization.

Secondly, we define the symbol $\tilde{B}_d(\xi)$ of the boundary operator B_d by

$$\tilde{B}_d(\xi) \equiv (q_h \tilde{B})(\xi).$$

Third, we choose $g_d = F_d^{-1}(q_h \tilde{g})$, where

$$(F_d^{-1}\tilde{u}_d)(\tilde{x}) = \frac{1}{(2\pi)^m} \int\limits_{\hbar \mathbf{T}^m} e^{i\tilde{x}\cdot\xi} \tilde{u}_d(\xi) d\xi, \quad \tilde{x} \in h\mathbf{Z}^m.$$

Lemma 1 *Let the boundary symbol $\tilde{B}(\xi)$ satisfy the condition (10) with order β. Then the following estimate*

$$|\tilde{b}_d(\xi') - \tilde{b}(\xi')| \leq ch^{æ-1-\beta}$$

holds.

Proof We give corresponding estimates:

$$|b(\xi') - b_d(\xi')| = \left| \int\limits_{-\infty}^{+\infty} \tilde{B}(\xi', \xi_m)\tilde{A}_+^{-1}(\xi', \xi_m)d\xi_m - \int\limits_{-\hbar\pi}^{\hbar\pi} \tilde{B}(\xi', \xi_m)\tilde{A}_{d,+}^{-1}(\xi', \xi_m)d\xi_m \right| =$$

$$\left| \left(\int\limits_{-\infty}^{-\hbar\pi} + \int\limits_{\hbar\pi}^{+\infty} \right) \tilde{B}(\xi', \xi_m)\tilde{A}_+^{-1}(\xi', \xi_m)d\xi_m \right|.$$

Two integrals have the same estimate and we consider the second one.

$$\int\limits_{\hbar\pi}^{+\infty} |\tilde{B}(\xi', \xi_m)A_+^{-1}(\xi', \xi_m)|d\xi_m \leq c_5 \int\limits_{\hbar\pi}^{+\infty} (1 + |\xi'| + |\xi_m|)^{\beta-æ} d\xi_m =$$

$$\frac{c_5}{æ - 1 - \beta}(1 + |\xi'| + \hbar\pi)^{1-æ} \leq ch^{æ-1-\beta}.$$

Theorem 2 *Let $f \equiv 0, v_d \equiv 0,$ $g \in H^{s-\beta-1/2}(\mathbf{R}^{m-1})$, $g_d \in H^{s-\beta-1/2}(h\mathbf{Z}^{m-1})$, $s-\beta > 1/2$, $æ > 1+\beta$, and*

$$\inf_{\xi' \in \mathbf{R}^{m-1}} |b(\xi')| > 0, \quad \inf_{\xi' \in \mathbf{T}^{m-1}, h>0} |b_d(\xi')| > 0.$$

Then boundary value problems (1) and (5) have unique solutions in spaces $H^s(\mathbf{R}^m_+)$ and $H^s(h\mathbf{Z}^m_+)$ respectively.

If $g \in L_1(\mathbf{R}^{m-1})$ then we have the estimate

$$|\tilde{u}_d(\xi) - \tilde{u}(\xi)| \leq ch^{æ-1-\beta}, \quad \xi \in \hbar\mathbf{T}^m.$$

Proof The existence and uniqueness for the problems was proved in [1] for continuous case and in [8] for discrete case, and here we have described the construction for solving discrete boundary value problem. Therefore we need to prove the estimate. We have

$$\tilde{u}(\xi) - \tilde{u}_d(\xi) = b^{-1}(\xi')\tilde{g}(\xi')A_+^{-1}(\xi', \xi_m) - b_d^{-1}(\xi')\tilde{g}_d(\xi')A_{d,+}^{-1}(\xi', \xi_m) =$$

$$(b^{-1}(\xi') - b_d^{-1}(\xi'))\tilde{g}_d(\xi')A_{d,+}^{-1}(\xi', \xi_m), \quad \xi \in \hbar\mathbf{T}^m,$$

and using Lemma 1 and boundedness of $\tilde{g}$ we complete the estimate.

References

1. Eskin, G.I.: Boundary value problems for elliptic pseudodifferential equations. AMS Providence (1981)
2. Taylor, M.E.: Pseudo-Differential Operators. Princeton Univ. Press Princeton (1980)
3. Treves, F.: Introduction to Pseudodifferential Operators and Fourier Integral Operators. Springer New York (1980)
4. Samarskii, A.A.: The Theory of Difference Schemes. CRC Press Boca Raton (2001)
5. Ryaben'kii, V.S.: Method of Difference Potentials and its Applications. Springer-Verlag Berlin–Heidelberg (2002)
6. Vasilyev, A.V., Vasilyev, V.B.: Periodic Riemann problem and discrete convolution equations. Differ. Equ. **51**, 652–660 (2015)
7. Vasilyev, A.V., Vasilyev, V.B.: Pseudo-differential operators and equations in a discrete half-space. Math. Model. Anal. **23** 492–506 (2018)
8. Vasilyev, A.V., Vasilyev, V.B.: On some discrete boundary value problems in canonical domains. In: Differential and Difference Equations and Applications. Springer Proc. Math. Stat. V. 230, pp.569–579, Cham: Springer (2018)
9. Vasilyev, A.V., Vasilyev, V.B.: On some discrete potential like operators. Tatra Mt. Math. Publ. **71** 195–212 (2018)
10. Vasilyev, A.V., Vasilyev, V.B.: On a digital approximation for pseudo-differential operators. Proc. Appl. Math. Mech. **17** 763–764 (2017)

11. Vasilyev, V.B.: Discreteness, periodicity, holomorphy and factorization. In: Constanda, C., Dalla Riva, M., Lamberti, P.D., Musolino, P. (eds.) Integral Methods in Science and Engineering. V.1, pp. 315–324. Theoretical Technique. Springer, Cham. (2017)
12. Vasilyev, A., Vasilyev, V.: Digital Operators, Discrete Equations and Error Estimates. In: Radu, F., Kumar, K., Berre, I., Nordbotten, J., Pop, I. (eds.) Numerical Mathematics and Advanced Applications ENUMATH 2017. Lecture Notes in Computational Science and Engineering, vol 126, pp. 983–991. Springer, Cham. (2019)
13. Vasilyev, V.B.: Digital Approximations for Pseudo-Differential Equations and Error Estimates. In: Nikolov, G., Kolkovska, N., Georgiev, K. (eds.) Numerical Methods and Applications. NMA 2018. Lecture Notes in Computer Science, vol 11189, pp. 483–490. Springer, Cham. (2019)
14. Vasilyev, V.B.: On a Digital Version of Pseudo-Differential Operators and Its Applications. In: Dimov, I., Faragó, I., Vulkov, L. (eds.) Finite Difference Methods. Theory and Applications. FDM 2018. Lecture Notes in Computer Science, vol 11386, pp. 596–603. Springer, Cham. (2019)
15. Vasilyev, V.: The periodic Cauchy kernel, the periodic Bochner kernel, and discrete pseudo-differential operators, *AIP Conf. Proc.* **1863** 140014 (2017)

Mathematical and Numerical Models of Atherosclerotic Plaque Progression in Carotid Arteries

Silvia Pozzi and Christian Vergara

Abstract We propose a mathematical model for the description of plaque progression in carotid arteries. This is based on the coupling of a fluid-structure interaction problem, arising between blood and vessel wall, and differential problems for the cellular evolution. A numerical model is also proposed. This is based on the splitting of the coupled problem based on a suitable strategy to manage the multiscale-in-time nature of the problem. We present some preliminary numerical results both in ideal and real scenarios.

1 Introduction

Atherosclerosis consists in the formation of plaques at bifurcation sites. Carotid arteries are one of the preferential sites of atherosclerotic plaque formation. The main complications related to plaque formation are the partial or total occlusion of the internal carotids with consequent cerebral ischemia possibly leading to stroke, the rupture of the plaque with consequent embolization of fragments in the brain vessels, and the formation of a thrombus whose detachment leads to embolism.

The mechanism of plaque formation can be briefly summarized as follows. In regions where the viscous forces exerted by the fluid on the arterial wall (wall shear stresses, WSS) are low and oscillating, the permeability of the internal vessel layer (intima) to low-density lipoprotein (LDL) increases [10]. Once in the intima, LDL can oxidize, leading to a pathological inflammation. To remove this, macrophages are recruited. Due to the ingestion of large amounts of oxidized LDL,

S. Pozzi
MOX, Dipartimento di Matematica, Politecnico di Milano, Milano, Italy
e-mail: silvia.pozzi@polimi.it

C. Vergara (✉)
LABS, Dipartimento di Chimica, Materiali e Ingegneria Chimica, Politecnico di Milano, Milano, Italy
e-mail: christian.vergara@polimi.it

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_109

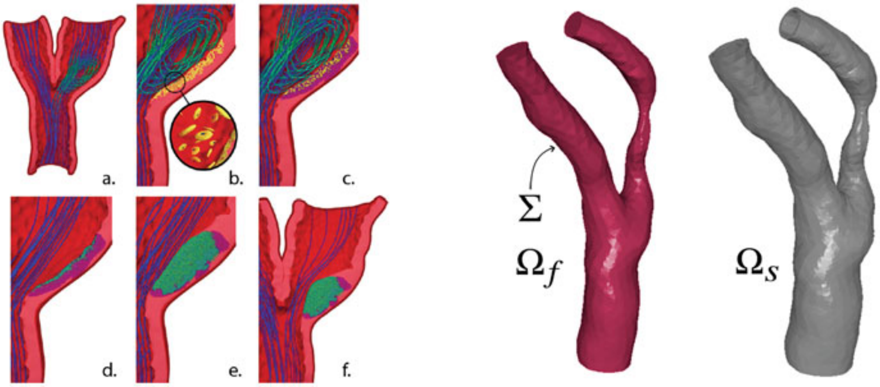


Fig. 1 Left: Schematic view of plaque formation and progression. (**a**) Carotid with blood recirculation; (**b**) LDL (in yellow) penetrates in the vessel wall; (**c**) oxidized LDL (in purple); (**d**–**f**) macrophages (in green) accumulate. Right: fluid and structure computational domains

macrophages differentiate into foam cells, which are responsible for the growth of a sub-endothelial plaque. If this inflammatory process persists for a sufficient time, the plaque can emerge in the lumen. In Fig. 1, left, these steps are schematically reported.

The prediction of the formation and evolution of the plaque in carotids is of utmost importance. For this reason, in recent years some studies have focused on the mathematical description of plaque progression [1, 5, 17]. The main characteristics that a mathematical model should account for are:

(i) A detailed description of blood dynamics, which plays a crucial role in plaque progression;
(ii) A description of the mutual dependence between macro and micro spatial scales;
(iii) The coupling between models describing events that occur with different characteristic times, that is seconds (blood dynamics) and years (plaque progression).

Regarding point (i), some works consider blood dynamics in rigid walls [1, 2, 5], whereas more recent studies include fluid-structure interaction (FSI) to better describe blood dynamics and include a growth tensor in the vessel wall dynamics to account for the plaque development [15, 16] (point (ii), *micro-to-macro* scales feedback). Another choice consists in using a plaque growth law [1, 2, 5, 17]. Regarding the *macro-to-micro* scales feedback in point (ii), most of the studies derive a relation between WSS and variation of permeability of cellular quantities [1, 5, 15]. A wide class of works considers a macroscopic description of cellular events by means of suitable partial differential equations (PDEs) for LDL, macrophages and foam cells evolution [1, 2, 5, 15, 16], whereas other works consider for them a cellular description [17]. Regarding point (iii), we point out that no specific techniques have been considered so far to manage the different temporal scales. We also notice that most of the works consider ideal geometries for the numerical experiments [1, 2, 5, 16], whereas in [15] an application to real geometries of mice is considered.

Starting from the studies cited above, in this work we propose a mathematical model for plaque progression in carotids, based on an FSI model, PDEs for LDL, macrophages and foam cells evolution, spatial feedbacks between macro and micro scales based on WSS and a growth law. At the numerical level, we propose a new strategy to deal with the multiscale-in-time nature of the problem. The model is applied to 3D cases, both in ideal and real geometries.

2 Mathematical Models

The mathematical model for plaque progression is based on two groups of differential problems and on their coupling. In the first group, we consider the "short time scale" model, that is the FSI problem, in the unknowns fluid velocity and pressure $(\boldsymbol{u}, p)$ and structure displacement $\boldsymbol{d}$, which occurs with a characteristic time of 1 s (the heartbeat). In the second group, we have the "long time scales" models, interacting with one another, that is two time dependent diffusion-reaction (DR) problems for LDL and macrophages concentrations (c_{LDL} and c_{macr}) [1, 15] and one ordinary differential equation (ODE) for the foam cells concentration c_{FC} [5].

The coupling between these two groups of models is provided by the time-averaged WSS (TAWSS), which influences the LDL and macrophages permeabilities [1] (*macro-to-micro* feedback) and by c_{FC}, which determines the plaque growth $\boldsymbol{d}_G$ [17] (*micro-to-macro* feedback). In Fig. 2 we report a diagram of the mathematical model. Notice that the fluid and structure domains Ω_f and Ω_s depend on the plaque growth $\boldsymbol{d}_G$.

$$\begin{tikzcd}
 & \begin{array}{c}\textbf{Short time scales}\\ (\boldsymbol{u},p,\boldsymbol{d})=\mathcal{FS}\left(\Omega_f\left(\boldsymbol{d}_G\right),\Omega_s\left(\boldsymbol{d}_G\right)\right)\end{array} \arrow[dr, bend left] & \\
\boldsymbol{d}_G=g_2\left(c_{FC}\right) \arrow[ur, bend left] & & TAWSS=g_1\left(\boldsymbol{u}\right) \arrow[dl, bend left] \\
 & \begin{array}{c}\textbf{Long time scales}\\ c_{LDL}=\mathcal{DR}\left(TAWSS;\Omega_s\left(\boldsymbol{d}_G\right)\right)\\ c_{macr}=\mathcal{DR}\left(TAWSS,c_{LDL},c_{FC};\Omega_s\left(\boldsymbol{d}_G\right)\right)\\ c_{FC}=\mathcal{ODE}\left(c_{LDL},c_{macr};\Omega_s\left(\boldsymbol{d}_G\right)\right)\end{array} \arrow[ul, bend left] &
\end{tikzcd}$$

Fig. 2 Diagram of the mathematical model of plaque progression

Referring to Fig. 1, right, we detail in what follows the equations of the submodels.

- **FSI problem**:

$$\begin{cases} \rho_f\left(\partial_t \boldsymbol{u} + \boldsymbol{u}\cdot\nabla\boldsymbol{u}\right) - \nabla\cdot\boldsymbol{T}_f(\boldsymbol{u},p) = \boldsymbol{0} & \text{in } \Omega_f(\boldsymbol{d},\boldsymbol{d}_G),\\ \nabla\cdot\boldsymbol{u} = 0 & \text{in } \Omega_f(\boldsymbol{d},\boldsymbol{d}_G),\\ \boldsymbol{u} = \partial_t\boldsymbol{d} & \text{on } \Sigma(\boldsymbol{d},\boldsymbol{d}_G),\\ \boldsymbol{T}_f(\boldsymbol{u},p)\boldsymbol{n} = \boldsymbol{T}_s(\boldsymbol{d})\boldsymbol{n} & \text{on } \Sigma(\boldsymbol{d},\boldsymbol{d}_G),\\ \rho_s\partial_{tt}\widehat{\boldsymbol{d}} - \nabla\cdot\widehat{\boldsymbol{T}}_s = \boldsymbol{0} & \text{in } \widehat{\Omega}_s(\boldsymbol{d}_G), \end{cases} \tag{1}$$

together with initial and boundary conditions and where ρ_f and ρ_s are the fluid and structure densities, $\boldsymbol{T}_f = \mu(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T)\boldsymbol{n} - p\boldsymbol{I}$ is the fluid Cauchy stress tensor, μ the fluid viscosity, $\widehat{\boldsymbol{T}}_s$ is the wall first Piola-Kirkhhoff tensor representing an hyperelastic material, $\boldsymbol{n}$ is the unit normal vector, and $\widehat{\ }$ denotes quantities in the Lagrangian framework;

- **Time-averaged wall shear stress**:

$$TAWSS = \frac{1}{T}\int_0^T \mu\sum_{j=1}^{2}\sqrt{\left(\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right)\boldsymbol{n}\cdot\boldsymbol{\tau}_j\right)^2 dt} = g_1(\boldsymbol{u}), \tag{2}$$

with $\boldsymbol{\tau}_j$ the tangential unit vectors;

- **Cellular differential problems**:

$$\begin{cases} \partial_t c_{LDL} - \nabla\cdot(D_{LDL}\nabla c_{LDL}) + r_{ox}c_{LDL} = 0 & \text{in } \Omega_s(\boldsymbol{d}_G),\\ \zeta_{LDL}c_{LDL} - D_{LDL}\nabla c_{LDL}\cdot\boldsymbol{n} = -\zeta_{LDL}c_{LDL,f} & \text{on } \Sigma(\boldsymbol{d},\boldsymbol{d}_G),\\ \zeta_{LDL} = \dfrac{\zeta_{LDL}^{ref}}{ln(2)}ln\left(1 + \dfrac{2TAWSS^{ref}}{TAWSS + TAWSS^{ref}}\right); \end{cases} \tag{3}$$

$$\begin{cases} \partial_t c_{macr} - \nabla\cdot(D_{macr}\nabla c_{macr}) + (r_{ox}c_{LDL})\,c_{macr} = 0 & \text{in } \Omega_s(\boldsymbol{d}_G),\\ \zeta_{macr}c_{macr} - D_{macr}\nabla c_{macr}\cdot\boldsymbol{n} = -\zeta_{macr}c_{macr,f} & \text{on } \Sigma(\boldsymbol{d},\boldsymbol{d}_G),\\ \zeta_{macr} = \dfrac{\zeta_{macr}^{ref}}{ln(2)}ln\left(1 + \dfrac{31/30TAWSS^{ref}}{TAWSS + 1/30TAWSS^{ref}}\right),\\ D_{macr} = D_{macr}^{dis} + \left(D_{macr}^{healthy} - D_{macr}^{dis}\right)e^{-c_{FC}}; \end{cases} \tag{4}$$

$$\partial_t c_{FC} = r_{ox}c_{LDL}c_{macr} \qquad \text{in } \Omega_s(\boldsymbol{d}_G), \tag{5}$$

where D are the diffusion tensors, r_{ox} the oxidation rate, ζ the permeabilities, index ^{ref} means reference, index ^{dis} means diseased. Notice that both diffusion-reaction problems are equipped by a Robin condition at the interface Σ to account for the equilibrium with the fluid concentrations, which are here supposed to be known constants $(c_{LDL,f},\ c_{macr,f})$;

- **Growth function**

$$\boldsymbol{d}_G = \kappa c_{FC}\boldsymbol{n} = g_2(c_{FC}) \qquad \text{on } \Sigma(\boldsymbol{d}_G), \tag{6}$$

that is a growth of the interface that occurs in the normal direction and where κ is a parameter regulating the growth rate [17]. The growth $\boldsymbol{d}_g$ is then extended in the whole $\Omega_s(\boldsymbol{d}_g)$ by means of an harmonic extension.

3 Numerical Methods

For the numerical solution of the coupled problem (1)–(2)–(3)–(4)–(5)–(6), we propose a way to treat its multiscale-in-time nature, that in fact decouples the subproblems (FSI, LDL, macrophages, foam cells) which could be solved by means of separate/pre-existing codes. This strategy is summarized in Fig. 3.

The blue region, characterized by a time discretization parameter Δt, is devoted to the discretized-in-time long time scale problems (3)–(4)–(5), which are solved for K time instants at the current block m in the domain $\Omega_{s,m-1}$ obtained at the previous block.

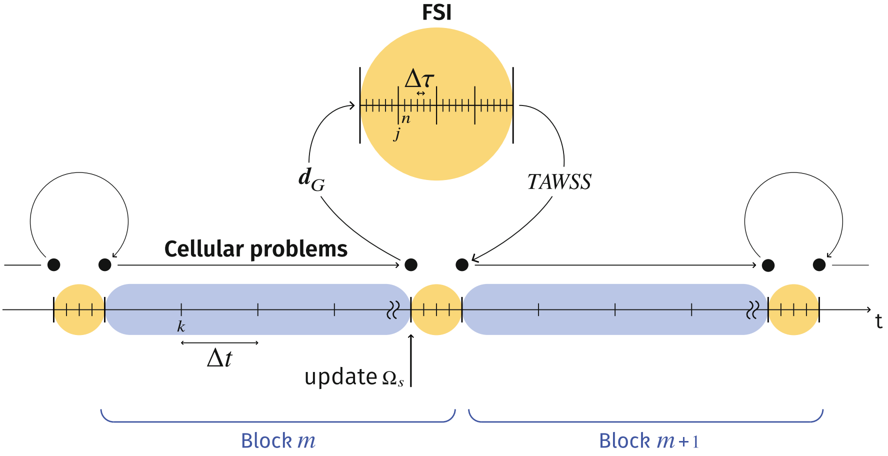


Fig. 3 Schematic representation of the numerical treatment of the multiscale-in-time coupled problem

After K time steps we update the structure domain by means of (6), obtaining $\Omega_{s,m}$ and then we solve for J heartbeats the discretized-in-time FSI problem (1) in this domain and in the corresponding fluid domain, with a discretization parameter $\Delta\tau << \Delta t$ (yellow region). After J heartbeats, we compute TAWSS by means of (2) and we update accordingly the permeabilities ζ_{LDL} and ζ_{macr}. Then the new block $m+1$ starts. These steps are detailed in Algorithm 1.

Algorithm 1 Numerical solution of the plaque progression coupled problem

```
Let $m$ be the block index, $j$ the heartbeat index, $n$ the short scales time step index, $k$ the long scale
time step index.
  for $m = 1 : M$
      for $k = 1 : K$
          Solve the discretized-in-time cellular problems at block $m$, time instant $k$:
              $c^k_{LDL,m} = \mathcal{DR}^k(TAWSS_{m-1}; \Omega_{s,m-1})$,
              $c^k_{macr,m} = \mathcal{DR}^k(TAWSS_{m-1}, c^k_{LDL,m}, c^{k-1}_{FC,m}; \Omega_{s,m-1})$,
              $c^k_{FC,m} = \mathcal{ODE}^k(c^k_{LDL,m}, c^k_{macr,m}, \Omega_{s,m-1})$;
      end for
      Update the structure domain:
              $\boldsymbol{d}_{G,m} = g_2(c_{FC,m}), \qquad \Omega_{s,m} = \Omega_{s,m-1} + \boldsymbol{d}_{G,m}$;
      for $j = 1 : J$
          for $n = 1 : N$
              Solve the discretized-in-time FSI problem at heartbeat $j$, time instant $n$:
                  $\left(\boldsymbol{u}^{j,n}_m, p^{j,n}_m, \boldsymbol{d}^{j,n}_m\right) = \mathcal{FS}\left(\Omega^{j,n-1}_{f,m-1}, \widehat{\Omega}_{s,m-1}\right)$;
          end for
      end for
      Compute the TAWSS:
                  $TAWSS_m = g_1\left(\boldsymbol{u}_m\right)$.
  end for
```

We discuss in what follows the numerical strategies used for the solution of both the cellular and FSI subproblems.

Regarding cellular problems, we consider BDF1 for time discretization and P1 Finite Elements for space discretization; linearization of the subproblems is performed by an explicit treatment of c_{FC} in the evolution of macrophages, whereas the other non-linearities are solved by the sequential solution of the 3 subproblems, see Algorithm 1.

For the numerical solution of the FSI problem we consider a first order time discretization for fluid, structure and kinematic conditions, with a semi-implicit treatment of the fluid convective term. The fluid geometry problem is determined by means of an harmonic extension of the interface displacement in an *Arbitrary Lagrangian-Eulerian* formulation [6, 9], whereas the geometric coupling is treated explicitly, a strategy which is known to be stable and accurate in hemodynamics, see e.g. [7, 11, 12, 14]. The resulting FSI problem is solved monolithically by means of P2-P1 Finite Elements, with an inexact Newton method given by a block approximation of the Jacobian, leading to the split solution of fluid velocity, pressure and vessel wall unknowns, see [3, 4]. This method has been shown to be highly scalable in the hemodynamic regime.

All the strategies have been implemented in the Finite Elements library *LifeV* (www.lifev.org).

4 Numerical Results

We present in what follow some 3D preliminary numerical results. In all the experiments we considered the linear Hooke law for the structure constitutive relation with Young modulus $E = 3 \cdot 10^5$ Pa and Poisson ratio $\nu = 0.49$. We used the following values for the other physical parameters: $\rho_f = 1.0\,\mathrm{g/cm}^3$, $\rho_s = 1.1\,\mathrm{g/cm}^3$, $\mu = 0.035\,\mathrm{P}$, $D_{LDL} = 1.2 \cdot 10^{-7}\,\mathrm{cm}^2/\mathrm{s}$, $r_{ox} = 0.5 \cdot 10^{-2}\,\mathrm{s}^{-1}$, $c_{LDL,f} = 1.9 \cdot 10^{-3}\,\mathrm{g/cm}^3$, $\zeta_{LDL}^{ref} = 1.7 \cdot 10^{-11}\,\mathrm{cm/s}$, $c_{macr,f} = 5 \cdot 10^{-5}\,\mathrm{g/cm}^3$, $\zeta_{macr}^{ref} = 1.1 \cdot 10^{-12}\,\mathrm{cm/s}$ [1], $D_{macr}^{dis} = 5.0 \cdot 10^{-9}\,\mathrm{cm}^2/\mathrm{s}$, $D_{macr}^{healthy} = 1.0 \cdot 10^{-9}\,\mathrm{cm}^2/\mathrm{s}$ [16]. Moreover, we set $\Delta t = 3\,\mathrm{h}$, $\Delta\tau = 10^{-3}\,\mathrm{s}$, $J = 3$, $N = 800$, $K = 750$ (corresponding to about 3 months). The value of $TAWSS^{ref} = 2.1$ Pa has been estimated by the Poiseuille solution.

At the inlet, we prescribed a physiological representative flow rate taken from [8], whereas at the outlet sections we considered absorbing boundary conditions [13].

In the first experiment, we consider as initial configuration an ideal vessel given by a cylinder of radius 0.5 cm and length 6.5 cm, where a 60% eccentric stenosis of length 1.8 cm has been included 2.5 cm far from the inlet. This leads to recirculation regions downstream the stenosis, leading to low values of TAWSS that should induce the mechanism of LDL, macrophages and foam cells accumulation.

From the results reported in Fig. 4, we observe in fact that, in correspondence of regions of low TAWSS given by the stagnation of blood flow, a growth of about 0.2 mm occurs after block 2, that is after 6 months. These results, although qualitative, show that our methodology is stable from the numerical point of view and that it is able to produce significant plaque growth values which are in the expected ranges.

In the second experiment, we apply the proposed method to a real carotid reconstructed from MRI data. In this case, we start from a healthy geometric condition and we expect that possibly stagnation regions at the bifurcation may lead to plaque growth. This is confirmed by the results reported in Fig. 5, where TAWSS is very low in correspondence of recirculation regions and, accordingly, plaque growth occurs. These results highlight the ability of our method to be applied to real scenarios, allowing to obtain significant results from the quantitative point of view (0.1 mm in 6 months).

These results are preliminary and are the first step towards a concrete application of our method. Of course validation of the method is mandatory. To this aim,

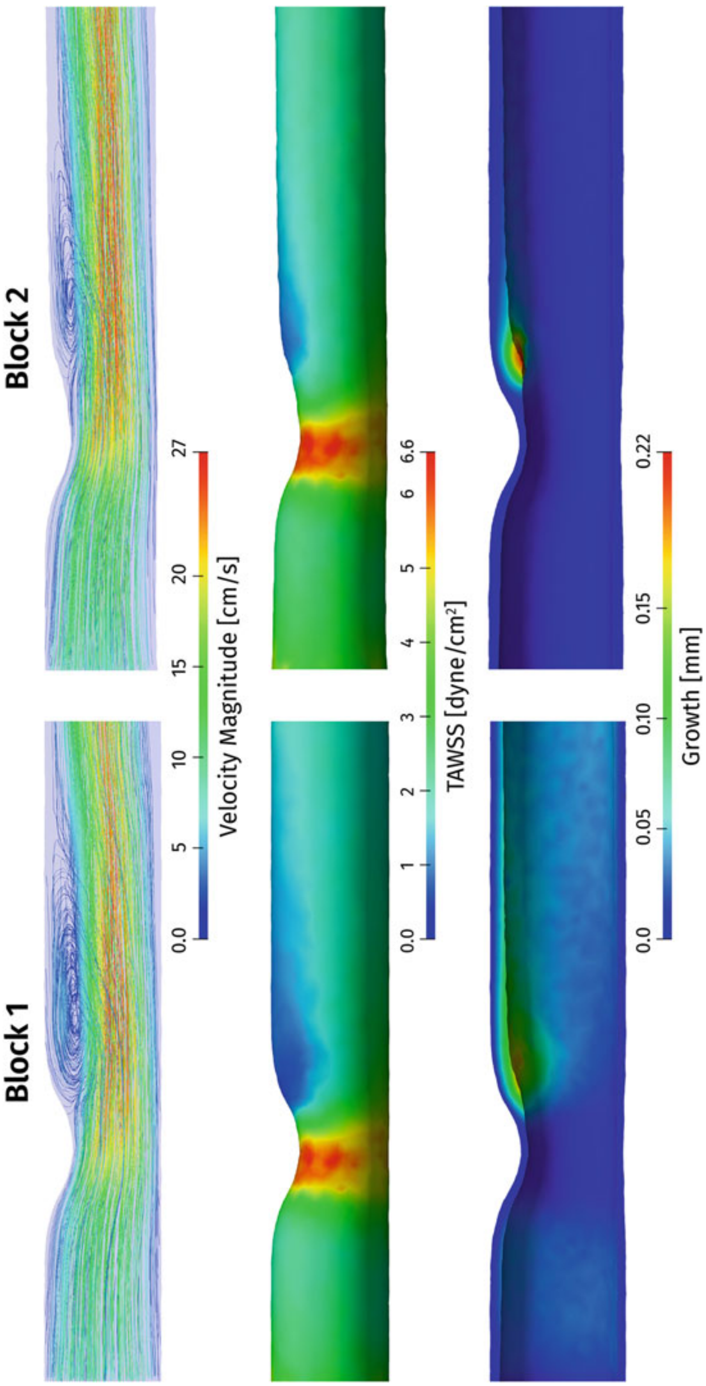


Fig. 4 Top: Streamlines of blood velocity field. Middle: TAWSS. Bottom: Plaque growth. Left: Block 1. Right: Block 2

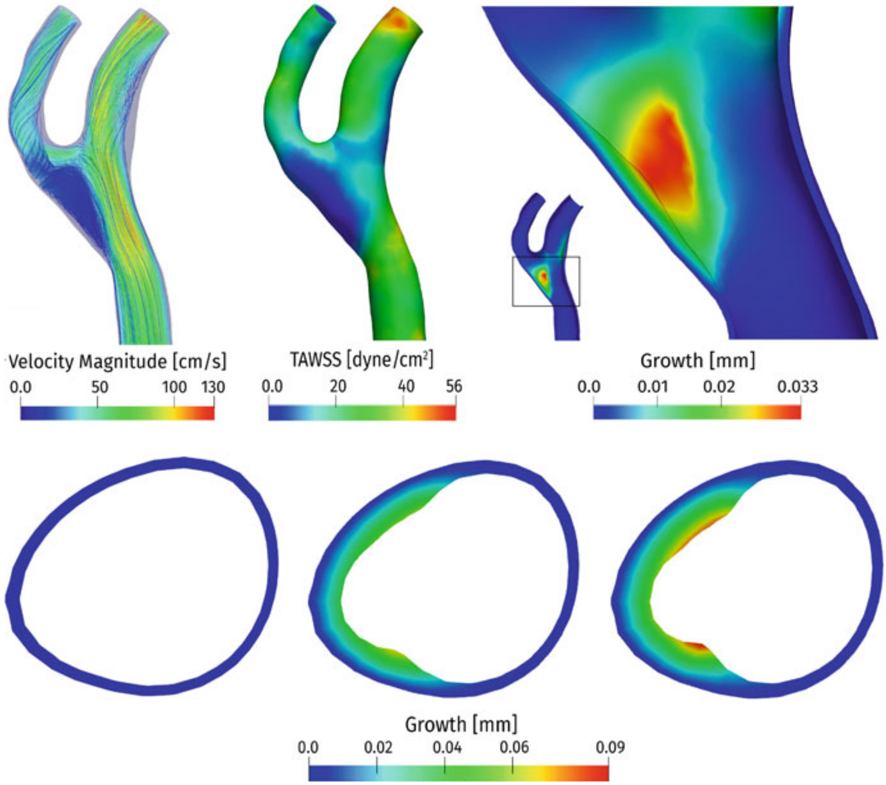


Fig. 5 Top: Streamlines of blood velocity (left), TAWSS (middle), plaque growth (right), block 1. Bottom: Plaque growth on a section at the bifurcation at the starting time instant (left), after block 1 (middle) and after block 2 (right)

comparisons of the results with clinical analysis of patients before and after the plaque progression shall be considered.

References

1. V. Calvez, J.G. Houot, N. Meunier, A. Raoult, and G. Rusnakova. Mathematical and numerical modeling of early atherosclerotic lesions. *ESAIM: Proceedings*, 30, 2010.
2. M. Cilla, E. Peña, and M. Martínez. Mathematical modelling of atheroma plaque formation and development in coronary arteries. *Journal of the Royal Society, Interface/the Royal Society*, 11:20130866, 2014.
3. P. Crosetto, S. Deparis, G. Fourestey, and A. Quarteroni. Parallel algorithms for fluid-structure interaction problems in haemodynamics. *SIAM J. Sci. Comput.*, 33:1598–1622, 2011.
4. S. Deparis, D. Forti, G. Grandperrin, and A. Quarteroni. Facsi: A block parallel preconditioner for fluid-structure interaction in hemodynamics. *Journal of Computational Physics*, (327):700–718, 2016.

5. G. Di Tomaso, V. Diaz-Zuccarini, and C. Pichardo-Almarza. A multiscale model of atherosclerotic plaque formation at its early stage. *IEEE transactions on bio-medical engineering*, 58:3460–3, 2011.
6. J. Donea, S. Giuliani, and J.P. Halleux. An arbitrary Lagrangian-Eulerian finite element method for transient dynamic fluid-structure interactions. *Computer Methods in Applied Mechanics and Engineering*, 33:689–723, 09 1982.
7. M.A. Fernández, J.F. Gerbeau, and C. Grandmont. A projection semi-implicit scheme for the coupling of an elastic structure with an incompressible fluid. *International Journal for Numerical Methods in Engineering*, 69(4):794–821, 2007.
8. B. Guerciotti, C. Vergara, L. Azzimonti, L. Forzenigo, A. Buora, P. Biondetti, and M. Domanin. Computational study of the fluid-dynamics in carotids before and after endarterectomy. *Journal of Biomechanics*, 49, 11 2015.
9. C.W Hirt, A.A Amsden, and J.L Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14:227–253, 03 1974.
10. D. Ku, D. Giddens, C. Zarins, and S. Glagov. Pulsatile flow and atherosclerosis in the human carotid bifurcation - positive correlation between plaque location and low and oscillating shear-stress. *Arteriosclerosis (Dallas, Tex.)*, 5:293–302, 05 1985.
11. F. Nobile, M. Pozzoli, and C. Vergara. Time accurate partitioned algorithms for the solution of fluid-structure interaction problems in haemodynamics. *Computers and Fluids*, 86, 11 2013.
12. F. Nobile, M. Pozzoli, and C. Vergara. Inexact accurate partitioned algorithms for fluid-structure interaction problems with finite elasticity in haemodynamics. *Journal of Computational Physics*, 273:598–617, 2014.
13. F. Nobile and C. Vergara. An effective fluid-structure interaction formulation for vascular dynamics by generalized robin conditions. *SIAM J. Scientific Computing*, 30:731–763, 01 2008.
14. E.W. Swim and P. Seshaiyer. A nonconforming finite element method for fluid-structure interaction problems. *Computer Methods in Applied Mechanics and Engineering*, 195(17–18):2088–2099, 2006.
15. M. Thon, A. Hemmler, A. Glinzer, M. Mayr, M. Wildgruber, A. Zernecke-Madsen, and M. Gee. A multiphysics approach for modeling early atherosclerosis. *Biomechanics and Modeling in Mechanobiology*, 17, 2017.
16. Y. Yang, M. Jager, W. and Neuss-Radu, and T. Richter. Mathematical modeling and simulation of the evolution of plaques in blood vessels. *Journal of mathematical biology*, 72, 2015.
17. T. Zohdi, G. Holzapfel, and S.A. Berger. A phenomenological model for atherosclerotic plaque growth and rupture. *Journal of theoretical biology*, 227:437–43, 2004.

Equilibrium Path Analysis Including Bifurcations with an Arc-Length Method Avoiding A Priori Perturbations

H. M. Verhelst, M. Möller, J. H. Den Besten, F. J. Vermolen, and M. L. Kaminski

Abstract Wrinkling or pattern formation of thin (floating) membranes is a phenomenon governed by buckling instabilities of the membrane. For (post-) buckling analysis, arc-length or continuation methods are often used with a priori applied perturbations in order to avoid passing bifurcation points when traversing the equilibrium paths. The shape and magnitude of the perturbations, however, should not affect the post-buckling response and hence should be chosen with care. In this paper, our primary focus is to develop a robust arc-length method that is able to traverse equilibrium paths and post-bifurcation branches without the need for a priori applied perturbations. We do this by combining existing methods for continuation, solution methods for complex roots in the constraint equation, as well as methods for bifurcation point indication and branch switching. The method has been benchmarked on the post-buckling behaviour of a column, using geometrically non-linear isogeometric Kirchhoff-Love shell element formulations. Excellent results have been obtained in comparison to the reference results, from both bifurcation point and equilibrium path perspective.

H. M. Verhelst (✉)
Delft University of Technology, Department of Applied Mathematics, Delft, The Netherlands

Delft University of Technology, Department of Maritime and Transport Technology, Delft, The Netherlands
e-mail: h.m.verhelst@tudelft.nl

M. Möller · F. J. Vermolen
Delft University of Technology, Department of Applied Mathematics, Delft, The Netherlands
e-mail: m.moller@tudelft.nl; f.j.vermolen@tudelft.nl

J. H. Den Besten · M. L. Kaminski
Delft University of Technology, Department of Maritime and Transport Technology, Delft, The Netherlands
e-mail: henk.denbesten@tudelft.nl; m.l.kaminski@tudelft.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_110

1 Introduction

Linear buckling analysis of (maritime) structures is widely used in engineering to estimate the loads for which instabilities or even collapse will occur. Post-buckling analysis is often considered to assess the load carrying capacity after instability or collapse. For (floating) thin membranes-like offshore solar platforms [14], (post-) buckling analysis involves the wrinkling phenomenon when loads on the membrane exceed critical values [3, 8, 15, 18–20].

When modelling instabilities like wrinkling, a priori perturbations of some shape and magnitude are often applied to initiate post-buckling without passing bifurcation points. Perturbations are required since bifurcation points introduce singularities in the system matrix, meaning that commonly used solution procedures are not able to provide the post-buckling response. However, as previously reported by Taylor et al. [18], the magnitude of the initial perturbations might influence the final solution.

Hence, in this paper our primary focus is to develop a numerical procedure—based on a combination of the conventional and extended arc-length method [5, 6], solution methods for complex roots [12, 24], as well as methods for bifurcation point indication and branch switching [9] (Sect. 2). The performance of the proposed method is illustrated using a benchmark problem (Sect. 3) and conclusions are drawn to complete the work (Sect. 4).

2 The Arc-Length Method

The arc-length method, also known as a path-following algorithm or a continuation method, is a method to advance through a solution space $\mathbf{w}(\mathbf{u}, \lambda)$ of the system

$$\mathbf{G}(\mathbf{u}, \lambda) = \mathbf{N}(\mathbf{u}) - \lambda \mathbf{P} = \mathbf{0}, \tag{1}$$

where $\mathbf{N}(\mathbf{u})$ is a vector function in terms of solution vector $\mathbf{u}$ and $\mathbf{P}$ is a constant vector multiplied by scaling λ. Both $\mathbf{N}$ and $\mathbf{P}$ can follow from a finite element discretization of a system of partial differential equations based on the finite solution vector $\mathbf{u} \in \mathbb{R}^n$. The function $\mathbf{G}$ can thus be used to find the solution $\mathbf{u}$ for a particular scaling λ (i.e. "load control") or vice-versa (i.e. "displacement control"). Alternatively, one can use the function $\mathbf{G}$ and a constraint equation $f(\mathbf{w})$ to find the combination $\mathbf{w} = (\mathbf{u}, \lambda)$ that satisfies $\mathbf{G}(\mathbf{w}) = \mathbf{0}$ and $f(\mathbf{w}) = 0$. This principle is used in the *arc-length method* [4, 16], which will be used to obtain the solution of Eq. (1) in the case that the solution is not known to be unique.

2.1 Conventional and Extended Arc-Length Method

The constraint equation $f(\mathbf{w})$ is often imposed on the solution increment $\Delta\mathbf{w} = (\delta\mathbf{u}_k, \delta\lambda_k)$ and can take different forms, e.g. using Riks' method [16] or Crisfield's method [4]. The latter imposes:

$$f(\mathbf{w}) = \delta\mathbf{u}_k^{\top}\delta\mathbf{u}_k + \Psi^2\delta\lambda_k^2\mathbf{P}^{\top}\mathbf{P} = \Delta l^2. \tag{2}$$

Here, Δl is the arc-length or the radius of the constraint equation, Ψ is a scaling factor to incorporate the dimensionality of the system in the factor λ. The constraint equation of Crisfield was used because this method always finds a solution, despite the curvature of the equilibrium path. The disadvantage, however, is that two solutions are found per iteration, and hence, that a particular solution needs to be selected. Note that the square root of the constraint equation, $\sqrt{f(\mathbf{w})}$, is a proper norm.

Crisfield [4] originally used $\Psi = 0$, referred to as a spherical constraint, but the elliptical constraint is used to maintain displacement and load steps in the same order of magnitude for different refinements:

$$\Psi^2 = \mathbf{u}_0^{\top}\mathbf{u}_0/\lambda_0^2\mathbf{P}^{\top}\mathbf{P}. \tag{3}$$

Here, λ_0 and $\mathbf{u}_0$ correspond to the solutions on a previous equilibrium point (i.e. a converged point). In the origin $\mathbf{w}_0 = (\mathbf{u}_0, \lambda_0) = (\mathbf{0}, 0)$, a slightly different procedure is used [12]. As a consequence of the constraint equation, the system matrix, if banded, loses its banded nature hence affecting convergence behaviour of nonlinear solvers [21]. Therefore, the system of equations is solved in a segregated way. To this extent, Eq. (1) is considered in terms of the unknown increments $\delta\lambda_k$ and $\delta\mathbf{u}_k$ at iteration k, such that

$$\mathrm{K}\delta\mathbf{u}_k = \mathbf{G}(\mathbf{u}, \delta\lambda_k) = \mathbf{N}(\mathbf{u}) - \delta\lambda_k\mathbf{P}. \tag{4}$$

Where the splitting of the incremental displacement $\delta\mathbf{u}_k$ in terms of a standard load-controlled Newton-Raphson method $\delta\bar{\mathbf{u}}_k$ and a component from the increment $\delta\lambda_k$ being $\delta\hat{\mathbf{u}}_k$ is used:

$$\delta\mathbf{u}_k = \beta\delta\bar{\mathbf{u}}_k + \delta\lambda_k\delta\hat{\mathbf{u}}_k. \tag{5}$$

The line-search parameter β is relevant when dealing with complex roots (see Sect. 2.2) and is equal to 1.0 otherwise. Then, for iteration k,

$$\mathrm{K}\delta\bar{\mathbf{u}}_k = \mathbf{G}(\mathbf{w}_k), \tag{6}$$

$$\mathrm{K}\delta\hat{\mathbf{u}}_k = \mathbf{P}. \tag{7}$$

where K is the Jacobian of the system to be solved and has to be computed once. A disadvantage is that no solutions can be found on limit points, since the Jacobian is singular there [6]. At each iteration, the load and displacement increments are updated using

$$\Delta \mathbf{w}_k = (\Delta \mathbf{u}_k, \Delta\lambda_k) = (\delta \mathbf{u}_{k-1}, \delta\lambda_{k-1}) + (\delta \mathbf{u}_k, \delta\lambda_k) \tag{8}$$

Using the constraint equation from Eq. (2) and using the fact that the iterative increment $\delta \mathbf{u}_k$ is depending on the unknown $\delta\lambda_k$, the constraint equation can be written as a polynomial in $\delta\lambda_k$:

$$a\delta\lambda_k^2 + b\delta\lambda_k + c = 0, \tag{9}$$

With,

$$\begin{aligned} a &= \delta\hat{\mathbf{u}}_k^\top \delta\hat{\mathbf{u}}_k + \Psi^2 \mathbf{P}^\top \mathbf{P} = a_0, \\ b &= 2\left(\delta\hat{\mathbf{u}}_k^\top \Delta\mathbf{u} + \Delta\lambda\Psi^2\mathbf{P}^\top\mathbf{P}\right) + 2\beta\delta\hat{\mathbf{u}}_k^\top\delta\bar{\mathbf{u}}_k = b_0 + \beta b_1, \\ c &= \beta^2\delta\bar{\mathbf{u}}_k^\top\delta\bar{\mathbf{u}}_k + 2\beta\delta\bar{\mathbf{u}}_k^\top\Delta\mathbf{u} + \Delta\mathbf{u}^\top\Delta\mathbf{u} + \Delta\lambda^2\Psi^2\mathbf{P}^\top\mathbf{P} - \Delta l^2 \\ &= c_0 + \beta c_1 + \beta^2 c_2. \end{aligned} \tag{10}$$

where $\Delta\mathbf{u} = (\Delta\mathbf{u}, \Delta\lambda)$ (indices omitted) denotes the increment in the previous load step. Since $\mathbf{u}_t$ and $\bar{\mathbf{u}}$ are known from Eqs. (6) and (7), the only unknown in Eq. (9) is the load increment $\delta\lambda$. Therefore, Eq. (9) is a scalar quadratic equation that is easily solved for $\delta\lambda_k$ and has two solutions. The choice of the solution is based on the 'angle' between the arc-length increment $\Delta\mathbf{w}$ of the previous load step and the current $\Delta\mathbf{w}_k$. Since this term is minimised for the increment $\delta\lambda_k$, it is sufficient to look at the following roots [17]:

$$\Theta_r = \delta\lambda_r\left(\Delta\mathbf{u}^\top\delta\hat{\mathbf{u}}_k + \Psi^2\Delta\lambda\right) \quad r = 1, 2. \tag{11}$$

The root $\delta\lambda_r$ for which Θ_r is largest is the selected root. In the original work of Crisfield [4] a different method was proposed, where the increment $\Delta\mathbf{u}_k$ is computed for both values of $\Delta\lambda_r$ and the largest inner-product is taken. Both methods were implemented and no major changes in the robustness of the methods were observed. By comparing the current increment with the previous load increment, both methods are robust as long as no sharp snap-back behaviour is present with respect to the chosen arc-length Δl.

In the first iteration of a new load step, the vector $\delta\mathbf{u}_{k-1}$ and the scalar $\delta\lambda_{k-1}$ are equal to zero. Hence, the trivial solution is found for Eq. (9). Therefore, the following method is used to initialize the method in a new load step. Note that by Eqs. (6) and (7) $\delta\hat{\mathbf{u}}_k$ is non-zero and $\bar{\mathbf{u}}$ is zero since the residual in the first iteration

$\mathbf{G}(\mathbf{w}_0)$ is zero. Therefore, the load increment in the first iteration is defined as [5, 12]:

$$\Delta\lambda_0 = \begin{cases} \Delta l / \sqrt{2\delta\hat{\mathbf{u}}_k^\top \delta\hat{\mathbf{u}}_k}, & \text{if } (\mathbf{u}_0, \lambda_0) = (\mathbf{0}, 0) \\ \Delta l / \sqrt{\delta\hat{\mathbf{u}}_k^\top \delta\hat{\mathbf{u}}_k + \Psi^2 \mathbf{P}^\top \mathbf{P}} & \text{otherwise.} \end{cases} \tag{12}$$

Its sign is determined by the previous load increment $\Delta\mathbf{w}$ [7]:

$$\text{sign}(\Delta\lambda_0) = \text{sign}(\Delta\mathbf{u}^\top \delta\hat{\mathbf{u}}_k + \Delta\lambda\Psi^2 \mathbf{P}^\top \mathbf{P}). \tag{13}$$

2.2 Solution Methods for Complex Roots

In the case of complex roots for Eq. (9), i.e. when $b^2 - 4ac < 0$, the numerical procedure as discussed in the previous section fails [2]. Complex roots occur when the equilibrium path is strongly curved in the region that is covered by one step. As a solution to complex roots, the arc-length can simply be bisected until real roots are found [1] or by utilising a pseudo line-search technique [12, 24]. The methods in the latter works are slightly different in the choice of the line-search parameter as will be detailed later.

As complex roots occur when $b^2 - 4ac < 0$ in Eq. (9), a line-search parameter $\tilde{\beta} \neq 1$ exists such that $b^2 - 4ac \geq 0$ is satisfied. Substitution of the coefficients from Eq. (10) in this condition provides a quadratic equation in terms of the unknown line-search parameter $\tilde{\beta}$:

$$a_s\tilde{\beta}^2 + b_s\tilde{\beta} + c_s \geq 0,$$

with [17]:

$$a_s = b_1^2 - 4a_0c_2, \quad b_s = 2b_0b_1 - 4a_0c_1 \quad \text{and} \quad c_s = b_0^2 - 4a_0c_0,$$

and which can be solved for the equality. When the parameter $\tilde{\beta}$ is obtained, Eq. (9) can again be solved to find the roots for $\delta\lambda_k$. Selection of $\tilde{\beta}$ can be done using $0 < \tilde{\beta} \leq \tilde{\beta}_{\max}$, where $\tilde{\beta}_{\max} = \min(1, \tilde{\beta}_2)$, since the solutions $\tilde{\beta}_{1,2}$ ($\tilde{\beta}_1 < \tilde{\beta}_2$) are of opposite sign and if $\tilde{\beta}$ is between those roots (i.e. if $a_sc_s > 0$), the constraint equation is satisfied. If $\tilde{\beta}$ is close to zero, the iterative method becomes inefficient and it is recommended to cut the arc-length [17, 24].

2.3 Methods for Bifurcation Point Indication and Branch Switching

When applying the arc-length method on buckling analysis, singular points indicate a transition between stability and instability. The singular points can be characterised as either limit points or bifurcation points. The tangential stiffness matrix K is singular, i.e. the determinant of this matrix is equal to zero. Additionally, the first eigenvector $\boldsymbol{\phi}_1$ of the tangential stiffness matrix on a singular point represents the buckling mode shape in case of a bifurcation point. Limit points and bifurcation points are distinguished by considering the inner product $\boldsymbol{\phi}_1^\top \mathbf{P}$. If this product is non-zero, a limit point is found [22].

When passing a singular point, the determinant of this matrix becomes negative, or equivalently, the product of the diagonal entries of the diagonal matrix D of the $\mathrm{LDL}^\top$ Cholesky decomposition changes sign. Unless a bifurcation point is exactly passed—which rarely occurs in practice—the matrix K is symmetric positive-definite. In this case, the $\mathrm{LDL}^\top$ decomposition can be used to factorise and solve Eqs. (6) and (7) and bifurcation points can be pinpointed by considering the sign of the lowest values of the diagonal matrix D. These determine the sign of the determinant of K and thus the stability of the system [21].

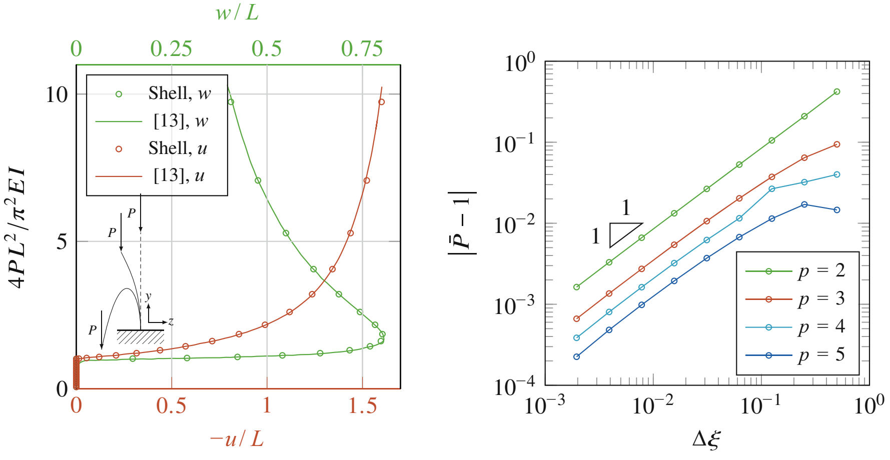
The bifurcation points are approached using the extended arc-length [23], which provides the solution $\mathbf{w}$ and the first eigenvector $\boldsymbol{\phi}_1$ of corresponding to the bifurcation point. This method converges quadratically since it is based on a Newton-Raphson method for solving the equilibrium equations $\mathbf{G}(\mathbf{u}, \lambda) = \mathbf{0}$, the singularity condition $\mathrm{K}(\mathbf{u}, \lambda)\boldsymbol{\phi}_1 = \mathbf{0}$ and a constraint equation to prevent the trivial solution $\boldsymbol{\phi}_1 = \mathbf{0}$ to be found [21–23].

When a bifurcation point $\mathbf{w}_P = (\mathbf{u}_P, \lambda_P)$ is found within a specified tolerance of the extended arc-length method, the eigenvector $\boldsymbol{\phi}_1$ is known from this method and the method can switch to the bifurcation branch by applying perturbation using the buckling mode shape, i.e. using $\boldsymbol{\phi}_1$. Branch switching is simply done by perturbing the displacements $\mathbf{u}_P$ at the bifurcation point by the normalized eigenvector $\bar{\boldsymbol{\phi}}_1$ multiplied by a factor τ. This factor can be chosen arbitrarily small [21].

3 Benchmark Problem

The geometrically linear isogeometric Kirchhoff-Love shell [11] formulation in the open-source Geometry+Simulation Modules (`G+Smo`[1]) [10] are used to model a thin shell. The benchmark is a column, i.e. a beam fixed at one side and loaded in-plane at the other side [13]. The column has length 1 $[m]$, thickness 0.01 $[m]$ and Young's modulus of 75 $[MPa]$. In both models, 32 elements of order 2 over

[1] The source of G+Smo can be found on github.com/gismo.



(a)

(b)

Fig. 1 Deformation of a column subject to a vertical end load. (**a**) Horizontal (u) and vertical (w) displacement (bottom and top axis, resp.) of the end-point versus the applied load. The inset represents the undeformed (dashed) and deformed (solid) configuration. (**b**) Convergence of the present arc-length method to the buckling load, for different knot vector spacings $\Delta\xi$ and B-spline orders p

the length and one element of order 2 in other directions are used. The eigenvector perturbation factor τ is 10^{-3}.

The results obtained with the arc-length method (Fig. 1a) show excellent agreement with the reference results [13] for both bifurcation point prediction and post-buckling behaviour. Furthermore, Fig. 1b shows the convergence of the extended iterations to the buckling point for both models with respect to $\bar{P} = 4\lambda P_{\text{ref}} L^2/\pi^2 EI$, where P_{ref} is the applied reference load. Convergence of the first order to the analytical solution is observed irrespective of the B-spline order. Hence the speed of convergence is not depending on the spline order p, but the magnitude of the error is.

4 Conclusions

In this paper, an arc-length method that does not require a priori perturbations was presented. The procedure is based on the Crisfield arc-length method with extensions for complex roots in the constraint equation for more robustness, and is able to find bifurcation branches without the need for a priori applied perturbations. For benchmarking, the model was applied on buckling and post-buckling analysis of a column with a compressive end load, modelled using isogeometric Kirchhoff-Love shell elements. The benchmark results show that the present method is able to provide accurate results in both path following as well as bifurcation point prediction. In future work, we will apply the present model on modelling wrinkles in thin (supported) sheets subject to large strains for validation and verification with previous studies [3, 8, 15, 18–20].

Acknowledgments The authors are grateful to Delft University of Technology for its support. Additionally, the authors are grateful to the community of the Geometry and Simulation Modules (G+Smo) for laying the basis of the developed code.

References

1. Bellini, P.X., Chulya, A.: An improved automatic incremental algorithm for the efficient solution of nonlinear finite element equations. Computers & Structures **26**(1–2), 99–110 (1987)
2. Carrera, E.: A study on arc-length-type methods and their operation failures illustrated by a simple model. Computers & Structures **50**(2), 217–229 (1994)
3. Cerda, E., Ravi-Chandar, K., Mahadevan, L.: Wrinkling of an elastic sheet under tension. Nature **419**(6907), 579–580 (2002)
4. Crisfield, M.: A Fast Incremental/Iterative Solution Procedure That Handles “Snap-Through”. In: Computational Methods in Nonlinear Structural and Solid Mechanics, pp. 55–62. Pergamon (1981)
5. Crisfield, M.A.: An arc-length method including line searches and accelerations. International Journal for Numerical Methods in Engineering **19**(9), 1269–1289 (1983)
6. Crisfield, M.A.: Non-linear finite element analysis of solids and structures - Volume 1: Essentials. John Wiley & Sons, Ltd (1991)
7. Feng, Y.T., Perić, D., Owen, D.R.J.: Determination of travel directions in path-following methods. Mathematical and Computer Modelling **21**(7), 43–59 (1995)
8. Fu, C., Wang, T., Xu, F., Huo, Y., Potier-Ferry, M.: A modeling and resolution framework for wrinkling in hyperelastic sheets at finite membrane strain. Journal of the Mechanics and Physics of Solids **124**, 446–470 (2019)
9. Fujii, F., Ramm, E.: Computational bifurcation theory: path-tracing, pinpointing and path-switching. Engineering Structures **19**(5), 385–392 (1997)
10. Jüttler, B., Langer, U., Mantzaflaris, A., Moore, S.E., Zulehner, W.: Geometry + Simulation Modules: Implementing Isogeometric Analysis. PAMM **14**(1), 961–962 (2014)
11. Kiendl, J., Bletzinger, K.U., Linhard, J., Wüchner, R.: Isogeometric shell analysis with Kirchhoff–Love elements. Computer Methods in Applied Mechanics and Engineering **198**(49–52), 3902–3914 (2009)
12. Lam, W.F., Morley, C.T.: Arc-Length Method for Passing Limit Points in Structural Calculation. Journal of Structural Engineering **118**(1), 169–185 (1992)
13. Pagani, A., Carrera, E.: Unified formulation of geometrically nonlinear refined beam theories. Mechanics of Advanced Materials and Structures **25**(1), 15–31 (2018)
14. Patterson, B.D., Mo, F., Borgschulte, A., Hillestad, M., Joos, F., Kristiansen, T., Sunde, S., van Bokhoven, J.A.: Renewable CO2 recycling and synthetic fuel production in a marine environment. Proceedings of the National Academy of Sciences of the United States of America **116**(25), 12212–12219 (2019)
15. Pocivavsek, L., Dellsy, R., Kern, A., Johnson, S., Lin, B., Lee, K.Y.C., Cerda, E.: Stress and fold localization in thin elastic membranes. Science (2008)
16. Riks, E.: The Application of Newton’s Method to the Problem of Elastic Stability. Journal of Applied Mechanics **39**(4), 1060 (1972)

17. Ritto-Corrêa, M., Camotim, D.: On the arc-length and other quadratic control methods: Established, less known and new implementation procedures. Computers & Structures **86**(11–12), 1353–1368 (2008)
18. Taylor, M., Bertoldi, K., Steigmann, D.J.: Spatial resolution of wrinkle patterns in thin elastic sheets at finite strain. Journal of the Mechanics and Physics of Solids **62**, 163–180 (2014)
19. Taylor, M., Davidovitch, B., Qiu, Z., Bertoldi, K.: A comparative analysis of numerical approaches to the mechanics of elastic sheets. Journal of the Mechanics and Physics of Solids **79**, 92–107 (2015)
20. Wang, T., Fu, C., Xu, F., Huo, Y., Potier-Ferry, M.: On the wrinkling and restabilization of highly stretched sheets. International Journal of Engineering Science **136**, 1–16 (2019)
21. Wriggers, P.: Nonlinear finite element methods. Springer (2008)
22. Wriggers, P., Simo, J.C.: A general procedure for the direct computation of turning and bifurcation points. International Journal for Numerical Methods in Engineering **30**(1), 155–176 (1990)
23. Wriggers, P., Wagner, W., Miehe, C.: A quadratically convergent procedure for the calculation of stability points in finite element analysis. Computer Methods in Applied Mechanics and Engineering **70**(3), 329–347 (1988)
24. Zhou, Z., Murray, D.W.: An incremental solution technique for unstable equilibrium paths of shell structures. Computers & Structures **55**(5), 749–759 (1995)

Some Mathematical Properties of Morphoelasticity

Ginger Egberts, Daan Smits, Fred Vermolen, and Paul van Zuijlen

Abstract We consider a morphoelastic framework that models permanent deformations. The text treats a stability assessment in one dimension and a preservation of symmetry in multiple dimensions. Next, we treat the influence of uncertainty in some of the field variables onto the predicted behaviour of tissue.

1 Introduction

Growth phenomena are well-studied topic in (medical) biology. Examples are tumor growth, organ development, embryonic growth or the evolution of skin. Organ development is a very interesting research or futuristic scientific development in which one tries to cultivate human and mammalian organs as an alternative to the need of donors for organ transplantation. In many cases, organs from donors will undergo repellence as a result of the immune system of the host. Therefore development of organs on the basis of the DNA from the host is of scientific interest. Further interest comes from modern meat industry in which meat is to be development outside the animal, such that slaughtering animals is no longer necessary. Of course, these topics are still visionary, however, in the future, these topics are expected to gain further research interest, including breakthroughs, and even will be implemented at a certain stage.

G. Egberts · D. Smits · F. Vermolen (✉)
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands
e-mail: F.J.Vermolen@tudelft.nl

P. van Zuijlen
Department of Plastic, Reconstructive and Hand Surgery, MOVE Research Institute, VU University Medical Centre, Amsterdam, The Netherlands

Burn Centre, Red Cross Hospital, Beverwijk, The Netherlands

Department of Plastic, Reconstructive and Hand Surgery, Red Cross Hospital, Beverwijk, The Netherlands

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_111

Mathematical models for growth exist in different levels of complexity, such as growth models that are based on curvature or on surface processes on the boundary of the growing object. Examples are particle growth and phase transitions in grains or the closure of a shallow scrub wound, in which the epidermis (upper skin) grows over the wounded area as a result of localised migration and proliferation of keratinocytes (cells in the epidermis). Other growth processes take place as a result of processes that are happening all over the body of the growing object. In biological applications, one may think of embryonic growth or tumor growth. In all these cases, in-body growth induces mechanical stresses and strains in the body. In large skin wounds, such as serious burn injuries, where skin contraction takes place, the skin may undergo changes such that permanent deformations remain. To deal with these mechanical processes, one composes a balance of momentum and one uses a constitutive law that couples the stresses and strains in the body. If one uses classical elasticity with Hooks's Law, then the deformations will always vanish as the forces are released. Hence growth and/or permanent deformations cannot be predicted with classical mechanical balances only. For this reason, one incorporates growth through morphoelasticity, which was described very clearly by Hall [1] and introduced earlier by Rodriguez et al. [2]. Here, one uses the following principle: the total deformation is decomposed into a deformation as a result of growth and a deformation as a result of mechanical forces. In a mathematical context, one considers the following three coordinate systems: $\mathbf{X}$, $\mathbf{X}_e(t)$, and $\mathbf{x}(t)$, which, respectively, represent the initial coordinate system, the equilibrium at time t that results due to growth or shrinkage, and the current coordinate system that results due to growth or shrinkage and mechanical deformation. The deformation gradient tensor is factorised into $\mathbf{F} = \mathbf{A}\,\mathbf{Z}$: $\mathbf{Z}$ a deformation gradient tensor due to (permanent) growth or shrinkage; and $\mathbf{A}$ a deformation gradient tensor due to (current) mechanical forces.

Another complication that is often encountered in biological systems is the fact that many of the biological variables change from individual to individual. Even changes within the same individual over time and location are not uncommon. These variations, both microscopic (local) and from individual to individual make the biological system suffer from a large degree of uncertainty and therefore many of the biological simulation frameworks should be designed such that they allow the estimation of likelihood that certain scenarios (such as metastasis of tumors or skin contraction after wounding) take place.

As far as we know, the morphoelastic system has not yet been analysed mathematically, and therefore we give some preliminary results for stability and symmetry of the strain tensor. Furthermore, we will show how to quantify the impact of uncertainty in the input parameters on the dynamics of tissue.

2 The Model for Morphoelasticity

Hall [1] derived a set of PDEs that integrate growth/shrinkage with mechanical forces in a two-field formalism for the displacement velocity and the effective Eulerian strain between the current equilibrium configuration and current configuration, based on the deformation gradient tensor $\mathbf{A}$. Let $\frac{D(.)}{Dt}$ denote the material time derivative of a quantify, then we consider the following differential equations for the displacement velocity $\mathbf{v}$ and the effective Eulerian strain $\boldsymbol{\epsilon}$ in an open Lipschitz domain $\Omega(t)$:

$$\begin{aligned} \rho\left(\frac{D\mathbf{v}}{Dt}+\mathbf{v}(\nabla\cdot\mathbf{v})\right)-\nabla\cdot\boldsymbol{\sigma}&=\mathbf{f}, \\ \frac{D\boldsymbol{\epsilon}}{Dt}+\boldsymbol{\epsilon}\,\mathrm{skw}(\mathbf{L})-\mathrm{skw}(\mathbf{L})\,\boldsymbol{\epsilon}+(\mathrm{tr}(\boldsymbol{\epsilon})-1)\mathrm{sym}(\mathbf{L})&=-\mathbf{G}. \end{aligned} \tag{1}$$

Here $\boldsymbol{\sigma}$, $\mathbf{L}$, $\mathbf{G}$, $\mathbf{f}$, respectively, denote the stress tensor, deformation gradient velocity tensor, growth tensor and body force that are given by

$$\begin{aligned} \mathbf{L}=\nabla\mathbf{v},\quad \mathbf{G}&=\alpha\boldsymbol{\epsilon},\qquad \alpha\in\mathbb{R}, \\ \boldsymbol{\sigma}=\mu_1\mathrm{sym}(\mathbf{L})+\mu_2\mathrm{tr}(\mathrm{sym}(\mathbf{L}))\mathbf{I}&+\frac{E}{1+\nu}\left(\boldsymbol{\epsilon}+\frac{\nu}{1-2\nu}\mathrm{tr}(\boldsymbol{\epsilon})\mathbf{I}\right). \end{aligned} \tag{2}$$

Here E, μ_1, μ_2, ν, respectively, represent the Youngs modulus (stiffness), kinematic and dynamic viscosity and Poisson ratio. Further, sym($\mathbf{L}$) and skw($\mathbf{L}$), respectively, denote the symmetric and skew-symmetric part of the tensor L. Equations (1) are solved for $\mathbf{v}$ and $\boldsymbol{\epsilon}$, and need boundary conditions for $\mathbf{v}$ and initial conditions for both $\mathbf{v}$ and $\boldsymbol{\epsilon}$. The displacement is postprocessed by integration of $\mathbf{v}$ over t.

3 Symmetry and Stability

3.1 Symmetry of the Strain Tensor

First, we demonstrate that if the strain tensor $\boldsymbol{\epsilon}$ is initially symmetric then it remains symmetric at all later times.

Theorem 1 *Let the second equation in Eq. (1) hold on open Lipschitz domain Ω for $t > 0$, suppose that $\boldsymbol{\epsilon}$ is symmetric on $t = 0$, then $\boldsymbol{\epsilon}$ remains symmetric for $t > 0$.*

Proof Taking the transpose of the second equation in Eq. (1), gives

$$\begin{aligned}\frac{D\boldsymbol{\epsilon}}{Dt} + \boldsymbol{\epsilon}\ \mathrm{skw}(\mathbf{L}) - \mathrm{skw}(\mathbf{L})\ \boldsymbol{\epsilon} + (\mathrm{tr}(\boldsymbol{\epsilon}) - 1)\mathrm{sym}(\mathbf{L}) &= -\alpha\boldsymbol{\epsilon},\\ \frac{D\boldsymbol{\epsilon}^T}{Dt} + \boldsymbol{\epsilon}^T\ \mathrm{skw}(\mathbf{L}) - \mathrm{skw}(\mathbf{L})\ \boldsymbol{\epsilon}^T + (\mathrm{tr}(\boldsymbol{\epsilon}) - 1)\mathrm{sym}(\mathbf{L}) &= -\alpha\boldsymbol{\epsilon}^T.\end{aligned} \tag{3}$$

Note that we used $\mathrm{sym}(\mathbf{L})^T = \mathrm{sym}(\mathbf{L})$ and $\mathrm{skw}(\mathbf{L})^T = -\mathrm{skw}(\mathbf{L})$, subtraction gives

$$\frac{D}{Dt}(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^T) + (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^T)\ \mathrm{skw}(\mathbf{L}) - \mathrm{skw}(\mathbf{L})\ (\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^T) = -\alpha(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^T). \tag{4}$$

From the above equation, it is clear that $(\boldsymbol{\epsilon} - \boldsymbol{\epsilon}^T) = \mathbf{0}$ represents an equilibrium, and hence symmetry of $\boldsymbol{\epsilon}$ represents an equilibrium, by which we conclude that initial symmetry implies no changes of symmetry for later times. □

The actual stability of this symmetry is another question worth investigating. We postpone this matter to future studies. Symmetry of the strain tensor warrants symmetry of the stress tensor, see Eq. (2), which implies zero torque and hence there is no spin.

3.2 Linear Stability of 1D Morphoelasticity

Next we consider the one-dimensional counterpart of Eqs. (1), which after processing the material time derivative, is given by

$$\begin{aligned}\rho\left(\frac{\partial v}{\partial t} + 2v\frac{\partial v}{\partial x}\right) - \mu\frac{\partial^2 v}{\partial x^2} - E\frac{\partial \epsilon}{\partial x} &= f,\\ \frac{\partial \epsilon}{\partial t} + v\frac{\partial \epsilon}{\partial x} + (\epsilon - 1)\frac{\partial v}{\partial x} &= -G.\end{aligned} \tag{5}$$

The domain is given by $\Omega(t) = (0, 1)$ where we use $v(t, 0) = v(t, 1) = 0$ as boundary conditions, which implies that the domain is fixed and that deformations can only form locally. We analyse stability of constant states in the above one-dimensional problem. To this extent, we analyse perturbations around the equilibria $v = 0$ and $\epsilon = \epsilon_0 \in \mathbb{R}$ for the case that $f = 0$ and $G = 0$. Linearisation of the above equations around these equilibria, gives

$$\rho\frac{\partial \tilde{v}}{\partial t} - \mu\frac{\partial^2 \tilde{v}}{\partial x^2} - E\frac{\partial \tilde{\epsilon}}{\partial x} = 0, \qquad \frac{\partial \tilde{\epsilon}}{\partial t} + (\epsilon_0 - 1)\frac{\partial \tilde{v}}{\partial x} = 0, \tag{6}$$

where $\tilde{v}$ and $\tilde{\epsilon}$ are perturbations around $v = 0$ and $\epsilon = \epsilon_0$. We write the perturbations in terms of a complex Fourier series, that is, we set

$$\tilde{v}(t,x) = \sum_{j=-\infty}^{\infty} c_j^v(t) e^{2i\pi jx}, \qquad \tilde{\epsilon}(t,x) = \epsilon_0 + \sum_{j=-\infty}^{\infty} c_j^{\epsilon}(t) e^{2i\pi jx}, \tag{7}$$

where we are to find coefficients c_j^v and c_j^{ϵ}, and where i represents the imaginary unit number. The use of Fourier Series for stability assessment was also described in, among others, [3]. Substitution into Eqs. (6), gives

$$\begin{aligned} \rho \sum_{j=-\infty}^{\infty} \dot{c}_j^v(t) e^{2i\pi jx} + \mu \sum_{j=\infty}^{\infty} (2\pi j)^2 c_j^v(t) e^{2i\pi jx} - iE \sum_{j=-\infty}^{\infty} (2\pi j) c_j^{\epsilon}(t) e^{2i\pi jx} &= 0, \\ \sum_{j=-\infty}^{\infty} \dot{c}_j^{\epsilon}(t) e^{2i\pi jx} + i(\epsilon_0 - 1) \sum_{j=\infty}^{\infty} (2\pi j) c_j^v(t) e^{2i\pi jx} &= 0. \end{aligned} \tag{8}$$

Orthonormality over $\Omega = (0,1)$, implies after multiplication by $e^{-2i\pi kx}$ and integration over Ω that

$$\begin{aligned} \dot{c}_k^v(t) + \frac{(2\pi k)^2 \mu}{\rho} c_k^v(t) - i\, \frac{2\pi k E}{\rho} c_k^{\epsilon}(t) &= 0, \\ \dot{c}_k^{\epsilon}(t) + i\ 2\pi k(\epsilon_0 - 1) c_k^v(t) &= 0. \end{aligned} \tag{9}$$

The above equations are in the form $y' + Ay = 0$, then $A = \begin{pmatrix} \frac{(2\pi k)^2 \mu}{\rho} & -i\frac{2\pi k E}{\rho} \\ i(\epsilon_0 - 1)2\pi k & 0 \end{pmatrix}$. This matrix has the following eigenvalues

$$\lambda_{\pm} = \frac{(2\pi k)^2 \mu}{2\rho} \pm \frac{1}{2}\sqrt{(\frac{(2\pi k)^2 \mu}{\rho})^2 + 4\frac{(2\pi k)^2 E}{\rho}(\epsilon_0 - 1)}.$$

This implies that linear stability is obtained for $\epsilon_0 \leq 1$, else a saddle point problem is obtained if $\lambda_{\pm} \in \mathbb{R}$. The eigenvalues are real-valued as long as $\mu \geq \frac{\sqrt{\rho E(1-\epsilon_0)}}{\pi}$ $(k = 1)$. The constant case $k = 0$ implies $\lambda_{\pm} = 0$, which reflects the trivial case in which there is no dynamics. This also implies that $\epsilon_0 = 0$ is a stable equilibrium state. Next to this, integration of Eqs. (6) over Ω, gives

$$\begin{aligned} \rho \frac{d}{dt} \int_0^1 \tilde{v} dx &= \left[\mu \frac{\partial \tilde{v}}{\partial x} + E\tilde{\epsilon} \right]_0^1, \\ \frac{d}{dt} \int_0^1 \tilde{\epsilon} dx + (\epsilon_0 - 1) \left[\tilde{v} \right]_0^1 = 0 \Longrightarrow \frac{d}{dt} \int_0^1 \tilde{\epsilon} dx &= 0 \Longrightarrow \int_0^1 \tilde{\epsilon} dx = \epsilon_0. \end{aligned} \tag{10}$$

Note that the boundary conditions $v(0,t) = v(1,t) = 0$ have been used in the second relation of the above equations. The solution $\tilde{\epsilon}$ to Eq. (6) converges towards ϵ_0 under conservation of $\tilde{\epsilon}$ such that $\epsilon_0 = \int_0^1 \tilde{\epsilon}(0,x)dx$. We summarise these results in Theorem 2, where we remark that one easily generalises the observations to a generic fixed domain $\Omega \subset \mathbb{R}$:

Theorem 2 *Let (v, ϵ) satisfy Eqs. (5), under the boundary conditions that $v = 0$ on the boundaries of open, connected domain $\Omega \subset \mathbb{R}$, then*

1. *The equilibria $(v, \epsilon) = (0, \epsilon_0)$, $\epsilon_0 \in \mathbb{R}$, are linearly stable if and only if $\epsilon_0 < 1$;*
2. *Given $\epsilon_0 < 1$, then the eigenvalues are real-valued if and only if $\mu \geq \frac{\sqrt{\rho E(1-\epsilon_0)}}{\pi}|\Omega|$ $(k = 1)$, where $|\Omega|$ denotes the size (measure) of Ω;*
3. *Convergence takes place through $\epsilon_0|\Omega| = \int_\Omega \tilde{\epsilon}(0,x)dx$;*

If $\epsilon_0 < 1$ and if $\mu < \frac{\sqrt{\rho E(1-\epsilon_0)}}{\pi}|\Omega|$ then convergence from perturbations around ϵ_0 will occur in a nonmonotonic way over time due to the fact that the eigenvalues of the linearised dynamical system are not real-valued. Furthermore, if $G = \alpha\epsilon$ for $\alpha > 0$, then the only stable equilibrium is $(v, \epsilon) = (0, 0)$.

4 Computer Simulations

First the numerical method and typical results are briefly explained. This is followed by results from a stochastic stiffness.

4.1 The Numerical Method and Typical Results

The solution to the model equations (1) and (2) is approximated by the finite-element method using linear triangles. Time integration is done by backward Euler in which a monolithic approach is used with Picard inner iterations. In cases that the triangles become ill-shaped, remeshing is applied. In three dimensions, the same is done for linear tetrahedra and bricks. A more detailed treatment is beyond the scope of the current paper, and can be found in [4]. We consider the example of a contracting wound. The results have been shown in Fig. 1, in which the left plot displays the area of a wound that first contracts due to cellular (fibroblast) forces, and subsequently retracts due to the release of cellular forces. In the case of viscoelasticity, it can be seen that the retraction proceeds until the boundaries coincide with the initial boundaries. It can also be seen that morphoelasticity predicts a permanent deformation in the sense that the area of the inflicted region does not converge to the initial configuration. The plot on the right shows how the maximum displacement and the dynamic equilibrium (due to deformation gradient tensor Z) evolves.

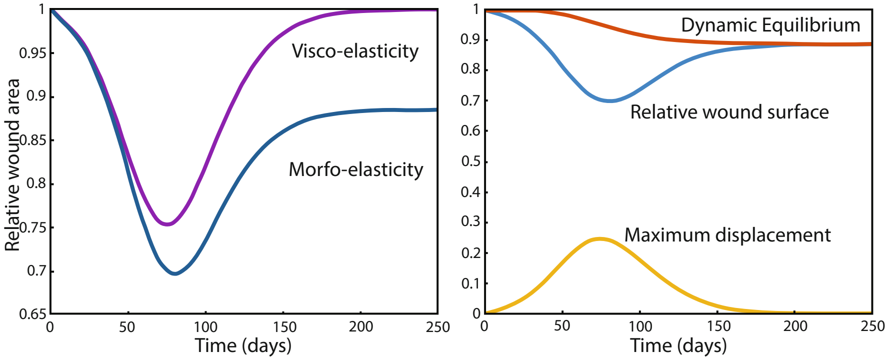


Fig. 1 Left: the relative wound area over time using the viscoelastic approach and morphoelastic approach; Right: the morphoelastic approach with the wound area, equilibrium and maximum displacement as a function of time

4.2 *Quantification of Uncertainty*

Since tissues contain unpredictable spatial microscopic variations, we assume that E, ρ, forcing f and α are random field variables over X consisting of lognormally distributed perturbations around their means. The fields of the aforementioned parameters are obtained through the following truncated Karhunen-Loève expansion over the spatial variable X

$$\hat{u}(X) = \sum_{j=1}^{n} \hat{Z}_j \sqrt{\frac{2}{n}} \sin((2j-1)\frac{\pi}{2L}X), \text{ where } \hat{Z}_j \sim \mathcal{N}(0, 1).$$

Here $\hat{Z}_j$ defines a set of *iid* stochastic variables that follow the standard normal distribution. The stochastic field variable $\hat{u}(X)$ is used to evaluate the field variables E, ρ, f and α. We explain the regeneration procedure for $\hat{E}(X)$:

$$\log(\hat{E}(X)) = \mu_E + \sigma_E \hat{u}(X) \Longrightarrow \hat{E}(X) = \exp(\mu + \sigma \hat{u}(X)),$$

where μ_E and σ_E^2 are the mean (expected value) and variance of $\hat{E}$. The mean and variance are related to the *arithmetic* sample mean $\mathcal{M}$ and *arithmetic* sample standard deviation $\mathcal{S}$ by

$$\mu_E = \ln(\frac{\mathcal{M}^2}{\sqrt{1 + \frac{\mathcal{S}^2}{\mathcal{M}^2}}}), \text{ and } \sigma_E = \sqrt{\ln(1 + \frac{\mathcal{S}^2}{\mathcal{M}^2})}. \tag{11}$$

Figure 2 shows histograms and an estimated cumulative probability distribution for the minimal reduction of area and the final reduction of area after having computed

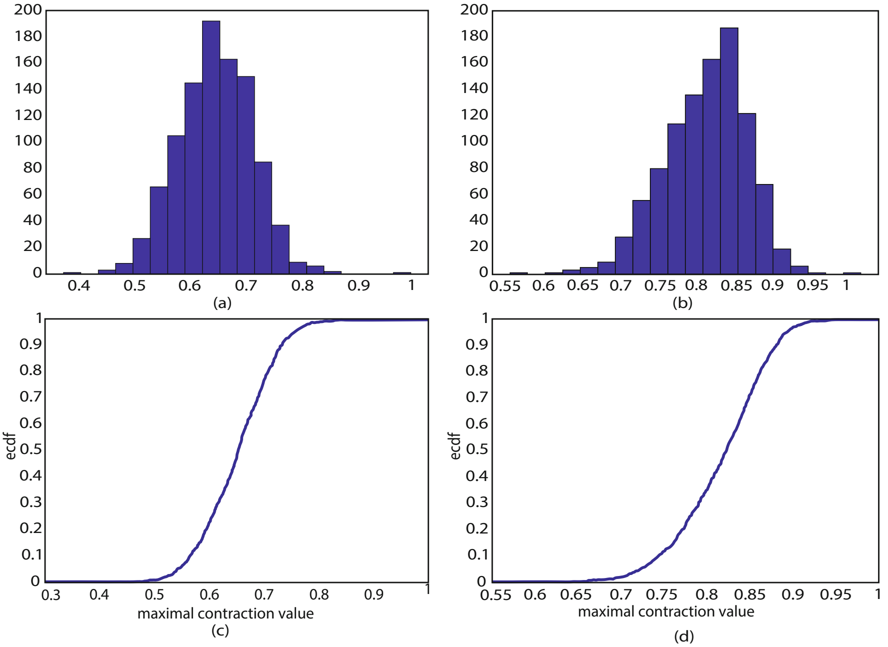


Fig. 2 Results from 1000 samples with $\mathcal{M}_E = 31$ N/(g cm)$^{1/2}$, $\mathcal{S}_E = 11$, $\mathcal{M}_\mu = 10^2$ (N day)/cm, $\mathcal{S}_\mu = 1$, $\mathcal{M}_\rho = 1.02$ g/cm, $\mathcal{S}_\rho = 0.2$, $\mathcal{M}_f = 4$ N/cm, $\mathcal{S}_f = 2$, $\mathcal{M}_\alpha = 0.05$ (−), $\mathcal{S}_\alpha = 0.02$, (**a**) histogram of the maximum wound contraction, that is the minimal wound area; (**b**) histogram of the final contraction, that is the final wound area; (**c**) cumulated probability density for the minimal wound area; (**d**) cumulated probability density for the final wound area

1000 samples. From Fig. 2 the likelihood that the contraction is worse than a certain threshold can be estimated. For instance, from Fig. 2d, the likelihood that the final wound area is smaller than 80% of its original value is about 0.28 (28%).

5 Conclusions

We have shown that morphoelasticity in combination with linear Hooke's Law implies that if the Eulerian effective strain tensor is initially symmetric, then it remains symmetric at all later times. Further, a stability analysis for the one-dimensional case revealed that all Eulerian effective strains smaller than one in combination with zero displacement velocity, represent linearly stable states. Further a condition for monotonicity of convergence over time has been derived. Next to these issues, a Karhunen-Loève expansion has been used for several variables involved to estimate the likelihood that contraction exceed a certain threshold. The model is subject to further uncertainty quantification.

Acknowledgments The authors are grateful for the financial support by the Dutch Burns Foundation under Project 17.105. Furthermore, the suggestions from the referee to improve the presentation of the manuscript are also warmly acknowledged.

References

1. Hall, C.L.: Modelling of some biological materials using continuum mechanics. PhD-thesis at the Queensland University of Technology (2008)
2. Rodriguez E., Hoger A., McCulloch A.: Stress-dependent finite growth in soft elastic tissues. J Biomech 27, 455–467 (1994)
3. Prokharau, P., Vermolen, F.J.: Stability analysis for a peri-implant osseointegration model. J. Math. Biol, 66, 351–382 (2013)
4. Koppenol, D.C., Vermolen, F.J.: Biomedical implications from a morphoelastic continuum model for the simulation of contracture formation in skin-grafts that cover excised burns. Biomech Model Mechanobiol, 16 (4), 1187–1206 (2017)

Approximating Eigenvectors with Fixed-Point Arithmetic: A Step Towards Secure Spectral Clustering

Lisa Steverink, Thijs Veugen, and Martin B. van Gijzen

Abstract We investigate the adaptation of the spectral clustering algorithm to the privacy preserving domain. Spectral clustering is a data mining technique that divides points according to a measure of connectivity in a data graph. When the matrix data are privacy sensitive, cryptographic techniques can be applied to protect the data. A pivotal part of spectral clustering is the partial eigendecomposition of the graph Laplacian. The Lanczos algorithm is used to approximate the eigenvectors of the Laplacian. Many cryptographic techniques are designed to work with positive integers, whereas the numerical algorithms are generally applied in the real domain. To overcome this problem, the Lanczos algorithm is adapted to be performed with fixed-point arithmetic. Square roots are eliminated and floating-point computations are transformed to fixed-point computations. The effects of these adaptations on the accuracy and stability of the algorithm are investigated using standard datasets. The performance of the original and the adapted algorithm is similar when few eigenvectors are needed. For a large number of eigenvectors loss of orthogonality affects the results.

1 Introduction

Computing eigenvectors of matrices has many important applications. One example is principal component analysis, a technique that is used to study large data sets such as those encountered in bioinformatics, data mining, chemical research, psychology,

L. Steverink · M. B. van Gijzen
Delft University of Technology, Delft, The Netherlands
e-mail: M.B.vanGijzen@TUDelft.nl

T. Veugen (✉)
TNO, Unit ICT, The Hague, The Netherlands

Cryptology Department, CWI, Amsterdam, The Netherlands
e-mail: thijs.veugen@tno.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_112

and in marketing. Another example is the characterisation of DNA sequences [17] in bioinformatics. Large graphs have become an important data source for applications from social networks, mobile and web applications to biomedical research, providing great value in both business and scientific research. Particularly, spectral analysis of graphs gives important results pertinent to community detection, PageRank, and spectral clustering.

Especially when the matrix data are sensitive, security measures should be taken to overcome undesired leakage of data during the computation of eigenvectors. The data could be commercially sensitive, but also privacy sensitive, as is often the case with medical data. As data may be collected from different sources, and data processing is increasingly performed in the cloud or by external parties which are not allowed to learn the contents, techniques like data perturbation, homomorphic encryption [10], or secret sharing [1], are frequently used. Unfortunately, such cryptographic techniques are designed to work with integers, whereas the numerical algorithms that are used to compute eigenvectors are designed to work with real numbers. This means that these floating-point based algorithms have to be transformed to fixed-point based algorithms. This has a great influence on the accuracy and stability of the existing, often iterative, approaches.

In this paper, we investigate the effect of approximating eigenvectors with fixed-point arithmetic, and focus on the accuracy and stability of the adjusted numerical algorithms. Although we do not design the complete cryptographic protocols for computing eigenvectors in the encrypted domain, we pay attention to avoid complex operations on encrypted (or secret-shared) numbers, such as square roots and integer divisions [4, 15]. We perform spectral clustering, and compare the results of our adapted numerical algorithms in $\mathbb{Z}_N$ to the original algorithms in $\mathbb{R}$ on three datasets.

The paper is organized as follows. First, related work and preliminaries will be discussed. Then we present the adapted Lanczos algorithm that works on positive integers. Subsequently, the accuracy analysis of secure spectral clustering that makes use of both the original and adapted algorithm, is given. We end with the conclusions.

This paper is based on the research described in [14], which contains many additional algorithmic details and experimental results.

1.1 Related Work

Power methods are known in cryptography for computing square roots or dividing integers [5]. Although they can also be used to find eigenvectors, there is not much previous work done on the numerical analysis of finding eigenvectors in the integer domain. Nikolaenko et al. presented a privacy preserving way of factorising a matrix for recommendation purposes [8], by combining homomorphic encryption and garbled circuits. Erkin et al. designed a secure method for performing k-means clustering [2] by means of additively homomorphic encryption, but this does not require computing eigenvectors. Sharma and Chen [12, 13] recently showed how spectral analysis could be securely done in the cloud, using additively homomorphic encryption and differential privacy. The focus of all related work is on the computational complexity, while we focus on accuracy, with complexity in mind.

1.2 Preliminaries

Spectral Clustering

The spectral clustering algorithm is able to find k, not necessarily convex clusters of similar points by mapping the data points to a k-dimensional space in which the similar points form convex sets. These convex sets can be clustered with a k-means algorithm. In spectral clustering, the dataset is represented as a graph G with weighted edges [16]. We aim to maximize the weights within the clusters, while the weights between clusters are low. A Laplacian matrix L is defined, which contains information about the connected components of G. The first k eigenvectors of Laplacian L approach indicator vectors of the connected components of G, and form convex clusters. Therefore, we are interested in finding the k eigenvectors of L that correspond to the k smallest eigenvalues. The complexity of computing the entire eigendecomposition of $L \in \mathbb{R}^{n \times n}$ is $O(n^3)$. Moreover, if the data set needs to be clustered into k clusters, only k eigenvectors are required. Therefore, we use numerical algorithms to approximate the k smallest eigenvalues and their corresponding eigenvectors.

The Lanczos Algorithm in $\mathbb{R}$

The Lanczos algorithm is used to reduce the Laplacian matrix L to a tridiagonal matrix T (the Ritz matrix) of which the eigenvalues (the Ritz values) approximate the eigenvalues of L. The Lanczos algorithm is shown in Algorithm 1 [3]. The inner product is indicated by a $\cdot$ between two vectors.

Algorithm 1: The Lanczos algorithm

```
Set $v_0 = \underline{0}$ and $\beta_1 = 1$.
Generate a random vector $v_1 \in (0, 1)^n \subset \mathbb{R}^n$.
for $j = 1, 2, \dots, m - 1$ do
    $\alpha_j \leftarrow (Lv_j \cdot v_j)/(v_j \cdot v_j)$
    $r_j \leftarrow Lv_j - \alpha_j v_j - \beta_j v_{j-1}$
    $\beta_{j+1} \leftarrow \|r_j\|_2$
    $v_{j+1} \leftarrow r_j/\beta_{j+1}$
end
$\alpha_m \leftarrow (Lv_m \cdot v_m)/(v_m \cdot v_m)$
```

After m iterations, Algorithm 1 yields Ritz matrix T:

$$T = \begin{pmatrix} \alpha_1 & \beta_2 & & & 0 \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \ddots & \ddots & \ddots & \\ & & \beta_{m-1} & \alpha_{m-1} & \beta_m \\ 0 & & & \beta_m & \alpha_m \end{pmatrix}. \qquad (1)$$

In exact arithmetic, the vectors $v_1, \ldots v_m$ form an orthonormal basis for the so-called Krylov subspace $\mathcal{K}_m(L, v_1)$ of dimension m, which is defined as

$$\mathcal{K}_m(L, v_1) = \text{span}\{v_1, Lv_1, \cdots, L^{m-1}v_1\}.$$

The eigenvalues of T are increasingly better estimates of the eigenvalues of L as its size grows. The extremal Ritz values are the first to converge in the spectrum of T.

Computing in the Integer Domain

Cryptographic techniques are designed to work on positive integers. Therefore, we translate the Lanczos algorithm to $\mathbb{Z}_N$, which is the set $\{0, 1, \ldots, N - 1\}$ with modular arithmetic. Because of security requirements, N is an odd 2048-bit number. The domain $\mathbb{Z}_N$ forms the *message space* of messages that can be encrypted. Modular arithmetic is used on $\mathbb{Z}_N$. Integer division is defined as follows:

Definition 1 Let $a, b \in \mathbb{Z}$. The integer division $a \div b$ is defined as the integer q such that $a = qb + r$ with *remainder* $r \in \mathbb{Z}$, where $0 \leq r < b$.

Fixed-point arithmetic is used to represent fractions as signed integers [4]. By multiplying fractions with 10^d, a signed integer is obtained, where d is the scaling parameter that determines the number of decimals that will be stored. Scaling fractions with 10^d has implications for the operations in the integer domain. To preserve the scaling parameter 10^d when dividing two numbers, the numerator is first multiplied by 10^d. We assume that each integer division on numbers in fixed-point arithmetic has this implicit additional multiplication. Moreover, we define the fixed-point arithmetic multiplication operations as follows:

Definition 2 Let a and b be fixed-point integers. The fixed-point integer multiplication $*$ is defined as

$$a * b = ab10^{-d}.$$

Indeed, multiplying $a10^{-d}$ and $b10^{-d}$ gives $ab10^{-2d} = (a * b)10^{-d}$, so $a * b$ is the scaled version of the product. The operator $*$ is also used to denote fixed-point matrix multiplications. Finally, $\langle v_j, v_j \rangle$ denotes the inner product or a matrix-vector product that makes use of the fixed-point integer multiplication. The following map ψ can encode signed integers (with absolute value less than $N/2$) as positive integers (less than N):

$$\psi : \{-(N-1)/2, \dots, 0, \dots, (N-1)/2\} \longrightarrow \mathbb{Z}_N, \tag{2}$$

$$x \longmapsto x \mod N. \tag{3}$$

Informally stated, the upper half of the domain $\mathbb{Z}_N$ is used to represent the negative integers of maximum bit length 2047. Using these definitions, we adapt the Lanczos algorithm to the integer domain. All computations in this algorithm are performed modulo N. In the algorithm, "mod N" will be omitted.

2 Lanczos Algorithm on Integers

The standard Lanczos algorithm in Algorithm 1 incorporates a normalization of the Lanczos vectors (see line 7). However, the square root operation (within line 6) is expensive in a finite field [7]. Therefore, we propose to perform an unnormalized version of the Lanczos algorithm [9] in the integer domain. Due to this lack of normalization, the entries of v_j tend to grow as the algorithm progresses. Thus, there is a danger of overflow of message space $\mathbb{Z}_N$. The unnormalized Lanczos algorithm in $\mathbb{Z}_N$ is given in Algorithm 2. The entries of starting vector v_1 are chosen randomly from (0, 1) and scaled by 10^d to integers. The Laplacian matrix L contains integer values and is unscaled. Note that this alternative Lanczos algorithm yields an unsymmetric matrix T, because the β_j from Algorithm 1 are now constants:

$$T = \begin{pmatrix} \alpha_1 & \gamma_2 & & & 0 \\ 10^d & \alpha_2 & \gamma_3 & & \\ & 10^d & \alpha_3 & \gamma_4 & \\ 0 & & \ddots & \ddots & \ddots \end{pmatrix}. \tag{4}$$

The above algorithm computes basis vectors $v_1 \cdots v_m$ for the Krylov subspace, and a matrix T_m whose eigenvalues (called Ritzvalues) converge to the eigenvalues of L. Additionally, [14] explains how to use this information to compute the Ritz values and corresponding Ritz vectors (approximating the eigenvectors) in the integer domain.

Algorithm 2: Unnormalized fixed-point Lanczos algorithm in $\mathbb{Z}_N$

```
$v_0 \leftarrow \underline{0}$ and $\beta_1 \leftarrow 0$
$\gamma_1 \leftarrow 0$
Generate a random vector $v_1 \in \{1, \ldots, 10^d\}^n$
for $j = 1, 2, \ldots, m-1$ do
    $L_j \leftarrow \langle L, v_j \rangle$
    $\alpha_j \leftarrow \langle v_j, L_j \rangle \div \langle v_j, v_j \rangle$
    $\beta_{j+1} \leftarrow 1$
    $v_{j+1} \leftarrow L_j - \alpha_j * v_j - \gamma_j * v_{j-1}$
    $\gamma_{j+1} \leftarrow \beta_{j+1} \langle v_{j+1}, v_{j+1} \rangle \div \langle v_j, v_j \rangle$
end
$L_m \leftarrow \langle L, v_m \rangle$
$\alpha_m \leftarrow \langle v_m, L_m \rangle \div \langle v_m, v_m \rangle$
```

3 Accuracy Analysis

In order to investigate the influence of adapting the Lanczos algorithm to the integer domain, the performance of the algorithm in $\mathbb{R}$ and $\mathbb{Z}_N$ is compared. The performance is measured by computing the accuracy of the Ritz values and Ritz vectors, the clustering accuracy and a measure of compactness. The value of N is chosen to comprise 2048 bits. Therefore, we say that overflow occurs when a number becomes larger than 2047 bits, since we need one bit to represent negative numbers. The algorithms were implemented in Python 3.6 and tested on three real datasets. Three datasets from the UCI Machine Learning Repository were used to assess the spectral clustering algorithm in $\mathbb{Z}_N$: the Wisconsin Breast Cancer Dataset, the Yeast5 Dataset and the Yeast10 Dataset [6]. These datasets were chosen for their variety in size and number of clusters. Moreover, a suitable Laplacian could be constructed in the integer domain for these datasets. The Wisconsin Breast Cancer Dataset has size 699 × 9 and should be clustered into two clusters. The Yeast5 Dataset has size 384 × 17 and contains five clusters. Finally, the Yeast10 Dataset is a 1484 × 8 dataset in which ten clusters can be distinguished. Below, we only give the numerical results for the Wisconsin Breast Cancer Dataset. We refer to [14] for a complete description of the numerical results for the other two data sets.

The accuracy of the Ritz value θ_i to eigenvalue λ_i of L is assessed with the absolute error:

$$|\theta_i - \lambda_i|. \tag{5}$$

The accuracy of the corresponding Ritz vector $\tilde{u}_i$ to an eigenvector u_i of L is measured with the absolute cosine of the angle α between the vectors:

$$|\cos(\alpha)| = \frac{|\tilde{u}_i \cdot u_i|}{\|\tilde{u}_i\| \cdot \|u_i\|}. \tag{6}$$

The *silhouette value* is a measure of the compactness and separation of clusters [11]. The distance of the data point to other data points in the same cluster is compared to the distance to data points in other clusters. Formally, the silhouette value of data point i is defined as

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}, \tag{7}$$

where $a(i)$ is the average distance from point i to other points in the same cluster, and $b(i)$ is the minimum average distance from point i to points in a different cluster. The squared Euclidean distance is used in the computation of the silhouette value. From the above definition it follows that

$$-1 \leq s(i) \leq 1 \tag{8}$$

for each data point i. A positive silhouette value indicates that the data point is clustered well. From a negative silhouette value we conclude that a data point has been misclassified.

3.1 Wisconsin Breast Cancer Dataset

A scaling parameter $d = 6$ is required to obtain sufficient accuracy in $\mathbb{Z}_N$. Table 1 shows the relative accuracy of the first two Ritz values. Both in $\mathbb{R}$ and in $\mathbb{Z}_N$ the eigenvalues are approximated well. The accuracy is higher in $\mathbb{R}$. Furthermore, the cosine of the angle between the Ritz vectors and the exact eigenvectors is shown. The values show that the eigenvectors are approximated with high accuracy. Table 2 shows the cluster quality. Both in $\mathbb{R}$ and in $\mathbb{Z}_N$, the first two eigenvectors are approximated well enough to form the correct convex clusters. The maximum entry bit length is 51 in matrix T and 76 in matrix V.

Table 1 The absolute error of the two smallest Ritz values ($\lambda_1 = 2.92700358$ and $\lambda_2 = 9.03710093\text{e}4$) and the accuracy of the corresponding Ritz vectors for the Wisconsin Breast Cancer dataset. Parameters: $d = 6$, $m = 6$

i	$\lvert\theta_i - \lambda_i\rvert\ \mathbb{R}$	$\lvert\theta_i - \lambda_i\rvert\ \mathbb{Z}_N$	$\lvert\cos\alpha\rvert\ \mathbb{R}$	$\lvert\cos\alpha\rvert\ \mathbb{Z}_N$
1	1.3157e−11	1.1549e−4	1.00000000	1.00000000
2	1.5449e−6	2.9566e−5	1.00000000	1.00000000

Table 2 Cluster quality of the Wisconsin Breast Cancer dataset. Parameters: $k = 2$, $d = 6$, $m = 6$

	Lanczos $\mathbb{R}$	Lanczos $\mathbb{Z}_N$
Cluster accuracy	95.85%	95.85%
Silhouette value	0.9118	0.9118

4 Conclusions

We conclude that a few of the smallest eigenvalues of the Laplacian could be approximated well in the integer domain. The accuracy of the algorithm in $\mathbb{R}$ and $\mathbb{Z}_N$ is similar, and the eigenvectors that correspond to the computed eigenvalues are approximated with high accuracy. For a small number of clusters, a good performance of the spectral clustering algorithm is achieved. As a higher number of clusters requires more iterations of the Lanczos algorithm, the loss of orthogonality may affect the accuracy of the spectral clustering algorithm in both domains, see [14].

References

1. Ben-David, A., Nisan, N., Pinkas, B.: FairplayMP - a secure multi-party computation system. In: ACM CCS (2008)
2. Erkin, Z., Veugen, T., Toft, T., Lagendijk, R.L.: Privacy-preserving user clustering in a social network. In: IEEE International Workshop on Information Forensics and Security (2009)
3. Golub, G., Van Loan, C.: Matrix Computations. Johns Hopkins University Press (1996)
4. Hoogh de, S.J.A.: Design of large scale applications of secure multiparty computation: secure linear programming. Ph.D. thesis, Eindhoven University of Technology (2012)
5. Jakobsen, T.: Secure multi-party computation on integers (2006)
6. Lichman, M.: UCI machine learning repository. http://archive.ics.uci.edu/ml (2013)
7. Liedel, M.: Secure distributed computation of the square root and applications. In: International Conference on Information Security Practice and Experience, pp. 277–288. Springer (2012)
8. Nikolaenko, V., Ioannidis, S., Weinsberg, U., Joye, M., Taft, N., Boneh, D.: Privacy-preserving matrix factorization. In: Proceedings of the 2013 ACM SIGSAC conference on Computer and communications security, pp. 801–812. ACM (2013)
9. Paige, C.C.: The computation of eigenvalues and eigenvectors of very large sparse matrices. Ph.D. thesis, University of London (1971)
10. Paillier, P.: Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of Eurocrypt 1999, *Lecture Notes in Computer Science*, vol. 1592, pp. 223–238. Springer-Verlag (1999). citeseer.ist.psu.edu/article/paillier99publickey.html
11. Rousseeuw, P.J.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of computational and applied mathematics **20**, 53–65 (1987)
12. Sharma, S., Chen, K.: Privategraph: a cloud-centric system for spectral analysis of large encrypted graphs. In: IEEE 37th International Conference on Distributed Computing Systems, pp. 2507–2510. IEEE Computer Society (2017)
13. Sharma, S., Powers, J., Chen, K.: Privacy-preserving spectral analysis of large graphs in public clouds. In: Proceedings of the 11th ACM on Asia Conference on Computer and Communications Security, pp. 71–82. ACM (2016)
14. Steverink, M.L.: Secure spectral clustering: the approximation of eigenvectors in the integer domain. Master's thesis, Delft University of Technology (2017). http://resolver.tudelft.nl/uuid:284fc7f2-440d-4435-ae04-fea83d12c12f
15. Veugen, T.: Encrypted integer division and secure comparison. International Journal of Applied Cryptography **3**, 166–180 (2014)
16. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and computing **17**(4), 395–416 (2007)
17. Yu, H.J., Huang, D.S.: Graphical representation for DNA sequences via joint diagonalization of matrix pencil. IEEE Journal of Biomedical and Health Informatics **17**(3), 503–511 (2013)

Modelling Turbulent Combustion Coupled with Conjugate Heat Transfer in OpenFOAM

Mohamed el Abbassi, Domenico Lahaye, and Kees Vuik

Abstract This paper verifies a mathematical model that is developed for the open source CFD-toolbox OpenFOAM, which couples turbulent combustion with conjugate heat transfer. This feature already exists in well-known commercial codes. It permits the prediction of the flame's characteristics, its emissions, and the consequent heat transfer between fluids and solids via radiation, convection, and conduction. The verification is based on a simplified 2D axisymmetric cylindrical reactor. In the first step, the combustion part of the solver is compared against experimental data for an open turbulent flame. This shows good agreement when using the full GRI 3.0 reaction mechanism. Afterwards, the flame is confined by a cylindrical wall and simultaneously conjugate heat transfer is activated and analysed. It is shown that the combustion and conjugate heat transfer are successfully coupled.

1 Introduction

Industrial furnaces such as kilns are pyroprocessing devices in which a heat source is generated via fuel combustion. In order to make a numerical prediction of the temperature distribution along a solid (e.g. the material bed, furnace walls, or heat exchanger), one must model the coupled effects of the occurring physical phenomena. The heat released by the turbulent flame may be transferred to the solid through all heat transfer modes: thermal radiation, conduction, and convection. Thermal radiation is transmitted to the solid directly from the flame, or indirectly from the hot exhaust and other solids. Conduction occurs within solids and through contact with other solid particles, while convective heat may be exchanged via any contact between gas and solids. In return, the fluctuating heat transfer affects the turbulent flow and flame characteristics. Controlling the flame enables achieving

M. el Abbassi · D. Lahaye · K. Vuik (✉)
Delft Institute of Applied Mathematics, Delft University of Technology, Delft, The Netherlands
e-mail: M.elAbbassi@tudelft.nl; D.J.P.Lahaye@tudelft.nl; C.Vuik@tudelft.nl

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_113

the desired heat distribution with minimum emissions. Coupling combustion and heat transfer is essential to find optimal solutions to these conflicting interests, particularly in view of increasing environmental concerns (which view reducing the furnace emissions and fuel consumption as urgent), along with the growing demand for an increase in furnace production rate.

Incorporating the heat transfer between fluids and solids into one mathematical problem may be referred as conjugate heat transfer (CHT). CHT is implemented in many popular CFD codes. Before this project, there were no publications on coupling turbulent combustion and CHT with the open source CFD-toolbox OpenFOAM. OpenFOAM sets a structured object-oriented framework and includes numerous applications to solve different kinds of CFD-related problems.

An implementation was recently proposed and developed for OpenFOAM by Tonkomo LLC [1, 2], that combines the turbulent-non-premixed-combustion solver reactingFoam with the CHT-solver chtMultiRegionFoam. This provides new opportunities for modelling furnaces or any other combustion and heat transfer related problem. In our work, the capabilities of the new solver are investigated by testing it on the 2D axisymmetric case of the open turbulent flame from the Sandia laboratory, by means of RANS simulation.

Our presented work is structured as follows. First the governing equations of the problem are highlighted. We describe the physical models of OpenFOAM that are needed to solve them and how the regions are coupled for energy transport. Afterwards, the cases and their boundary conditions are presented, followed by a discussion of the results.

2 Governing Equations and Numerical Models

In the fluid domain, the Favre-averaged transport equations of mass, momentum, sensible enthalpy and chemical species [3] are respectively described by

$$\frac{\partial(\bar{\rho})}{\partial t} + \nabla \cdot (\bar{\rho}\widetilde{u}) \quad = 0, \tag{1}$$

$$\frac{\partial(\bar{\rho}\widetilde{u})}{\partial t} + \nabla \cdot (\bar{\rho}\widetilde{u}\widetilde{u}) \;= -\nabla\bar{p} + \nabla \cdot \mu\nabla\widetilde{u} \quad - \nabla \cdot \bar{\rho}\widetilde{u''u''}, \tag{2}$$

$$\frac{\partial(\bar{\rho}\widetilde{Y}_\alpha)}{\partial t} + \nabla \cdot (\bar{\rho}\widetilde{u}\widetilde{Y}_\alpha) = \qquad \nabla \cdot \bar{\rho}\Gamma\nabla\widetilde{Y}_\alpha - \nabla \cdot \bar{\rho}\widetilde{Y_\alpha''u''} + \widetilde{R}_\alpha, \tag{3}$$

$$\frac{\partial(\bar{\rho}\widetilde{h})}{\partial t} + \nabla \cdot (\bar{\rho}\widetilde{u}\widetilde{h}) \;= \frac{D}{Dt}\bar{p} + \nabla \cdot \frac{\lambda}{c_p}\nabla\widetilde{h} \;\; - \nabla \cdot \bar{\rho}\widetilde{h''u''} \;\; + \widetilde{Q}_c + \widetilde{Q}_r, \tag{4}$$

where ρ is the density, u the velocity, p the pressure, μ the laminar dynamic viscosity, Y_α the species mass fraction of species α, Γ the species diffusion coefficient, R the reaction rate of species α, h the specific sensible enthalpy, λ the thermal conductivity and c_p the specific heat capacity at constant pressure. The heat source terms Q_c and Q_r are due to combustion and thermal radiation, respectively. The over-bar and tilde notations stand for the average values, while the double quotation marks denote the fluctuating components due to turbulence. Note that several source terms (such as body forces and viscous heating) are neglected.

For solid regions, only the energy transfer needs to be solved and therefore the equation of enthalpy for solids, which is the following heat equation, has to be added to the list of transport Eqs. (1)–(4):

$$\frac{\partial(\bar{\rho}h)}{\partial t} = \nabla \cdot (\lambda \nabla T). \tag{5}$$

To couple the thermal energy transport between the fluid and solid domains, two important conditions are required at the interface of the domains to ensure continuity of both the temperature and heat flux:

$$T_{f,int} = T_{s,int} \tag{6}$$

and

$$\lambda_f \frac{\partial T_f}{\partial y}\bigg|_{int,y=+0} = \lambda_s \frac{\partial T_s}{\partial y}\bigg|_{int,y=-0}, \tag{7}$$

where the subscripts f, s and int respectively stand for fluid, solid and interface. y is the local coordinate normal to the solid. Unclosed terms appear in the transport equations of the fluid domain due to Favre averaging. These will be treated in this section, followed by the elaboration of the heat transfer at the interface.

2.1 Turbulence

The unknown Reynolds stresses (last term of Eq. (2)) are solved by employing the Boussinesq hypothesis that is based on the assumption that in turbulent flows, the relation between the Reynolds stress and viscosity is similar to that of the stress tensor in laminar flows, but with increased (turbulent) viscosity:

$$-\nabla \cdot \bar{\rho}\widetilde{u_i{}''u_j{}''} = \mu_t \left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) - \frac{2}{3}\left(\rho k + \mu_t \frac{\partial u_k}{\partial x_k}\right)\delta_{ij}, \tag{8}$$

where μ_t is the turbulent viscosity and k the turbulent kinetic energy. The Reynolds stresses are closed with the Realizable k-ϵ turbulence model, which is widely known for its superior capability over the Standard and RNG k-ϵ models in predicting the mean of the more complex flow features. The model solves two additional transport equations: one for the turbulent kinetic energy k, and the other for its dissipation rate ϵ

$$\frac{\partial(\bar{\rho}k)}{\partial t} + \nabla \cdot (\bar{\rho}\tilde{u}k) = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\theta_k}\right)\nabla k\right] + \mu_t\left(\frac{\partial u_i}{\partial x_j}\right)^2 - \bar{\rho}\epsilon, \tag{9}$$

$$\frac{\partial(\bar{\rho}\epsilon)}{\partial t} + \nabla \cdot (\bar{\rho}\tilde{u}\epsilon) = \nabla \cdot \left[\left(\mu + \frac{\mu_t}{\theta_\epsilon}\right)\nabla \epsilon\right] + \bar{\rho}c_1 S\epsilon - \bar{\rho}c_2\frac{\epsilon^2}{k + \sqrt{\nu\epsilon}}, \tag{10}$$

where θ_k,θ_ϵ and c_2 are constants. S is the modulus of the mean strain rate tensor, defined as $S = \sqrt{2S_{ij}S_{ij}}$ and c_1 is a function of k, ϵ and S. Again, note that the effect of buoyancy and other source terms are neglected. With k and ϵ, the turbulent viscosity can be determined by the following relation:

$$\mu_t = \bar{\rho}c_\mu\frac{k^2}{\epsilon}, \tag{11}$$

where in the Realizable k-ϵ model, c_μ is a function of k, ϵ, the mean strain rate and the mean rotation rate. This is one of the major differences compared to the other k-ϵ models where c_μ is a constant.

The turbulent scalar fluxes $\bar{\rho}\widetilde{\phi"u"}$ for the scalar chemical species and scalar sensible enthalpy (both denoted as ϕ) are closed with the Gradient diffusion assumption

$$-\bar{\rho}\widetilde{\phi"u"} = \nabla \cdot (\Gamma_t\tilde{\phi}), \tag{12}$$

where Γ_t is the turbulent diffusivity determined by (assuming Lewis number = 1) the turbulent viscosity μ_t and turbulent Prandtl number Pr_t:

$$\Gamma_t = \frac{\mu_t}{Pr_t}. \tag{13}$$

2.2 Combustion

The mean chemical source term $\widetilde{R}_\alpha$ is closed with the Partially Stirred Reactor (PaSR) model. The model developed at Chalmers university allows for the detailed Arrhenius chemical kinetics to be incorporated in turbulent reacting flows. It assumes that each cell is divided into a non-reacting part and a reaction zone that is treated as a perfectly stirred reactor. The fraction is proportional to the ratio of the chemical reaction time t_c to the total conversion time $t_c + t_{mix}$:

$$\gamma = \frac{t_c}{t_c + t_{mix}}. \tag{14}$$

The turbulence mixing time t_{mix} characterizes the exchange process between the reacting and non-reacting mixture, and is determined via the k-ϵ model as

$$t_{mix} = c_{mix}\sqrt{\frac{\mu_{eff}}{\bar{\rho}\epsilon}}, \tag{15}$$

where c_{mix} is a constant and μ_{eff} is the sum of the laminar and turbulent viscosity. Then the mean source term is calculated as $\widetilde{R}_\alpha = \gamma R_\alpha$, where R_α is the laminar reaction rate of species α and is computed as the sum of the Arrhenius reaction rates over the N_R reactions that the species participate in:

$$R_\alpha = \sum_{r=1}^{N_R} \hat{R}_{\alpha,r}, \tag{16}$$

where $\hat{R}_{\alpha,r}$ is the Arrhenius rate of creation/destruction of species α in reaction r. For a reversible reaction, the Arrhenius rate is given by

$$\hat{R}_{\alpha,r} = \psi_{f,r}\prod_{r=1}^{N_R}[C_{\beta,r}]^{\eta'_{\ell,r}} - \psi_{b,r}\prod_{r=1}^{N_R}[C_{\beta,r}]^{\eta''_{m,r}}, \tag{17}$$

where $C_{\beta,r}$ is the concentration of species β in reaction r, $\eta'_{\ell,r}$ is the rate exponent for reactant ℓ in reaction r, $\eta''_{m,r}$ is the stoichiometric coefficient for product m in reaction r, and $\psi_{f,r}$ and $\psi_{b,r}$ are respectively the forward and backward rate constants given by the Arrhenius expressions.

The chemical time scale can be determined with the following relation:

$$\frac{1}{t_c} = max\left(\frac{-\partial R_\alpha}{\partial Y_\alpha}\frac{1}{\bar{\rho}}\right). \tag{18}$$

2.3 Energy

The thermal conductivity λ in the averaged transport of the specific sensible enthalpy (Eq. (4)) is replaced by the effective conductivity λ_{eff}, which incorporates the unknown turbulent scalar flux. From Eqs. (12) and (13), λ_{eff} is defined by the Standard and Realizable k-ϵ models as

$$\lambda_{eff} = \frac{\mu}{Pr} + \frac{\mu_t}{Pr_t}, \tag{19}$$

where the turbulent Prandtl number, from experimental data, has an average value of 0.85. The heat release due to combustion $\widetilde{Q_c}$ follows from the calculations of $\widetilde{R_\alpha}$

$$\widetilde{Q_c} = -\sum_{\alpha=1}^{N} \Delta h^o_{f,\alpha} \widetilde{R_\alpha}, \tag{20}$$

where $\Delta h^o_{f,\alpha}$ is the formation enthalpy of species α, and N is the total number of species.

2.4 Thermal Radiation

To obtain the mean radiation source term $\widetilde{Q_r}$ for the enthalpy transport equation, we employ the P1 approximation in accordance with the previous work, which solves the following partial differential equation for a non-scattering medium

$$-\nabla \cdot \left(\frac{1}{3\kappa} \nabla G \right) = \kappa (4\sigma T^4 - G), \tag{21}$$

where the radiation source term appears on the LHS of the equation. G is the total incident radiation, κ is the absorption coefficient of the medium and σ is the Stefan–Boltzmann constant. The P1 approximation is subject to the following boundary condition of the third kind

$$-\frac{1}{3\kappa} \mathbf{n} \cdot \nabla G = -\frac{\kappa_w}{2(2-\kappa_w)} (4\sigma T_w^4 - G_w). \tag{22}$$

The absorption coefficients of the gas mixture (κ) is determined using a built-in gray gas model, whereas the wall has a constant value κ_w of 0.6. Adding radiation to the problem alters the interface condition (Eq. (7)) to

$$\lambda_{eff} \frac{\partial T_f}{\partial y}\bigg|_{int,y=+0} + q_{r,in} - q_{r,out} = \lambda_s \frac{\partial T_s}{\partial y}\bigg|_{int,y=-0}, \tag{23}$$

where $q_{r,in}$ is the incident radiative heat flux absorbed by the solid and $q_{r,out}$ is the reflected and emitted radiative heat flux leaving the solid.

Table 1 Boundary and initial conditions for Sandia Flame D. zG stands for the Neumann boundary condition zeroGradient. The axial-velocities are expressed in m/s, and the temperatures in K. Species are denoted in mass fractions

Variable	Fuel jet	Pilot jet	Co-flow	Gas-wall interface	Outer wall surface	Side wall surfaces
U_{axial} (m/s)	49.6	11.4	0.9	0	–	–
T (K)	294	1880	291	Coupled	291	zG
Y_{CH_4}	0.1561	0	0	zG	–	–
Y_{O_2}	0.1966	0.054	0.23	zG	–	–
Y_{N_2}	0.6473	0.742	0.77	zG	–	–
Y_{H_2O}	0	0.0942	0	zG	–	–
Y_{CO_2}	0	0.1098	0	zG	–	–

3 Numerical Set-Up

3.1 Test Cases

The solver is tested on two methane-air combustion cases. In the first case, the implementation of combustion in the new solver is validated with experimental data from a turbulent piloted diffusion flame from the Sandia National Laboratories (Sandia Flame D). The burner dimensions can be found here [4].

For the second case, CHT is incorporated and the Sandia Flame D is confined by a cylindrical wall made of refractory material, with inner and outer diameters of respectively 300 and 360 mm. The axial length of the calculation domain (excluding fuel and pilot channels) is 600 mm. The boundary conditions of the two cases can be found in Table 1. The wall has the following thermal properties: a density of 2800 kg m^{-3}, a thermal conductivity (λ_s) of 2.1 W m^{-1} K^{-1}, a specific heat capacity (c_p) of 860 J kg^{-1} K^{-1} and a radiative emissivity (κ_s) of 0.6 m^{-1}.

The computational domains of cases 1 to 2 consist of respectively 38,000 and 43,000 quadrilateral cells.

4 Results and Discussion

4.1 Case 1

In Fig. 1 the temperature along the axis of symmetry is plotted. It shows that the multiRegionReactingFoam's prediction is identical to that of reactingFoam, as would be expected when CHT is switched of. Both solvers over-predict the ignition delay, temperature rise and peak temperature with the 2-step reaction mechanism. When using the full GRI reaction mechanism, these features are better captured and show good agreement.

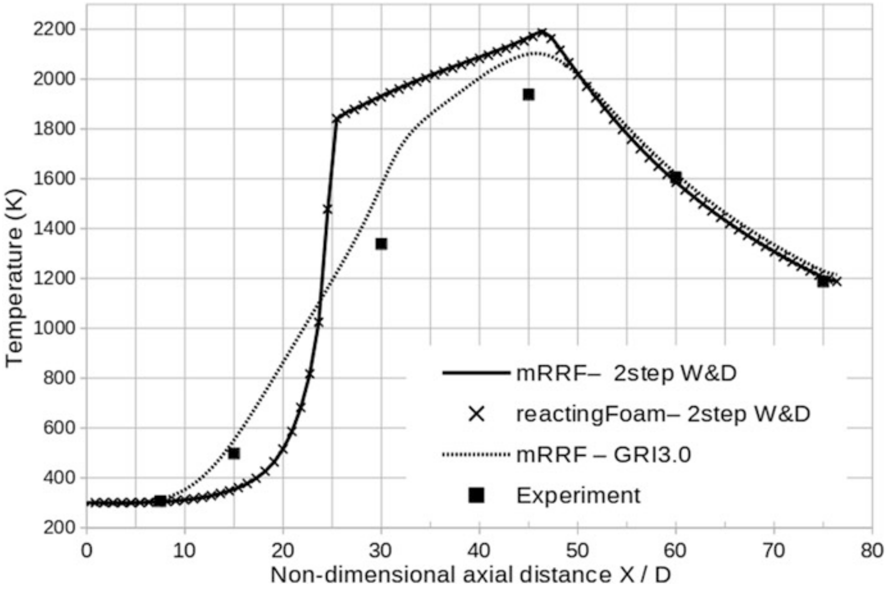


Fig. 1 Temperature progression along the centre line (Case 1)

4.2 Case 2

Now that a wall is introduced around the Sandia Flame D, it absorbs some of the energy, as can be seen in Fig. 2. Figure 3 shows a decomposition of the heat transfer to the wall in which the wall is being heated only due to thermal radiation. The wall is not heated via convection due to the fact that the hot gas heated by the flame leaves the domain before coming into contact with the wall. In fact, the convective heat transfer part plays a cooling role by transferring some of the wall's heat to the cold adjacent air, hence the negative contribution.

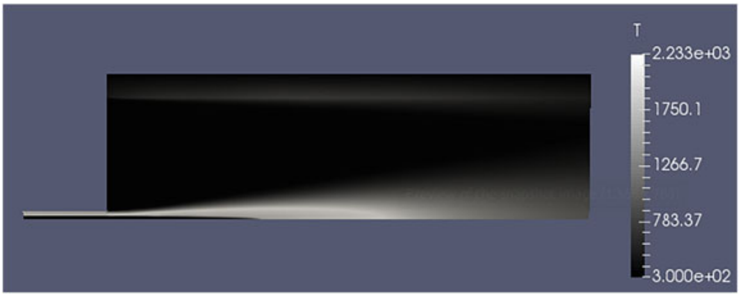


Fig. 2 Contour plot of the temperature (Case 2)

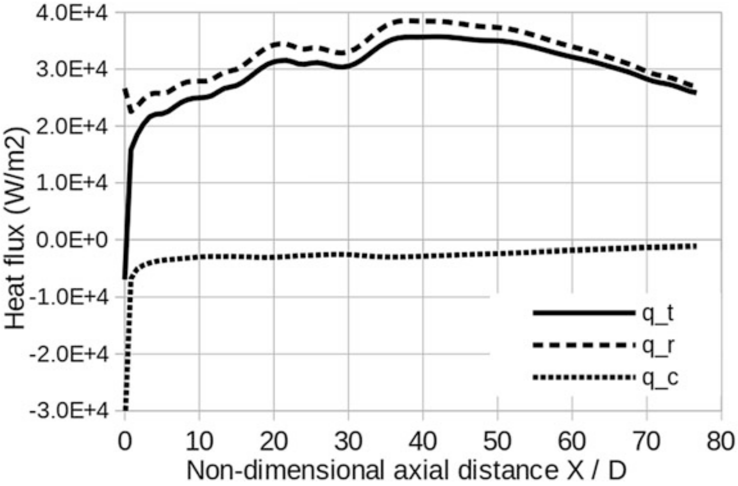


Fig. 3 Heat flux along the inner wall surface (Case 2). q_t, q_r and q_c are respectively the total, radiative and convective heat fluxes

5 Conclusions

This work has shown that OpenFOAM's standard solvers reactingFoam and chtMultiRegionFoam are successfully implemented in the new solver multiRegionReactingFoam. This enables the modelling of combustion with conjugate heat transfer. The results of the new solver, with conjugate heat transfer turned off, are identical to reactingFoam and good agreement is shown with experiments when using the full GRI mechanism. Also the flame-wall interaction is shown when enabling conjugate heat transfer. This still requires to be validated.

Acknowledgments The authors would like to thank Eric Daymo from Tonkomo LLC for developing the solver multiRegionReactingFoam and for his collaboration on debugging the solver in order to make it more robust.

References

1. Source code of chtMultiRegionReactingFoam. https://github.com/TonkomoLLC. Accessed: 2017-01-15.
2. E.A. Daymo and M. Hettel. Chemical reaction engineering with DUO and chtMultiRegionReactingFoam. 4th OpenFOAM User Conference 2016, Cologne-Germany, 2016.
3. T. Poinsot and D. Veynante. *Theoretical and Numerical Combustion*. R.T. Edwards, Inc., Philadelphia, 2nd edition, 2005.
4. Sandia flame D test description on the ERCOFTAC QNET-CFD wiki forums. http://qnet-ercoftac.cfms.org.uk/w/index.php/Description_AC2-09. Accessed: 2016-07-01.

Higher Order Regularity Shifts for the p-Poisson Problem

Anna Kh. Balci, Lars Diening, and Markus Weimar

Abstract We discuss new local regularity estimates related to the p-Poisson equation $-\operatorname{div}(A(\nabla u)) = -\operatorname{div} F$ for $p > 2$. In the planar case $d = 2$ we are able to transfer local interior Besov and Triebel-Lizorkin regularity up to first order derivatives from the forcing term F to the flux $A(\nabla u) = |\nabla u|^{p-2}\,\nabla u$. In case of higher dimensions or systems we have a smallness restriction on the corresponding smoothness parameter. Apart from that, our results hold for all reasonable parameter constellations related to weak solutions $u \in W^{1,p}(\Omega)$ including quasi-Banach cases with applications to adaptive finite element analysis.

1 Introduction

In this paper, for $1 < p < \infty$ and $F \in L^{p'}(\Omega)$ with $1/p + 1/p' = 1$, we consider weak solutions $u \in W^{1,p}(\Omega)$ to the p-Poisson equation

$$-\operatorname{div}(A(\nabla u)) = -\operatorname{div} F \quad \text{in} \quad \Omega, \tag{1}$$

where $\Omega \subset \mathbb{R}^d$ is a domain in dimension $d \geq 2$ and

$$A(Q) := |Q|^{p-2}\,Q.$$

In fact, in what follows we focus on scalar solutions in the super-linear case $p \geq 2$ with special attention to planar domains. Equations of this type play a role in many

A. Kh. Balci · L. Diening
University Bielefeld, Bielefeld, Germany
e-mail: akhripun@math.uni-bielefeld.de; lars.diening@uni-bielefeld.de

M. Weimar (✉)
Ruhr University Bochum, Bochum, Germany
e-mail: markus.weimar@rub.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_114

applications such as, e.g., in non-Newtonian fluid theory, non-Newtonian filtering, turbulent flows of gas in porous media, rheology, or radiation of heat.

By now, it is fairly well-known that the performance of numerical solvers for operator equations like (1) is intimately related to the maximal smoothness of the true solution u in certain scales of (quasi-) Banach spaces of functions or distributions which generalize the scale of Sobolev Hilbert spaces H^s; see, e.g., [6, 8, 10, 13]. For the general theory of such Besov and Triebel-Lizorkin type function spaces we refer to [20–22]. However, let us stress the fact that these scales include several classical function spaces such as, e.g., Hölder-Zygmund, Bessel-potential, or Sobolev-Slobodeckij spaces, as special cases [19]. Therefore, an extensive regularity analysis of the p-Poisson equation in these scales is of interest in its own right.

Although the non-linear case $p \neq 2$ is much harder than the ordinary Poisson problem, in the last decades several regularity results for solutions to (1) have been derived [2, 4, 5, 7, 9, 11, 12, 14–18, 24]. Anyhow, these papers do not contain shift theorems in *general* Besov or Triebel-Lizorkin spaces of higher order. The following results partially fill this gap. The presentation is based on [3].

2 Main Results

Before we can state our results we have to introduce some notation: For balls $B \subset \Omega$, we denote by $\mathbf{B}^s_{\varrho,q}(B)$ and $\mathbf{F}^s_{\varrho,q}(B)$, the Besov space, resp. Triebel-Lizorkin space, of functions or distributions on B with differentiability $s > 0$, integrability $0 < \varrho \leq \infty$, and fine index $0 < q \leq \infty$ (with $\varrho < \infty$ for the $\mathbf{F}$-scale). We use $\|\cdot\|_{\mathbf{B}^s_{\varrho,q}(B)}$ to denote the corresponding (quasi-) norm and $|\cdot|_{\mathbf{B}^s_{\varrho,q}(B)}$ for the (quasi-) semi norm describing the part of the s-order derivatives. Likewise we do for the $\mathbf{F}$-scale. As usual, we let $(x)_+ := \max\{0, x\}$ for $x \in \mathbb{R}$. Moreover,

$$\langle g \rangle_M := \fint_M g(x)\,\mathrm{d}x, \qquad g \in L^1_{\mathrm{loc}}(\mathbb{R}^d),$$

defines the mean value of g over $M \subset \mathbb{R}^d$, where $\fint_M \ldots \,\mathrm{d}x := |M|^{-1} \int_M \ldots \,\mathrm{d}x$ denotes the average integral with $|M|$ being the volume of M. We write $f \lesssim g$ if there exists a constant such that $f \leq c\, g$. Finally, we use $f \sim g$ if $f \lesssim g$ and $g \lesssim f$.

Our main result is the following local regularity transfer from F to $A(\nabla u)$, stated here for $d = 2$ (for an extension to higher dimensions see Sect. 3 below):

Theorem 1 (Regularity Shift) *Given $2 \leq p < \infty$, a domain $\Omega \subset \mathbb{R}^2$ in $d = 2$, and $F \in L^{p'}(\Omega)$ let $u \in W^{1,p}(\Omega)$ be a (scalar) weak solution to (1). Further, let $s > 0$ and $\varrho, q \in (0, \infty]$ be such that*

$$d\left(\frac{1}{\varrho} - \frac{1}{p'}\right)_+ < s < 1. \tag{2}$$

Then for any ball B with $2B \subset \Omega$ there holds

$$|A(\nabla u)|_{\boldsymbol{B}^s_{\varrho,q}(B)^d} \lesssim |F|_{\boldsymbol{B}^s_{\varrho,q}(2B)^d} + \left(\fint_{2B} |A(\nabla u) - \langle A(\nabla u)\rangle_{2B}|^{p'} \,\mathrm{d}x \right)^{1/p'}. \tag{3}$$

If additionally $\varrho < \infty$ and

$$d\left(\frac{1}{q} - \frac{1}{p'}\right)_+ < s < 1,$$

then the same estimate (3) holds true when $\boldsymbol{B}^s_{\varrho,q}$ is replaced by $\boldsymbol{F}^s_{\varrho,q}$.

Note that condition (2) ensures the compact embedding $\mathbf{B}^s_{\varrho,q}(B) \hookrightarrow\hookrightarrow L^{p'}(B)$ which particularly allows to characterize these spaces of smoothness $s < 1$ in terms of lowest order local oscillations

$$\begin{aligned} \operatorname{osc}^B_w f(x,t) &:= \inf_{c\in\mathbb{R}} \left(\fint_{B_t(x)\cap B} |f - c|^w \,\mathrm{d}x \right)^{1/w}, \\ &\sim \left(\fint_{B_t(x)\cap B} \left|f - \langle f\rangle_{B_t(x)\cap B}\right|^w \,\mathrm{d}x \right)^{1/w}, \; 1 \le w \le p',\, t > 0,\, x \in B, \end{aligned}$$

where $B_t(x) := \left\{ y \in \mathbb{R}^d \,\middle|\, |y - x| < t \right\}$. Indeed, under these conditions we have

$$\mathbf{B}^s_{\varrho,q}(B) = \left\{ g \in L^{\max\{\varrho,p'\}}(B) \,\middle|\, \|g\| := \left\| g \,\middle|\, L^\varrho(B) \right\| + |g|_{\mathbf{B}^s_{\varrho,q}(B)} < \infty \right\}$$

in the sense of equivalent (quasi-) norms with

$$|g|_{\mathbf{B}^s_{\varrho,q}(B)} := \left(\int_0^1 \left[t^{-s} \left\| \operatorname{osc}^B_w g(\cdot,t) \,\middle|\, L^\varrho(B) \right\| \right]^q \frac{\mathrm{d}t}{t} \right)^{1/q}$$

(appropriately modified if $q = \infty$); see, e.g., [21, Thm. 5.2.1]. Similar results remain true in the $\mathbf{F}$-scale, as well as for vector-valued spaces.

For our proof of Theorem 1 we combine these characterizations with the subsequent fundamental oscillation decay estimate for $A(\nabla u)$ which is of independent interest. For details we refer to [3, Sect. 3 and 4].

Theorem 2 (Oscillation Decay) *Let the assumptions of Theorem 1 be satisfied. Then for all $\beta \in (0, 1)$ there exists some $\theta_0 \in (0, 1)$ and $c = c(\beta, \theta_0) > 0$ such that for all balls $B \subset \Omega$ there holds*

$$\begin{aligned} &\left(\fint_{\theta_0 B} \left|A(\nabla u) - \langle A(\nabla u)\rangle_{\theta_0 B}\right|^{p'} \,\mathrm{d}x \right)^{\frac{1}{p'}} \\ &\quad\le \theta_0^\beta \left(\fint_B |A(\nabla u) - \langle A(\nabla u)\rangle_B|^{p'} \,\mathrm{d}x \right)^{\frac{1}{p'}} + c \left(\fint_B |F - \langle F\rangle_B|^{p'} \,\mathrm{d}x \right)^{\frac{1}{p'}}. \end{aligned}$$

3 Further Results on Oscillations and Open Questions

In this section, we collect several oscillation decay estimates which are of interest in their own.

First of all we can iterate the bound from Theorem 2 in order to obtain an oscillation decay for arbitrary reduction factors $\theta \in (0, 1)$ by means of standard estimates. The price to pay is an additional constant factor.

Corollary 1 *Let u, p, F, and β be as in Theorem 2. Then for all $\theta \in (0, 1)$ and all balls $B_t(x) \subset \Omega$ there holds*

$$\operatorname{osc}^{\Omega}_{p'} A(\nabla u)(x, \theta t) \lesssim \theta^{\beta} \operatorname{osc}^{\Omega}_{p'} A(\nabla u)(x, t) + \theta^{\beta} \int_{\theta}^{1} \lambda^{-\beta} \operatorname{osc}^{\Omega}_{p'} F(x, \lambda t) \frac{\mathrm{d}\lambda}{\lambda}.$$

The proof of Theorem 2 above relies (at least partially) on a local comparison of u to some p-harmonic function h (which satisfies certain boundary conditions). That is, $h \in W^{1,p}_{\mathrm{loc}}(\Omega)$ solves

$$-\operatorname{div} A(\nabla h) = 0 \tag{4}$$

in the distributional sense. Regularity studies for such functions date back at least for 50 years. Our contribution to this topic is the following almost linear osc_1-decay for $A(\nabla h)$; cf. [3, Thm. 2.2]. The basic idea of its proof can be traced back to the seminal paper of Uhlenbeck [23] that splits the analysis into two regimes: In the so-called *non-degenerate* case, ∇h (locally) is close to a constant such that the problem behaves like a linear equation with constant coefficients. In contrast, in the remaining *degenerate* regime, we have to deal with a fully non-linear problem.

Theorem 3 (Decay for p-Harmonic Functions I) *Given $2 \le p < \infty$, let $h \colon \Omega \to \mathbb{R}$ be p-harmonic on $\Omega \subset \mathbb{R}^d$ with $d = 2$. Then for all $\beta \in (0, 1)$ there exists $c_\beta > 0$ such that for all balls $B_t(x) \subset \Omega$ and every $\theta \in (0, 1)$ there holds*

$$\operatorname{osc}_1 A(\nabla h)(x, \theta t) \le c_\beta \, \theta^{\beta} \operatorname{osc}_1 A(\nabla h)(x, t).$$

It is clear that Theorem 3 is optimal in the sense that oscillations can never decay faster than linear. However, the limiting case $\beta = 1$ of linear decay is excluded by our method of proof. Therefore, we raise the question:

Question 1 Does Theorem 3 also hold with $\beta = 1$?

Further, it can be shown easily that the result of Theorem 3 does not extend to the case $1 < p < 2$. In this regime, the natural object to look at is ∇h rather than

$A(\nabla h)$. Indeed, using duality in the sense of differential forms, we can show the following analogue of Theorem 3, see [3, Thm. 2.3].

Theorem 4 (Decay for p-Harmonic Functions II) *Given $1 < p \leq 2$, let $h \colon \Omega \to \mathbb{R}$ be p-harmonic with on $\Omega \subset \mathbb{R}^d$ with $d = 2$. Then for all $\beta \in (0,1)$ there exists $c_\beta > 0$ such that for all balls $B_t(x) \subset \Omega$ and every $\theta \in (0,1)$ there holds*

$$\operatorname{osc}_1 \nabla h(x, \theta t) \leq c_\beta\, \theta^\beta \operatorname{osc}_1 \nabla h(x,t).$$

Again this assertion does not extend to the super-linear regime $p > 2$ and it remains open what happens if $\beta = 1$. However, it seems that a unified statement, for the full parameter range $1 < p < \infty$, might be possible in terms of the related vector field $V(\nabla h)$, where

$$V(Q) := |Q|^{\frac{p-2}{2}}\, Q.$$

Indeed, there are good reasons to conjecture that the following question has an affirmative answer.

Question 2 For $d, n \in \mathbb{N}$ and $1 < p < \infty$ let $h \colon \Omega \to \mathbb{R}^n$ be a p-harmonic function on $\Omega \subset \mathbb{R}^d$. Is it true that $V(\nabla h) \in C^1(\Omega)$ and that there holds a linear decay estimate,

$$\operatorname{osc}_2 V(\nabla h)(x, \theta t) \lesssim \theta \operatorname{osc}_2 V(\nabla h)(x,t), \qquad \theta \in (0,1],$$

on all balls $B_t(x) \subset \Omega$?

Let us mention that from $V(\nabla h) \in C^1$ it follows that $\nabla h \in C^1$ for $p \leq 2$. Moreover, for $p \geq 2$ it implies $A(\nabla h) \in C^1$ and $\nabla h \in C^{\frac{1}{p-1}}$, i.e., $h \in C^{p'}$ in the sense of Hölder spaces. Thus the conjecture is stronger than the well-known p'-conjecture; see [1].

Finally, note that our Theorems 1–4 can be generalized to higher dimensions and/or vectorial solutions to (1). However, in this setting we have to restrict ourselves to $s, \beta \in (0, \beta_0)$, where $\beta_0 \leq 1$ is some unknown small number. The reason is that our method of proof is based on decay estimates for p-harmonic functions which are worse in this general situation.

4 Regularity of Vector Field Deformations

The local regularity transfer in Theorem 1 tells us that the mapping $F \mapsto A(\nabla u)$ maintains integrability and smoothness (up to first order) such that we can formally cancel the divergence operator in (1). So, in order to formulate regularity statements for the solution u, given F, it remains to analyze how these properties behave under the vector field deformation $A(\nabla u) \mapsto \nabla u$.

To this end, for $\alpha > 0$ and $n \in \mathbb{N}$, let us define the transformation

$$T_\alpha \colon \mathbb{R}^n \to \mathbb{R}^n, \qquad Q \mapsto T_\alpha(Q) := |Q|^\alpha \, \frac{Q}{|Q|} \tag{5}$$

with $T_\alpha(0) := 0$. Then, under composition, $\{T_\alpha \,|\, \alpha > 0\}$ forms a group with T_1 being the identity and $(T_\alpha)^{-1} = T_{\frac{1}{\alpha}}$. So, for $\alpha, \beta > 0$ we have $T_{\alpha\beta}(Q) = T_\alpha(T_\beta(Q))$ and thus

$$\nabla u = T_{\frac{2}{p}}(V(\nabla u)) = T_{\frac{1}{p-1}}(A(\nabla u)) \qquad \text{and} \qquad V(\nabla u) = T_{\frac{p'}{2}}(A(\nabla u)).$$

Since $\frac{2}{p}, \frac{1}{p-1}, \frac{p'}{2} \in (0, 1]$ if $p \geq 2$, we are especially interested in the mapping properties of T_α for small α and vector-valued functions. The following result extends the local assertion [3, Prop. 4.4] to a global one.

Theorem 5 *If $d, n \in \mathbb{N}$, $\alpha \in (0, 1]$, and $\Omega \subset \mathbb{R}^d$ is a bounded Lipschitz domain or $\Omega = \mathbb{R}^d$ itself, the following statements for T_α from (5) hold true:*

(1) For $0 < r \leq \infty$ we have $T_\alpha \colon L^r(\Omega)^n \to L^{\frac{r}{\alpha}}(\Omega)^n$ with

$$\left\| T_\alpha(G) \,\middle|\, L^{\frac{r}{\alpha}}(\Omega)^n \right\| \sim \left\| G \,\middle|\, L^r(\Omega)^n \right\|^\alpha .$$

(2) If $0 < \varrho, q \leq \infty$ and $d\big(\frac{1}{\varrho} - 1\big)_+ < s < 1$, then $T_\alpha \colon \boldsymbol{B}^s_{\varrho,q}(\Omega)^n \to \boldsymbol{B}^{\alpha s}_{\frac{\varrho}{\alpha},\frac{q}{\alpha}}(\Omega)^n$ with

$$\left\| T_\alpha(G) \,\middle|\, \boldsymbol{B}^{\alpha s}_{\frac{\varrho}{\alpha},\frac{q}{\alpha}}(\Omega)^n \right\| \lesssim \left\| G \,\middle|\, \boldsymbol{B}^s_{\varrho,q}(\Omega)^n \right\|^\alpha .$$

(3) For $0 < \varrho < \infty$, $0 < q \leq \infty$, and $d\big(\frac{1}{\min\{\varrho,q\}} - 1\big)_+ < s < 1$ there holds $T_\alpha \colon \boldsymbol{F}^s_{\varrho,q}(\Omega)^n \to \boldsymbol{F}^{\alpha s}_{\frac{\varrho}{\alpha},\frac{q}{\alpha}}(\Omega)^n$ with

$$\left\| T_\alpha(G) \,\middle|\, \boldsymbol{F}^{\alpha s}_{\frac{\varrho}{\alpha},\frac{q}{\alpha}}(\Omega)^n \right\| \lesssim \left\| G \,\middle|\, \boldsymbol{F}^s_{\varrho,q}(\Omega)^n \right\|^\alpha .$$

Proof Assertion (1) directly follows from the definition of the (quasi-) norm of $G\colon \Omega \to \mathbb{R}^n$ in vector-valued Lebesgue spaces,

$$\left\| G \,\middle|\, L^r(\Omega)^n \right\| := \sum_{j=1}^{n} \left\| G_j \,\middle|\, L^r(\Omega) \right\| \sim \left\| |G| \,\middle|\, L^r(\Omega) \right\|, \qquad 0 < r \leq \infty,$$

and the fact that $|T_\alpha(Q)| = |Q|^\alpha$ for all $\alpha > 0$.

The proof of the remaining assertions is based on the characterization of Besov- and Triebel-Lizorkin spaces under consideration in terms of first-order ball means

$$\left(d_{t,v}^{1,\Omega} G\right)(x) := \left(t^{-d} \int_{h \in V^1(x,t)} \left| \left(\Delta_h^1 G\right)(x) \right|^v \,\mathrm{d}h \right)^{1/v}, \quad x \in \Omega,\ t > 0,\ 0 < v \leq \infty,$$

(correspondingly modified if $v = \infty$), where $\Delta_h^1 G$ denotes the first-order difference of G with stepsize $h \in \mathbb{R}^d$ and $V^1(x,t) := \{h \in B_t(0) \,|\, x + \tau h \in \Omega \text{ for } \tau \in [0,1]\}$; cf. [22, Sect. 1.11.9]. To this end, note that for $\alpha \in (0,1]$ there exists $c_{\alpha,n} > 0$ such that $|T_\alpha(P) - T_\alpha(Q)| \leq c_{\alpha,n}\, |P - Q|^\alpha$ for all $P, Q \in \mathbb{R}^n$, see [3, Formula (4.10)]. This allows to pointwise bound the first-order differences of $T_\alpha(G)$,

$$\left| \left(\Delta_h^1 T_\alpha(G)\right)(x) \right| \lesssim \left| \left(\Delta_h^1 G\right)(x) \right|^\alpha, \qquad h \in \mathbb{R}^d,\ x \in \Omega_h := \{x \in \Omega \mid x + h \in \Omega\},$$

as well as the corresponding ball means with $v := 1/\alpha \geq 1$

$$\left(d_{t,1/\alpha}^{1,\Omega} T_\alpha(G)\right)(x) \lesssim \left(d_{t,1}^{1,\Omega} G\right)(x)^\alpha, \qquad x \in \Omega,\ t > 0.$$

Let us consider the case of Besov spaces with $q < \infty$ on domains. In this situation, the characterization of interest is given by Triebel [22, Thm. 1.118(ii)], where we set $r := v$ and $M := 1$. Then the condition $d\left(\frac{1}{\varrho/\alpha} - \frac{1}{r}\right)_+ < \alpha s$ needed for $\mathbf{B}^{\alpha s}_{\frac{\varrho}{\alpha},\frac{q}{\alpha}}(\Omega)^n$ is equivalent to our assumption $d\left(\frac{1}{\varrho} - 1\right)_+ < s$. Moreover, $s < 1 = M$ implies $\alpha s < M$. So, for the (quasi-) semi norm we obtain

$$\begin{aligned} |T_\alpha(G)|_{\mathbf{B}^{\alpha s}_{\frac{\varrho}{\alpha},\frac{q}{\alpha}}(\Omega)^n} &= \left(\int_0^1 \left[t^{-\alpha s} \left\| d_{t,1/\alpha}^{1,\Omega} T_\alpha(G) \,\middle|\, L^{\varrho/\alpha}(\Omega) \right\| \right]^{q/\alpha} \frac{\mathrm{d}t}{t} \right)^{\alpha/q} \\ &\lesssim \left(\int_0^1 \left[t^{-\alpha s} \left\| d_{t,1}^{1,\Omega} G \,\middle|\, L^{\varrho}(\Omega) \right\|^\alpha \right]^{q/\alpha} \frac{\mathrm{d}t}{t} \right)^{\alpha/q} = \left(|G|_{\mathbf{B}^s_{\varrho,q}(\Omega)^n} \right)^\alpha. \end{aligned}$$

In addition, for the lower order part of the (quasi-) norm, we have to bound $\left\| T_\alpha(G) \,\middle|\, L^{\overline{\varrho/\alpha}}(\Omega)^n \right\|$ in terms of $\left\| G \,\middle|\, L^{\overline{\varrho}}(\Omega)^n \right\|^\alpha$, where we set $\overline{w} := \max\{w, 1\}$ for $w > 0$. This can be done by combining Assertion (1) with Hölder's inequality.

If $q = \infty$ or if we deal with Triebel-Lizorkin spaces on domains, the proof is based on exactly the same arguments as before. In contrast, for spaces on $\Omega = \mathbb{R}^d$, some care is needed as the lower order term has to be replaced by

$$\left\| T_\alpha(G) \,\middle|\, L^{\overline{\varrho/\alpha}}(\Omega)^n \right\|^* := \left\| T_\alpha(G) \,\middle|\, L^{\varrho/\alpha}(\Omega)^n \right\| + \left\| T_\alpha(G) \,\middle|\, L^{\overline{\varrho/\alpha}}(\Omega)^n \right\|$$

which needs to be controlled by $\left\| G \,\middle|\, L^{\overline{\varrho}}(\Omega)^n \right\|^*$ to the power α. If $\overline{\varrho/\alpha} = 1$, i.e., $0 < \varrho < \alpha \leq 1$, this additionally requires the estimate

$$\begin{aligned} \left\| G \,\middle|\, L^\alpha(\mathbb{R}^d)^n \right\| &\lesssim \left\| G \,\middle|\, L^\varrho(\mathbb{R}^d)^n \right\|^{1-\Theta} \left\| G \,\middle|\, L^1(\mathbb{R}^d)^n \right\|^{\Theta} \\ &\lesssim \left\| G \,\middle|\, L^\varrho(\mathbb{R}^d)^n \right\| + \left\| G \,\middle|\, L^1(\mathbb{R}^d)^n \right\| = \left\| G \,\middle|\, L^{\overline{\varrho}}(\mathbb{R}^d)^n \right\|^*, \end{aligned}$$

for which we used the log-convexity of Lebesgue (quasi-) norms together with Young's inequality with suitable parameters. □

Corollary 2 *Under the assumptions of Theorem 1 there holds*

$$\left\| \nabla u \,\middle|\, \boldsymbol{B}^{\frac{s}{p-1}}_{\varrho(p-1),q(p-1)}(B)^d \right\| \lesssim \left\| F \,\middle|\, \boldsymbol{B}^s_{\varrho,q}(2B)^d \right\|^{\frac{1}{p-1}}$$

and similarly for $\boldsymbol{B}$ *replaced by* $\boldsymbol{F}$.

Acknowledgments The research of Anna Kh. Balci was partly supported by Russian Foundation for Basic Research project 19-01-00184 in Vladimir State University named after Alexander and Nikolay Stoletovs and by the German Research Foundation (DFG) through the CRC 1283.

References

1. Araújo, D.J., Teixeira, E.V., and Urbano, J.M.: A proof of the $C^{p'}$-regularity conjecture in the plane. Adv. Math. **316**, 541–553 (2017)
2. Avelin, B., Kuusi, T., and Mingione, G.: Nonlinear Calderón-Zygmund theory in the limiting case. Arch. Rational Mech. Anal. **214**(2), 663–714 (2018)
3. Balci, A.Kh., Diening, L., and Weimar, M.: Higher order Calderón-Zygmund estimates for the p-Laplace equation. J. Differential Equations **268**, 590–635 (2020)
4. Breit, D., Cianchi, A., Diening, L., Kuusi, T., and Schwarzacher, S.: Pointwise Calderón-Zygmund gradient estimates for the p-Laplace system. J. Math. Pures Appl. **114**, 146–190 (2018)
5. Cianchi, A., and Maz'ya, V.G.: Second-order two-sided estimates in nonlinear elliptic problems. Arch. Rational Mech. Anal. **229**(2), 569–599 (2018)
6. Cioica-Licht, P., and Weimar, M.: On the limit regularity in Sobolev and Besov scales related to approximation theory. J. Fourier Anal. Appl. **26**(1), Art. 10, 1–24 (2020)
7. Clop, A., Giova, R., and Passarelli di Napoli, A.: Besov regularity for solutions of p-harmonic equations. Adv. Nonlinear Anal. **8**(1), 762–778 (2017)
8. Dahlke, S., Dahmen, W., and DeVore, R.: Nonlinear approximation and adaptive techniques for solving elliptic operator equations. In: Dahmen, W., Kurdila, A., and Oswald, P. (eds)

Multiscale Wavelet Methods for Partial Differential Equations, pp. 237–283. Academic Press, San Diego (1997)

9. Dahlke, S., Diening, L., Hartmann, C., Scharf, B., and Weimar M.: Besov regularity of solutions to the p-Poisson equation. Nonlinear Anal. **130**, 298–329 (2016)
10. DeVore, R.A.: Nonlinear approximation. Acta Numer. **7**, 51–150 (1998)
11. DiBenedetto, E., and Manfredi, J.: On the higher integrability of the gradient of weak solutions of certain degenerate elliptic systems. Amer. J. Math. **115**(5), 1107–1134 (1993)
12. Diening, L., Kaplický, P., and Schwarzacher, S.: BMO estimates for the p-Laplacian. Nonlinear Anal. **75**(2), 637–650 (2012)
13. Gaspoz, F., and Morin, P.: Approximation classes for adaptive higher order finite element approximation. Math. Comp. **83**(289), 2127–2160 (2014)
14. Hartmann, C., and Weimar, M.: Besov regularity of solutions to the p-Poisson equation in the vicinity of a vertex of a polygonal domain. Results Math. **73**(41), 1–28 (2018)
15. Iwaniec, T.: Projections onto gradient fields and L^p-estimates for degenerated elliptic operators. Studia Math. **75**(3), 293–312 (1983).
16. Kuusi, T., and Mingione, G.: Linear potentials in nonlinear potential theory. Arch. Rational Mech. Anal. **207**(1), 215–246 (2013)
17. Kuusi, T., and Mingione, G.: Vectorial nonlinear potential theory. J. Eur. Math. Soc. **20**(4), 929–1004 (2018)
18. Lindgren, E., and Lindqvist, P.: Regularity of the p-Poisson equation in the plane. J. Anal. Math. **132**(1), 217–228 (2017)
19. Runst, T., and Sickel, W.: Sobolev Spaces of Fractional Order, Nemytskij Operators, and Nonlinear Partial Differential Equations. Walter de Gruyter & Co., Berlin (1996)
20. Triebel, H.: Theory of Function Spaces. Birkhäuser, Basel (1983)
21. Triebel, H.: Theory of Function Spaces II. Birkhäuser, Basel (1992)
22. Triebel, H.: Theory of Function Spaces III. Birkhäuser, Basel (2006)
23. Uhlenbeck, K.: Regularity for a class of non-linear elliptic systems. Acta Math. **138**, 219–240 (1977)
24. Weimar, M.: On the lack of interior regularity of the p-Poisson problem with $p > 2$, Math. Nachr., (2020), https://arxiv.org/abs/1907.12805

A Low-Rank Approach for Nonlinear Parameter-Dependent Fluid-Structure Interaction Problems

Peter Benner, Thomas Richter, and Roman Weinhandl

Abstract Parameter-dependent discretizations of linear fluid-structure interaction problems can be approached with low-rank methods. When discretizing with respect to a set of parameters, the resulting equations can be translated to a matrix equation since all operators involved are linear. If nonlinear fluid-structure interaction problems are considered, a direct translation to a matrix equation is not possible. We present a method that splits the parameter set into disjoint subsets and, on each subset, computes an approximation of the problem related to the upper median parameter by means of the Newton iteration. This approximation is then used as initial guess for one Newton step on a subset of problems.

1 Introduction

Fluid-structure interaction (FSI) problems depend on parameters such as the solid shear modulus, the fluid density and the fluid viscosity. Parameter-dependent FSI discretizations allow to observe the reaction of an FSI model to a change of such parameters. A parameter-dependent discretization of a linear FSI problem yields many linear systems to be approximated. These equations can be translated to one single matrix equation. The solution, a matrix, can be approximated by a low-rank method as discussed in [5]. But as soon as nonlinear FSI problems are considered, such a translation is not possible anymore.

P. Benner · R. Weinhandl (✉)
Max Planck Institute for Dynamics of Complex Technical Systems, Magdeburg, Germany

Otto von Guericke University Magdeburg, Magdeburg, Germany
e-mail: benner@mpi-magdeburg.mpg.de; weinhandl@mpi-magdeburg.mpg.de

T. Richter
Otto von Guericke University Magdeburg, Magdeburg, Germany
e-mail: thomas.richter@ovgu.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_115

The proposed method extends the low-rank framework of [5] to nonlinear problems. It splits the parameter set into disjoint subsets. On each of these subsets, the Newton approximation for the problem related to the upper median parameter is computed to approximate the Jacobian matrix for all problems related to the subset. This allows to formulate a Newton step as a matrix equation. The Newton update, a matrix, can be approximated by a low-rank method and the global approximation to the parameter-dependent nonlinear FSI problem is achieved by stacking the approximations on the disjoint subsets column-wise.

2 The Nonlinear Problem

Let $d \in \{2, 3\}$, Ω, F, S be open subsets of $\mathbb{R}^d$ with $\bar{F} \cup \bar{S} = \bar{\Omega}$, $F \cap S = \emptyset$. We use the stationary Navier-Stokes equations [4, Section 2.4.5.3] to model the fluid part in F and the stationary Navier-Lamé equations [4, Problem 2.23] for the solid part in S. The interface is $\Gamma_{\text{int}} = \partial F \cap \partial S$, the boundary part where Neumann outflow conditions hold $\Gamma_f^{\text{out}} \subset \partial F \setminus \partial S$ and the boundary part where Dirichlet conditions hold $\Gamma_f^D = \partial F \setminus (\Gamma_f^{\text{out}} \cup \Gamma_{\text{int}})$. The weak formulation of the coupled nonlinear FSI problem with a vanishing right hand side f reads

$$\begin{aligned} \langle \nabla \cdot v, \xi \rangle_F &= 0, \\ \mu_s \langle \nabla u + \nabla u^T, \nabla\varphi \rangle_S + \lambda_s \langle \operatorname{tr}(\nabla u) I, \nabla\varphi \rangle_S & \\ + \rho_s \langle (v \cdot \nabla) v, \varphi \rangle_F + \nu_f \rho_f \langle \nabla v + \nabla v^T, \nabla\varphi \rangle_F - \langle p, \nabla \cdot \varphi \rangle_F &= 0 \quad \text{and} \\ \langle \nabla u, \nabla\psi \rangle_F &= 0. \end{aligned} \tag{1}$$

With $v_{\text{in}} \in H^1(\Omega)^d$, an extension of the Dirichlet data on Γ_f^D, the trial function $v \in v_{\text{in}} + H_0^1(\Omega, \Gamma_f^D \cup \Gamma_{\text{int}})^d$ is the velocity, $u \in H_0^1(\Omega)^d$ the deformation and $p \in L^2(F)$ the pressure. The test functions are $\xi \in L^2(F)$ (divergence equation), $\varphi \in H_0^1(\Omega, \partial\Omega \backslash \Gamma_f^{\text{out}})^d$ (momentum equation) and $\psi \in H_0^1(F)^d$ (deformation equation). The $\mathcal{L}^2$ scalar product on F and S is denoted by $\langle \cdot, \cdot \rangle_F$ and $\langle \cdot, \cdot \rangle_S$, respectively. The parameters involved are the kinematic fluid viscosity $\nu_f \in \mathbb{R}$, the fluid density $\rho_f \in \mathbb{R}$, the solid shear modulus $\mu_s \in \mathbb{R}$ and the first Lamé parameter $\lambda_s \in \mathbb{R}$.

3 Discretization and Linearization

Assume we are interested in discretizing the nonlinear FSI problem described in (1) parameter-dependently with respect to $m_1 \in \mathbb{N}$ shear moduli given by the set

$$S_\mu := \{\mu_s^{i_1}\}_{i_1 \in \{1,\ldots,m_1\}} \subset \mathbb{R}^+, \quad \text{with} \quad \mu_s^1 < \ldots < \mu_s^{m_1}.$$

Consider a finite element discretization on Ω_h, a matching mesh of the domain Ω, with a total number of $N \in \mathbb{N}$ degrees of freedom. Let $A_0 \in \mathbb{R}^{N\times N}$ be a discretization matrix of all *linear* operators involved in (1) with fixed parameters ν_f, ρ_f, μ_s and λ_s. Let $A_1 \in \mathbb{R}^{N\times N}$ be the discretization matrix of the operator

$$\langle \nabla u + \nabla u^T, \nabla\varphi\rangle_S. \tag{2}$$

The nonlinear part in (1), the convection term, requires a linearization technique.

3.1 Linearization with Newton Iteration

For a linearization by means of the Newton iteration, we need the Jacobian matrix of the operator $\langle (v\cdot\nabla)v, \varphi\rangle_F$. In our finite element space, every unknown $x_h = (p_h, v_h, u_h)^T \in \mathbb{R}^N$ consists of a pressure p_h, a velocity v_h and a deformation u_h. The discrete test space also has dimension N and every unknown there can be written as $(\xi_h, \phi_h, \psi_h)^T \in \mathbb{R}^N$. The Jacobian matrix of $\langle (v\cdot\nabla)v, \varphi_h\rangle_F$ in our finite element space, evaluated at x_h, is

$$J_{\langle (v\cdot\nabla)v,\varphi_h\rangle_F}(x_h) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \frac{\partial\langle (v\cdot\nabla)v,\varphi_h\rangle_F}{\partial v}\Big|_{v=v_h} & 0 \\ 0 & 0 & 0 \end{pmatrix} =: A_{\text{conv}}(x_h) \in \mathbb{R}^{N\times N}.$$

Let $b_D \in \mathbb{R}^N$ be the right hand side vector that depends on the desired Dirichlet boundary conditions of the nonlinear FSI problem. Consider the FSI problem related to a fixed shear modulus $\mu_s^{i_1} \in S_\mu$ for some $i_1 \in \{1,\dots,m_1\}$ first.

If we start with an initial guess $x_0^{i_1} \in \mathbb{R}^N$, for instance $x_0^{i_1} = b_D$, at Newton step $j \in \mathbb{N}$, the equation

$$\big(A_0 + (\mu_s^{i_1} - \mu_s)A_1 + \rho_f A_{\text{conv}}(x_{j-1}^{i_1})\big)s = -g(x_{j-1}^{i_1}, \mu_s^{i_1}) \tag{3}$$

is to be solved for $s \in \mathbb{R}^N$. The approximation at linearization step j then is

$$x_j^{i_1} = x_{j-1}^{i_1} + s.$$

$g(x_{j-1}^{i_1}, \mu_s^{i_1})$ evaluates all operators in (1) at the pressure, velocity and deformation of the approximation of the previous linearization step $x_{j-1}^{i_1}$ and the shear modulus $\mu_s^{i_1}$.

4 Newton Iteration and Low-Rank Methods

In order to approximate a set of problems at one time, Eq. (3) has to be translated to a matrix equation. For this, first of all, we split the parameter set S_μ into disjoint subsets.

If we perform a Newton step for a set of problems at one time, the same Jacobian matrix is used for the whole set. Therefore, the solutions to these different problems should not differ too much from each other. The method suggested in this paper splits the given parameter set into $K \in \mathbb{N}$ disjoint subsets, each of them containing adjacent parameters.

$$S_\mu = \dot{\bigcup_{k=1}^{K}} \mathcal{I}_k.$$

By $\tilde{m}_k$, we denote the index of the upper median parameter of the set $\mathcal{I}_k$. After the parameter set is split into the subsets $\{\mathcal{I}_k\}_{k\in\{1,\dots,K\}}$, we compute the Newton approximation $x_{\epsilon_N}^{\tilde{m}_k}$ of the problem related to the upper median parameter $\mu_s^{\tilde{m}_k}$ for all $k \in \{1,\dots,K\}$ up to some given accuracy $\epsilon_N > 0$. $x_{\epsilon_N}^{\tilde{m}_k}$ is then used as initial guess for one Newton step.

4.1 *The Matrix Equation*

With $D_{1,k} := \operatorname{diag}(\mathcal{I}_k) - \mu_s I^{|\mathcal{I}_k|\times|\mathcal{I}_k|}$ and $v_{\mathcal{I}_k} := (\mu_s^{i_1})_{i_1\in\mathcal{I}_k} \in \mathbb{R}^{|\mathcal{I}_k|}$, the matrix equation that is to be solved for $S_k \in \mathbb{R}^{N\times|\mathcal{I}_k|}$ on every subset $\mathcal{I}_k$ is

$$A_0 S_k + A_1 S_k D_{1,k} + \rho_f A_{\text{conv}}(x_{\epsilon_N}^{\tilde{m}_k}) S_k = \underbrace{-g(x_{\epsilon_N}^{\tilde{m}_k}, 0) \otimes (1,\dots,1) - A_1 x_{\epsilon_N}^{\tilde{m}_k} \otimes v_{\mathcal{I}_k}^T}_{=:B_k}. \tag{4}$$

$I^{|\mathcal{I}_k|\times|\mathcal{I}_k|}$ denotes the identity matrix of size $|\mathcal{I}_k| \times |\mathcal{I}_k|$. In (4), the initial guess for the Newton step is

$$X_{\text{initial}}^k := x_{\epsilon_N}^{\tilde{m}_k} \otimes (1,\dots,1).$$

The approximation at the next linearization step is

$$X^k := X_{\text{initial}}^k + S_k. \tag{5}$$

The global approximation for the whole parameter-dependent problem then is

$$\tilde{X} := [X^1|\cdots|X^K].$$

Remark 1 The initial guess for the Newton step (4), X^k_{initial}, has rank 1 and the operator (2) is linear. This is why the rank of the right hand side matrix B_k in (4) is not bigger than 2.

Remark 2 If multiple Newton steps like (4) were performed, two main difficulties would come up. At step 2, the approximation of the previous linearization step would be given by X^k from (5).

The Right Hand Side X^k is not a matrix of low rank and $g(\cdot,\cdot)$ would have to be evaluated for all columns of X^k separately in a second Newton step. Thus, the right hand side matrix B_k would not have low-rank structure either.

The Jacobian Matrix Since all columns of the initial guess X^k_{initial} coincide, the Jacobian matrix in (4) is correct for all equations related to the parameter set $\mathcal{I}_k$. But the columns of X^k differ from each other. A second Newton step would then become what is, in the literature, often called an inexact Newton step [4, Remark 5.7].

4.2 Low-Rank Methods

Let $k \in \{1,\ldots,K\}$,

$$\tilde{A}_{\text{conv}} := A_{\text{conv}}(x^{\tilde{m}_k}_{\epsilon_N}) \quad \text{and} \quad b_g := g(x^{\tilde{m}_k}_{\epsilon_N}, 0).$$

Consider only the column related to the parameter index $i_1 \in \mathcal{I}_k$ in (4):

$$\underbrace{\left(A_0 + (\mu_s^{i_1} - \mu_s)A_1 + \rho_f \tilde{A}_{\text{conv}}\right)}_{=:A(\mu_s^{i_1})} s^{i_1} = \underbrace{-b_g - \mu_s^{i_1} A_1 x^{\tilde{m}_k}_{\epsilon_N}}_{=:b(\mu_s^{i_1})}, \qquad \text{with} \qquad s^{i_1} \in \mathbb{R}^N.$$

Assume that $x^{\tilde{m}_k}_{\epsilon_N}$ is fixed and $A(\mu_s^{i_1})$ is invertible for all $\mu_s^{i_1} \in \mathcal{I}_k$. $A(\mu_s^{i_1})$ and $b(\mu_s^{i_1})$ depend linearly on $\mu_s^{i_1}$. $A(\cdot)$ and $b(\cdot)$ are analytic matrix- and vector-valued functions, respectively. Due to [2, Theorem 2.4], the singular value decay of the matrix S_k in (4) is exponential. Algorithm 1 exploits this fact and approximates S_k in (4) by a low-rank matrix.

5 Numerical Results

A $3d$ jetty flow in a channel with the geometric configuration

$$\Omega := (0, 12) \times (0, 4) \times (0, 4), S := (2, 3) \times (0, 2) \times (0, 4) \quad \text{and} \quad F := \Omega \setminus \bar{S}$$

is considered. The left Dirichlet inflow is given by the velocity profile

$$v = \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} \frac{1}{10} y(4-y)z(4-z) \\ 0 \\ 0 \end{pmatrix} \in \mathbb{R}^3 \quad \text{at} \quad x = 0.$$

At $x = 12$, the do nothing boundary condition holds. At $z = 0$, deformation and velocity in normal direction is prohibited. Everywhere else on $\partial\Omega$, the velocity and the deformation vanish. For the Navier-Stokes equations, stabilized Stokes elements [4, Lemma 4.47] are used.

Algorithm 1 Low-rank Method for One-parameter Nonlinear FSI

```
Require: Accuracy $\epsilon_N > 0$ for Newton method, ranks $R_k \in \mathbb{N}$ for $k \in \{1, \ldots, K\}$
Ensure: The rank-$\sum_{k=1}^{K} R_k$ approximation $\hat{X}$ of the parameter-dependent FSI discretization
  Split the parameter set $S_\mu$ into the subsets $\dot{\bigcup}_{k=1}^{K} \mathcal{I}_k$.
  for $k = 1, \ldots, K$ do
      Compute the Newton approximation of the upper median parameter problem related to a
      shear modulus of $\mu_s^{\tilde{m}_k}$ with accuracy $\epsilon_N \curvearrowright x_{\epsilon_N}^{\tilde{m}_k}$.
      Use $x_{\epsilon_N}^{\tilde{m}_k} \otimes (1, \ldots, 1) \in \mathbb{R}^{N \times |\mathcal{I}_k|}$ as initial guess for one Newton step on $\mathcal{I}_k$. Find a rank-
      $(R_k - 1)$ approximation $\hat{S}_k$ that approximates $S_k \in \mathbb{R}^{N \times |\mathcal{I}_k|}$ from (4) by a low-rank method
      from [5].
      Build the sum
          $$\hat{X}_k = x_{\epsilon_N}^{\tilde{m}_k} \otimes (1, \ldots, 1) + \hat{S}_k.$$
  end for
  $\hat{X} := [\hat{X}_1 | \ldots | \hat{X}_K]$
```

5.1 Parameters

The nonlinear FSI problem (1) is discretized with Q_1 elements (compare [4, Section 4.2.1]) with respect to

$$1500 \text{ shear moduli } \mu_s^{i_1} \in S_\mu \subset [20{,}000, 60{,}000].$$

The fixed first Lamé parameter is $\lambda_s = 200{,}000$. With these parameters, solid configurations with Poisson ratios between 0.38462 and 0.45455 are covered. The fluid density is $\rho_f = 12.5$ and the kinematic fluid viscosity is $\nu_f = 0.04$.

5.2 Comparison ChebyshevT with Standard Newton

A server operating CentOS 7 with 2 AMD EPYC 7501 and 512GB RAM, MATLAB® 2017b in combination with the htucker MATLAB toolbox [3] and the finite element toolkit GASCOIGNE [1] was used to compare Algorithm 1 with 1500 Newton iterations applied consecutively. The parameter set S_μ was split into $K = 15$ subsets.

Preconditioner and Eigenvalue Estimation

After the Newton approximations $x_{\epsilon_N}^{\tilde{m}_k}$ are available for all $k \in \{1, \ldots, 15\}$, the LU decomposition of the mean-based preconditioner P_T^k [5, Section 3.2] of $A(\mu_s^{i_1})$ is computed separately on every subset $\mathcal{I}_k$. To estimate the parameters d and c for the ChebyshevT method [5, Algorithm 3], the eigenvalues of the matrices

$$(P_T^k)^{-1} A\Big(\max_{i_1 \in \mathcal{I}_k} (\mu_s^{i_1})\Big) \qquad \text{and} \qquad (P_T^k)^{-1} A\Big(\min_{i_1 \in \mathcal{I}_k} (\mu_s^{i_1})\Big)$$

are taken into consideration. For all $K = 15$ subsets, they were estimated to $d = 1$ and $c = 0.1$ for a small number of $N = 945$ degrees of freedom within 28.3 s (computation time for the Newton approximations included). Every approximation is related to a certain shear modulus. Therefore, all problems differ by the Poisson ratio of the solid. The y-axis in Fig. 1, on the other hand, corresponds to the relative residual norm

$$\frac{\|g(x^{i_1}, \mu_s^{i_1})\|_2}{\|g(b_D, \mu_s^{i_1})\|_2}$$

of the approximation x^{i_1} for $i_1 \in \{1, \ldots, m_1\}$. Algorithm 1 was applied with $\epsilon_N = 10^{-4}$ and $R_k = 10\ \forall k \in \{1, \ldots, 15\}$ to a problem with $N = 255{,}255$. Therefore, the global approximation rank is $R = 150$. In comparison to this, standard Newton iterations were applied to the 1500 separate problems consecutively where for every Newton iteration, the last approximation served as initial guess for the next Newton iteration.

The approximations obtained by the Standard Newton iterations within 238 h (1507 Newton steps) provided, as visualized in Fig. 1, relative residuals with norms smaller than 10^{-12} each. Algorithm 1 took 519 min (35 Newton steps) to

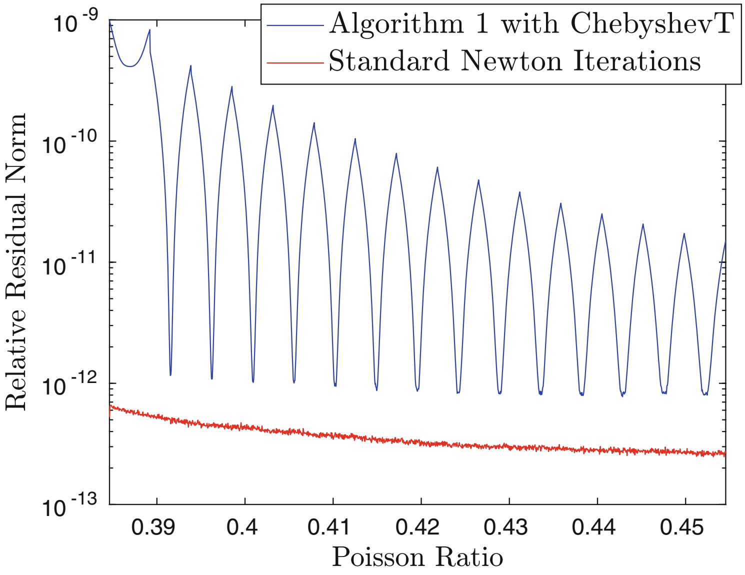


Fig. 1 Comparison of the approximations provided by Algorithm 1 and 1500 standard Newton iterations applied consecutively

compute the low-rank approximation. In addition to the 28.3 s for the eigenvalue estimation, the 20 Newton steps to compute $x_{\epsilon_N}^{\tilde{m}_k}$ for all $k \in \{1, \dots, 15\}$ took, in total, 195.65 min and the 15 Newton steps for the matrix equations (4) took, in total, 323.3 min.

6 Conclusions

Low-rank methods can be used to compute approximations to parameter-dependent nonlinear FSI discretizations, in particular, if each of the subsets, the parameter set is split into, does contain problems that do not differ too much from each other. The Newton step on the subset uses the same Jacobian matrix and the same initial guess for the whole subset. It has to provide acceptable convergence within one single step not only for the upper median problem.

Whether the results can be improved by choosing the subsets $\mathcal{I}_k$ or the approximation ranks on these subsets adaptively, is still open. Moreover, how these low-rank methods can be applied to fully nonlinear FSI problems that use, in addition to the Navier-Stokes equations on the fluid, for instance, the St. Venant Kirchhoff model equations [4, Definition 2.18] on the solid is an open problem. Then, the right hand side in (4) would have to be approximated.

Acknowledgments This work was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—314838170, GRK 2297 MathCoRe.

References

1. Becker, R., Braack, M., Meidner, D., Richter, T., Vexler, B.: The finite element toolkit GASCOIGNE. http://www.uni-kiel.de/gascoigne
2. Kressner, D., Tobler, C.: Low-rank tensor Krylov subspace methods for parametrized linear systems. SIAM J. Matrix Anal. Appl. **32**(4), 1288–1316 (2011). https://doi.org/10.1137/100799010
3. Kressner, D., Tobler, C.: Algorithm 941: `htucker`–a Matlab toolbox for tensors in hierarchical Tucker format. ACM Trans. Math. Software **40**(3), Art. 22, 22 (2014). https://doi.org/10.1145/2538688
4. Richter, T.: Fluid-structure Interactions, *Lecture Notes in Computational Science and Engineering*, vol. 118. Springer International Publishing, Cham, Switzerland (2017). https://doi.org/10.1007/978-3-319-63970-3
5. Weinhandl, R., Benner, P., Richter, T.: Low-rank linear fluid-structure interaction discretizations. e-Print, arXiv (2019). https://arxiv.org/abs/1905.11000

Simulating Two-Dimensional Viscoelastic Fluid Flows by Means of the "Tensor Diffusion" Approach

Patrick Westervoß and Stefan Turek

Abstract In this work, the novel "Tensor Diffusion" approach for simulating viscoelastic fluids is proposed, which is based on the idea, that the extra-stress tensor in the momentum equation of the flow model is replaced by a product of the strain-rate tensor and a tensor-valued viscosity. At least for simple flows, this approach offers the possibility to reduce the full nonlinear viscoelastic model to a generalized "Tensor Stokes" problem, avoiding the need of considering a separate stress tensor in the solution process. Besides fully developed channel flows, the "Tensor Diffusion" approach is evaluated as well in the context of general two-dimensional flow configurations, which are simulated by a suitable four-field formulation of the viscoelastic model respecting the "Tensor Diffusion".

1 Introduction

Numerical simulations of viscoelastic fluids are still a challenging task, especially due to the involved constitutive equations describing the complex material behaviour of the flow. From a numerical point of view, constitutive equations of differential type are quite straightforward to apply in combination with the Stokes equations, but being applicable only for a limited range of flow configurations [1–3].

An alternative modelling approach in numerical flow simulations is offered by considering integral constitutive equations, which are often of the so-called time-separable Rivlin-Sawyers (or Kaye-BKZ) type [4, 5], where the stress tensor is written as an infinite integral of the form

$$\boldsymbol{\sigma}\,(t) = \int_{-\infty}^{t} M\left(t - t'\right)\left[\phi_1\,(I_1, I_2)\,\mathbf{B}_{t'}\,(t) + \phi_2\,(I_1, I_2)\,\mathbf{B}_{t'}^{-1}(t)\right] dt' \qquad \text{(1a)}$$

P. Westervoß (✉) · S. Turek
Institute for Applied Mathematics (LS III), TU Dortmund University, Dortmund, Germany
e-mail: pwesterv@math.tu-dortmund.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_116

In the above stress integral, ϕ_1, ϕ_2 are empirical functions to model nonlinear effects depending on the two non-trivial invariants I_1, I_2 of the Finger tensor $\mathbf{B}$. One of the most suitable approaches to handle integral material models in combination with the Stokes equations is the so-called "Deformation Fields Method" (DFM, [6–8]). A central object in this scheme is the Finger tensor, which is evolved in time depending on the velocity field $\mathbf{u}$ according to the differential equation

$$\frac{\partial}{\partial s}\mathbf{B}_{t'}(s) + (\mathbf{u}(s)\cdot\nabla)\,\mathbf{B}_{t'}(s) - \nabla\mathbf{u}(s)^{\top}\cdot\mathbf{B}_{t'}(s) - \mathbf{B}_{t'}(s)\cdot\nabla\mathbf{u}(s) = \mathbf{0} \quad \text{(1b)}$$

in $s \in \left[t', t\right]$ for fixed t', where $\mathbf{B}_{t'}\left(t'\right) = \mathbf{I}$.

However both, the differential as well as integral material model, give rise to numerical challenges due to the complex rheology of the considered viscoelastic fluids. On the one hand, in the differential case, the well-known "High Weissenberg Number Problem" (HWNP, [1, 2]) together with the need of considering multiple modes [3] has to be taken into account. On the other hand, for integral constitutive equations, a suitable numerical treatment of the resulting integro-differential set of equations needs to be derived resp. requires further improvement [6–8].

Therefore, in this work, the novel "Tensor Diffusion" approach is introduced, offering the possibility to remove the complex rheology of the fluid from the set of equations and to establish a straightforward numerical treatment of viscoelastic fluids.

2 The "Tensor Diffusion" Approach

As outlined above, many difficulties and challenges in simulating viscoelastic fluids arise from the complex rheology of the fluid characterized by both, differential and integral constitutive equations. Consequently, avoiding the need of considering such an equation at all would probably improve the general numerical treatment of such fluids. Thus, the underlying assumption of the novel "Tensor Diffusion" approach is the existence of a decomposition of the extra-stress tensor according to

$$\boldsymbol{\sigma} = \boldsymbol{\mu}\cdot\mathbf{D}(\mathbf{u}) \quad (2)$$

where $\boldsymbol{\mu} \in \mathbb{R}^{2\times 2}$ in two-dimensional settings. Inserting the stress decomposition (2) into the stationary Stokes equations gives the so-called "Tensor Stokes" problem

$$-\frac{1}{2}\nabla\cdot\left(\boldsymbol{\mu}\cdot\mathbf{D}(\mathbf{u}) + \mathbf{D}(\mathbf{u})\cdot\boldsymbol{\mu}^{\top}\right) + \nabla p = \mathbf{0}, \qquad \nabla\cdot\mathbf{u} = 0 \quad (3)$$

Note, that a symmetrized version of the "Tensor Stokes" problem is considered here, since the "Tensor Diffusion" $\boldsymbol{\mu}$ is in general not symmetric as shown in Sect. 3.1 (for details, see [9, 10]).

Assuming, that the so-called "Tensor Diffusion" $\boldsymbol{\mu}$—corresponding to an actual viscoelastic flow problem—is known or given, the "nonlinear" velocity and pressure solution, originally resulting from the (direct steady) nonlinear differential or integral viscoelastic model, can be computed by simply solving the "Tensor Stokes" problem (3) in $(\mathbf{u}, p)$. Thus, the constitutive equation or the complex rheology of such fluids is removed from the system and the corresponding stresses are computed in post-processing fashion based on the velocity solution calculated from Eq. (3). Furthermore, a robust, efficient, accurate and stable numerical scheme can be used for solving the "Tensor Stokes" problem (3), since typical solution techniques for (generalized) Stokes problems, i.e. problems in $(\mathbf{u}, p)$ only, are applicable in this context.

Obviously, the "Tensor Stokes" problem represents an extension of classical generalized Stokes equations involving a shear-rate dependent *scalar viscosity* (c.f. [11]), since besides the corresponding "shear thinning" effect, in principle the full viscoelastic material behaviour is covered by the *tensor-valued viscosity* $\boldsymbol{\mu}$ (see Sect. 3). Thus, one of the main potential benefits of the novel "Tensor Diffusion" approach is the possibility to express the complex rheology by means of a "Tensor Diffusion" $\boldsymbol{\mu}$ instead of solving a nonlinear constitutive equation. In the following section, the validity of this concept will be shown for Poiseuille-like flows, followed by an evaluation for complex flow configurations like the "Flow around cylinder"-benchmark in Sect. 4.

3 Proof of Concept

In the following, the validity of the underlying assumption, that a stress decomposition according to Eq. (2) exists, is investigated by checking the ability of the "Tensor Diffusion" approach to reproduce viscoelastic flow characteristics usually resulting from differential or integral material models. Therefore, steady-state two-dimensional fully developed channel flows for viscoelastic fluids are considered, where the same velocity profile is obtained at any cutline over the channel height, i.e. in y-direction. Thus, the velocity field consists only of a y-dependent contribution in x-direction, i.e. the channel length. Similarly, the components of stress and Finger tensors depend on y only, but not on x.

3.1 Fully Developed Channel Flows for UCM

Considering the differential steady-state version of the Upper-Convected Maxwell model (UCM, [4]) in the above setting, the corresponding unknowns can be given analytically, especially leading to a parabolic velocity profile. Furthermore, the corresponding (symmetric) strain-rate as well as stress tensors read

$$\boldsymbol{\sigma} = \begin{pmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{12} & \sigma_{22} \end{pmatrix} = \begin{pmatrix} 2\eta_p \Lambda u_y^2 & \eta_p u_y \\ \eta_p u_y & 0 \end{pmatrix}, \ \mathbf{D}\left(\mathbf{u}\right) = \frac{1}{2}\begin{pmatrix} 2u_x & v_x + u_y \\ v_x + u_y & 2v_y \end{pmatrix} = \frac{1}{2}\begin{pmatrix} 0 & u_y \\ u_y & 0 \end{pmatrix} \quad (4)$$

Consequently it is realized, that indeed a matrix- or tensor-valued quantity reading

$$\boldsymbol{\mu} = 2\eta_p \begin{pmatrix} 1 & 2\Lambda u_y \\ 0 & 1 \end{pmatrix} \quad (5)$$

can be derived even analytically, relating $\boldsymbol{\sigma}$ and $\mathbf{D}$ according to Eq. (2).

In principle, the same can be done in case of the steady-state integral version of UCM, where only one single Finger tensor needs to be considered due to the stationary velocity field. However, inserting the analytical expressions for the components of the Finger tensor—derived for fully developed channel flows—into the single-mode “stationary” stress integral for UCM (c.f. [4]) yields

$$\begin{aligned} \boldsymbol{\sigma} &= \int_0^\infty \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right)\left(\mathbf{B}\left(s\right) - \mathbf{I}\right) ds \\ &= \left[2\int_0^\infty \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right)\begin{pmatrix} s & s^2 u_y \\ 0 & s \end{pmatrix} ds\right]\left[\frac{1}{2}\begin{pmatrix} 0 & u_y \\ u_y & 0 \end{pmatrix}\right] = \boldsymbol{\mu} \cdot \mathbf{D}\left(\mathbf{u}\right) \quad (6) \end{aligned}$$

with the same “Tensor Diffusion” $\boldsymbol{\mu}$ as calculated from the differential version. Particularly, a stress decomposition according to Eq. (2) can be derived for differential as well as integral viscoelastic models.

3.2 *Poiseuille-Like Flow for the Wagner Model*

In the following, a nonlinear integral model is considered, in detail the Wagner model [12], which—for two-dimensional stationary flow configurations—results in a stress integral of the form

$$\begin{aligned} \boldsymbol{\sigma} = \int_0^\infty \frac{\eta_p}{\Lambda^2} \exp\left(-\frac{s}{\Lambda}\right)\Big[f \exp\left(-n_1\sqrt{I-2}\right) + \ldots \\ (1-f)\exp\left(-n_2\sqrt{I-2}\right)\Big]\mathbf{B}\left(s\right) ds \quad (7) \end{aligned}$$

including the single non-trivial invariant I of the Finger tensor $\mathbf{B}$. For fully developed channel flows, the stress integral can be converted into

$$\boldsymbol{\sigma} = \boldsymbol{\mu} \cdot \mathbf{D}\left(\mathbf{u}\right) + \boldsymbol{\nu} \quad (8)$$

where $\boldsymbol{\mu}, \boldsymbol{\nu} \in \mathbb{R}^{2\times 2}$ and

$$\mu_{11} = 2\eta_p \left[f \left(1 + n_1 \Lambda \sqrt{u_y^2}\right)^{-2} + (1-f)\left(1 + n_2 \Lambda \sqrt{u_y^2}\right)^{-2} \right] \tag{9a}$$

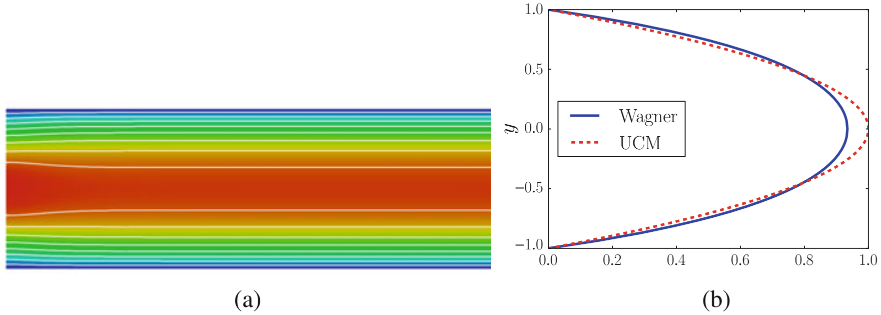
$$\mu_{12} = 4\eta_p \Lambda u_y \left[f \left(1 + n_1 \Lambda \sqrt{u_y^2}\right)^{-3} + (1-f)\left(1 + n_2 \Lambda \sqrt{u_y^2}\right)^{-3} \right] \tag{9b}$$

$$\nu = \frac{\eta_p}{\Lambda} \left[f \left(1 + n_1 \Lambda \sqrt{u_y^2}\right)^{-1} + (1-f)\left(1 + n_2 \Lambda \sqrt{u_y^2}\right)^{-1} \right] \tag{9c}$$

besides $\mu_{22} = \mu_{11}$ and $\mu_{21} = 0$ as well as $\nu_{11} = \nu_{22} = \nu$ and $\nu_{12} = \nu_{21} = 0$. Consequently, a "generalized" stress decomposition compared to UCM in Eq. (6) is derived. However, by introducing the modified pressure $P = p - \nu$, a similar version of the "Tensor Stokes" problem in Eq. (3) is obtained, but now replacing the original pressure p by the modified pressure P, since the operator $\nabla \cdot \boldsymbol{\nu}$ occurring in the "Tensor Stokes" problem can be considered as $\nabla \nu$ and thus be absorbed into the pressure gradient.

In the following, a modified Poiseuille flow is considered in Finite Element simulations, where the velocity on the inflow edge is set to take a parabolic profile. At the same time, the "Tensor Diffusion" corresponding to a fully developed channel flow is prescribed globally, which is why the flow should evolve to its fully developed nonlinear shape away from the inflow.

Obviously, the flow profiles obtained from the Wagner model for the material parameters given in [12] recover the shear-thinning effect regarding the velocity profile as depicted in Fig. 1, which is a typical material behaviour of viscoelastic fluids. Furthermore, this velocity profile, resulting from two-dimensional simulations, matches the solution of the one-dimensional version of the full integral model derived for fully developed channel flows [10]. This indicates, that especially for nonlinear integral models, viscoelastic flow characteristics in fully developed channel flows are reproduced by simply solving a generalized Stokes-like problem



(a) (b)

Fig. 1 Channel flow for Wagner model, $\Lambda = 1.0, f = 0.57, n_1 = 0.31, n_2 = 0.106$. (**a**) x-velocity from 2D. (**b**) x-velocity at x_{mid}

of the form (3) in the unknowns $(\mathbf{u}, p)$, where the complex rheology arising from the stress integral is completely hidden in the "Tensor Diffusion".

In principle, the same procedure can be done also for other nonlinear viscoelastic constitutive equations like the Giesekus model [13] in the differential or the PSM model [14] in the integral case. However, for none of these two cases, the "Tensor Diffusion" $\boldsymbol{\mu}$ can be given in closed form, since it can be derived only "semi analytically" or numerically. But nevertheless, similar results for comparing solutions of one- and two-dimensional simulations can also be obtained for other viscoelastic models than Wagner, further outlining the basic validity of the proposed "Tensor Diffusion" approach [9, 10].

4 Complex Flow Configurations

So far, the proposed "Tensor Diffusion" approach is analyzed only in the context of fully developed channel flows, for which it is possible to derive and verify the validity of this novel approach. When more general two-dimensional flow configurations shall be investigated in terms of this novel approach, an explicit derivation of the corresponding tensor-valued viscosity $\boldsymbol{\mu}$ is not (yet?) possible.

Instead, a straightforward implementation for determining the "Tensor Diffusion" numerically is obtained by complementing the original differential steady-state viscoelastic model by an additional algebraic equation regarding $\boldsymbol{\mu}$ and inserting symmetrized version of the stress decomposition (2) into the momentum equation of the flow model. Consequently, to evaluate the applicability of the "Tensor Diffusion" approach in the context of general two-dimensional flow configurations, the well-known "Flow around cylinder" benchmark [1, 2, 15] is simulated by means of the four-field formulation of the above "Tensor Stokes" problem reading

$$-2\eta_s \nabla \cdot \mathbf{D}(\mathbf{u}) - \frac{1}{2}\nabla \cdot \left(\boldsymbol{\mu} \cdot \mathbf{D}(\mathbf{u}) + \mathbf{D}(\mathbf{u}) \cdot \boldsymbol{\mu}^\top\right) + \nabla p = \mathbf{0} \tag{10a}$$

$$\nabla \cdot \mathbf{u} = 0 \tag{10b}$$

$$(\mathbf{u} \cdot \nabla)\boldsymbol{\sigma} - \nabla \mathbf{u}^\top \cdot \boldsymbol{\sigma} - \boldsymbol{\sigma} \cdot \nabla \mathbf{u} + \mathbf{f}\left(\Lambda, \eta_p, \boldsymbol{\sigma}\right) = 2\frac{\eta_p}{\Lambda}\mathbf{D}(\mathbf{u}) \tag{10c}$$

$$\boldsymbol{\mu} \cdot \mathbf{D}(\mathbf{u}) - \boldsymbol{\sigma} = 0 \tag{10d}$$

which is discretized within the Finite Element framework presented in [2], where the "Tensor Diffusion" is approximated by elementwise constant polynomials [10].

Within the typical benchmark configuration of a present solvent contribution of $\eta_s = 0.59$, the drag coefficients $C_D(\mathbf{T})$, which are computed based on the total stress tensor $\mathbf{T}$, are analyzed for evaluating the quality of the simulation results for several Weissenberg numbers $\mathrm{We} = \Lambda U_{\mathrm{mean}}/R$. Therefore, the drag coefficients calculated from the "Tensor Diffusion" are compared to reference results as well as

Table 1 Oldroyd-B model [4, 16]

We	$C_D\,(\mathbf{T}_\sigma)$	$C_D\,(\mathbf{T}_\mu)$	Ref. [1]
0.1	130.342	130.348	130.36
0.2	126.605	126.624	126.62
0.3	123.172	123.212	123.19
0.4	120.553	120.549	120.59
0.5	118.747	118.751	118.83

Table 2 Giesekus model, $\alpha = 0.1$ [13]

We	$C_D\,(\mathbf{T}_\sigma)$	$C_D\,(\mathbf{T}_\mu)$	Ref. [15]
0.1	125.567	125.572	125.58
0.5	103.717	103.733	103.73
1.0	95.536	95.568	95.55
5.0	85.210	85.243	–
10.0	83.047	83.068	–

Table 3 UCM ($\alpha = 0.0$) or Giesekus model

We	α	$C_D\,(\mathbf{T}_\sigma)$	$C_D\,(\mathbf{T}_\mu)$
0.1	0.0	127.373	127.403
0.5	0.0	96.046	98.054
0.1	0.1	115.377	115.508
0.5	0.1	60.804	61.992

results based on the original approach validated in [2]. In the following, $\mathbf{T}_\sigma$ denotes the total stress tensor arising from the original viscoelastic model and $\mathbf{T}_\mu$ the one corresponding to the "Tensor Stokes" problem, where in principle $\boldsymbol{\sigma}$ is replaced by the symmetrized stress-decomposition to obtain $\mathbf{T}_\mu$ from $\mathbf{T}_\sigma$.

A summary of the drag coefficients resulting from the above configuration is given in Tables 1 and 2, which illustrates, that the drag coefficients obtained from the four-field formulation (10) of the "Tensor Stokes" problem show a good agreement to the results computed by means of the original method as well as the reference results [1, 15] for both, the Oldroyd-B and Giesekus model. For the latter, reference results apparently are available only up to We $=$ 1.0, which is why the "Tensor Stokes" results for higher Weissenberg numbers are evaluated by a comparison with the original approach only.

The more challenging configuration is represented by considering the "no solvent" case in the above setting, where $\eta_s = 0$ in Eq. (10a). Unfortunately, no reference results are available for this flow configuration, which is why the "Tensor Stokes" results are again compared only against the results of the original approach. When analyzing the calculated drag coefficients given in Table 3, again the "Tensor Stokes" results show a good agreement to the results of the original problem—especially for lower We for both, the UCM as well as Giesekus model. Besides, for the Giesekus model it was not possible to reach significantly larger Weissenberg numbers as in case of UCM, which again illustrates the complexity of this flow configuration.

Additionally, recall that $\boldsymbol{\mu}$ is approximated in Q_0 only, which is of lower order than the corresponding approximation of $\boldsymbol{\sigma}$ in Q_2. Naturally, results obtained from the original problem are expected to be of higher accuracy anyway. But nevertheless, applying the “Tensor Diffusion” approach gives simulation results of a similar quality as the original approach, even for this complex flow configuration.

5 Conclusion

In this work, the novel “Tensor Diffusion” approach is introduced, where in principle the extra-stress tensor in the momentum equation of the viscoelastic model is replaced by a product of the so-called “Tensor Diffusion” and the strain-rate tensor.

The underlying assumption, that such a stress decomposition exists in general, is verified in a first step for fully developed channel flows, where the full viscoelastic model can be reduced to a so-called “Tensor Stokes” problem. Consequently, the nonlinear viscoelastic solution might be simply computed from a generalized Stokes-like problem including a tensor-valued viscosity.

Furthermore, the applicability suitable of the “Tensor Diffusion” approach is evaluated within the two-dimensional “Flow around cylinder” benchmark. Here, the drag coefficients resulting from the original viscoelastic model as well as reference results are reproduced quite well by means of a four-field formulation of the “Tensor Stokes” problem. But nevertheless, as a main goal of future work, the full viscoelastic flow model shall be reduced to a pure “Tensor Stokes” problem.

References

1. M. A. Hulsen, R. Fattal, R. Kupferman, Flow of viscoelastic fluids past a cylinder at high Weissenberg number: Stabilized simulations using matrix logarithms, Journal of Non-Newtonian Fluid Mechanics 127 (1) (2005) 27–39. https://doi.org/10.1016/j.jnnfm.2005.01.002
2. H. Damanik, J. Hron, A. Ouazzi, S. Turek, A monolithic FEM approach for the log-conformation reformulation (LCR) of viscoelastic flow problems, Journal of Non-Newtonian Fluid Mechanics 165 (19) (2010) 1105–1113. https://doi.org/10.1016/j.jnnfm.2010.05.008
3. J. Kroll, S. Turek, P. Westervoß, Evaluation of nonlinear differential models for the simulation of polymer melts, Kautschuk Gummi Kunststoffe (317) (2017) 48–52.
4. R. G. Larson, Constitutive Equations for Polymer Melts and Solutions, Butterworths Series in Chemical Engineering, Butterworth-Heinemann, 1988.
5. R. Keunings, Finite element methods for integral viscoelastic fluids, 2003.
6. E. Peters, M. Hulsen, B. van den Brule, Instationary Eulerian viscoelastic flow simulations using time separable Rivlin–Sawyers constitutive equations, Journal of Non-Newtonian Fluid Mechanics 89 (1) (2000) 209–228. https://doi.org/10.1016/S0377-0257(99)00026-9
7. M. Hulsen, E. Peters, B. van den Brule, A new approach to the deformation fields method for solving complex flows using integral constitutive equations, Journal of Non-Newtonian Fluid Mechanics 98 (2) (2001) 201–221. https://doi.org/10.1016/S0377-0257(01)00110-0
8. M. A. Hulsen, P. D. Anderson, The deformation fields method revisited: Stable simulation of instationary viscoelastic fluid flow using integral models, Journal of Non-Newtonian Fluid Mechanics 262 (2018) 68–78. https://doi.org/10.1016/j.jnnfm.2018.03.001

9. P. Westervoß, S. Turek, H. Damanik, A. Ouazzi, The "tensor diffusion" approach for simulating viscoelastic fluids, Tech. rep., Department of Mathematics, technical report of the Institute for Applied Mathematics, No. 617 (Nov. 2019). http://dx.doi.org/10.17877/DE290R-20363
10. P. Westervoß, A new approach for simulating viscoelastic fluids, Ph.D. thesis, TU Dortmund (to be submitted in 2020).
11. A. Ouazzi, S. Turek, Numerical methods and simulation techniques for flow with shear and pressure dependent viscosity, in: M. Feistauer, V. Dolejsi, P. Knobloch, K. Najzar (Eds.), Numerical Mathematics and Advanced Applications, Springer, 2003, pp. 668–676, enumath 2003 Prague; ISBN-Nr. 3-540-21460-7.
12. H. M. Laun, Description of the non-linear shear behaviour of a low density polyethylene melt by means of an experimentally determined strain dependent memory function, Rheologica Acta 17 (1) (1978) 1–15. http://dx.doi.org/10.1007/BF01567859
13. H. Giesekus, Die elastizität von flüssigkeiten, Rheologica Acta 5 (1966) 29–35.
14. C. W. Macosko, R. G. Larson, K. (Firm), Rheology : principles, measurements, and applications, New York : VCH, 1994, originally published as ISBN 1560815795.
15. S. Claus, T. Phillips, Viscoelastic flow around a confined cylinder using spectral/hp element methods, Journal of Non-Newtonian Fluid Mechanics 200 (2013) 131–146, special Issue: Advances in Numerical Methods for Non-Newtonian Flows. https://doi.org/10.1016/j.jnnfm.2013.03.004
16. J. G. Oldroyd, On the formulation of rheological equations of state, Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 200 (1063) (1950) 523–541.

Dynamic and Weighted Stabilizations of the L-scheme Applied to a Phase-Field Model for Fracture Propagation

Christian Engwer, Iuliu Sorin Pop, and Thomas Wick

Abstract We consider a phase-field fracture propagation model, which consists of two (nonlinear) coupled partial differential equations. The first equation describes the displacement evolution, and the second is a smoothed indicator variable, describing the crack position. We propose an iterative scheme, the so-called L-scheme, with a dynamic update of the stabilization parameters during the iterations. Our algorithmic improvements are substantiated with two numerical tests. The dynamic adjustments of the stabilization parameters lead to a significant reduction of iteration numbers in comparison to constant stabilization values.

1 Introduction

This work is an extension of [3] in which an L-type iterative scheme (see [5, 8]) with stabilizing parameters for solving phase-field fracture problems was proposed. In [3], the stabilization parameters were chosen as constants throughout an entire computation. With these choices, the convergence of the scheme has been proven rigorously. The resulting approach performs well in the sense that an unlimited number of iterations compared to a truncated scheme yields the same numerical solution. The results were validated by investigating the load-displacements curves. Moreover, the robustness of the scheme w.r.t. spatial mesh refinement was shown.

C. Engwer
Institut für Numerische und Angewandte Mathematik Fachbereich Mathematik und Informatik der Universität Münster, Münster, Germany
e-mail: christian.engwer@uni-muenster.de

I. S. Pop
Faculty of Sciences, Universiteit Hasselt, Diepenbeek, Belgium
e-mail: sorin.pop@uhasselt.be

T. Wick (✉)
Leibniz Universität Hannover, Institut für Angewandte Mathematik, Hannover, Germany
e-mail: thomas.wick@ifam.uni-hannover.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_117

Nonetheless, the iteration numbers (for an unlimited number of iterations) remained high.

In this work, we propose and compare two extensions of the aforementioned scheme. First, we update the L scheme parameters dynamically. Second, we use an adaptive weight depending on the fracture location inside the domain. For the latter idea, we use the phase-field variable to weight L locally.

The outline of this work is as follows: In Sect. 2 the model is stated whereas Sect. 3 presents the dynamic choice of the stabilization parameters. In Sect. 4, we present two numerical tests to study the performance of the proposed scheme.

2 The Phase-Field Fracture Model

We consider an elliptic problem stemming from the crack propagation model proposed in [3]. $\Omega \subset \mathbb{R}^d$ is a d-dimensional, polygonal and bounded domain. We use the spaces $W^{1,\infty}(\Omega)$, containing functions having essentially bounded weak derivatives in any direction, and $H_0^1(\Omega)$ containing functions vanishing at the boundary of Ω (in the sense of traces) and having square integrable weak derivatives. $(\cdot,\cdot)$ stands for the $L^2(\Omega)$ inner product. For the ease of writing we use the notations $V := (H_0^1(\Omega))^d$ and $W := W^{1,\infty}(\Omega)$. The vector-valued displacements are denoted by u. For modeling fracture propagation in Ω, a phase field variable φ is used. This approximates the characteristic function of the intact region of Ω. Written in weak form, we solve the following problems iteratively

- **Problem** 1^i : Given $(u^{i-1}, \varphi^{i-1}) \in V \times W$, find $u^i \in V$ s.t. for all $v \in V$

$$a_u(u^i, v) := L_u(u^i - u^{i-1}, v) + \Big(g(\varphi^{i-1})\boldsymbol{\sigma}^+(u^i), \mathbf{e}(v)\Big) + \Big(\boldsymbol{\sigma}^-(u^i), \mathbf{e}(v)\Big) = 0. \tag{1}$$

- **Problem** 2^i : Given $(\varphi^{i-1}, u^i, \bar{\varphi}) \in W \times V \times W$, find $\varphi^i \in W$ s.t. for all $\psi \in W$

$$\begin{aligned} a_\varphi(\varphi^i, \psi) := {} & L_\varphi(\varphi^i - \varphi^{i-1}, \psi) + G_c\varepsilon(\nabla\varphi^i, \nabla\psi) - \frac{G_c}{\varepsilon}(1 - \varphi^i, \psi) \\ & + (1-\kappa)(\varphi^i \boldsymbol{\sigma}^+(u^i) : \mathbf{e}(u^i), \psi) + (\Xi + \gamma[\varphi^i - \bar{\varphi}]^+, \psi) = 0. \end{aligned} \tag{2}$$

In case of convergence, the first terms in the above are vanishing, and the limit pair $(u, \varphi) \in V \times W$ solves a time discrete counterpart of the model in [3], if $\bar{\varphi}$ is interpreted as the phase field at the previous time step. In this context, with $\Xi \in L^2(\Omega)$ and $\gamma > 0$, the last term in (2) is the augmented Lagrangian penalization proposed in [9] for the irreversibility constraint of the fracture propagation.

Furthermore, in the above, ε is a (small) phase-field regularization parameter, $G_c > 0$ is the critical elastic energy restitution rate, and $0 < \kappa \ll 1$ is a regularization parameter used to avoid the degeneracy of the elastic energy. The latter is similar to replacing the fracture with a softer material. Next, $g(\varphi) :=$

$(1-\kappa)\varphi^2+\kappa$ is the degradation function, and $\mathbf{e} := \frac{1}{2}(\nabla u + \nabla u^T)$ is the strain tensor.

The stress tensor in the above is split into a tensile and compressive part,

$$\boldsymbol{\sigma}^+ := 2\mu_s \mathbf{e}^+ + \lambda_s[\mathrm{tr}(\mathbf{e})]^+ I, \quad \boldsymbol{\sigma}^- := 2\mu_s(\mathbf{e}-\mathbf{e}^+) + \lambda_s\big(\mathrm{tr}(\mathbf{e}) - [\mathrm{tr}(\mathbf{e})]^+\big)I,$$

where $[\cdot]^+$ stands for the positive cut of the argument. Further, $\mathbf{e}^+ = \mathbf{P}\boldsymbol{\Lambda}^+\mathbf{P}^T$, with $\mathbf{P}$ being the matrix containing the unit eigenvectors corresponding to the eigenvalues of the strain tensor $\mathbf{e}$. In particular, for $d = 2$ one has $\mathbf{P} = [v_1, v_2]$ and

$$\boldsymbol{\Lambda}^+ := \boldsymbol{\Lambda}^+(u) := \begin{pmatrix} [\lambda_1(u)]^+ & 0 \\ 0 & [\lambda_2(u)]^+ \end{pmatrix}.$$

3 The L-scheme with Dynamic Updates of the Stabilization Parameters

The iteration (1)–(2) is essentially the scheme proposed in [3], in which the stabilization parameters L_u and L_φ are taken constant. To improve the convergence behaviour of the scheme, we propose a dynamic update of these parameters.

Dynamic Update at Each Iteration/Constant in Space The iteration discussed in [3] uses constant parameters L_u and L_φ. With this choice, the convergence has been proved rigorously. However, the number of iterations can remain high. High iteration numbers for phase-field fracture problems were also reported in [4, 10]. To improve the efficiency, we suggest in this work to update L_u and L_φ at each iteration i:

$$L_i = a(i)L_{i-1}, \qquad \text{where } L_i := L_{u,i} = L_{\varphi,i}.$$

Inspired by numerical continuation methods in e.g. [1], one would naturally choose a large L_0 and $a(i) := a < 1$ to obtain a decreasing sequence $L_0 > L_1 > L_2 > \ldots$, updated until a lower bound L_- is reached. However, this seems not to be a good choice in phase-field fracture since the system does not have a unique solution. Consequently, with increasing i the iterations would oscillate in approaching one or another solution, and the algorithm convergence deteriorates. For this reason, we propose the other way around: the closer the iteration is to some solution, the larger the stabilization parameters is chosen, so that the iterations remain close to this solution. We choose $a(i) := a > 1$, yielding $L_0 < L_1 < L_2 < \ldots$ up to a maximal L_*.

On the Specific Choice of the Parameters A possible choice for a is $a(i) := 5^i$ $(i = 0, 1, 2, \ldots)$, while $L_0 := 10^{-10}$. This heuristic choice and may be improved by using the solution within the iteration procedure, or a-posteriori error estimates for the iteration error. Moreover, $a(i) := 5^i$ is motivated as follows. Higher values greater than 5 would emphasize too much the stabilization. On the other hand, too low values, do not lead to any significant enhancement of the convergence behaviour. We substantiate these claims by also using $a(i) = 10^i$ and $a(i) = 20^i$ in our computations.

Dynamic Update Using the Iteration An extension of the strategy is to adapt the L-scheme parameters in space by using the phase-field variable $\varphi^{n,i-1}$. We still take $L_i = aL_{i-1}$, but now $a := a(i, \varphi^{n,i-1})$. Away from the fracture, we have $\varphi \approx 1$ and essentially only the elasticity component (2) is being solved. On the other hand, the stabilization is important in the fracture region, for which we take

$$L_i = a(i, \varphi^{i-1})L_{i-1}, \qquad \text{with } a(i, \varphi^{i-1}) := (1 - \varphi^{i-1})a.$$

Recalling that the fracture is characterised by $\varphi \approx 0$, it becomes clear that the stabilization parameters are acting mainly in the fracture region. Finally, to improve further the convergence behaviour of the scheme we adapt Ξ at each iteration. In this case we take $\Xi_i = \Xi_{i-1} + \gamma[\varphi^{i-1} - \bar{\varphi}]^+$.

Algorithm 1 Dynamic variant of the L-scheme for a phase-field fracture

```
Choose $\gamma > 0$, $a > 1$, as well as $\Xi^0$ and $L_0$. Set $i = 0$.
repeat
  Let $i = i + 1$;
  Solve the two problems, namely
     Solve the nonlinear elasticity problem in (1)
     Solve the nonlinear phase-field problem in (2)
  Update  $L^i = aL^{i-1}$
  Update  $\Xi^i = \Xi^{i-1} + \gamma[\varphi^i - \bar{\varphi}]^+$
until
     $\max\{\|a_u(u^i, v)\|, \|a_\varphi(\varphi^i, \psi)\|/v \in V, \psi \in W\} \leq \text{TOL}$,
```

The Final Algorithm The algorithm is based on the iterative procedure for phase-field fracture originally proposed in [9]. Therein, the inequality constraint is realized by an augmented Lagrangian iteration. Within this loop we update the L scheme parameters too. The resulting is sketched in Algorithm 1, in which TOL $= 10^{-6}$ is taken, and $L = L_u = L_\varphi$.

Remark 1 For the solution of both nonlinear subproblems (1) and (2), we use a monotonicity-based Newton method (details see e.g., in [10]) with the tolerance 10^{-8}. Inside Newton's method, we solve the linear systems with a direct solver.

4 Numerical Tests

We consider two test examples. Details for the first test van be found in [7]. The setup of the second test can be found for instance in [6]. Both examples were already computed in [3] and the results therein are compared to the ones obtained here. The scheme is implemented in a code based on the deal.II library [2].

Single Edge Notched Shear Test The configuration is shown in Fig. 1 and a final simulation result in Fig. 2 (left). Specifically, we use $\mu_s = 80.77\,\text{kN/mm}^2$, $\lambda_s = 121.15\,\text{kN/mm}^2$, and $G_c = 2.7\,\text{N/mm}$. The crack growth is driven by a non-homogeneous Dirichlet condition for the displacement field on Γ_{top}, the top

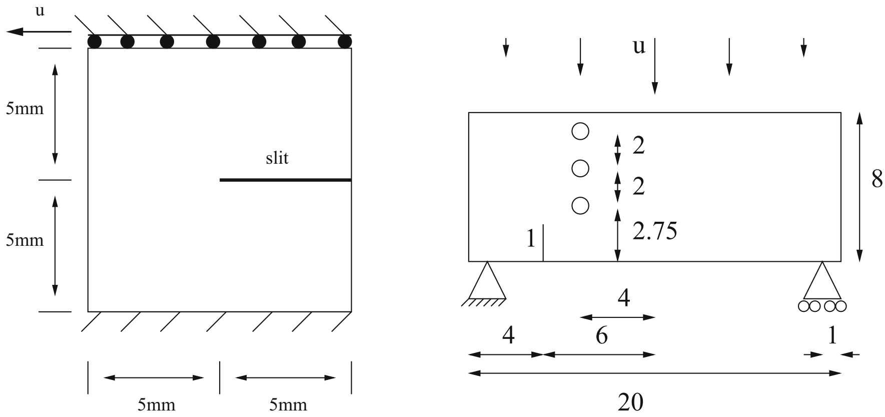


Fig. 1 Examples 1 and 2. The following conditions are prescribed: on the left and right boundaries, $u_y = 0\,\text{mm}$ and traction-free in x-direction. On the bottom part, $u_x = u_y = 0\,\text{mm}$. On Γ_{top}, $u_y = 0\,\text{mm}$ and u_x is as stated in (3). Finally, the lower part of the slit is fixed in y-direction, i.e., $u_y = 0\,\text{mm}$. Right: Asymmetric notched three point bending test. The three holes have each a diameter of 0.5. All units are in *mm*

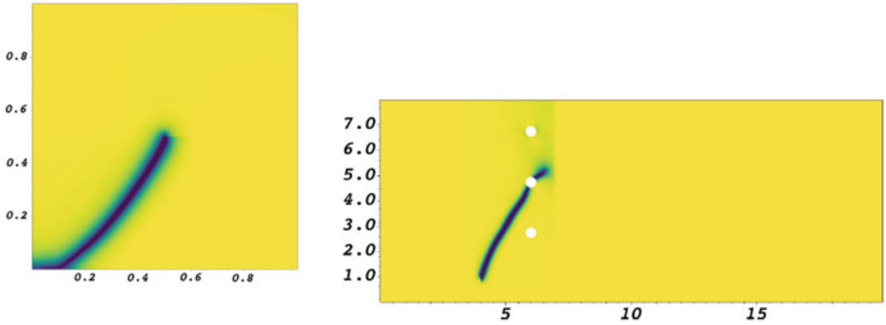


Fig. 2 Examples 1 and 2. Numerical solutions on the finest meshes and at the end time. The cracks are displayed in dark blue color

boundary of B. We increase the displacement on Γ_{top} over time, namely we apply non-homogeneous Dirichlet conditions:

$$u_x = t\bar{u}, \quad \bar{u} = 1\,\text{mm/s}, \tag{3}$$

where t denotes the current loading time. Furthermore, we set $\kappa = 10^{-10}$ [mm] and $\varepsilon = 2h$ [mm]. We evaluate the surface load vector on the Γ_{top} as

$$\tau = (F_x, F_y) := \int_{\Gamma_{\text{top}}} \boldsymbol{\sigma}(u)\nu \, \mathrm{d}s, \tag{4}$$

with normal vector ν, and we are particularly interested in the shear force F_x. Three different meshes with 1024 (Ref. 4), 4096 (Ref. 5) and 16,384 (Ref. 6) elements are observed in order to show the robustness of the proposed schemes. The results are shown in Fig. 6.

Our findings are summarized in Figs. 3 and 4. The numerical solutions for all four different strategies for choosing L are practically identical, only the number of iterations being different. Here, $L = 0$ and $L = 1e - 2$ denote tests in

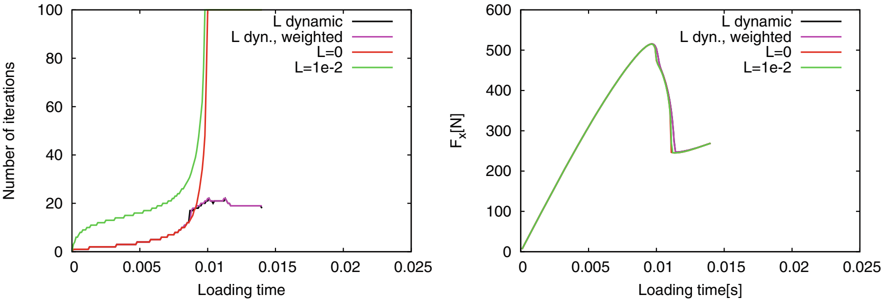


Fig. 3 Example 1. Comparison of dynamic L updates, the weighted version, and constant L. Left: number of iterations. Right: load-displacement curves

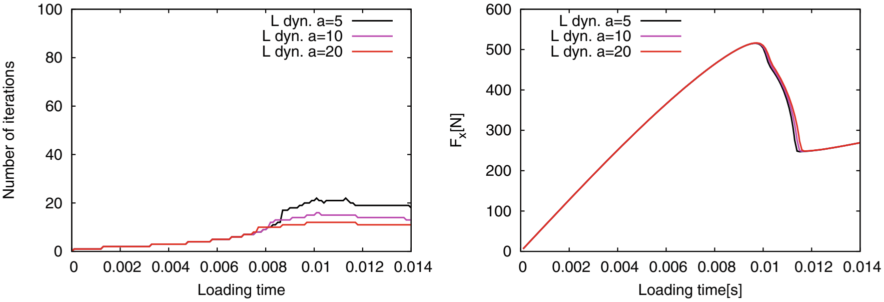


Fig. 4 Example 1. Comparison of different a for the dynamic L scheme

which $L = L_u = L_\varphi$ are taken constant throughout the entire computation. The newly proposed dynamic versions are denoted by `L dynamic` and `L dyn. weighted`. We observe a significant reduction in the computational cost when using the dynamic L-schemes. The maximum number of iterations is 21 for both the weighted version and the spatially-constant L-scheme. This number is reduced to 12 iterations using $a = 20$ while the accuracy only slightly changes.

Asymmetrically Notched Three Point Bending Test The configuration is shown in Fig. 1 (right). The initial mesh is 3, 4 and 5 times uniformly refined, yielding 3904, 15,616 and 62,464 mesh elements with the minimal mesh size parameter $h_3 = 0.135, h_4 = 0.066$ and $h_5 = 0.033$. As material parameters, we use $\mu_s = 8\,\text{kN/mm}^2$, $\lambda_s = 12\,\text{kN/mm}^2$, and $G_c = 1 \times 10^{-3}\,\text{kN/mm}$. Furthermore, we set $k = 10^{-10}h[\text{mm}]$ and $\varepsilon = 2h$.

Figure 5 presents the number of iterations and the load-displacement curves. The number of iterations is decreasing from 500 (in the figures cut to 100) for the classical L-scheme, to a maximum of 25 when using the dynamic updates. The choice of weighting does not seem to have a significant influence on the number of iterations though. The crack starts growing a bit later when using the dynamic updates, which can be inferred from the right plot in Fig. 5. Thus, the stabilization parameters have a slight influence on the physical solution. This can be explained in the following way. In regions where $\varphi = 0$ the solution component u is not uniquely defined. This leads to a sub-optimal convergence behaviour of the L-scheme. With the dynamic L-scheme we regain uniqueness, but at the cost of a slightly modified physical problem.

Remark 2 Noteworthy, the number of iterations for the dynamic L-scheme is robust with respect to the mesh refinement, as shown in Fig. 6. This is in line with the analysis in [3, 5, 8], where it is proved that the convergence rate does not depend on the spatial discretization.

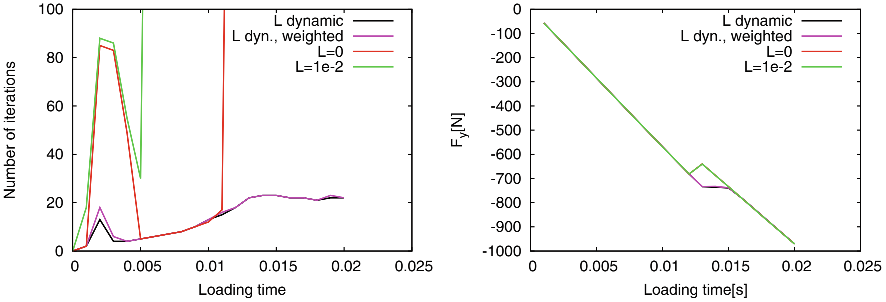


Fig. 5 Example 2: Left: The number of iterations for the different schemes; the results for $L = 0$ and $L = 1e-2$ are taken from [3]. Right: The load-displacement curves; a slight difference can be observed in the results, indicating that the dynamic updates lead to a slight delay in the prediction of the starting time for the fracture growth

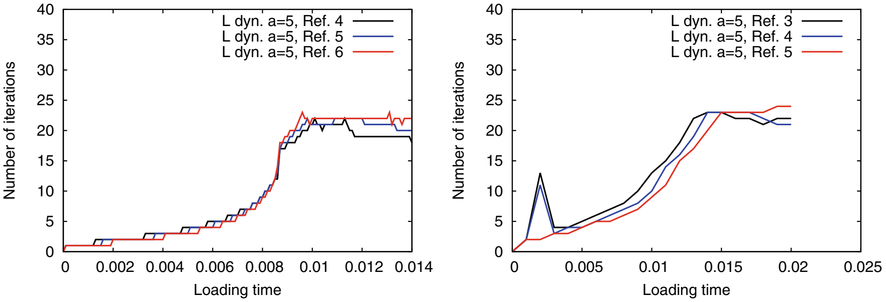


Fig. 6 Examples 1 and 2 for the dynamic L scheme using $a = 5$; three different mesh levels are used in order to verify the robustness of the proposed scheme. The results indicate that the mesh size does not influence the number of the iterations

Acknowledgments TW is supported by the German Research Foundation, Priority Program 1748 (DFG SPP 1748) under the project No. 392587580. CE is supported by the German Research Foundation, via Priority Program 1648 (DFG SPP 1648) under the grant No. EN-1042/2-2 and via EXC 2044-390685587, Mathematics Münster: Dynamics-Geometry-Structure. ISP is supported by the Research Foundation-Flanders (FWO), Belgium through the Odysseus programme (project G0G1316N).

References

1. E. L. Allgower and K. Georg. *Numerical continuation methods: an introduction*. Springer, 1990.
2. D. Arndt, W. Bangerth, T. C. Clevenger, D. Davydov, M. Fehling, D. Garcia-Sanchez, G. Harper, T. Heister, L. Heltai, M. Kronbichler, R. M. Kynch, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. The `deal.II` library, version 9.1. *J. Numer. Math.*, 27(4):203–213, 2019.
3. M. K. Brun, T. Wick, I. Berre, J. M. Nordbotten, and F. A. Radu. An iterative staggered scheme for phase field brittle fracture propagation with stabilizing parameters. *Comp. Meth. Appl. Mech. Engrg.*, 361:112752, 2020.
4. T. Gerasimov and L. D. Lorenzis. A line search assisted monolithic approach for phase-field computing of brittle fracture. *Comp. Meth. Appl. Mech. Engrg.*, 312:276–303, 2016.
5. F. List and F. A. Radu. A study on iterative methods for solving Richards' equation. *Comput. Geosci.*, 20(2):341–353, 2016.
6. A. Mesgarnejad, B. Bourdin, and M. Khonsari. Validation simulations for the variational approach to fracture. *Comp. Meth. Appl. Mech. Engrg.*, 290:420–437, 2015.
7. C. Miehe, F. Welschinger, and M. Hofacker. Thermodynamically consistent phase-field models of fracture: variational principles and multi-field FE implementations. *Int. J. Numer. Methods Engrg.*, 83:1273–1311, 2010.
8. I. S. Pop, F. Radu, and P. Knabner. Mixed finite elements for the Richards' equation: linearization procedure. *J. Comput. Appl. Math.*, 168(1–2):365–373, 2004.
9. M. Wheeler, T. Wick, and W. Wollner. An augmented-Lagangrian method for the phase-field approach for pressurized fractures. *Comp. Meth. Appl. Mech. Engrg.*, 271:69–85, 2014.
10. T. Wick. An error-oriented Newton/inexact augmented Lagrangian approach for fully monolithic phase-field fracture propagation. *SIAM J. Sci. Comput.*, 39(4):B589–B617, 2017.

Adaptive Numerical Simulation of a Phase-Field Fracture Model in Mixed Form Tested on an L-shaped Specimen with High Poisson Ratios

Katrin Mang, Mirjam Walloth, Thomas Wick, and Winnifried Wollner

Abstract This work presents a new adaptive approach for the numerical simulation of a phase-field model for fractures in nearly incompressible solids. In order to cope with locking effects, we use a recently proposed mixed form where we have a hydro-static pressure as additional unknown besides the displacement field and the phase-field variable. To fulfill the fracture irreversibility constraint, we consider a formulation as a variational inequality in the phase-field variable. For adaptive mesh refinement, we use a recently developed residual-type a posteriori error estimator for the phase-field variational inequality which is efficient and reliable, and robust with respect to the phase-field regularization parameter. The proposed model and the adaptive error-based refinement strategy are demonstrated by means of numerical tests derived from the L-shaped panel test, originally developed for concrete. Here, the Poisson's ratio is changed from the standard setting towards the incompressible limit $\nu \to 0.5$.

1 Introduction

Crack propagation is one of the major research topics in mechanical, energy, and environmental engineering. A well-established variational approach for Griffith's [5] quasi-static brittle fracture was introduced by Francfort and Marigo [3]. Miehe et al. [10] introduced the name 'phase-field modeling' for this variational approach. If the observed solid is assumed to be nearly incompressible, the classical phase-field fracture model fails due to volume-locking. In this work, we combine the mixed problem formulation, recently proposed by the authors in [7], with the

K. Mang · T. Wick (✉)
Institute of Applied Mathematics, Leibniz Universität Hannover, Hannover, Germany
e-mail: katrin.mang@ifam.uni-hannover.de; thomas.wick@ifam.uni-hannover.de

M. Walloth · W. Wollner
Department of Mathematics, Technische Universität Darmstadt, Darmstadt, Germany
e-mail: walloth@mathematik.tu-darmstadt.de; wollner@mathematik.tu-darmstadt.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_118

adaptive numerical solution based on a residual-type error estimator for the arising phase-field variational inequality [6, 11]. This allows to simulate crack propagation on adaptive refined meshes in nearly incompressible materials by using the phase-field method.

2 A Phase-Field Model for Nearly Incompressible Solids

2.1 Notation and Spaces

We emanate from a two-dimensional, open and smooth domain $\Omega \subset \mathbb{R}^2$. Let I be a loading interval $[0, T]$, where $T > 0$ is the end time value. A displacement function $u : (\Omega \times I) \to \mathbb{R}^2$ is defined on the domain Ω. On a subset $\Gamma_D \subset \partial\Omega$ of the boundary, we enforce Dirichlet boundary conditions. For the phase-field variable $\varphi : (\Omega \times I) \to [0, 1]$ with $\varphi = 0$ in the crack and $\varphi = 1$ in the unbroken material, we have homogeneous Neumann values $\nabla\varphi \cdot n = 0$ on the whole boundary, where n is the unit outward normal to the boundary. The physics of the underlying problem ask to enforce crack irreversibility, i.e., that φ is monotone non-increasing with respect to $t \in I$.

By $(a, b) := \int_\Omega a \cdot b \, dx$ for vectors a, b, the L^2 scalar-product is denoted. The Frobenius scalar product of two matrices of the same dimension is defined as $A : B := \sum_i \sum_j a_{ij} b_{ij}$ and therewith the L^2-scalar product is given by $(A, B) := \int_\Omega A : B \, dx$.

For a weak problem formulation, we consider a subdivision $0 = t_0 < \ldots < t_N = T$ of the interval I. In each time step, we define approximations $(u^n, \varphi^n) \approx (u(t_n), \varphi(t_n))$ and hence the irreversibility condition is approximated by $\varphi^n \leq \varphi^{n-1}$ for all $n = 1, \ldots, N$. To simplify the notation, we omit the superscript $(\cdot)^n$ and set $u := u^n$ and $\varphi := \varphi^n$, whenever it is clear from the context. The phase-field space is $\mathcal{W} := H^1(\Omega)$ with a feasible set $\mathcal{K} := \{\psi \in \mathcal{W} \mid \psi \leq \varphi^{n-1} \leq 1\}$. Further, we define the function spaces $\mathcal{V} := (H_0^1(\Omega))^2 := \{w \in (H^1(\Omega))^2 \mid w = 0 \text{ a.e. on } \Gamma_D\}$, $\mathcal{U} := L_2(\Omega)$, and $\mathcal{X} := \{\Lambda \in \mathcal{W}^* \mid \Lambda \geq 0\}$, where $\mathcal{W}^*$ is the dual space of $\mathcal{W}$. Further, let $u_D \in (H^1(\Omega))^2 \cap C^0(\Gamma_D)$ be a continuation of the Dirichlet-data. For the classical phase-field fracture model, we refer to Miehe et al. [10]. In the next section, the mixed form of the phase-field fracture model is formulated.

2.2 Mixed Phase-Field Fracture Model

The stress tensor $\sigma(u)$ is given by $\sigma(u) := 2\mu E_{\text{lin}}(u) + \lambda \text{tr}(E_{\text{lin}}(u))\mathbf{I}$ with the Lamé coefficients $\mu, \lambda > 0$. The linearized strain tensor therein is defined as $E_{\text{lin}}(u) := \frac{1}{2}(\nabla u + \nabla u^T)$. By $\mathbf{I}$, the two-dimensional identity matrix is denoted. For a mixed formulation of the problem, we define

$$p := \lambda \nabla \cdot u,$$

with $p \in \mathcal{U}$, such that the pure elasticity equation reads as follows:
Find $u \in \mathcal{V}$ and $p \in \mathcal{U}$ such that

$$\begin{aligned} 2\mu(E_{\mathrm{lin}}(u), E_{\mathrm{lin}}(w)) + (\nabla \cdot w, p) &= 0 \quad \forall w \in \mathcal{V}, \\ (\nabla \cdot u, q) - {}^1\!/\!_\lambda(p, q) &= 0 \quad \forall q \in \mathcal{U}. \end{aligned}$$

Following [9], we consider the tensile ($\sigma^+(u, p)$) and compressive ($\sigma^-(u, p)$) parts of the stress tensor. For this reason, the positive part of the pressure $p^+ \in L_2(\Omega)$ has to be defined as $p^+ := \max\{p, 0\}$, and $E^+_{\mathrm{lin}}(u)$ is given as the projection of $E_{\mathrm{lin}}(u)$ onto positive semidefinite matrices. Now, we can split the stress tensor $\sigma(u, p)$ as:

$$\begin{aligned} \sigma^+(u, p) &:= 2\mu E^+_{\mathrm{lin}}(u) + p^+\mathbf{I}, \\ \sigma^-(u, p) &:= 2\mu(E_{\mathrm{lin}}(u) - E^+_{\mathrm{lin}}(u)) + (p - p^+)\mathbf{I}. \end{aligned}$$

In the following, the critical energy release rate is denoted by G_c and a degradation function is defined as $g(\varphi) := (1-\kappa)\varphi^2+\kappa$, with a small regularization parameter $\kappa > 0$. Next, we can formulate the mixed phase-field problem in incremental form [7]:

Problem 1 (Mixed Phase-Field Formulation) Given the initial data $\varphi^{n-1} \in \mathcal{K}$, find $u := u^n \in \{u_D + \mathcal{V}\}$, $p := p^n \in \mathcal{U}$ and $\varphi := \varphi^n \in \mathcal{K}$ for loading steps $n = 1, 2, \ldots, N$ such that

$$\begin{aligned} g(\varphi^{n-1})(\sigma^+(u, p), E_{\mathrm{lin}}(w)) + (\sigma^-(u, p), E_{\mathrm{lin}}(w)) &= 0 \quad \forall\, w \in \mathcal{V}, \\ (\nabla \cdot u, q) - {}^1\!/\!_\lambda(p, q) &= 0 \quad \forall\, q \in \mathcal{U}, \\ (1-\kappa)(\varphi\sigma^+(u, p) : E_{\mathrm{lin}}(u), \psi - \varphi) + G_c(-{}^1\!/\!_\epsilon(1-\varphi), \psi - \varphi) & \\ +G_c\epsilon(\nabla\varphi, \nabla(\psi - \varphi)) &\geq 0 \quad \forall\, \psi \in \mathcal{K} \subset \mathcal{W}, \end{aligned}$$

where $\epsilon > 0$ describes the bandwidth of the transition zone between broken and unbroken material. This weak formulation in Problem 1 can be reformulated to a complementarity system by introducing a Lagrange multiplier $\Lambda \in \mathcal{X}$, see [6, 7].

The numerical treatment of the phase-field system in a monolithic fashion including the discretization as well as the adaptive refinement strategy are discussed in the following.

2.3 Numerical Treatment and Programming Code

Based on the complementarity formulation of Problem 1, with the help of a Lagrange multiplier, the crack irreversibility constraint is enforced, see [7, Section 4.1]. For the discretization in space, we employ a Galerkin finite element method in each loading step. To this end, the domain Ω is partitioned into quadrilaterals. To fulfill a discrete inf-sup condition, Taylor-Hood elements with biquadratic shape functions (Q_2) for the displacement field u and bilinear shape functions (Q_1) for the pressure variable p as well as for the phase-field variable are used. For further details on the stable mixed form of the classical phase-field fracture model as well as the handling of the crack irreversibility condition and the numerical solving steps, we refer to [7].

The overall implementation is done in DOpElib [2, 4] using the finite element library deal.II [1].

2.4 Adaptive Refinement

A residual-type a posteriori error estimator η for the classical phase-field fracture model, presented and tested in [6], provides a robust upper bound. Here, robust means that the unknown constant in the bound does not depend on ϵ such that the quality of the estimator is independent of ϵ. The mesh adaptation is realized using extracted local error indicators from the a posteriori error estimator in [6, Section 3.2] on the given meshes over all loading steps.

In the following, $\mathfrak{M}^n$ denotes the mesh in the incremental step n and I_h^n is the corresponding nodal interpolation operator on the mesh $\mathfrak{M}^n$. The searched discrete quantities are denoted by an index $(\cdot)_h$, i.e., the displacement u_h^n, the phase-field variable φ_h^n, the pressure p_h^n, and the Lagrange multiplier Λ_h^n. The adaptive solution strategy is given in the following.

Algorithm Given a partition in time $t_0 < \ldots < t_N$, and an initial mesh $\mathfrak{M}^n = \mathfrak{M}$ for all $n = 0, \ldots, N$.

1. Set $\varphi_h^0 = I_h^0 \varphi^0$ and solve the discrete complementarity system to obtain the discrete solutions $\boldsymbol{u}_h^n, \varphi_h^n, p_h^n, \Lambda_h^n$ for all $n = 1, \ldots, N$.
2. Evaluate the error estimator in order to obtain η^n for each incremental step.
3. Stop, if $\sum_{n=1}^N (\eta^n)^2$ and $\|I_h^n \varphi^{n-1} - \varphi^{n-1}\|$ are small enough for all $n = 1, \ldots, N$.
4. For each $n = 1, \ldots N$, mark elements in $\mathfrak{M}^n$ based on η^n according to an optimization strategy, as implemented in deal.II [1].
5. Refine the meshes according to the marking and satisfaction of the constraints on hanging nodes.
6. Repeat from step 1.

3 Numerical Results

In this section, the mixed phase-field model formulation is applied to simulate crack propagation in an L-shaped specimen with the help of adaptive refined meshes. First, the setup of the L-shaped panel test and the corresponding material and numerical parameters are given. Afterwards, the load-displacement curves and the crack paths are discussed for three different Poisson ratios from the standard setting towards the incompressible limit $\nu \to 0.5$.

3.1 Configuration of the L-shaped Panel Test

The L-shaped panel test was originally developed by Winkler [12] to test the crack pattern of concrete experimentally and numerically. Concrete is compressible with a Poisson ratio of $\nu = 0.18$. To simulate fracture propagation in nearly incompressible materials, within this work, the Poisson's ratio is increased towards an incompressible solid.

In Fig. 1, the test geometry of the L-shaped panel test is declared. In the right corner Γ_{u_y} on a small stripe of 30 mm at the boundary, a special displacement condition is defined as a loading-dependent non-homogeneous Dirichlet condition:

$$u_y = t \cdot \text{mm/s}, \quad \text{for } t \in I := [0; 0.4\ \text{s}],$$

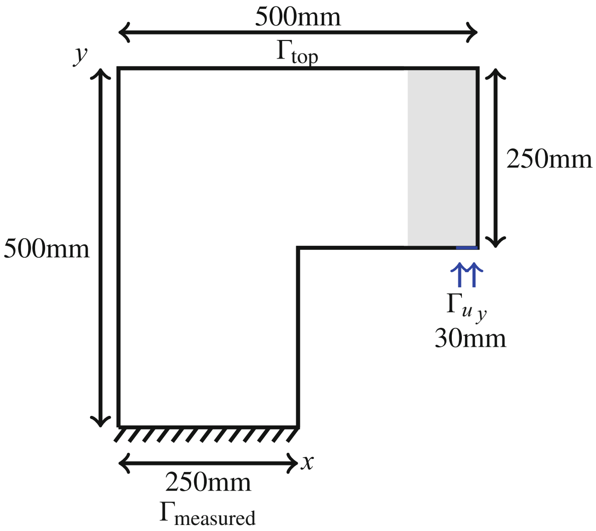


Fig. 1 Geometry and boundary conditions of the L-shaped panel test

where t denotes the total time and $T = 0.4$ s is the end time which corresponds to a displacement of 0.4mm. The time interval I is divided into steps of the loading size δt.

Remark 1 To avoid developing unphysical cracks in the singularity on the boundary Γ_{u_y}, the domain where the phase-field inequality is solved, is constrained to the subset given by $x <= 400$ mm similar to [8]. For $x > 400$ mm we assume $\varphi = 1$.

In Table 1, the required material and numerical parameters for the L-shaped panel test are listed. Keep in mind, that the given values for μ and λ fit to the original material concrete and are changed for other values of ν in the following numerical tests, as listed in Table 2. Further, the discretization parameter h in Table 1 changes within the refinement steps, so h is the starting mesh parameter on the coarsest mesh before adaptive refining.

In Table 3, the minimal and maximal number of degrees of freedom are given for three different test cases $\nu = 0.18$, $\nu = 0.40$ and $\nu = 0.49$. The adaptive computations are based on a three times uniform refined mesh and three adaptive refinement steps. For comparison, also the load-displacement curves for tests, executed on a four times uniform refined mesh, are added in Fig. 2. The load-displacement curves in Fig. 2 indicate that the higher the Poisson ratio, the higher

Table 1 Standard settings of the material and numerical parameters for the L-shaped panel test

Parameter	Description	Value
μ	Lamé coefficient	$10.95\ \mathrm{kN/mm^2}$
λ	Lamé coefficient	$6.16\ \mathrm{kN/mm^2}$
ν	Poisson's ratio	0.18
G_c	Critical energy rate	8.9×10^{-5} kN/mm
h	Discretization parameter	7.289 mm
ϵ	Bandwidth	14.0 mm
δt	Incremental size	10^{-4} s
$\mathcal{I}$	End time	0.4 s
κ	Regularization parameter	10^{-10}

Table 2 Tests with different Poisson's ratios

ν	μ	λ
0.18	$10.95 \cdot 10^3$	$6.18 \cdot 10^3$
0.40	$10.95 \cdot 10^3$	$42.36 \cdot 10^3$
0.49	$10.95 \cdot 10^3$	$51.89 \cdot 10^4$

Table 3 The minimal and maximal number of degrees of freedom (DoF) per incremental step on adaptive meshes

ν	min. #DoF	max. #DoF
0.18	Uniform	213,445
0.18	53,925	125,599
0.40	Uniform	213,445
0.40	53,925	121,709
0.49	Uniform	213,445
0.49	53,925	91,574

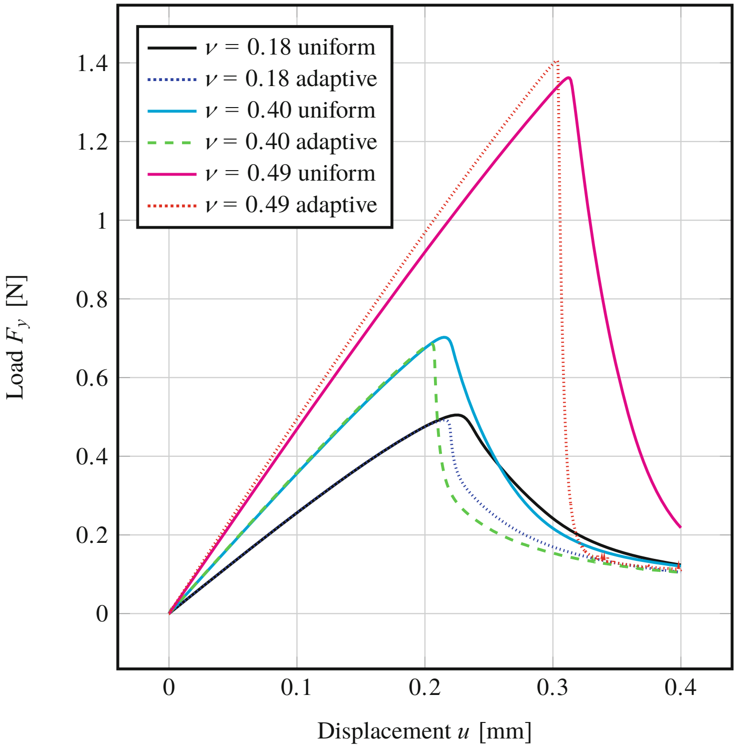


Fig. 2 Load-displacement curves for the L-shaped panel test with different Poisson ratios and adaptively refined meshes versus uniform refinement. Weighted loading measured on the lower left boundary Γ_{measured} labeled in Fig. 1

is the maximal loading value before the crack starts propagating. Further, the path of the load-displacement curves for $\nu = 0.18$, in particular for the adaptive test run in Fig. 2, coincide with the numerical and experimental results in concrete [12]. In general, the adaptive computations exhibit a faster crack growth as it is expected in brittle materials, and may call for additional adaptive refinement of the time discretization for which models and indicators still need to be developed. As a second quantity of interest, in Fig. 3, the crack path can be observed in certain incremental steps on adaptive refined meshes, exemplary for $\nu = 0.40$. The refinement strategy based on the error indicators steers the resolution of the crack area, especially of the crack tip as visualized in Fig. 4.

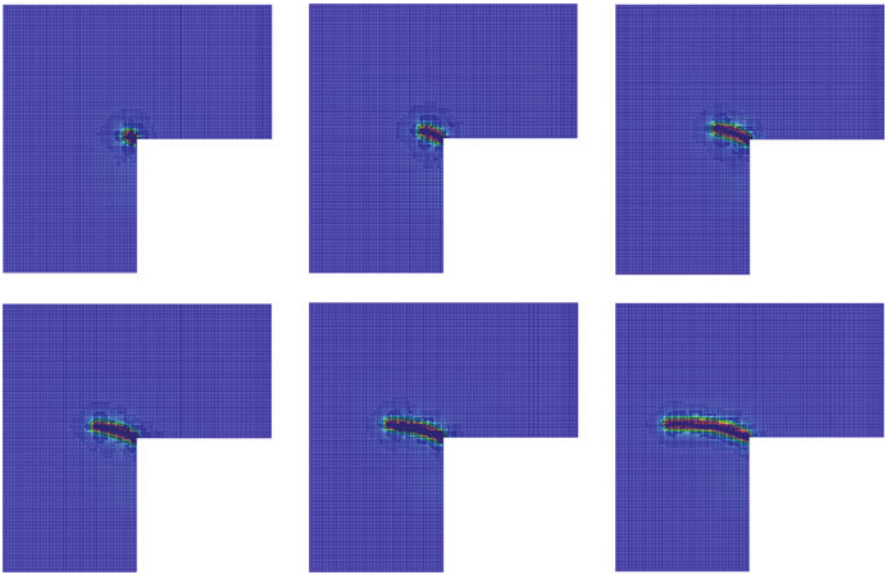


Fig. 3 Poisson's ratio $\nu = 0.40$. Snapshots of the phase-field function after three adaptive refinement steps in the incremental steps 0.2082, 0.209, 0.2099, 0.2136, 0.2323 and 0.2997s on the current adaptive mesh

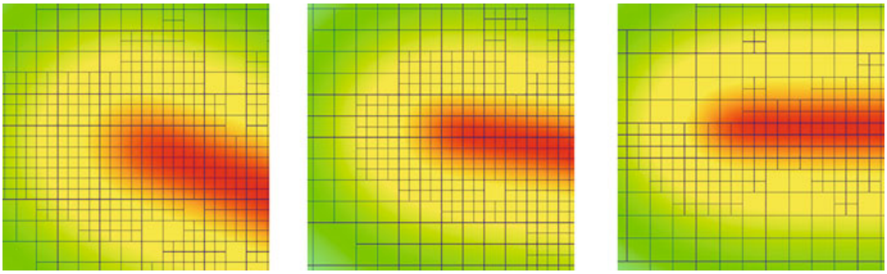


Fig. 4 Poisson's ratio $\nu = 0.40$. Enhanced extract of the phase-field function in the crack tip after three adaptive refinement steps in the incremental steps 0.2099, 0.2176 and 0.2997s

4 Conclusion

We have combined and extended [7] and [6] to adaptive refinement based on robust residual-type a posteriori error estimators for phase-field model for fractures in nearly incompressible materials. The method is demonstrated on a numerical test for the L-shaped panel test. Therefore, we proposed three test cases in Sect. 3 with different Poisson ratios ν approximating the incompressible limit $\nu = 0.5$. The load-displacement curves of the three tests show a correlation between an increasing Poisson ratio and a stronger loading force. In view of mesh adaptivity, we observed very convincing findings: the mesh refinement is localized in the area of the (a priori unknown) fracture path and allows to resolve the crack tip region. Further, our adaptive refined meshes allow for a faster crack growth compared to uniformly refined meshes.

Acknowledgments This work was funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation)—392587580. It is part of the Priority Program 1748 (DFG SPP 1748) *Reliable Simulation Techniques in Solid Mechanics. Development of Non-standard Discretization Methods, Mechanical and Mathematical Analysis*.

References

1. D. Arndt, W. Bangerth, D. Davydov, T. Heister, L. Heltai, M. Kronbichler, M. Maier, J.-P. Pelteret, B. Turcksin, and D. Wells. The `deal.II` library, version 8.5. *Journal of Numerical Mathematics*, 25(3):137–146, 2017.
2. The Differential Equation and Optimization Environment: DOpElib. http://www.dopelib.net.
3. G. A. Francfort and J.-J. Marigo. Revisiting brittle fracture as an energy minimization problem. *Journal of the Mechanics and Physics of Solids*, 46(8):1319–1342, 1998.
4. C. Goll, T. Wick, and W. Wollner. DOpElib: Differential equations and optimization environment; A goal oriented software library for solving PDEs and optimization problems with PDEs. *Archive of Numerical Software*, 5(2):1–14, 2017.
5. A. A. Griffith. The phenomena of flow and rupture in solids. *Philosophical Transactions of the Royal Society A*, 221:163–198, 1921.
6. K. Mang, M. Walloth, T. Wick, and W. Wollner. Mesh adaptivity for quasi-static phase-field fractures based on a residual-type a posteriori error estimator. *GAMM Mitteilungen*, 43:e202000003, 2020.
7. K. Mang, T. Wick, and W. Wollner. A phase-field model for fractures in nearly incompressible solids. *Computational Mechanics*, 65:61–78, 2020.
8. A. Mesgarnejad, B. Bourdin, and M. M. Khonsari. Validation simulations for the variational approach to fracture. *Computer Methods in Applied Mechanics and Engineering*, 290:420–437, 2015.
9. C. Miehe, M. Hofacker, and F. Welschinger. A phase field model for rate-independent crack propagation: Robust algorithmic implementation based on operator splits. *Computer Methods in Applied Mechanics and Engineering*, 199(45–48):2765–2778, 2010.
10. C. Miehe, F. Welschinger, and M. Hofacker. Thermodynamically consistent phase-field models of fracture: variational principles and multi-field FE implementations. *International Journal for Numerical Methods in Fluids*, 83(10):1273–1311, 2010.
11. M. Walloth. Residual-type a posteriori estimators for a singularly perturbed reaction-diffusion variational inequality – reliability, efficiency and robustness. Technical Report 1812.01957, arXiv, 2018.
12. B. J. Winkler. *Traglastuntersuchungen von unbewehrten und bewehrten Betonstrukturen auf der Grundlage eines objektiven Werkstoffgesetzes für Beton*. Innsbruck University Press, 2001.

Convergence Rates for Matrix P-Greedy Variants

Dominik Wittwar and Bernard Haasdonk

Abstract When using kernel interpolation techniques for constructing a surrogate model from given data, the choice of interpolation points is crucial for the quality of the surrogate. When dealing with vector-valued target functions which are approximated by matrix-valued kernel models, the selection problem is further complicated as not only the choice of points but also the directions in which the data is projected must be determined.

We thus propose variants of Matrix P-greedy algorithms that enable us to iteratively select suitable sets of point-direction pairs with which the approximation space is enriched. We show that the selected pairs result in quasi-optimal convergence rates. Experimentally, we investigate the approximation quality of the different variants.

1 Introduction

Kernel methods are useful tools for constructing surrogate models using scattered data [9]. One important task for constructing these surrogates is to determine where a target function should be sampled to obtain sparse surrogates with high accuracy. As it was recently shown for scalar-valued kernels, the data sites determined by the P-greedy algorithm result in quasi-optimal decay rates [8]. When dealing with vector-valued target functions an approximation approach based on matrix-valued kernels is beneficial as correlations in the target function components can be incorporated in contrast to an approach using scalar-valued kernels for each individual target function component.

In Sect. 2 we give a short introduction to matrix-valued kernels. We extend the scalar P-greedy procedure to the matrix-valued case in Sect. 3 and continue

D. Wittwar (✉) · B. Haasdonk
University of Stuttgart, Stuttgart, Germany
e-mail: dominik.wittwar@ians.uni-stuttgart.de; bernard.haasdonk@ians.uni-stuttgart.de

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_119

in Sect. 4 by showing that the proposed variants result in approximations which maintain the same quasi-optimal convergence rates. We conclude with a numerical example highlighting the different benefits of the variants.

2 Approximation with Matrix-Valued Kernels

We give a short overview on matrix-valued kernels. For a more extensive introduction to this topic we refer to [1, 5, 6]. A positive definite matrix-valued kernel is a bivariate function $K : \Omega \times \Omega \subset \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{m\times m}$ such that $K(x, y) = K(y, x)^T$ for any $x, y \in \Omega$ and for all pairwise distinct point sets $X = \{x_1, \dots, x_n\} \subset \Omega$, $n \in \mathbb{N}$ the block matrix $A \in \mathbb{R}^{mn\times mn}$ given by blocks

$$A_{ij} := K(X, X)_{ij} := K(x_i, x_j) \in \mathbb{R}^{m\times m} \tag{1}$$

is positive semi-semidefinite. In particular $K(x, x)$ is symmetric and positive semi-definite for all $x \in \Omega$. Each such kernel corresponds to a Hilbert space $\mathcal{H}$ of vector-valued functions $f : \Omega \to \mathbb{R}^m$, the so called Reproducing Kernel Hilbert space (RKHS), which can be uniquely characterised by the following properties

$$K(\cdot, x)\alpha \in \mathcal{H} \text{ and } \langle f, K(\cdot, x)\alpha)\rangle_{\mathcal{H}} = f(x)^T\alpha, \quad \text{for all } f \in \mathcal{H}, x \in \Omega, \alpha \in \mathbb{R}^m. \tag{2}$$

Any subspace $\mathcal{N}(X) := \text{span}\{K(\cdot, x)\alpha | x \in X, \alpha \in \mathbb{R}^m\}$ consisting of the span of columns of the evaluation of the kernel at finite point sets $X \subset \Omega$ is then again a RKHS and its corresponding kernel $K_{\mathcal{N}(X)}$ can be evaluated using K and the orthogonal projection $\Pi_{\mathcal{N}(X)} : \mathcal{H} \to \mathcal{N}(X)$, i.e.

$$K_{\mathcal{N}(X)}(x, y)\alpha = (\Pi_{\mathcal{N}(X)}K(\cdot, y)\alpha)(x), \quad \text{for all } x, y \in \Omega, \alpha \in \mathbb{R}^m. \tag{3}$$

With the gram matrix $A = K(X, X)$ this can alternatively be expressed as

$$K_{\mathcal{N}(X)}(x, y) = K(x, X)A^{-1}K(X, y), \tag{4}$$

where $K(x, X) \in \mathbb{R}^{m\times nm}$ is a concatenation of the matrices $K(x, \hat{x})$, $\hat{x} \in X$. In a similar fashion we can express the best approximation of f in $\mathcal{N}(X)$, which coincides with the interpolant of f on X, by

$$\Pi_{\mathcal{N}(X)} f(x) = K(x, X)A^{-1} f(X). \tag{5}$$

Furthermore, the reproducing property (2) leads to a bound on the pointwise error between any $f \in \mathcal{H}$ and its best approximation in $\mathcal{N}(X)$ in terms of the Power-function matrix $\mathbf{P}_{\mathcal{N}(X)}(x) = K(x,x) - K_{\mathcal{N}(X)}(x,x)$ via

$$\left(f(x) - \Pi_{\mathcal{N}(X)} f(x)\right)^T \alpha \leq \left(\alpha^T \mathbf{P}_{\mathcal{N}(X)}(x)\alpha\right)^{1/2} \|f\|_{\mathcal{H}} \quad \text{for all } x \in \Omega, \alpha \in \mathbb{R}^m.$$

Depending on the choice of α this leads to bounds in the different p-norms:

$$\|f(x) - \Pi_{\mathcal{N}(X)} f(x)\|_1 \leq \operatorname{trace}(\mathbf{P}_{\mathcal{N}(X)}(x))^{1/2} \|f\|_{\mathcal{H}}, \tag{6}$$

$$\|f(x) - \Pi_{\mathcal{N}(X)} f(x)\|_\infty \leq (\max \operatorname{diag}(\mathbf{P}_{\mathcal{N}(X)}))^{1/2} \|f\|_{\mathcal{H}}, \tag{7}$$

$$\|f(x) - \Pi_{\mathcal{N}(X)} f(x)\|_2 \leq (\lambda_{\max}(\mathbf{P}_{\mathcal{N}(X)}(x)))^{1/2} \|f\|_{\mathcal{H}}. \tag{8}$$

In the scalar case, i.e. $m = 1$ all these bounds are equivalent and $\mathbf{P}^{1/2}_{\mathcal{N}(X)}$ is equal to the so called Power-function.

Remark 1 Instead of all columns of $K(\cdot, x)$ one may also only consider certain directions $K(\cdot, x)\alpha$. This leads to subspaces of the form

$$\mathcal{N} = \operatorname{span}\{K(\cdot, x_i)\alpha_i | (x_i, \alpha_i) \in \Omega \times \mathbb{R}^m\}$$

for a finite set of tuples $\{(x_i, \alpha_i)\}_{i=1}^n$. In this case similar results for (3)–(8) hold, where the blocks of A given in (1) are replaced by

$$A_{ij} = \alpha_i^T K(x_i, x_j)\alpha_j.$$

3 Matrix P-greedy Algorithms

In this section we want to address how the interpolation points X can be chosen. To this end we employ P-greedy algorithms which have been shown to result in quasi-optimal approximation rates for the approximation of scalar-valued functions [8]. The basic principle of the P-greedy algorithm using matrix-valued kernels (Matrix P-greedy) is outlined in Algorithm 1.

Algorithm 1 Matrix P-greedy Algorithm

```
Require: finite sampling of the input domain $\Omega_N \subset \Omega$, kernel $K : \Omega \times \Omega \to \mathbb{R}^{m \times m}$, initial
  approximation space $\mathcal{N}$, error indicator function $E$, tolerance $\varepsilon > 0$, space extension routine
  "extend"
  while $\max_{x \in \Omega_N} E(\mathbf{P}_{\mathcal{N}}(x)) \geq \varepsilon$ do
      $x^* = \arg\max_{x \in \Omega_N} E(\mathbf{P}_{\mathcal{N}}(x))$
      $\mathcal{N} = \text{extend}(\mathcal{N}, K(\cdot, x^*))$
  end while
  return $\mathcal{N}$
```

For the error indicator function $E : \mathbb{R}^{m \times m} \to \mathbb{R}$ we propose, based on the bounds given in (6)–(8), the following three variants

$$E_1(B) := \frac{1}{m}\text{trace}(B), \quad E_\infty(B) := \max \text{diag}(B), \quad E_2(B) := \lambda_{\max}(B). \tag{9}$$

For the extension routine we select

$$\begin{aligned} \text{extend}_{\text{full}}(\mathcal{N}, K(\cdot, x)) &:= \mathcal{N} + \text{colspan}(K(\cdot, x)), \\ \text{extend}_{\text{eig}}(\mathcal{N}, K(\cdot, x)) &:= \mathcal{N} + \text{span}(K(\cdot, x)\alpha_{\max}), \\ \text{extend}_{\text{diag}}(\mathcal{N}, K(\cdot, x)) &:= \mathcal{N} + \text{span}(K(\cdot, x)e_{\max}), \end{aligned}$$

where $\alpha_{\max}$ denotes an eigenvector to the largest eigenvalue and $e_{\max}$ the standard basis vector to the largest diagonal value of $\mathbf{P}_{\mathcal{N}}(x)$, respectively.

As the name suggests extend$_{\text{full}}$ extends the approximation space by an m dimensional subspace, which leads to a rapid increase in the dimension of the approximation space for large m. However, all components of the target function data $f(X)$ are incorporated into the approximation, see (5). In contrast, extend$_{\text{eig}}$ and extend$_{\text{diag}}$ increase the dimension only by 1. Therefore, when approximating a target function with a space constructed by extend$_{\text{eig}}$ or extend$_{\text{diag}}$, this might require a larger number of individual target function evaluations compared to a space construction by extend$_{\text{full}}$. This happens as only projections of the target function data $f(x_i)^T\alpha_i$ are used in the construction of the approximant. We will consider all possible combinations and denote the combined routine as greedy$_{p,\text{type}}$ with $p \in \{1, 2, \infty\}$ and type $\in$ {full, eig, diag}.

4 Convergence Rates for *k*-Dimensional Greedy Space Extensions

The different P-greedy variants described in Sect. 3 can be interpreted as weak greedy algorithms [2] which enrich the approximation space with a subspace of dimension $k \geq 1$ in the following sense: Assume that $\mathcal{H}$ is a Hilbert space and let $\mathcal{F} \subset \mathcal{H}$ be a compact set. Starting with an initial space V_0 the weak greedy with constant $0 < \gamma \leq 1$ and subspace increment dimension k iteratively selects a sequence $(W_n)_{n\in\mathbb{N}}$ of k-dimensional subspaces in the following way:

1. Select $W_n \subset \text{span}\{\mathcal{F}\}$, $\dim(W_n) = k$ such that

$$\max_{f \in W_n} \|f - \Pi_{V_{n-1}} f\|_{\mathcal{H}} \geq \gamma \max_{f \in \mathcal{F}} \|f - \Pi_{V_{n-1}} f\|_{\mathcal{H}}$$

.

2. Extend the space $V_n := V_{n-1} + W_n$.

Building on the results provided in [3] for the case of $k = 1$, the approximation quality of the best approximation in V_n with dimension $N = n \cdot k$ can be related to the Kolmogorov N-width $d_N(\mathcal{F})$ given by

$$d_N(\mathcal{F}) = \inf_{\mathcal{N}} \sup_{f \in \mathcal{F}} \|f - \Pi_{\mathcal{N}} f\|_{\mathcal{H}}$$

where the infimum is taken over all N-dimensional subspaces $\mathcal{N} \subset \mathcal{H}$.

Theorem 1 *Let $(W_n)_{n\in\mathbb{N}}$ be the sequence of subspace increments and $(V_n)_{n\in\mathbb{N}_0}$ be the sequence of spaces chosen by a weak greedy algorithm with constant γ. Let*

$$\sigma_n := \max_{f \in \mathcal{F}} \|f - \Pi_{V_n} f\|_{\mathcal{H}}$$

and $d_N(\mathcal{F})$ be the Kolmogorov N-width for the set $\mathcal{F}$ then it holds with $C_0, c_0, \alpha > 0$

1. *If $d_N(\mathcal{F}) \le C_0 N^{-\alpha}$ then $\sigma_n \le C_1 N^{-\alpha}$ with $C_1 := 2^{7\alpha+1} k^{\alpha} \gamma^{-2} C_0$*
2. *If $d_N(\mathcal{F}) \le C_0 e^{-c_0 N^{\alpha}}$ then $\sigma_n \le \sqrt{2C_0}\gamma^{-1} e^{-c_1 N^{\alpha}}$ with $c_1 = 2^{-1-4\alpha} k^{-\alpha} c_0$.*

Proof Let

$$g_n^1 := \arg \max_{f \in W_n} \|g_n^i - \Pi_{V_{n-1}} g_n^i\|_{\mathcal{H}}$$

and $g_n^2, \ldots, g_n^k$ such that $g_n^1, \ldots, g_n^k$ form a Basis of W_n. Let now $\{\hat{g}_n^i | n \in \mathbb{N},\ i = 1, \ldots, k\}$ denote the orthonormal system generated by applying the Gram-Schmidt orthonormalization to $\{g_1^1, g_1^2, \ldots, g_1^k, g_2^1, \ldots\}$. Using the results provided in [3] it is sufficient to show that the infinite lower-triangular matrix A given by

$$A := (a_{ij})_{i,j=0}^{\infty}, \quad a_{i,j} = \langle g_i^1, \hat{g}_j^1 \rangle_{\mathcal{H}}$$

meets the following conditions

1. The diagonal elements satisfy $\gamma\sigma_{n-1} \le |a_{nn}| \le \sigma_{n-1}$.
2. For every $M \ge n$, one has $\sum_{j=n}^{M} a_{Mj}^2 \le \sigma_{n-1}^2$.

By construction of $\hat{g}_n^1$ we have

$$\hat{g}_n^1 = \left(g_n^1 - \Pi_{V_{n-1}} g_n^1\right) / \left\| g_n^1 - \Pi_{V_{n-1}} g_n^1 \right\|$$

and therefore

$$a_{nn} = \langle g_n^1, \hat{g}_n^1 \rangle_{\mathcal{H}} = \left\| g_n^1 - \Pi_{V_{n-1}} g_n^1 \right\| .$$

Hence the first condition is satisfied by definition of σ_n and choice of g_n^1:

$$\gamma\sigma_{n-1} = \gamma \max_{f\in\mathcal{F}} \|f - \Pi_{V_{n-1}} f\|_{\mathcal{H}} \leq \|g_n^1 - \Pi_{V_{n-1}} g_n^1\|_{\mathcal{H}} \leq \max_{f\in\mathcal{F}} \|f - \Pi_{V_{n-1}} f\|_{\mathcal{H}} = \sigma_{n-1}.$$

Since the sequence of spaces is nested, i.e. $V_n \subset V_{n+1}$ we have

$$\sum_{j=n}^{M} a_{Mj}^2 = \sum_{j=n}^{M} \langle g_M^1, \hat{g}_j^1\rangle_{\mathcal{H}}^2 \leq \sum_{j=n}^{M}\sum_{i=1}^{m} \langle g_M^1, \hat{g}_j^i\rangle_{\mathcal{H}}^2$$

$$= \left\| g_M^1 - \Pi_{V_{n-1}} g_M^1 \right\|^2 \leq \max_{f\in\mathcal{F}} \|f - \Pi_{V_{n-1}} f\|_{\mathcal{H}} = \sigma_{n-1}^2.$$

this concludes the proof. □

To apply the results of Theorem 1 we now only have to verify that the Matrix P-greedy algorithm with the indicator functions given in (9) and the different extension routines is indeed a weak greedy with k-dimensional increments. To this end let $(V_n)_{n\in\mathbb{N}}$ denote the nested sequence of spaces selected by the Matrix P-greedy algorithm. We first note that in our case we have $\mathcal{F} = \{K(\cdot, x)\alpha | x \in \Omega, \alpha \in \mathbb{R}^m, \|\alpha\| = 1\}$ and, therefore, by applying (3)

$$\begin{aligned}\sigma_n = \max_{f\in\mathcal{H}} \|f - \Pi_{V_n} f\| &= \max_{(x,\alpha)\in\Omega\times\mathbb{R}^m, \|\alpha\|=1} \|K(\cdot, x)\alpha - K_{V_n}(\cdot, x)\alpha\| \\ &= \max_{(x,\alpha)\in\Omega\times\mathbb{R}^m, \|\alpha\|=1} (\alpha^T(K(x,x) - K_{V_n}(x,x))\alpha)^{1/2} \\ &= \max_{x\in\Omega} (\lambda_{\max}(\mathbf{P}_{V_n}(x)))^{1/2}.\end{aligned}$$

We immediately conclude, that greedy$_{2,\text{full}}$ and greedy$_{2,\text{eig}}$ lead to a strong, i.e. $\gamma = 1$ greedy. For the remaining combinations we make use of the following inequality chain for symmetric positive definite matrices $B \in \mathbb{R}^{m\times m}$

$$\max \operatorname{diag}(B) \leq \lambda_{\max}(B) \leq \operatorname{trace}(B) \leq m \max \operatorname{diag}(B) \leq m\lambda_{\max}(B),$$

which leads to $\gamma \geq m^{-2}$ in the remaining cases. Please note that the routine extend$_{1,\text{eig}}$ is equivalent to the POD-greedy algorithm used in the context of reduced basis approximation for which similar rates have been obtained [4].

Note that similar to [7] bounds on the Kolmogorov N-width can be obtained for matrix-valued kernels.

5 Numerical Experiment

For a test case we consider the domain $\Omega = [-1, 1]$ and the kernel $K : \Omega \times \Omega \to \mathbb{R}^{10\times 10}$ given by

$$K(x, y) = e^{-4\|x-y\|^2} A_1 + e^{-10\|x-y\|^2} A_2$$

for two random, symmetric positive definite matrices $A_1, A_2 \in \mathbb{R}^{10\times 10}$. As a target function $f \in \mathcal{H}$ we consider

$$f(x) = \sum_{i=1}^{10} K(x, x_i)\alpha_i,$$

where we chose 10 randomly selected points $x_1, \dots, x_{10} \in \Omega$ and coefficient vectors $\alpha_1, \dots, \alpha_{10} \in \mathbb{R}^{10}$. For a first test, we run the Matrix P-greedy for the different indicator functions and the full extension routine for 30 iterations. The decay of the maximum indicator function values are depicted in Fig. 1. As can be seen, the maximum values decay at a similar rate. However, using E_2 results in a higher computational effort, as in every iteration multiple eigenvalue problems have to be solved which further necessitates the entire Power-function matrix $\mathbf{P}$ to be evaluated. In contrast E_1 and E_∞ only rely on the diagonal values of $\mathbf{P}$ and no eigenvalue problems have to be solved.

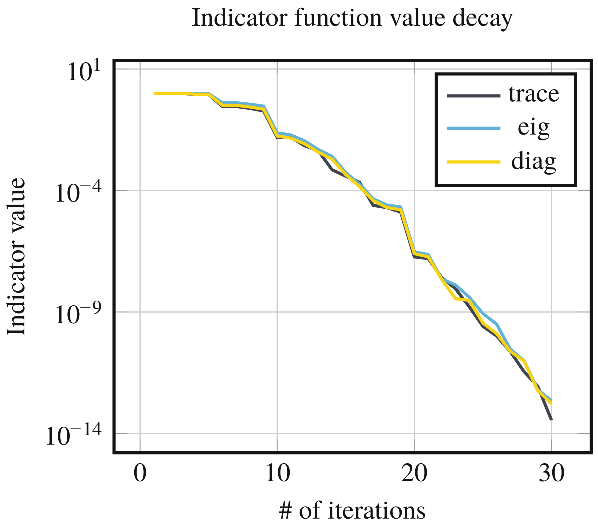


Fig. 1 Decay of the maximum indicator function value with respect to the number of iterations of the Matrix P-greedy

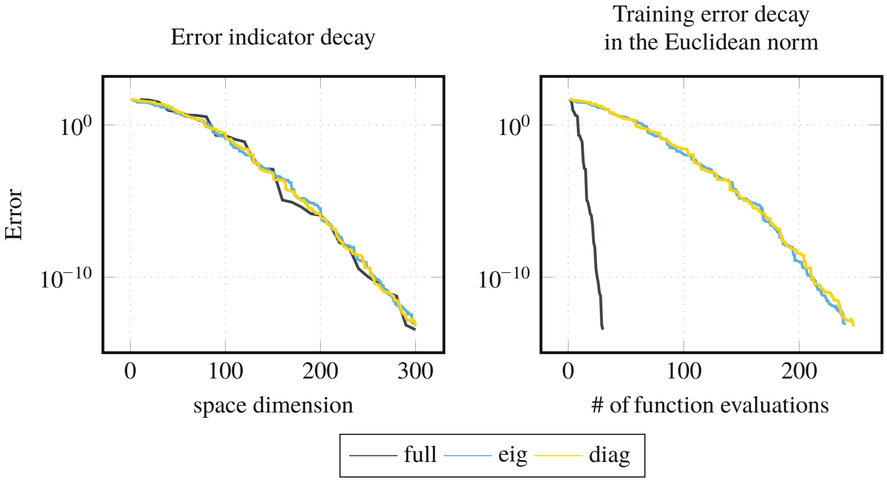


Fig. 2 Error decay rates with respect to the approximation space dimension (left) and number of function evaluations (right)

We thus decide on using E_1 for investigating the performance of the different extension routines. We compute sequences of interpolants $(s^n_{1,\text{full}})_{1\le n\le 30}$, $(s^n_{1,\text{eig}})_{1\le n\le 300}$ and $(s^n_{1,\text{diag}})_{1\le n\le 300}$ as well as the sequences of errors in the squared native space norm

$$\Delta^n_{1,\text{type}} = \|f - s^n_{1,\text{type}}\|^2_{\mathcal{H}}, \quad \text{type} \in \{\text{full, eig, diag}\}.$$

Recall that for the full extension routine the approximation space dimension is increased by 10 in every iteration. Hence the sequences for the other extension routines have $300 = 30 \cdot 10$ elements to result in the same target dimension $N = k \cdot n$. The error decay rates with respect to the approximation space dimension and the number of target function evaluations necessary for the construction of the interpolant are depicted in Fig. 2. As we can see the quality of the interpolant is almost identical for all extension methods, However, extend$_{\text{eig}}$ and extend$_{\text{diag}}$ require 239 and 246 unique function evaluation, respectively in contrast to extend$_{\text{full}}$ which only requires 30. Thus, extend$_{\text{full}}$ seems to be the preferred choice, if the target function is expensive to evaluate.

Acknowledgments The authors acknowledge funding by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy—EXC 2075—390740016.

References

1. Alvarez, M., Rosasco, L., Lawrence, N.D.: Kernels for vector-valued functions: a review. Foundations and Trends in Machine Learning **4**(3), 195–266 (2012)
2. Binev, P., Cohen, A., Dahmen, W., DeVore, R., Petrova, G., Wojtaszczyk, P.: Convergence rates for greedy algorithms in reduced basis methods. SIAM J. Math. Anal. **43**(3), 1457–1472 (2011). http://dx.doi.org/10.1137/100795772
3. DeVore, R., Petrova, G., Wojtaszczyk, P.: Greedy algorithms for reduced bases in Banach spaces. Constr. Approx. **37**(3), 455–466 (2013). http://dx.doi.org/10.1007/s00365-013-9186-2
4. Haasdonk, B.: Convergence rates of the POD–Greedy method. ESAIM: Mathematical Modelling and Numerical Analysis **47**(3), 859–873 (2013). http://dx.doi.org/10.1051/m2an/2012045
5. Micchelli, C.A., Pontil, M.: On learning vector-valued functions. Neural Comput. **17**(1), 177–204 (2005). http://dx.doi.org/10.1162/0899766052530802
6. Reisert, M., Burkhardt, H.: Learning equivariant functions with matrix valued kernels. J. Mach. Learn. Res. **8**, 385–408 (2007). http://dl.acm.org/citation.cfm?id=1248659.1248674
7. Rieger, C., Zwicknagl, B.: Sampling inequalities for infinitely smooth functions, with applications to interpolation and machine learning. Adv. Comput. Math. **32**(1), 103–129 (2008). http://dx.doi.org/10.1007/s10444-008-9089-0
8. Santin, G., Haasdonk, B.: Convergence rate of the data-independent P-greedy algorithm in kernel-based approximation. Dolomites Res. Notes Approx. **10**, 68–78 (2017). www.emis.de/journals/DRNA/9-2.html
9. Wendland, H.: Scattered Data Approximation. Cambridge University Press (2004). http://dx.doi.org/10.1017/CBO9780511617539. Cambridge Books Online

Efficient Solvers for Time-Periodic Parabolic Optimal Control Problems Using Two-Sided Bounds of Cost Functionals

Monika Wolfmayr

Abstract This article is devoted to presenting efficient solvers for time-periodic parabolic optimization problems. The solvers are based on deriving two-sided bounds for the cost functional. Here, we especially employ the time-periodic nature of the problem discussed in order to obtain fully computable and guaranteed upper and lower bounds for the cost functional. We present the multiharmonic finite element method as a proper approach for deriving a discretized solution of the time-periodic problem. The multiharmonic finite element functions can be used as initial guess for the arbitrary functions in the upper and lower bounds, which then can be minimized and maximized, respectively, in order to obtain an approximate solution of any desired accuracy. Finally, new numerical results are presented in order to show the efficiency of the method discussed also in practice.

1 Introduction

This article is devoted to the presentation of efficient solvers which are based on the time-periodic setting of the parabolic optimal control problem. For that, two-sided bounds for the cost functional are derived, which are guaranteed, fully computable and sharp. The two-sided bounds provide a new formulation of the optimization problem, since the minimization and maximization of the upper and lower bounds, respectively, or alternatively the direct minimization of their difference lead to the optimal value of the optimal control problem. The a posteriori estimates are of functional type. Functional a posteriori error estimates for parabolic problems were first presented in [12] and [3]. Later first a posteriori estimates of functional type for elliptic optimal control problems were derived in [2], for time-periodic problems in [7] and for time-periodic optimal controls after that in [8]. A new technique for deriving lower bounds for cost functionals was discussed in [13] by generalizing

M. Wolfmayr (✉)
Faculty of Information Technology, University of Jyväskylä, Jyväskylä, Finland
e-mail: monika.k.wolfmayr@jyu.fi

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_120

ideas from [10]. This article presents now upper as well as lower bounds which are called majorants and minorants for the cost functional and shows how efficient solvers can be build up in a functional setting for time-periodic parabolic problems. Further details as well as the discussion on a second optimal control problem can be found in [14].

The article is divided into the following parts: In Sect. 2, the optimal control problem as well as some Hilbert spaces are introduced in order to establish a proper functional setting for the problem discussed. The two-sided bounds are presented in Sect. 3. The discussion of the multiharmonic finite element method for the time-periodic problem is subject matter of Sect. 4, where also a short review on the preconditioned minimal residual method can be found. Finally, in Sects. 5 and 6, numerical results and conclusions are presented, respectively.

2 Time-Periodic Parabolic Optimal Control Problem

The time-periodic problem discussed in this article is formulated in a functional space setting. For that let us introduce the following Hilbert spaces first: $H^{1,0}(Q) := \{u \in L^2(Q) : \nabla u \in [L^2(Q)]^d\}$, $H^{0,1}(Q) := \{u \in L^2(Q) : \partial_t u \in L^2(Q)\}$, $H^{1,1}(Q) := \{u \in L^2(Q) : \nabla u \in [L^2(Q)]^d, \partial_t u \in L^2(Q)\}$, where the $d+1$-dimensional space-time domain Q is defined by $Q := \Omega \times (0, T)$ for $d = \{1, 2, 3\}$ with $\Omega \subset \mathbb{R}^d$ and the given time interval $(0, T)$. The boundary of Ω is defined by $\Gamma := \partial\Omega$, on which homogeneous Dirichlet boundary conditions are prescribed. The lateral surface of Q is denoted by $\Sigma := \Gamma \times (0, T)$. The model cost functional in this article is given by

$$\mathcal{J}(y, u) := \frac{1}{2}\|y - y_d\|^2_{L^2(Q)} + \frac{\lambda}{2}\|u\|^2_{L^2(Q)} \tag{1}$$

being minimized with respect to (w.r.t.) (y, u) subject to the PDE-constraints

$$\partial_t y(\boldsymbol{x}, t) - \Delta y(\boldsymbol{x}, t) = u(\boldsymbol{x}, t) \qquad (\boldsymbol{x}, t) \in Q, \tag{2}$$

$$y(\boldsymbol{x}, t) = 0 \qquad (\boldsymbol{x}, t) \in \Sigma, \tag{3}$$

$$y(\boldsymbol{x}, 0) = y(\boldsymbol{x}, T) \qquad \boldsymbol{x} \in \overline{\Omega}. \tag{4}$$

Here, y denotes the state and u the control function, and the target function $y_d \in L^2(Q)$ is a given, not necessarily time-periodic desired state. The cost parameter in the cost functional has been denoted by $\lambda > 0$.

The boundary and time-periodic conditions are included in the Hilbert spaces as follows: $H_0^{1,0}(Q) := \{u \in H^{1,0}(Q) : u = 0 \text{ on } \Sigma\}$, $H_{\text{per}}^{0,1}(Q) := \{u \in H^{0,1}(Q) : u(0) = u(T) \text{ in } \overline{\Omega}\}$. We also introduce the following space for the flux functions: $H(\text{div}, Q) := \{\boldsymbol{\tau} \in [L^2(Q)]^d : \nabla \cdot \boldsymbol{\tau}(\cdot, t) \in L^2(\Omega) \text{ for a.e. } t \in (0, T)\}$, and the space $H^{1,\frac{1}{2}}_{\text{per}}(Q) := \{u \in H^{1,0}(Q) : \|\partial_t^{1/2} u\|_{L^2(Q)} < \infty\}$, with the norm defined in Fourier space as $\|\partial_t^{1/2} u\|^2_{L^2(Q)} := \frac{T}{2}\sum_{k=1}^{\infty} k\omega \|\boldsymbol{u}_k\|^2_{L^2(\Omega)}$. We use the Fourier series representation with frequency $\omega = 2\pi/T$ and time-period T in the following form:

$$y(\boldsymbol{x}, t) := y_0^c(\boldsymbol{x}) + \sum_{k=1}^{\infty} \left(y_k^c(\boldsymbol{x}) \cos(k\omega t) + y_k^s(\boldsymbol{x}) \sin(k\omega t) \right), \tag{5}$$

where the Fourier coefficients y_0^c and $\boldsymbol{y}_k := (y_k^c, y_k^s)^T$, $k \in \mathbb{N}$, with $\tilde{\kappa}(k\omega t) = \cos(k\omega t)$ or $\tilde{\kappa}(k\omega t) = \sin(k\omega t)$ depending if $\kappa = c$ or $\kappa = s$, are given by

$$y_0^c(\boldsymbol{x}) := \frac{1}{T}\int_0^T y(\boldsymbol{x}, t)\, dt, \qquad y_k^\kappa(\boldsymbol{x}) := \frac{2}{T}\int_0^T y(\boldsymbol{x}, t)\tilde{\kappa}(k\omega t)\, dt.$$

3 Two-Sided Bounds for the Optimal Control Problem

In this section, we present the upper and lower bounds for the cost functional, which are guaranteed, fully computable and sharp. Direct minimization and maximization can be applied on the majorants and minorants as an efficient strategy leading to the exact solution of the optimal control problem.

3.1 Guaranteed Upper Bound for the Cost Functional

The upper bound for the cost functional (1) is derived by first obtaining an estimate for the approximation error in the $H^{1,\frac{1}{2}}_{\text{per}}(Q)$-seminorm $|y|^2_{1,\frac{1}{2}} := \|\nabla y\|^2_{L^2(Q)} + \|\partial_t^{1/2} y\|^2_{L^2(Q)}$ for problem (2)–(4) as shown in [7] and [8]. It is given by

$$|y(v) - \eta|_{1,\frac{1}{2}} \leq \frac{1}{\underline{\mu_1}} \left(C_F \|\mathcal{R}_1(\eta, \boldsymbol{\tau}, v)\|_{L^2(Q)} + \|\mathcal{R}_2(\eta, \boldsymbol{\tau})\|_{L^2(Q)} \right), \tag{6}$$

with $\underline{\mu_1} = 1/\sqrt{2}$, Friedrichs' constant $C_F > 0$ and the arbitrary functions $\eta \in H^{1,1}_{0,\text{per}}(Q)$, $\boldsymbol{\tau} \in H(\text{div}, Q)$ and $v \in L^2(Q)$. The residual functions are defined as $\mathcal{R}_1(\eta, \boldsymbol{\tau}, v) := \partial_t \eta - \nabla \cdot \boldsymbol{\tau} - v$ and $\mathcal{R}_2(\eta, \boldsymbol{\tau}) := \boldsymbol{\tau} - \nabla \eta$. The function $\boldsymbol{\tau}$ represents an image of the flux. Using estimate (6) as well as Cauchy-Schwarz' and Friedrichs' inequalities, we can derive the following fully computable, guaranteed and sharp estimate:

$$\mathcal{J}(y(v), v) \leq \mathcal{J}^{\oplus}(\alpha, \beta; \eta, \boldsymbol{\tau}, v) \quad \forall v \in L^2(Q)\, \forall \alpha, \beta > 0 \tag{7}$$

with α, β coming from applying Young's inequality, where the majorant is given by

$$\mathcal{J}^{\oplus}(\alpha, \beta; \eta, \boldsymbol{\tau}, v) := \frac{1+\alpha}{2}\|\eta - y_d\|^2_{L^2(Q)} + \frac{(1+\alpha)(1+\beta)C_F^2}{2\alpha\underline{\mu_1}^2}\|\mathcal{R}_2(\eta, \boldsymbol{\tau})\|^2_{L^2(Q)}$$
$$+ \frac{(1+\alpha)(1+\beta)C_F^4}{2\alpha\beta\underline{\mu_1}^2}\|\mathcal{R}_1(\eta, \boldsymbol{\tau}, v)\|^2_{L^2(Q)} + \frac{\lambda}{2}\|v\|^2_{L^2(Q)}. \tag{8}$$

The majorant is sharp:

$$\inf_{\substack{\eta \in H^{1,1}_{0,\text{per}}(Q), \boldsymbol{\tau} \in H(\text{div}, Q),\\ v \in L^2(Q), \alpha, \beta > 0}} \mathcal{J}^{\oplus}(\alpha, \beta; \eta, \boldsymbol{\tau}, v) = \mathcal{J}(y(u), u). \tag{9}$$

3.2 Guaranteed Lower Bound for the Cost Functional

The guaranteed and fully computable minorant for the cost functional is obtained by estimating approximation errors in a similar way as for the majorant for arbitrary functions and applying Cauchy-Schwarz' and Friedrichs' inequalities leading to the estimate

$$\mathcal{J}(y(u), u) \geq \mathcal{J}^{\ominus}(\eta, \zeta, \boldsymbol{\tau}, \boldsymbol{\rho}) \tag{10}$$

for all $\eta, \zeta \in H^{1,1}_{0,\text{per}}(Q)$ and $\boldsymbol{\tau}, \boldsymbol{\rho} \in H(\text{div}, Q)$ with the supremum being attained for the exact solution of the optimal control problem

$$\sup_{\eta, \zeta \in H^{1,1}_{0,\text{per}}(Q), \boldsymbol{\tau}, \boldsymbol{\rho} \in H(\text{div}, Q)} \mathcal{J}^{\ominus}(\eta, \zeta, \boldsymbol{\tau}, \boldsymbol{\rho}) = \mathcal{J}(y(u), u). \tag{11}$$

The function $\boldsymbol{\rho}$ represents an image of the flux for the adjoint function $p(\boldsymbol{x}, t)$, which is the solution of the adjoint equation $-\partial_t p - \Delta p = y - y_d$ appearing in the optimality equations of the optimal control problem discussed. The arbitrary function $\zeta \in H^{1,1}_{0,\text{per}}(Q)$ is an image of the adjoint and can in practice be chosen as its approximation. Here, the minorant $\mathcal{J}^{\ominus}$ is defined as

$$\begin{aligned}\mathcal{J}^{\ominus}(\eta,\zeta,\boldsymbol{\tau},\boldsymbol{\rho}) &:= \frac{1}{2}\|\eta - y_d\|^2_{L^2(Q)} - \int_Q \left(\nabla\eta\cdot\nabla\zeta + \partial_t\eta\,\zeta + \lambda^{-1}\zeta^2\right) d\boldsymbol{x}\,dt \\ &\quad - \frac{C_F^2}{\underline{\mu_1}^2\lambda}\left(C_F\|\mathcal{R}_3(\zeta,\boldsymbol{\rho},\eta)\|_{L^2(Q)} + \|\mathcal{R}_4(\zeta,\boldsymbol{\rho})\|_{L^2(Q)}\right)^2 + \frac{1}{2\lambda}\|\zeta\|^2_{L^2(Q)} \\ &\quad - \frac{1}{\underline{\mu_1}}(C_F\|\mathcal{R}_1(\eta,\boldsymbol{\tau},-\lambda^{-1}\zeta)\|_{L^2(Q)} + \|\mathcal{R}_2(\eta,\boldsymbol{\tau})\|_{L^2(Q)}) \\ &\qquad \times \left(C_F\|\mathcal{R}_3(\zeta,\boldsymbol{\rho},\eta)\|_{L^2(Q)} + \|\mathcal{R}_4(\zeta,\boldsymbol{\rho})\|_{L^2(Q)}\right). \end{aligned} \tag{12}$$

The new residual functions are defined as $\mathcal{R}_3(\zeta,\boldsymbol{\rho},\eta) := \eta - y_d + \partial_t\zeta + \nabla\cdot\boldsymbol{\rho}$ and $\mathcal{R}_4(\zeta,\boldsymbol{\rho}) := \boldsymbol{\rho} - \nabla\zeta$. They reflect naturally the adjoint equation of the optimality system of the optimal control problem.

4 Multiharmonic Finite Element Discretization and the Preconditioned Minimal Residual Method

Since the optimal control problem is time-periodic, the multiharmonic finite element method is a natural approach for its discretization as well as an initial guess for the functional a posteriori estimates. The idea of the multiharmonic finite element method is to expand the functions into Fourier series in time, truncate the Fourier series and to approximate the Fourier coefficients by the finite element method for instance similar as discussed in [1]. We note also that it is enough that the functions as for instance the given desired state are taken from $L^2(Q)$ in order to expand them into Fourier series with Fourier coefficients from $L^2(\Omega)$. Moreover, the cost functional (1) can be written w.r.t. the Fourier modes as follows

$$\mathcal{J}(y,u) = T\mathcal{J}_0(y_0^c,u_0^c) + \frac{T}{2}\sum_{k=1}^{\infty}\mathcal{J}_k(\boldsymbol{y}_k,\boldsymbol{u}_k), \tag{13}$$

where $\mathcal{J}_0(y_0^c,u_0^c) := \frac{1}{2}\|y_0^c - y_{d0}^c\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\|u_0^c\|^2_{L^2(\Omega)}$ and $\mathcal{J}_k(\boldsymbol{y}_k,\boldsymbol{u}_k) := \frac{1}{2}\|\boldsymbol{y}_k - \boldsymbol{y}_{dk}\|^2_{L^2(\Omega)} + \frac{\lambda}{2}\|\boldsymbol{u}_k\|^2_{L^2(\Omega)}$. After choosing the truncation index $N \in \mathbb{N}$ the Fourier coefficients $\boldsymbol{y}_k := (y_k^c, y_k^s)^T, \in H_0^1(\Omega)\times H_0^1(\Omega)$ (see also (5)) are approximated by finite element functions $\boldsymbol{y}_{kh} := (y_{kh}^c, y_{kh}^s)^T \in V_h \times V_h$ with the conforming finite element space $V_h := \text{span}\{\phi_1,\dots,\phi_n\}$, where piecewise linear and continuous elements are used as basis $\{\phi_i(\boldsymbol{x}) : i = 1,2,\dots,n_h\}$, $n := n_h = \dim V_h = O(h^{-d})$, the discretization parameter h, and the regular triangulation $\mathcal{T}_h$. Altogether the multiharmonic finite element functions can be denoted by $y_{Nh}(\boldsymbol{x},t) := y_{0h}^c(\boldsymbol{x}) + \sum_{k=1}^{N}(y_{kh}^c(\boldsymbol{x})\cos(k\omega t) + y_{kh}^s(\boldsymbol{x})\sin(k\omega t))$. The choice of the truncation index N depends on the smoothness of the given data w.r.t. time. The two-sided bounds presented in this paper provide the framework for a more detailed analysis on the choice of N and also for the discussion of an adaptive method in time. However, this is not subject matter of this work, but is discussed in more detail in [14] as well as in a successive paper. Similar to (13) the majorant and minorant can be represented w.r.t. the Fourier modes given for the majorant by

$$\begin{aligned}\mathcal{J}_0^{\oplus} := & \frac{1+\alpha_0}{2}\|y_{0h}^c - y_{d0}^c\|_{L^2(\Omega)}^2 + \frac{1}{2\lambda}\|p_{0h}^c\|_{L^2(\Omega)}^2 \\ & + \frac{(1+\alpha_0)(1+\beta_0)C_F^2}{2\alpha_0\underline{\mu_1}^2}\|\mathcal{R}_{20}^c\|_{L^2(\Omega)}^2 + \frac{(1+\alpha_0)(1+\beta_0)C_F^4}{2\alpha_0\beta_0\underline{\mu_1}^2}\|\mathcal{R}_{10}^c\|_{L^2(\Omega)}^2,\end{aligned} \tag{14}$$

where $\mathcal{R}_{10}^c := \nabla\cdot\boldsymbol{\tau}_{0h}^c - \lambda^{-1}p_{0h}^c$, $\mathcal{R}_{20}^c := \boldsymbol{\tau}_{0h}^c - \nabla y_{0h}^c$, and

$$\begin{aligned}\mathcal{J}_k^{\oplus} := & \frac{1+\alpha_k}{2}\|\boldsymbol{y}_{kh} - \boldsymbol{y}_{dk}\|_{L^2(\Omega)}^2 + \frac{1}{2\lambda}\|\boldsymbol{p}_{kh}\|_{L^2(\Omega)}^2 \\ & + \frac{(1+\alpha_k)(1+\beta_k)C_F^2}{2\alpha_k\underline{\mu_1}^2}\|\mathcal{R}_{2k}\|_{L^2(\Omega)}^2 + \frac{(1+\alpha_k)(1+\beta_k)C_F^4}{2\alpha_k\beta_k\underline{\mu_1}^2}\|\mathcal{R}_{1k}\|_{L^2(\Omega)}^2,\end{aligned} \tag{15}$$

where $\mathcal{R}_{1k} := (\mathcal{R}_{1k}^c, \mathcal{R}_{1k}^s)^T := (-k\omega\, y_{kh}^s + \nabla\cdot\boldsymbol{\tau}_{kh}^c - \lambda^{-1}p_{kh}^c, k\omega\, y_{kh}^c + \nabla\cdot\boldsymbol{\tau}_{kh}^s - \lambda^{-1}p_{kh}^s)^T$ and $\mathcal{R}_{2k} := (\mathcal{R}_{2k}^c, \mathcal{R}_{2k}^s)^T := (\boldsymbol{\tau}_{kh}^c - \nabla y_{kh}^c, \boldsymbol{\tau}_{kh}^s - \nabla y_{kh}^s)^T$. Then the overall majorant can be written as

$$\mathcal{J}^{\oplus}(\boldsymbol{\alpha}_{N+1}, \boldsymbol{\beta}_N; y_{Nh}, p_{Nh}, \boldsymbol{\tau}_{Nh}) := T\,\mathcal{J}_0^{\oplus} + \frac{T}{2}\sum_{k=1}^{N}\mathcal{J}_k^{\oplus} + \frac{1+\alpha_{N+1}}{2}\,\mathcal{E}_N, \tag{16}$$

where we have introduced $\boldsymbol{\alpha}_{N+1} := (\alpha_0, \ldots, \alpha_{N+1})^T$ and $\boldsymbol{\beta}_N := (\beta_0, \ldots, \beta_N)^T$, and the truncation's remainder term $\mathcal{E}_N := \frac{T}{2}\sum_{k=N+1}^{\infty}\left(\|y_{dk}^c\|_{L^2(\Omega)}^2 + \|y_{dk}^s\|_{L^2(\Omega)}^2\right)$ can always be computed for any accuracy because the desired state is known. Analogously as (16) for the majorant, we can formulate the minorant w.r.t. the Fourier modes as

$$\mathcal{J}^{\ominus}(y_{Nh}, p_{Nh}, \boldsymbol{\tau}_{Nh}, \boldsymbol{\rho}_{Nh}) := T\,\mathcal{J}_0^{\ominus} + \frac{T}{2}\sum_{k=1}^{N}\mathcal{J}_k^{\ominus} + \frac{\mathcal{E}_N}{2} \tag{17}$$

again with introducing residual functions corresponding to the Fourier modes. The flux functions $\boldsymbol{\tau}_{0h}^c$, $\boldsymbol{\rho}_{0h}^c$ and $\boldsymbol{\tau}_{kh}$, $\boldsymbol{\rho}_{kh}$, for all $k = 1, \ldots, N$, are reconstructed by lowest-order Raviart-Thomas elements mapping L^2-functions to $H(\mathrm{div}, \Omega)$ (as presented in [11]) yielding the overall flux functions $\boldsymbol{\tau}_{Nh}$, $\boldsymbol{\rho}_{Nh}$. Minimizing the majorant w.r.t. positive parameters α_k, β_k leads to optimized parameters $\boldsymbol{\alpha}_{N+1}$, $\boldsymbol{\beta}_N$.

In order to compute the approximate solution which serves as a first guess for the majorants and minorants, we use a robust preconditioned minimal residual method on the discretized problem, which is a system of linear equations with a saddle point structure. The structure comes from the saddle point nature of the optimality equations. Proper fast solvers for the linear systems are discussed in more detail in [4, 6, 9] as well as in [14].

5 Numerical Results

In this section, we present new numerical results for the optimal control problem (1)–(4). The computations have been performed in C++ on a laptop with Intel(R) Core(TM) i5-6267U CPU @ 2.90 GHz processor and 16 GB 2133 MHz LPDDR3 memory. We chose the 2-dimensional computational domain $\Omega = (0, 1)^2$. This leads to the Friedrichs constant $C_F = 1/(\sqrt{2}\pi)$. The frequency is chosen as $\omega = 1$ and $T = 2\pi/\omega$ is the corresponding time period. The multiharmonic finite element approximations are used for η, ζ and $\boldsymbol{\tau}$, $\boldsymbol{\rho}$ and RT^0-extensions (lowest-order standard Raviart-Thomas) are used for the fluxes leading to averaged fluxes being from $H(\text{div}, \Omega)$. A preconditioned minimal residual method was used with 8 iteration steps in all computations using an algebraic multilevel preconditioner with 4 inner iterations as discussed in [5, 6]. All computational times t^{sec} in seconds include the CPU times for computing the majorants and minorants, which are significantly smaller compared to the initialization and applying the preconditioned minimal residual method. In this numerical example, we have chosen a desired state, which is non-periodic in time however from $L^2(Q)$, given by $y_d(\boldsymbol{x}, t) = e^t(-0.2(1 + t)\cos(t) + ((-2 + 4\pi^4 t)0.2 + t)\sin(t))\sin(x_1\pi)\sin(x_2\pi)$, for which the exact solution is known being the state function $y(\boldsymbol{x}, t) = e^t t \sin(t)\sin(x_1\pi)\sin(x_2\pi)$. The cost parameter is chosen as $\lambda = 0.1$. The efficiency indices are defined as $I_{\text{eff}}^{\mathcal{J}_0^{\oplus}} := \mathcal{J}_0^{\oplus}/\mathcal{J}_0$, $I_{\text{eff}}^{\mathcal{J}_0^{\ominus}} := \mathcal{J}_0^{\ominus}/\mathcal{J}_0$, $I_{\text{eff}}^{\mathcal{J}_k^{\oplus}} := \mathcal{J}_k^{\oplus}/\mathcal{J}_k$, $I_{\text{eff}}^{\mathcal{J}_k^{\ominus}} := \mathcal{J}_k^{\ominus}/\mathcal{J}_k$, $I_{\text{eff}}^{\mathcal{J},0} := \mathcal{J}_0^{\oplus}/\mathcal{J}_0^{\ominus}$ and $I_{\text{eff}}^{\mathcal{J},k} := \mathcal{J}_k^{\oplus}/\mathcal{J}_k^{\ominus}$. Tables 1 and 2 show the numerical results for the Fourier modes $k = 0$ and $k = 1$ computed on meshes of different sizes

Table 1 Minorant $\mathcal{J}_0^{\ominus}$, majorant $\mathcal{J}_0^{\oplus}$ and their efficiency indices computed on meshes of different sizes

mesh	t^{sec}	$\mathcal{J}_0^{\ominus}$	$I_{\text{eff}}^{\mathcal{J}_0^{\ominus}}$	$\mathcal{J}_0^{\oplus}$	$I_{\text{eff}}^{\mathcal{J}_0^{\oplus}}$	$I_{\text{eff}}^{\mathcal{J},0}$
16 × 16	0.02	8.89e+06	0.89	9.92e+05	1.00	1.12
32 × 32	0.06	8.97e+06	0.90	9.98e+06	1.00	1.11
64 × 64	0.26	8.98e+06	0.90	9.99e+06	1.00	1.11
128 × 128	1.04	8.99e+06	0.90	9.99e+06	1.00	1.11
256 × 256	4.10	8.99e+06	0.90	9.99e+06	1.00	1.11

Table 2 Minorant $\mathcal{J}_1^{\ominus}$, majorant $\mathcal{J}_1^{\oplus}$ and their efficiency indices computed on meshes of different sizes

mesh	t^{sec}	$\mathcal{J}_1^{\ominus}$	$I_{\text{eff}}^{\mathcal{J}_1^{\ominus}}$	$\mathcal{J}_1^{\oplus}$	$I_{\text{eff}}^{\mathcal{J}_1^{\oplus}}$	$I_{\text{eff}}^{\mathcal{J},1}$
16×16	0.02	2.40e+07	0.92	2.62e+07	1.01	1.09
32×32	0.06	2.65e+07	0.90	2.95e+07	1.00	1.11
64×64	0.27	2.66e+07	0.90	2.95e+07	1.00	1.11
128×128	1.07	2.66e+07	0.90	2.95e+07	1.00	1.11
256×256	4.25	2.66e+07	0.90	2.95e+07	1.00	1.11

Table 3 Overall minorant and majorant, the respective minorants and majorants corresponding to the Fourier modes, and their efficiency indices computed on a mesh of size 256×256

Fourier mode	t^{sec}	$\mathcal{J}^{\ominus}$	$I_{\text{eff}}^{\mathcal{J}^{\ominus}}$	$\mathcal{J}^{\oplus}$	$I_{\text{eff}}^{\mathcal{J}^{\oplus}}$	$I_{\text{eff}}^{\mathcal{J}}$
$k = 2$	4.18	8.51e+06	0.90	9.43e+06	1.00	1.11
$k = 3$	4.10	2.24e+06	0.90	2.47e+06	1.00	1.11
$k = 4$	4.08	7.61e+05	0.91	8.37e+05	1.00	1.10
$k = 5$	4.25	3.21e+05	0.91	3.51e+05	1.00	1.09
$k = 6$	4.16	1.58e+05	0.92	1.71e+05	1.00	1.09
$k = 7$	4.16	8.63e+04	0.92	9.32e+04	1.00	1.08
$k = 8$	4.13	5.11e+04	0.93	5.49e+04	1.00	1.07
Overall ($N = 8$)	–	1.79e+08	0.90	1.98e+08	1.00	1.11

ranging between 16×16 and 256×256. Table 3 presents the numerical results computed on the 256×256 mesh for different Fourier modes (from $k = 2$ to $k = 8$) and the overall minorant and majorant for the full cost functional. The truncation's reminder term for a truncation index $N = 8$ for the given desired state of this example is given by $\mathcal{E}_8 = 786{,}901$.

6 Conclusions

The derived two-sided bounds are guaranteed, fully computable and sharp. They provide an alternative way to be used as subject of direct minimization (or maximization) in order to obtain the solution of the optimization problem. The cost functional discussed in this article is standard and is used in many real life applications. The minimization is w.r.t. a state and control function and the target is a given desired state. A more detailed discussion including theorems and proofs for the model problem of this article including the presentation of additional numerical tests can be found in [14]. Moreover, [14] discusses another cost functional, where the target is a given desired gradient.

Acknowledgments The author gratefully acknowledges the financial support by the Academy of Finland under the grant 295897 as well as by the Central Finland regional Fund of the Finnish Cultural Foundation.

References

1. Ciarlet, P. G., The Finite Element Method for Elliptic Problems, Studies in Mathematics and its Applications **4**, North-Holland, Amsterdam (1978). Republished by SIAM (2002)
2. Gaevskaya, A., Hoppe, R. H. W., Repin, S.: A posteriori estimates for cost functionals of optimal control problems. In: Numer. Math. Adv. Appl. Proc. ENUMATH 2005, pp. 308–316 (2006)
3. Gaevskaya, A. V., Repin, S. I.: A posteriori error estimates for approximate solutions of linear parabolic problems. Differ. Equ. **41**, 970–983 (2005)
4. Kollmann, M., Kolmbauer, Langer, U., Wolfmayr, M., Zulehner, W., A robust finite element solver for a multiharmonic parabolic optimal control problem, Comput. Math. Appl. **65**, 469–486 (2013)
5. Kraus, J., Additive Schur complement approximation and application to multilevel preconditioning, SIAM J. Sci. Comput. **34**, A2872–A2895 (2012)
6. Kraus, J., Wolfmayr, M., On the robustness and optimality of algebraic multilevel methods for reaction-diffusion type problems, Comput. Vis. Sci. **16**, 15–32 (2013)
7. Langer, U., Repin, S., Wolfmayr, M.: Functional a posteriori error estimates for parabolic time-periodic boundary value problems. Comput. Methods Appl. Math. **15**, 353–372 (2015)
8. Langer, U., Repin, S., Wolfmayr, M.: Functional a posteriori error estimates for time-periodic parabolic optimal control problems. Num. Func. Anal. Opt. **37**, 1267–1294 (2016)
9. Langer, U., Wolfmayr, M., Multiharmonic finite element analysis of a time-periodic parabolic optimal control problem, J. Numer. Math. **21**, 265–300 (2013)
10. Mikhlin, S. G.: Variational methods in mathematical physics, Pergamon Press Oxford (1964)
11. Raviart, P. A., Thomas, J. M., A mixed finite element method for 2-nd order elliptic problems, Mathematical Aspects of Finite Element Methods, Lect. Notes Math. **606**, 292–315 (1977)
12. Repin, S.: Estimates of deviation from exact solutions of initial-boundary value problems for the heat equation. Rend. Mat. Acc. Lincei **13**, 121–133 (2002)
13. Wolfmayr, M.: A note on functional a posteriori estimates for elliptic optimal control problems. Numer. Meth. Part. Differ. Equat. **33**, 403–424 (2017)
14. Wolfmayr, M.: Guaranteed lower bounds for cost functionals of time-periodic parabolic optimization problems. Comput. Math. Appl. **80**(5), 1050-1072 (2020) https://doi.org/10.1016/j.camwa.2020.04.021

Finite Element Approximation of a System Coupling Curve Evolution with Prescribed Normal Contact to a Fixed Boundary to Reaction-Diffusion on the Curve

Vanessa Styles and James Van Yperen

Abstract We consider a finite element approximation for a system consisting of the evolution of a curve evolving by forced curve shortening flow coupled to a reaction-diffusion equation on the evolving curve. The curve evolves inside a given domain $\Omega \subset \mathbb{R}^2$ and meets $\partial\Omega$ orthogonally. We present numerical experiments and show the experimental order of convergence of the approximation.

1 Introduction

We consider a curve $\Gamma(t)$ evolving by forced curve shortening flow inside a given bounded domain $\Omega \subset \mathbb{R}^2$, with the forcing being a function of the solution, $w : \Gamma(t) \to \mathbb{R}$, of a reaction-diffusion equation that holds on $\Gamma(t)$, such that

$$v = \kappa + f(w) \qquad \text{on } \Gamma(t),\ \ t \in (0, T], \tag{1}$$

$$\partial_t^\bullet w = w_{ss} + \kappa\, v\, w + g(v, w) \qquad \text{on } \Gamma(t),\ \ t \in (0, T], \tag{2}$$

subject to the initial data $\Gamma(0) = \Gamma_0$ and $w(\cdot, 0) = w_0$ on Γ_0.

Here v and κ respectively denote the normal velocity and mean curvature of $\Gamma(t)$, corresponding to the choice $\mathbf{n}$ of a unit normal, s is the arclength parameter on $\Gamma(t)$ and $\partial_t^\bullet w := w_t + v\frac{\partial w}{\partial \mathbf{n}}$ denotes the material derivative of w. In addition we impose that the curve meets the boundary $\partial\Omega$ orthogonally. To this end we assume that $\partial\Omega$ is given by a smooth function F such that

$$\partial\Omega = \{p \in \mathbb{R}^2 : F(p) = 0\} \ \text{ and } \ |\nabla F(p)| = 1 \ \ \forall\, p \in \partial\Omega.$$

V. Styles · J. Van Yperen (✉)
Department of Mathematics, University of Sussex, Brighton, England
e-mail: v.styles@sussex.ac.uk; j.vanyperen@sussex.ac.uk

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_121

Coupling the parametrisation of (1) and (2) that is presented in [1] for the setting in which $\Gamma(t)$ is a closed curve, with the formulation of (1) presented in [5] for the setting in which $\Gamma(t)$ meets the boundary $\partial\Omega$ orthogonally, yields the following system:

$$\alpha \mathbf{x}_t + (1-\alpha)(\mathbf{x}_t \cdot \mathbf{n})\,\mathbf{n} = \frac{\mathbf{x}_{\rho\rho}}{|\mathbf{x}_\rho|^2} + f(w)\,\mathbf{n}, \qquad (\rho, t) \in \mathbb{I} \times (0, T) \tag{3}$$

$$w_t - (\mathbf{x}_t \cdot \boldsymbol{\tau})\,\frac{w_\rho}{|\mathbf{x}_\rho|} - \frac{1}{|\mathbf{x}_\rho|}\left(\frac{w_\rho}{|\mathbf{x}_\rho|}\right)_\rho - \kappa\, v\, w = g(v, w), \qquad (\rho, t) \in \mathbb{I} \times (0, T) \tag{4}$$

$$\mathbf{x}(\rho, 0) = \mathbf{x}_0(\rho), \quad w(\rho, 0) = w_0(\rho) \qquad \rho \in \mathbb{I} \tag{5}$$

$$w(\rho, t) = w_b, \qquad (\rho, t) \in \{0, 1\} \times [0, T] \tag{6}$$

$$F(\mathbf{x}(\rho, t)) = 0, \qquad (\rho, t) \in \{0, 1\} \times [0, T] \tag{7}$$

$$(\mathbf{x}_\rho(\rho, t) \cdot \nabla^\perp F(\mathbf{x}(\rho, t)) = 0, \qquad (\rho, t) \in \{0, 1\} \times [0, T]. \tag{8}$$

Here $\alpha \in (0, 1]$, $\mathbb{I} := (0, 1)$, $\mathbf{x}(\cdot, t) : [0, 1] \to \mathbb{R}^2$, $w(\rho, t) := w(\mathbf{x}(\rho, t), t)$, $(\rho, t) \in [0, 1] \times [0, T]$, and the unit tangent and unit normal to $\Gamma(t)$ are respectively given by $\boldsymbol{\tau} = \mathbf{x}_s = \frac{\mathbf{x}_\rho}{|\mathbf{x}_\rho|}$ and $\mathbf{n} = \boldsymbol{\tau}^\perp$ where $(\cdot)^\perp$ denotes counter-clockwise rotation by $\frac{\pi}{2}$.

The formulation of curve shortening flow in the form of (3) for a closed curve in $\mathbb{R}^2$ was presented and analysed in [8], where the DeTurck trick is used in coupling the motion of the curve to the harmonic map heat flow, with the parameter $\alpha \in (0, 1]$ being such that $1/\alpha$ corresponds to the diffusion coefficient in the harmonic map heat flow. Setting $\alpha \in (0, 1]$ introduces a tangential part in the velocity which, at the numerical level, gives rise to a good distribution of the mesh points along the curve. Setting $\alpha = 1$ one recovers the formulation introduced and analysed in [4], while formally setting $\alpha = 0$ yields the approach introduced in [3]. In [6] the authors derive finite element approximations of a simplified version of the parametric coupled system (3)–(8), and two related models. In particular, the evolution law for the parametric system derived in [6], can be obtained from (3) by setting $\alpha = 1$, $F(x, y) = |x| - 1$, and considering a slightly different formulation of the reaction-diffusion equation (4). In [1] the authors prove optimal error bounds for a fully discrete finite element approximation of the coupled system (3)–(5) for the case where $\Gamma(t)$ is a closed curve in $\mathbb{R}^2$. While in [9] optimal error bounds are presented for a semi-discrete finite element approximation of an alternative formulation, which is introduced and analysed in [7], of the coupled system (3)–(5), for the case where $\Gamma(t)$ is a closed curve in $\mathbb{R}^2$. Setting $\alpha = 1$ and $f(w) = 0$ in (3) and coupling the resulting equation to (7) and (8) gives rise to the model presented and analysed in [5], in which optimal order error bounds for a semi-discrete finite element approximation of curve shortening flow with a prescribed normal contact to a fixed boundary are presented. In [2] the authors propose parametric finite element approximations of combined second and fourth order geometric evolution equations for curves that are connected via triple or quadruple junctions or that intersect external boundaries.

2 Weak Formulation and Finite Element Approximation

For a weak formulation of (3) we multiply it by $|\mathbf{x}_\rho|^2\boldsymbol{\xi}$, where $\boldsymbol{\xi} \in [H^1(\mathbb{I})]^2$ is a test function, integrate in space, use integration by parts and (8) to obtain $\forall\, \boldsymbol{\xi} \in [H^1(\mathbb{I})]^2$

$$\begin{aligned}\Big(|\mathbf{x}_\rho|^2\,[\alpha\,\mathbf{x}_t + (1-\alpha)(\mathbf{x}_t\cdot\mathbf{n})\,\mathbf{n}\,]\,,\boldsymbol{\xi}\Big) + \big(\mathbf{x}_\rho,\boldsymbol{\xi}_\rho\big)\\ = \big[(\mathbf{x}_\rho\cdot\nabla F(\mathbf{x}))(\boldsymbol{\xi}\cdot\nabla F(\mathbf{x}))\big]_0^1 + \Big(|\mathbf{x}_\rho|^2 f(w)\,\mathbf{n},\boldsymbol{\xi}\Big)\,,\end{aligned} \tag{9}$$

where $(\cdot,\cdot)$ denotes the standard $L^2(\mathbb{I})$ inner product. For a weak formulation of (4) we multiply it by $|\mathbf{x}_\rho|\,\eta$, where $\eta \in H_0^1(\mathbb{I})$ is a time-independent test function, integrate in space, use integration by parts and note that $\boldsymbol{\tau}_\rho = \kappa\,\mathbf{n}\,|\mathbf{x}_\rho|$ to obtain $\forall\,\eta \in H_0^1(\mathbb{I})$

$$\frac{d}{dt}\big(|\mathbf{x}_\rho|\,w,\eta\big) + \big(\psi\,w,\eta_\rho\big) + \left(\frac{w_\rho}{|\mathbf{x}_\rho|},\eta_\rho\right) = \big(|\mathbf{x}_\rho|\,g(v,w),\eta\big)\,. \tag{10}$$

Here ψ is the tangential velocity of $\Gamma(t)$, such that the normal and tangential velocities of $\Gamma(t)$ are given by $v = \mathbf{x}_t\cdot\mathbf{n}$ and $\psi = \mathbf{x}_t\cdot\boldsymbol{\tau}$. We now introduce a finite element approximation of (9) and (10). We first let $0 = t_0 < t_1 < \cdots < t_{N-1} < t_N = T$ be a partition of $[0,T]$ with $\Delta t_n := t_n - t_{n-1}$. Next we partition the interval $\mathbb{I}$ such that $\mathbb{I} = \cup_{j=1}^J \overline{\sigma_j}$, where $\sigma_j = (\rho_{j-1},\rho_j)$, with $h_j = \rho_j - \rho_{j-1}$. We set

$$\begin{aligned}V^h &:= \{\chi \in C(\mathbb{I}) : \chi_{|\sigma_j} \text{ is affine}, j = 1,\ldots,J\} \subset H^1(\mathbb{I})\\ V_0^h &:= \{\chi \in V^h : \chi(\rho_j) = 0, \text{ for } j \in \{0,J\}\}\end{aligned}$$

and denote the standard Lagrange interpolation operator by $I^h : C(\mathbb{I}) \to V^h$, where $(I^h\eta)(\rho_j) = \eta(\rho_j)$, for $j = 0,\ldots,J$. We define the discrete inner product $(\eta_1,\eta_2)^h$ by

$$(\eta_1,\eta_2)^h := \sum_{j=1}^J \int_{\sigma_j} I_j^h(\eta_1\,\eta_2)\ \mathrm{d}\rho,$$

where $I_j^h = I_{|\sigma_j}^h$ is the local interpolation operator. Our finite element approximation of (9) and (10) then takes the form:

Given $(\mathbf{X}^{n-1}, W^{n-1} - w_b) \in [V^h]^2 \times V_0^h$, find $(\mathbf{X}^n, W^n - w_b) \in [V^h]^2 \times V_0^h$ such that for all $(\boldsymbol{\xi}^h, \eta^h) \in [V^h]^2 \times V_0^h$ we have

$$\left(|\mathbf{X}_\rho^{n-1}|^2 \left[\alpha D_t \mathbf{X}^n + (1-\alpha)(D_t \mathbf{X}^n \cdot \mathcal{N}^{n-1})\, \mathcal{N}^{n-1}\right], \boldsymbol{\xi}^h\right)^h + \left(\mathbf{X}_\rho^n, \boldsymbol{\xi}_\rho^h\right)$$
$$= \left[(\mathbf{X}_\rho^n \cdot \nabla F(\mathbf{X}^n))(\boldsymbol{\xi}^h \cdot \nabla F(\mathbf{X}^n))\right]_0^1 + \left(|\mathbf{X}_\rho^{n-1}|^2 f(W^{n-1})\, \mathcal{N}^{n-1}, \boldsymbol{\xi}^h\right)^h \tag{11}$$

$$D_t\left[\left(|\mathbf{X}_\rho^n|\, W^n, \eta^h\right)^h\right] + \left(\frac{W_\rho^n}{|\mathbf{X}_\rho^n|}, \eta_\rho^h\right) + \left(\Psi^n\, W^n, \eta_\rho^h\right)^h$$
$$= \left(|\mathbf{X}_\rho^n|\, g(V^n, W^{n-1}), \eta^h\right)^h \tag{12}$$

with the additional boundary constraint

$$F(\mathbf{X}_0^n) = F(\mathbf{X}_J^n) = 0. \tag{13}$$

Here and in what follows we set $D_t(a^n) := (a^n - a^{n-1})/\Delta t_n$ and on σ_j, $j = 1, \ldots, J$, we set $\mathcal{T}^n = \frac{\mathbf{X}_\rho^n}{|\mathbf{X}_\rho^n|}$, $\mathcal{N}^n = (\mathcal{T}^n)^\perp$, $\Psi^n = D_t \mathbf{X}^n \cdot \mathcal{T}^n$ and $V^n = D_t \mathbf{X}^n \cdot \mathcal{N}^n$.

Remark 1 In [5], rather than use the nonlinear scheme presented above to approximate (3), (7) and (8), the authors present a linear scheme in which (7) is not necessarily satisfied but is instead approximated through the relation $0 = \frac{d}{dt}F(\mathbf{x}) = \mathbf{x}_t \cdot \nabla F(\mathbf{x})$.

3 Numerical Results

3.1 Solution of the Discrete System (11) and (13)

We solve the resulting system of nonlinear algebraic equations arising at each time level from the approximation (11) and (13), with $\boldsymbol{\xi}^h = \chi_j$, $j = 1, \ldots, J-1$, $\boldsymbol{\xi}^h = \nabla^\perp F(\mathbf{X}^n)\chi_0$ and $\boldsymbol{\xi}^h = \nabla^\perp F(\mathbf{X}^n)\chi_J$, using the following Newton scheme, where for ease of presentation we set $\alpha = 1$ and $f(w) = 0$:

Given $\mathbf{X}^{n,i-1}$, with $\mathbf{X}^{n,0} = \mathbf{X}^{n-1}$, we set $\mathbf{X}^{n,i} := \mathbf{X}^{n,i-1} + \boldsymbol{\delta}^i$ such that for $j = 1, \ldots, J-1$, $\boldsymbol{\delta}^i$ solves

$$\frac{1}{2}\left(q_j^{n-1} + q_{j-1}^{n-1}\right)\frac{\boldsymbol{\delta}_j^i}{\Delta t_n} - \left(\boldsymbol{\delta}_{j-1}^i - 2\boldsymbol{\delta}_j^i + \boldsymbol{\delta}_{j+1}^i\right)$$
$$= -\frac{1}{2}\left(q_j^{n-1} + q_{j-1}^{n-1}\right) D_t(\mathbf{X}_j^{n,i-1}) + \left(\mathbf{X}_{j-1}^{n,i-1} - 2\mathbf{X}_j^{n,i-1} + \mathbf{X}_{j+1}^{n,i-1}\right), \tag{14a}$$

$$\begin{aligned}
&\frac{1}{2}\, q_0^{n-1} \left[\left(\frac{\boldsymbol{\delta}_0^i}{\Delta t_n} \cdot \nabla^\perp F(\mathbf{X}_0^{n,i-1}) \right) + \left(D_t(\mathbf{X}_0^{n,i-1}) \cdot D_\perp^2 F(\mathbf{X}_0^{n,i-1})\, \boldsymbol{\delta}_0^i \right) \right] \\
&\quad - \left((\boldsymbol{\delta}_1^i - \boldsymbol{\delta}_0^i) \cdot \nabla^\perp F(\mathbf{X}_0^{n,i-1}) \right) - \left((\mathbf{X}_1^{n,i-1} - \mathbf{X}_0^{n,i-1}) \cdot D_\perp^2 F(\mathbf{X}_0^{n,i-1})\, \boldsymbol{\delta}_0^i \right) \\
&\quad = -\frac{1}{2}\, q_0^{n-1} \left(D_t(\mathbf{X}_0^{n,i-1}) \cdot \nabla^\perp F(\mathbf{X}_0^{n,i-1}) \right) + \left((\mathbf{X}_1^{n,i-1} - \mathbf{X}_0^{n,i-1}) \cdot \nabla^\perp F(\mathbf{X}_0^{n,i-1}) \right),
\end{aligned} \tag{14b}$$

$$\begin{aligned}
&\frac{1}{2}\, q_{J-1}^{n-1} \left[\left(\frac{\boldsymbol{\delta}_J^i}{\Delta t_n} \cdot \nabla^\perp F(\mathbf{X}_J^{n,i-1}) \right) + \left(D_t(\mathbf{X}_J^{n,i-1}) \cdot D_\perp^2 F(\mathbf{X}_J^{n,i-1})\, \boldsymbol{\delta}_J^i \right) \right] \\
&\quad - \left((\boldsymbol{\delta}_{J-1}^i - \boldsymbol{\delta}_J^i) \cdot \nabla^\perp F(\mathbf{X}_J^{n,i-1}) \right) - \left((\mathbf{X}_{J-1}^{n,i-1} - \mathbf{X}_J^{n,i-1}) \cdot D_\perp^2 F(\mathbf{X}_J^{n,i-1})\, \boldsymbol{\delta}_J^i \right) \\
&\quad = -\frac{1}{2}\, q_{J-1}^{n-1} \left(D_t(\mathbf{X}_J^{n,i-1}) \cdot \nabla^\perp F(\mathbf{X}_J^{n,i-1}) \right) + \left((\mathbf{X}_{J-1}^{n,i-1} - \mathbf{X}_J^{n,i-1}) \cdot \nabla^\perp F(\mathbf{X}_J^{n,i-1}) \right),
\end{aligned} \tag{14c}$$

$$\left(\nabla F(\mathbf{X}_0^{n,i-1}) \cdot \boldsymbol{\delta}_0^i \right) = -F(\mathbf{X}_0^{n,i-1}) \ \text{ and } \ \left(\nabla F(\mathbf{X}_J^{n,i-1}) \cdot \boldsymbol{\delta}_J^i \right) = -F(\mathbf{X}_J^{n,i-1}), \tag{14d}$$

where $q_j^{n-1} = |\mathbf{X}_{j+1}^{n-1} - \mathbf{X}_j^{n-1}|^2$, $D_\perp^2 = \begin{pmatrix} -\partial_{xy}^2 & -\partial_{yy}^2 \\ \partial_{xx}^2 & \partial_{xy}^2 \end{pmatrix}$, and in an abuse of notation we have redefined D_t from the previous section such that $D_t(\mathbf{X}_j^{n,i-1}) := (\mathbf{X}_j^{n,i-1} - \mathbf{X}_j^{n-1})/\Delta t_n$. We adopt the stopping criteria $\max\limits_{j=0,J} |F(\mathbf{X}_j^{n,i})| \leq \tau$ for some predetermined tolerance, τ.

3.2 *Experimental Order of Convergence of (11) and (13)*

We investigate the experimental order of convergence of (11) and (13) by monitoring the following errors:

$$\mathcal{E}_1 := \sup_{n=0,\dots,N} \| I^h(\mathbf{x}_\rho^n) - \mathbf{X}_\rho^n \|_{[L^2(\mathbb{I})]^2}^2, \quad \mathcal{E}_2 := \sum_{n=1}^{N} \Delta t_n \| D_t(I^h(\mathbf{x}^n) - \mathbf{X}^n) \|_{[L^2(\mathbb{I})]^2}^2.$$

In addition we show how the choice of α affects the size of the errors. In all examples we use a uniform mesh size $hJ = 1$ and a uniform time step size $\Delta t = h^2$.

Example 1 In the first example we set $T = 0.4$ and $\Omega := \mathbb{R} \times \mathbb{R}_{>0}$, such that $\partial\Omega$ is given by $F(x, y) = y$. Taking $\Gamma(0)$ to be a semi circle with radius 1, the explicit solution is given by

$$\mathbf{x}(\rho, t) = \sqrt{1 - 2t}\, (\cos(\pi\rho), \sin(\pi\rho))^T.$$

In the left-hand plot in Fig. 1 we display: $\mathbf{X}^0$ in black, $\mathbf{X}^n$ at $t^n = 0.08k$, $k = 1, \dots, 5$, in blue, and $\partial\Omega$ in red, while in Table 1 we display the values of $\mathcal{E}_i$,

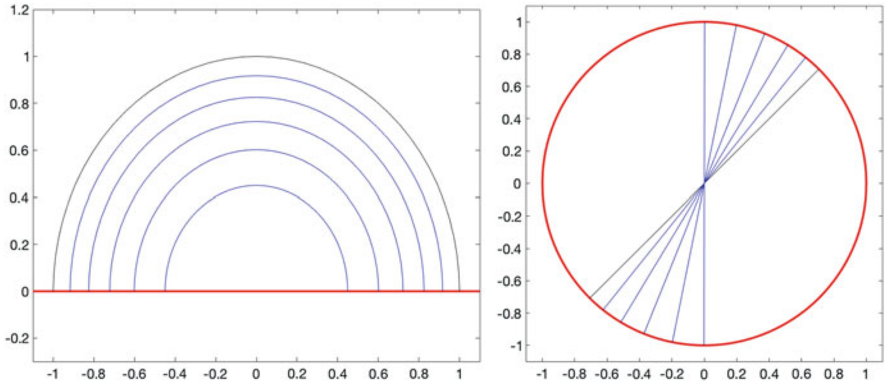


Fig. 1 $\mathbf{X}^n$ at $t^n = 0, 0.08, 0.16, 0.24, 0.32, 0.4$ for the shrinking semi-circle (left), and $t^n = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ for the rotating diameter (right)

Table 1 Errors and eocs for the shrinking semi circle with $\alpha = 1$ (left) and $\alpha = 0.5$ (right)

J	N	$\mathcal{E}_1 \times 10^3$	eoc_1	$\mathcal{E}_2 \times 10^4$	eoc_2
10	40	4.672	–	20.16	–
20	160	0.3997	3.55	1.859	3.44
40	640	0.02726	3.87	0.1298	3.84
80	2560	0.001742	3.97	0.008347	3.96

J	N	$\mathcal{E}_1 \times 10^3$	eoc_1	$\mathcal{E}_2 \times 10^4$	eoc_2
10	40	1.589	–	8.884	–
20	160	0.1389	3.52	0.8302	3.42
40	640	0.009514	3.87	0.05798	3.84
80	2560	0.0006087	3.97	0.003729	3.96

$i = 1, 2$, for $\alpha = 1$ (left) and $\alpha = 0.5$ (right). For both values of α we see eocs close to four, however we note that the errors for $\alpha = 0.5$ are significantly smaller than those for $\alpha = 1$.

Example 2 In the second example we set $T = 0.5$ and Ω to be the unit disc with centre (0, 0), such that $\partial\Omega$ is given by $F(x, y) = \frac{1}{2}\left(x^2 + y^2 - 1\right)$. In contrast to the previous example this example has been constructed so that $|\nabla F(p)| = 1$ is only satisfied on $\partial\Omega$. By setting $f(\rho, t) = \frac{4(\rho-\frac{1}{2})}{(1-2t)^2+1}$ the explicit solution is given by

$$\mathbf{x}(\rho, t) = \frac{2(\rho - \frac{1}{2})}{\sqrt{(1-2t)^2 + 1}} (1 - 2t, 1)^T ,$$

such that $\Gamma(t)$ is a rotating straight line that spans the diameter of Ω. In the right-hand plot of Fig. 1 we display: $\mathbf{X}^0$ in black, $\mathbf{X}^n$ at $t^n = 0.1k$, $k = 1, \ldots, 5$, in blue, and $\partial\Omega$ in red, while Table 2 displays the errors $\mathcal{E}_i$, $i = 1, 2$, for $\alpha = 1$ (left) and $\alpha = 0.5$ (right). As in Example 1, both values of α exhibit eocs close to four, with the errors obtained using $\alpha = 0.5$ being smaller than those obtain using $\alpha = 1$. However the difference in the errors for the two values of α in this example is much smaller than the difference in the errors for the two values of α in Example 1, we believe that this is due to the fact that in this example $\mathbf{x}$ is a linear function.

Table 2 Errors and eocs for the rotating diameter with $\alpha = 1$ (left) and $\alpha = 0.5$ (right)

J	N	$\mathcal{E}_1 \times 10^4$	eoc_1	$\mathcal{E}_2 \times 10^5$	eoc_2	J	N	$\mathcal{E}_1 \times 10^4$	eoc_1	$\mathcal{E}_2 \times 10^5$	eoc_2
10	50	1.440	–	3.040	–	10	50	1.181	–	2.710	–
20	200	0.09198	3.97	0.1925	3.98	20	200	0.07459	3.98	0.1716	3.98
40	800	0.005780	3.99	0.01207	4.00	40	800	0.004674	4.00	0.01076	4.00
80	3200	0.0003617	4.00	0.0007552	4.00	80	3200	0.0002923	4.00	0.0006727	4.00

Table 3 Errors and eocs for the rotating diameter for scheme proposed in [5]

J	N	$\mathcal{E}_1 \times 10^4$	eoc_1	$\mathcal{E}_2 \times 10^5$	eoc_2	$\mathcal{E}_3 \times 10^3$	eoc_3
10	50	43.83		76.20		5.771	
20	200	3.175	3.79	5.442	3.81	1.563	0.94
40	800	0.2076	3.93	0.3542	3.94	0.3989	0.99
80	3200	0.01317	3.98	0.02243	3.98	0.1003	1.00

To demonstrate Remark 1 in Sect. 2, we include Table 3 in which we display errors obtained using the scheme in [5]. In particular we display $\mathcal{E}_i$, $i = 1, 2, 3$, with

$$\mathcal{E}_3 := \sup_{n=0,\ldots,N} \sup_{j=0,J} |F(\mathbf{X}_j^n)|.$$

Comparing the errors in Tables 2 and 3 we see that the magnitude of the errors for the Newton scheme, (14a)–(14d), are significantly smaller than the errors for the linear scheme in [5].

3.3 Experimental Order of Convergence of the Coupled Scheme (11)–(13)

We conclude our numerical results by investigating the experimental order of convergence of the coupled scheme (11)–(13). In addition to monitoring the errors $\mathcal{E}_i$, $i = 1, 2$, we also monitor

$$\mathcal{E}_4 := \sup_{n=0,\ldots,N} \|I^h(w^n) - W^n\|^2_{L^2(\mathbb{I})}, \quad \mathcal{E}_5 := \sum_{n=1}^{n} \Delta t_n \|I^h(w^n_\rho) - W^n_\rho\|^2_{L^2(\mathbb{I})}.$$

We adopt the same set-up as in Example 2, with $T = 0.5$ and Ω being the unit disc with centre $(0, 0)$, such that $\partial\Omega$ is given by $F(x, y) = \frac{1}{2}\left(x^2 + y^2 - 1\right)$. Setting

Table 4 Errors and eocs for the parabola defined on the rotating diameter with $\alpha = 0.5$

J	N	$\mathcal{E}_1 \times 10^4$	eoc_1	$\mathcal{E}_2 \times 10^5$	eoc_2	$\mathcal{E}_4 \times 10^6$	eoc_4	$\mathcal{E}_5 \times 10^6$	eoc_5
10	50	1.205	–	3.756	–	1.207	–	3.073	–
20	200	0.07643	3.98	0.2453	3.94	0.07829	3.95	0.2010	3.93
40	800	0.004795	3.99	0.01551	3.98	0.004937	3.99	0.01271	3.98
80	3200	0.0003000	4.00	0.0009721	4.00	0.0003093	4.00	0.0007967	4.00

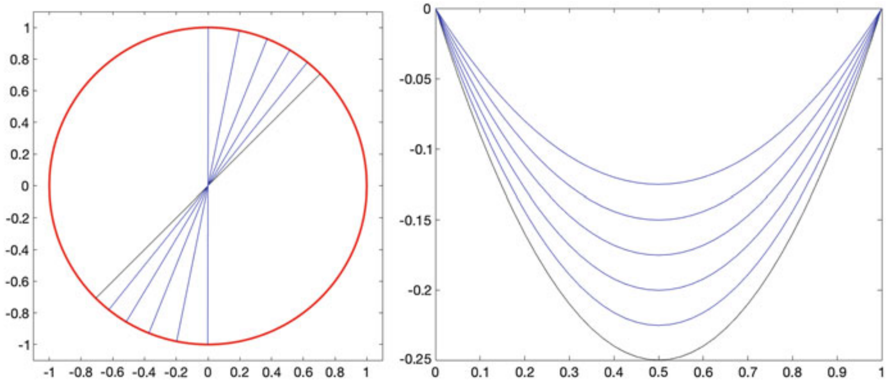


Fig. 2 $\mathbf{X}^n$ at $t^n = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ for the rotating diameter (left) and W^n at $t^n = 0, 0.1, 0.2, 0.3, 0.4, 0.5$ for the shrinking parabola (right)

$f(\rho, t) = \frac{4\left(\rho^2 - \frac{w(\rho,t)}{1-t} - \frac{1}{2}\right)}{(1-2t)^2+1}$ and $g = \frac{t-1}{2} - \frac{w(\rho,t)}{1-t}$, the explicit solution is given by

$$x = \frac{2(\rho - \frac{1}{2})}{\sqrt{(1-2t)^2+1}}(1-2t, 1)^T \ \text{ and } \ w = (1-t)\rho(\rho-1),$$

such that w describes a shrinking parabola and, as in Example 2, $\Gamma(t)$ is a rotating straight line that spans the diameter of Ω. In the left-hand plot of Fig. 2 we display: $\mathbf{X}^0$ in black, $\mathbf{X}^n$, at $t^n = 0.1k$, $k = 1, \ldots, 5$, in blue, and $\partial\Omega$ in red, while in the right-hand plot we display: W^0 in black and W^n, at $t^n = 0.1k$, $k = 1, \ldots, 5$ in blue. In Table 4 we present the experimental order of convergence for the errors obtained using $\alpha = 0.5$, we do not present the errors for $\alpha = 1$ since they are very similar to those obtained using $\alpha = 0.5$. For all the four errors we see eocs close to four.

Remark 2 In the three examples presented above if we take $\Delta t = Ch$ we observe eocs close to two rather than the eocs close to four that we observe above for $\Delta t = Ch^2$. Similar convergence behaviour was observed in [1].

Acknowledgments JVY gratefully acknowledges the support of the EPSRC grant 1805391. VS would like to thank the Isaac Newton Institute for Mathematical Sciences for support and hospitality during the programme *Geometry, compatibility and structure preservation in* *computational differential equations* when work on this paper was undertaken. This work was supported by: EPSRC grant number EP/R014604/1.

References

1. J.W. Barrett, K. Deckelnick and V. Styles: *Numerical analysis for a system coupling curve evolution to reaction diffusion on the curve*. SIAM Journal on Numerical Analysis, vol 55 (2017), p.1080–1100
2. J.W. Barrett, H. Garcke and R. Nürnberg: *On the variational approximation of combined second and fourth order geometric evolution equations* SIAM J. Scientific Comput. **29**, (2007), 1006-1041.
3. J.W. Barrett, H. Garcke and R. Nürnberg: *The approximation of planar curve evolutions by stable fully implicit finite element schemes that equidistribute*. Numerical Methods for Partial Differential Equations, **27** (2011), pp. 1–30.
4. K. Deckelnick and G. Dziuk: *On the approximation of the curve shortening flow*. Calculus of Variations, Applications and Computations, (Pitman, 1995), p.100–108
5. K. Deckelnick and C.M. Elliott: *Finite element error bounds for a curve shrinking with prescribed normal contact to a fixed boundary*. IMA Journal of Numerical Analysis, vol 18 (Oxford University Press, 1998), p.635–654
6. K. Deckelnick, C.M. Elliott and V. Styles: *Numerical diffusion-induced grain boundary motion*. Interfaces and Free Boundaries, vol 3 (1998), p.393–414
7. G. Dziuk: *Convergence of a semi-discrete scheme for the curve shortening flow*. Mathematical Models and Methods in Applied Sciences, vol 4 (World Scientific, 1994), p.589–606
8. C.M. Elliott and H. Fritz: *On approximations of the curve shortening flow and of the mean curvature flow based on the DeTurck trick*. IMA Journal of Numerical Analysis, vol 37 (2016), p.543–603
9. P. Pozzi and B. Stinner: *Curve shortening flow coupled to lateral diffusion*. Numerische Mathematik, vol 135 (Springer, 2017), p.1171–1205

The Newmark Method and a Space–Time FEM for the Second–Order Wave Equation

Marco Zank

Abstract For the second–order wave equation, we compare the Newmark Galerkin method with a stabilised space–time finite element method for tensor–product space–time discretisations with piecewise multilinear, continuous ansatz and test functions leading to an unconditionally stable Galerkin–Petrov scheme, which satisfies a space–time error estimate. We show that both methods require to solve a linear system with the same system matrix. In particular, the stabilised space–time finite element method can be solved sequentially in time as the Newmark Galerkin method. However, the treatment of the right–hand side of the wave equation is different, where the Newmark Galerkin method requires more regularity.

1 Introduction

In this work, we compare a stabilised space–time finite element method, analysed in [3, 5, 6], and the Newmark Galerkin method for the model problem of the homogeneous Dirichlet problem for the second–order wave equation,

$$\left.\begin{aligned} \partial_{tt}u(x,t) - \Delta_x u(x,t) &= f(x,t) && \text{for } (x,t) \in Q = \Omega \times (0,T), \\ u(x,t) &= 0 && \text{for } (x,t) \in \Sigma = \partial\Omega \times [0,T], \\ u(x,0) &= u_0(x) && \text{for } x \in \Omega, \\ \partial_t u(x,0) &= v_0(x) && \text{for } x \in \Omega, \end{aligned}\right\} \tag{1}$$

where $\Omega = (0, L)$ is an interval for $d = 1$, or Ω is polygonal for $d = 2$, or Ω is polyhedral for $d = 3$, $T > 0$ is a terminal time, $u_0 \in H^1_0(\Omega)$, $v_0 \in L^2(\Omega)$ are given initial conditions and $f \in L^1(0, T; L^2(\Omega))$ is a given right–hand side. To derive a

M. Zank (✉)
Universität Wien, Wien, Austria
e-mail: marco.zank@univie.ac.at

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_122

space–time variational formulation, we define the space–time Sobolev spaces by

$$H^{1,1}_{0;0,}(Q) := L^2(0,T;H^1_0(\Omega)) \cap H^1_{0,}(0,T;L^2(\Omega)) \subset H^1(Q),$$

$$H^{1,1}_{0;,0}(Q) := L^2(0,T;H^1_0(\Omega)) \cap H^1_{,0}(0,T;L^2(\Omega)) \subset H^1(Q),$$

$$H^{1,1}_{0;}(Q) := L^2(0,T;H^1_0(\Omega)) \cap H^1(0,T;L^2(\Omega)) \subset H^1(Q),$$

where $v \in H^1_{0,}(0,T;L^2(\Omega))$ satisfies $\|v(\cdot,0)\|_{L^2(\Omega)} = 0$ and $w \in H^1_{,0}(0,T;L^2(\Omega))$ fulfils $\|w(\cdot,T)\|_{L^2(\Omega)} = 0$, see [5] for more details. In addition, we introduce the bounded bilinear form $a(\cdot,\cdot)\colon\ H^1(Q) \times H^1(Q) \to \mathbb{R}$ by

$$a(u,w) := -\langle \partial_t u, \partial_t w\rangle_{L^2(Q)} + \langle \nabla_x u, \nabla_x w\rangle_{L^2(Q)}, \qquad u \in H^1(Q),\ w \in H^1(Q).$$

Due to [1, Chapter IV, Section 3], the wave equation (1) admits a unique solution $u \in H^{1,1}_{0;}(Q)$ with $u(\cdot,0) = u_0$ in $L^2(\Omega)$, fulfilling

$$\forall w \in H^{1,1}_{0;,0}(Q)\colon \quad a(u,w) = \langle f, w\rangle_{L^2(Q)} + \langle v_0, w(\cdot,0)\rangle_{L^2(\Omega)}.$$

To treat the initial condition u_0 in discretisations, we consider the splitting $u = \hat{u} + U_0$ with $U_0 \in H^{1,1}_{0;}(Q) \cap C([0,T];H^1_0(\Omega))$, see [1, Chapter IV, Theorem 4.2], satisfying

$$U_0(\cdot,0) = u_0 \text{ in } H^1_0(\Omega) \quad \text{and} \quad \forall w \in H^{1,1}_{0;,0}(Q)\colon \quad a(U_0,w) = 0, \tag{2}$$

and with $\hat{u} \in H^{1,1}_{0;0,}(Q)$, satisfying

$$\forall w \in H^{1,1}_{0;,0}(Q)\colon \quad a(\hat{u},w) = \langle f, w\rangle_{L^2(Q)} + \langle v_0, w(\cdot,0)\rangle_{L^2(\Omega)} - a(U_0,w). \tag{3}$$

Note that the variational formulations (2) and (3) are uniquely solvable, satisfying corresponding stability estimates, see [1, 4–6] for details.

2 Stabilised Space–Time Finite Element Method

In this section, we recall the stabilised space–time finite element method with piecewise multilinear, continuous functions for the wave equation (1) as analysed in [3, 5, 6], where we state also the resulting linear system.

For a tensor–product ansatz, we consider admissible decompositions

$$\overline{Q} = \overline{\Omega} \times [0,T] = \bigcup_{i=1}^{N_x} \overline{\omega_i} \times \bigcup_{\ell=1}^{N_t} [t_{\ell-1}, t_\ell]$$

with $N := N_x \cdot N_t$ space–time elements, where the time intervals $(t_{\ell-1}, t_\ell)$ with mesh size $h_{t,\ell}$ are defined via the decomposition

$$0 = t_0 < t_1 < t_2 < \cdots < t_{N_t - 1} < t_{N_t} = T \tag{4}$$

of the time interval $(0, T)$. For the spatial domain Ω, we consider a shape–regular sequence $(\mathcal{T}_\nu)_{\nu \in \mathbb{N}}$ of admissible decompositions $\mathcal{T}_\nu := \{\omega_i \subset \mathbb{R}^d : i = 1, \dots, N_x\}$ of Ω into finite elements $\omega_i \subset \mathbb{R}^d$ with mesh size $h_{x,i}$. The spatial elements ω_i are intervals for $d = 1$, triangles or quadrilaterals for $d = 2$, and tetrahedra or hexahedra for $d = 3$. Next, we introduce the finite element space

$$Q_h^1(Q) := V_{h_x,0}(\Omega) \otimes S_{h_t}^1(0, T)$$

of piecewise multilinear, continuous functions, i.e.

$$V_{h_x,0}(\Omega) = \operatorname{span}\{\psi_j\}_{j=1}^{M_x} \subset H_0^1(\Omega), \quad S_{h_t}^1(0, T) = \operatorname{span}\{\varphi_\ell\}_{\ell=0}^{N_t} \subset H^1(0, T).$$

In fact, $V_{h_x,0}(\Omega)$ is either the space $S_{h_x}^1(\Omega) \cap H_0^1(\Omega)$ of piecewise linear, continuous functions on intervals ($d = 1$), triangles ($d = 2$), and tetrahedra ($d = 3$), or $V_{h_x,0}(\Omega)$ is the space $Q_{h_x}^1(\Omega) \cap H_0^1(\Omega)$ of piecewise linear/bilinear/trilinear, continuous functions on intervals ($d = 1$), quadrilaterals ($d = 2$), and hexahedra ($d = 3$). Additionally, $M_{h_x} \in \mathbb{R}^{M_x \times M_x}$ and $A_{h_x} \in \mathbb{R}^{M_x \times M_x}$ denote mass and stiffness matrices defined via

$$M_{h_x}[i, j] = \langle \psi_j, \psi_i \rangle_{L^2(\Omega)}, \quad A_{h_x}[i, j] = \langle \nabla_x \psi_j, \nabla_x \psi_i \rangle_{L^2(\Omega)}, \; i, j = 1, \dots, M_x. \tag{5}$$

Moreover, the L^2 projection $Q_{h_x} : L^2(\Omega) \to V_{h_x,0}(\Omega)$ on the piecewise linear, continuous functions is given for functions $u \in L^2(\Omega)$ as the solution of the variational formulation to find $Q_{h_x} u \in V_{h_x,0}(\Omega)$ such that

$$\forall v_{h_x} \in V_{h_x,0}(\Omega) : \quad \langle Q_{h_x} u, v_{h_x} \rangle_{L^2(\Omega)} = \langle u, v_{h_x} \rangle_{L^2(\Omega)}. \tag{6}$$

Analogously, $Q_{h_x}^1 : H_0^1(\Omega) \to V_{h_x,0}(\Omega)$ is the H_0^1 projection defined by

$$\forall v_{h_x} \in V_{h_x,0}(\Omega) : \quad \langle \nabla_x Q_{h_x}^1 u, \nabla_x v_{h_x} \rangle_{L^2(\Omega)} = \langle \nabla_x u, \nabla_x v_{h_x} \rangle_{L^2(\Omega)} \tag{7}$$

for given $u \in H_0^1(\Omega)$. With these notations, we define the perturbed bilinear form

$$a_h(u_h, w_h) := -\langle \partial_t u_h, \partial_t w_h \rangle_{L^2(Q)} + \sum_{m=1}^{d} \langle \partial_{x_m} u_h, Q_{h_t}^0 \partial_{x_m} w_h \rangle_{L^2(Q)}, \; u_h, w_h \in Q_h^1(Q),$$

where $Q^0_{h_t}$ denotes the L^2 projection on the space of piecewise constant functions with respect to the time mesh (4). To approximate the solution of the problem (3), we consider the variational formulation to find $\widetilde{u}_h \in Q^1_h(Q) \cap H^{1,1}_{0;0,}(Q)$ such that

$$a_h(\widetilde{u}_h, w_h) = \langle f, w_h\rangle_{L^2(Q)} + \langle v_0, w_h(\cdot, 0)\rangle_{L^2(\Omega)} - a_h(I_{h_t} Q^1_{h_x} U_0, w_h) \tag{8}$$

for all $w_h \in Q^1_h(Q) \cap H^{1,1}_{0;\,,0}(Q)$, where U_0 is replaced by $I_{h_t} Q^1_{h_x} U_0 \in Q^1_h(Q)$, i.e.

$$I_{h_t} Q^1_{h_x} U_0(x,t) = \sum_{\ell=0}^{N_t} \sum_{j=1}^{M_x} (A^{-1}_{h_x} \underline{U}^\ell_0)[j]\, \psi_j(x)\varphi_\ell(t), \qquad \underline{U}^\ell_0[i] := \langle \nabla_x U_0(\cdot, t_\ell), \nabla_x \psi_i\rangle_{L^2(\Omega)}$$

which is the interpolant with respect to the temporal and the H^1_0 projection with respect to the spatial variables, see (7). The discrete variational formulation (8) admits a unique solution $\widetilde{u}_h \in Q^1_h(Q) \cap H^{1,1}_{0;0,}(Q)$ without any CFL condition, see [3, 5, 6]. Note that a CFL condition is required when the perturbed bilinear form $a_h(\cdot,\cdot)$ on the left–hand side in (8) is replaced by $a(\cdot,\cdot)$, see [4, 5].

Next, we rewrite the variational formulation (8) to state the corresponding linear system. The unique solution $\widetilde{u}_h \in Q^1_h(Q) \cap H^{1,1}_{0;0,}(Q)$ of the discrete variational formulation (8) admits the representation $\widetilde{u}_h(x,t) = \sum_{\ell=1}^{N_t} \sum_{j=1}^{M_x} \widetilde{u}^\ell_j \psi_j(x)\varphi_\ell(t)$. So, we define the function $\overline{u}_h \in Q^1_h(Q) \cap H^{1,1}_{0;0,}(Q)$ by

$$\overline{u}_h(x,t) := \sum_{\ell=1}^{N_t} \sum_{j=1}^{M_x} \overline{u}^\ell_j \psi_j(x)\varphi_\ell(t) := \sum_{\ell=1}^{N_t} \sum_{j=1}^{M_x} \left\{ \widetilde{u}^\ell_j + (A^{-1}_{h_x} \underline{U}^\ell_0)[j] \right\} \psi_j(x)\varphi_\ell(t),$$

satisfying the to (8) equivalent discrete variational formulation

$$a_h(\overline{u}_h, w_h) = \langle f, w_h\rangle_{L^2(Q)} + \langle v_0, w_h(\cdot, 0)\rangle_{L^2(\Omega)} - \sum_{j=1}^{M_x} (A^{-1}_{h_x} \underline{u}_0)[j]\, a_h(\psi_j \varphi_0, w_h) \tag{9}$$

for all $w_h \in Q^1_h(Q) \cap H^{1,1}_{0;\,,0}(Q)$ with $\underline{U}^0_0 =: \underline{u}_0 \in \mathbb{R}^{M_x}$. Then, the approximate solution $u \approx u_h := \widetilde{u}_h + I_{h_t} Q^1_{h_x} U_0 \in Q^1_h(Q)$ of the wave equation (1) is given by

$$u_h(x,t) = \varphi_0(t) \cdot (Q^1_{h_x} u_0)(x) + \overline{u}_h(x,t) = \sum_{\ell=0}^{N_t} \sum_{j=1}^{M_x} u^\ell_j \psi_j(x)\varphi_\ell(t) \tag{10}$$

with $u^0_j := (A^{-1}_{h_x} \underline{u}_0)[j]$ for $j = 1, \ldots, M_x$, and $u^\ell_j := \overline{u}^\ell_j$ for $j = 1, \ldots, M_x$, $\ell = 1, \ldots, N_t$. For $\ell = 0, \ldots, N_t$, set $\underline{U}^\ell = \left(u^\ell_1, u^\ell_2, \ldots, u^\ell_{M_x}\right)^\top \in \mathbb{R}^{M_x}$. Hence, we obtain that

$$A_{h_x}\underline{U}^0 = \underline{u}_0 \tag{11}$$

with $\underline{u}_0[j] = \underline{U}_0^0[j] = \langle \nabla_x U_0(\cdot, 0), \nabla_x \psi_j \rangle_{L^2(\Omega)} = \langle \nabla_x u_0, \nabla_x \psi_j \rangle_{L^2(\Omega)}$, $j = 1, \ldots, M_x$. So, the discrete variational formulation (9) is equivalent to solving the global linear system

$$\left(-A_{h_t} \otimes M_{h_x} + \widetilde{M}_{h_t} \otimes A_{h_x}\right)(\underline{U}^1, \underline{U}^2, \ldots, \underline{U}^{N_t})^\top = (\underline{\hat{F}}^0, \underline{\hat{F}}^1, \ldots, \underline{\hat{F}}^{N_t-1})^\top \tag{12}$$

with the spatial mass and stiffness matrices (5), the temporal matrices

$$A_{h_t} = \begin{pmatrix} \frac{-1}{h_{t,1}} & & & & \\ \frac{1}{h_{t,1}} + \frac{1}{h_{t,2}} & \frac{-1}{h_{t,2}} & & & \\ \frac{-1}{h_{t,2}} & \frac{1}{h_{t,2}} + \frac{1}{h_{t,3}} & \frac{-1}{h_{t,3}} & & \\ & \ddots & \ddots & \ddots & \\ & & \frac{-1}{h_{t,N_t-1}} & \frac{1}{h_{t,N_t-1}} + \frac{1}{h_{t,N_t}} & \frac{-1}{h_{t,N_t}} \end{pmatrix} \in \mathbb{R}^{N_t \times N_t},$$

$$\widetilde{M}_{h_t} = \frac{1}{4}\begin{pmatrix} h_{t,1} & & & & \\ h_{t,1} + h_{t,2} & h_{t,2} & & & \\ h_{t,2} & h_{t,2} + h_{t,3} & h_{t,3} & & \\ & \ddots & \ddots & \ddots & \\ & & h_{t,N_t-1} & h_{t,N_t-1} + h_{t,N_t} & h_{t,N_t} \end{pmatrix} \in \mathbb{R}^{N_t \times N_t}$$

and the vectors of the right–hand side, $k = 0, \ldots, N_t - 1$, $i = 1, \ldots, M_x$,

$$\underline{\hat{F}}^k[i] := \langle f, \psi_i \varphi_k \rangle_{L^2(Q)} + \varphi_k(0)\langle v_0, \psi_i \rangle_{L^2(\Omega)} - \sum_{j=1}^{M_x} (A_{h_x}^{-1}\underline{u}_0)[j]\, a_h(\psi_j \varphi_0, \psi_i \varphi_k).$$

After some calculations, we obtain that the linear system (11) and the global linear system (12) are equivalent to sequentially solving the linear systems

$$A_{h_x}\underline{U}^0 = \underline{u}_0, \tag{13}$$

$$\left(\frac{1}{h_{t,1}} M_{h_x} + \frac{h_{t,1}}{4} A_{h_x}\right)\underline{U}^1 = \underline{f}^0 + \underline{v}_0 + \left(\frac{1}{h_{t,1}} M_{h_x} - \frac{h_{t,1}}{4} A_{h_x}\right)\underline{U}^0, \tag{14}$$

$$\begin{aligned}\left(\frac{1}{h_{t,\ell}} M_{h_x} + \frac{h_{t,\ell}}{4} A_{h_x}\right)\underline{U}^\ell = \underline{f}^{\ell-1} &+ \left(\left(\frac{1}{h_{t,\ell-1}} + \frac{1}{h_{t,\ell}}\right) M_{h_x} - \frac{h_{t,\ell-1} + h_{t,\ell}}{4} A_{h_x}\right)\underline{U}^{\ell-1} \\ &- \left(\frac{1}{h_{t,\ell-1}} M_{h_x} + \frac{h_{t,\ell-1}}{4} A_{h_x}\right)\underline{U}^{\ell-2}\end{aligned} \tag{15}$$

for $\ell = 2, \dots, N_t$, where the vectors $\underline{u}_0$, $\underline{v}_0$, $\underline{f}^\ell \in \mathbb{R}^{M_x}$ are given by, $i = 1, \dots, M_x$, $\ell = 0, \dots, N_t - 1$,

$$\underline{u}_0[i] = \langle \nabla_x u_0, \nabla_x \psi_i \rangle_{L^2(\Omega)}, \quad \underline{v}_0[i] := \langle v_0, \psi_i \rangle_{L^2(\Omega)}, \quad \underline{f}^\ell[i] := \langle f, \psi_i \varphi_\ell \rangle_{L^2(Q)}.$$

The main results for the proposed space–time method, which leads to solve the linear systems (13)–(15), are the unconditional stability, i.e. no CFL condition is needed, and the space–time error estimate with $h_x = \max h_{x,i}$, $h_t = \max h_{t,\ell}$, which are summarised in the following theorem. The proof is analogous to the proofs in [3, 5] with some additional difficulties due to the handling of u_0. Since this is far behind the scope of this work, details will be discussed elsewhere.

Theorem 1 *Let the solution u of* (1) *and Ω be sufficiently regular. Then, for the approximation $u_h \approx u$, given in* (10)*, we have the space–time estimates*

$$\|u_h\|_{L^2(Q)} \le C(f, u_0, v_0, T, \Omega) \quad \textit{and} \quad \|u - u_h\|_{L^2(Q)} \le \widetilde{C}(u, T, \Omega) \cdot (h_x^2 + h_t^2).$$

3 Newmark Galerkin Method

In this section, we recall the Newmark Galerkin method for the wave equation (1) with a right–hand side $f \in C([0,T]; L^2(\Omega))$, where we state also the resulting linear systems. With the notations of Sect. 2, we introduce the approximations

$$U_{h_x,\ell}(x) := \sum_{i=1}^{M_x} U_i^\ell \psi_i(x) \approx u(x, t_\ell) \quad \text{and} \quad \hat{U}_{h_x,\ell}(x) := \sum_{i=1}^{M_x} \hat{U}_i^\ell \psi_i(x) \approx \partial_t u(x, t_\ell)$$

for $x \in \Omega$ and $\ell \in \{0, \dots, N_t\}$, where U_i^ℓ, $\hat{U}_i^\ell \in \mathbb{R}$ are the unknown coefficients of $U_{h_x,\ell}$, $\hat{U}_{h_x,\ell} \in V_{h_x,0}(\Omega) \subset H_0^1(\Omega)$. Furthermore, for $(x,t) \in Q$, we set

$$u_h(x,t) := \sum_{\ell=0}^{N_t} \sum_{i=1}^{M_x} U_i^\ell \psi_i(x) \varphi_\ell(t) = \sum_{\ell=0}^{N_t} U_{h_x,\ell}(x) \varphi_\ell(t) \approx u(x,t),$$

$$\hat{u}_h(x,t) := \sum_{\ell=0}^{N_t} \sum_{i=1}^{M_x} \hat{U}_i^\ell \psi_i(x) \varphi_\ell(t) = \sum_{\ell=0}^{N_t} \hat{U}_{h_x,\ell}(x) \varphi_\ell(t) \approx \partial_t u(x,t),$$

i.e. u_h, $\hat{u}_h \in Q_h^1(Q)$. For the wave equation (1), a conforming discretisation in space with $V_{h_x,0}(\Omega) \subset H_0^1(\Omega)$ in combination with the Newmark method, see [2, (8.6-4), (8.6-5), (8.6-6), page 205], leads to the so–called Newmark Galerkin method to find the functions $U_{h_x,\ell},\ \hat{U}_{h_x,\ell} \in V_{h_x,0}(\Omega) \subset H_0^1(\Omega)$ for $\ell \in \{0,\dots,N_t\}$ such that

$$U_{h_x,0} = Q_{h_x}^1 u_0, \quad \hat{U}_{h_x,0} = Q_{h_x} v_0 \tag{16}$$

and for $\ell = 1,\dots,N_t$,

$$\begin{aligned}\forall v_{h_x} \in V_{h_x,0}(\Omega)\colon \quad & \frac{1}{h_{t,\ell}^2}\langle U_{h_x,\ell} - U_{h_x,\ell-1} - h_{t,\ell}\hat{U}_{h_x,\ell-1}, v_{h_x}\rangle_{L^2(\Omega)} \\ +\frac{1}{4}\langle \nabla_x U_{h_x,\ell} + \nabla_x U_{h_x,\ell-1}, \nabla_x v_{h_x}\rangle_{L^2(\Omega)} &= \frac{1}{4}\langle f(\cdot,t_\ell) + f(\cdot,t_{\ell-1}), v_{h_x}\rangle_{L^2(\Omega)},\end{aligned} \tag{17}$$

$$\begin{aligned}\forall \hat{v}_{h_x} \in V_{h_x,0}(\Omega)\colon \quad & \frac{1}{h_{t,\ell}}\langle \hat{U}_{h_x,\ell} - \hat{U}_{h_x,\ell-1}, \hat{v}_{h_x}\rangle_{L^2(\Omega)} \\ +\frac{1}{2}\langle \nabla_x U_{h_x,\ell} + \nabla_x U_{h_x,\ell-1}, \nabla_x \hat{v}_{h_x}\rangle_{L^2(\Omega)} &= \frac{1}{2}\langle f(\cdot,t_\ell) + f(\cdot,t_{\ell-1}), \hat{v}_{h_x}\rangle_{L^2(\Omega)},\end{aligned} \tag{18}$$

where the projections Q_{h_x}, $Q_{h_x}^1$ are defined in (6) and (7). The Newmark Galerkin method (16)–(18) is equivalent to the linear systems

$$A_{h_x}\underline{U}^0 = \underline{u}_0, \quad M_{h_x}\underline{\hat{U}}^0 = \underline{v}_0, \tag{19}$$

and for all $\ell = 1,\dots,N_t$,

$$\begin{aligned}\left(M_{h_x} + \frac{h_{t,\ell}^2}{4}A_{h_x}\right)\underline{U}^\ell = \left(M_{h_x} - \frac{h_{t,\ell}^2}{4}A_{h_x}\right)\underline{U}^{\ell-1} \\ +h_{t,\ell}M_{h_x}\underline{\hat{U}}^{\ell-1} + \frac{h_{t,\ell}^2}{4}\left(\underline{F}^\ell + \underline{F}^{\ell-1}\right),\end{aligned} \tag{20}$$

$$M_{h_x}\underline{\hat{U}}^\ell = M_{h_x}\underline{\hat{U}}^{\ell-1} - \frac{h_{t,\ell}}{2}A_{h_x}\left(\underline{U}^\ell + \underline{U}^{\ell-1}\right) + \frac{h_{t,\ell}}{2}\left(\underline{F}^\ell + \underline{F}^{\ell-1}\right), \tag{21}$$

where $M_{h_x}, A_{h_x} \in \mathbb{R}^{M_x \times M_x}$ are the mass and stiffness matrices (5) and the vectors $\underline{u}_0,\ \underline{v}_0,\ \underline{F}^\ell \in \mathbb{R}^{M_x}$ are defined by

$$\underline{u}_0[i] := \langle \nabla_x u_0, \nabla_x \psi_i\rangle_{L^2(\Omega)}, \quad \underline{v}_0[i] := \langle v_0, \psi_i\rangle_{L^2(\Omega)}, \quad \underline{F}^\ell[i] := \langle f(\cdot,t_\ell), \psi_i\rangle_{L^2(\Omega)}$$

for $i = 1, \dots, M_x$, $\ell = 0, \dots, N_t$. Solving the linear systems (20) and (21) is equivalent to solving

$$\left(\frac{1}{h_{t,\ell}} M_{h_x} + \frac{h_{t,\ell}}{4} A_{h_x}\right) \underline{U}^{\ell} - \left(\frac{1}{h_{t,\ell}} M_{h_x} - \frac{h_{t,\ell}}{4} A_{h_x}\right) \underline{U}^{\ell-1} - M_{h_x} \underline{\hat{U}}^{\ell-1} = \frac{h_{t,\ell}}{4} \left(\underline{F}^{\ell} + \underline{F}^{\ell-1}\right), \tag{22}$$

$$M_{h_x} \underline{\hat{U}}^{\ell} - M_{h_x} \underline{\hat{U}}^{\ell-1} = -\frac{h_{t,\ell}}{2} A_{h_x} \left(\underline{U}^{\ell} + \underline{U}^{\ell-1}\right) + \frac{h_{t,\ell}}{2} \left(\underline{F}^{\ell} + \underline{F}^{\ell-1}\right) \tag{23}$$

for $\ell = 1, \dots, N_t$.

The difference of (22) for ℓ and $\ell - 1$ is given by

$$\begin{aligned} &\frac{h_{t,\ell}}{4} \left(\underline{F}^{\ell} + \underline{F}^{\ell-1}\right) - \frac{h_{t,\ell-1}}{4} \left(\underline{F}^{\ell-1} + \underline{F}^{\ell-2}\right) \\ &= \left(\frac{1}{h_{t,\ell}} M_{h_x} + \frac{h_{t,\ell}}{4} A_{h_x}\right) \underline{U}^{\ell} + \left(-\left(\frac{1}{h_{t,\ell}} + \frac{1}{h_{t,\ell-1}}\right) M_{h_x} + \frac{h_{t,\ell} - h_{t,\ell-1}}{4} A_{h_x}\right) \underline{U}^{\ell-1} \\ &\quad + \left(\frac{1}{h_{t,\ell-1}} M_{h_x} - \frac{h_{t,\ell-1}}{4} A_{h_x}\right) \underline{U}^{\ell-2} \qquad \underbrace{-M_{h_x} \underline{\hat{U}}^{\ell-1} + M_{h_x} \underline{\hat{U}}^{\ell-2}}_{= \frac{h_{t,\ell-1}}{2} A_{h_x} \left(\underline{U}^{\ell-1} + \underline{U}^{\ell-2}\right) - \frac{h_{t,\ell-1}}{2} \left(\underline{F}^{\ell-1} + \underline{F}^{\ell-2}\right)} \\ &= \left(\frac{1}{h_{t,\ell}} M_{h_x} + \frac{h_{t,\ell}}{4} A_{h_x}\right) \underline{U}^{\ell} + \left(-\left(\frac{1}{h_{t,\ell}} + \frac{1}{h_{t,\ell-1}}\right) M_{h_x} + \frac{h_{t,\ell} + h_{t,\ell-1}}{4} A_{h_x}\right) \underline{U}^{\ell-1} \\ &\quad + \left(\frac{1}{h_{t,\ell-1}} M_{h_x} + \frac{h_{t,\ell-1}}{4} A_{h_x}\right) \underline{U}^{\ell-2} - \frac{h_{t,\ell-1}}{2} \left(\underline{F}^{\ell-1} + \underline{F}^{\ell-2}\right) \end{aligned} \tag{24}$$

for $\ell = 2, \dots, N_t$, where (23) is used for $\ell - 1$.

Hence, with (19), $\ell = 1$ in (22) and (24), the Newmark Galerkin method (16)–(18) is equivalent to the linear systems

$$A_{h_x} \underline{U}^0 = \underline{u}_0, \tag{25}$$

$$\left(\frac{1}{h_{t,1}} M_{h_x} + \frac{h_{t,1}}{4} A_{h_x}\right) \underline{U}^1 = \frac{h_{t,1}}{4} \left(\underline{F}^1 + \underline{F}^0\right) + \underline{v}_0 + \left(\frac{1}{h_{t,1}} M_{h_x} - \frac{h_{t,1}}{4} A_{h_x}\right) \underline{U}^0, \tag{26}$$

$$\begin{aligned} \left(\frac{1}{h_{t,\ell}} M_{h_x} + \frac{h_{t,\ell}}{4} A_{h_x}\right) \underline{U}^{\ell} &= \frac{h_{t,\ell}}{4} \left(\underline{F}^{\ell} + \underline{F}^{\ell-1}\right) + \frac{h_{t,\ell-1}}{4} \left(\underline{F}^{\ell-1} + \underline{F}^{\ell-2}\right) \\ &\quad + \left(\left(\frac{1}{h_{t,\ell}} + \frac{1}{h_{t,\ell-1}}\right) M_{h_x} - \frac{h_{t,\ell} + h_{t,\ell-1}}{4} A_{h_x}\right) \underline{U}^{\ell-1} \\ &\quad - \left(\frac{1}{h_{t,\ell-1}} M_{h_x} + \frac{h_{t,\ell-1}}{4} A_{h_x}\right) \underline{U}^{\ell-2} \end{aligned} \tag{27}$$

for $\ell = 2, \dots, N_t$.

4 Comparison and Conclusions

In this section, we compare the stabilised space–time finite element method of Sect. 2 and the Newmark Galerkin method of Sect. 3. So, the Eqs. (13)–(15) and the Eqs. (25)–(27) show that the stabilised space–time finite element method in (9) differs from the Newmark Galerkin method (16)–(18) only in the treatment of the right–hand side f. For the stabilised space–time finite element method in (9), right–hand sides in $L^1(0, T; L^2(\Omega))$ are allowed, whereas in the Newmark Galerkin method (16)–(18), point evaluations $f(\cdot, t_\ell)$ occur, i.e. the right–hand side f must be continuous with respect to time. Note that in general, it is not possible to use a numerical integration formula for approximating the right–hand side parts in (14) and (15) to recover the Eqs. (26) and (27). On the other hand, for a constant right–hand side f, both methods are the same on the algebraic level. Hence, the space–time error estimate of Theorem 1 holds true also for the Newmark Galerkin method (16)–(18) and the stabilised space–time finite element method in (9) fulfils, as the Newmark Galerkin method, a conservation of total energy. In addition, the global linear system (12) can be solved sequentially in time as a two–step method. However, replacing $a_h(I_{h_t} Q^1_{h_x} U_0, w_h)$ in (8) with a different approximation of $a(U_0, w_h)$ may lead to a stabilised space–time finite element method, where the right–hand sides of the resulting linear systems differ from the Newmark Galerkin method also in the Eqs. (25)–(27).

Acknowledgments The author greatly appreciates the discussion with I. Perugia, J. Schöberl and P. Stocker.

References

1. Ladyzhenskaya, O.A.: The boundary value problems of mathematical physics, *Applied Mathematical Sciences*, vol. 49. Springer-Verlag, New York (1985)
2. Raviart, P.A., Thomas, J.M.: Introduction à l'analyse numérique des équations aux dérivées partielles. Collection Mathématiques Appliquées pour la Maîtrise. Paris etc.: Masson. 224 p. (1983)
3. Steinbach, O., Zank, M.: A stabilized space–time finite element method for the wave equation. In: Advanced Finite Element Methods with Applications. Selected papers from the 30th Chemnitz FEM Symposium 2017, (T. Apel, U. Langer, A. Meyer, O. Steinbach eds.), Lecture Notes in Computational Science and Engineering, pp. 315–342. Springer (2019)
4. Steinbach, O., Zank, M.: Coercive space-time finite element methods for initial boundary value problems. Electron. Trans. Numer. Anal. **52**, 154–194 (2020)
5. Zank, M.: Inf–sup stable space–time methods for time–dependent partial differential equations. volume 36 of Monographic Series TU Graz: Computation in Engineering and Science (2020)
6. Zlotnik, A.A.: Convergence rate estimates of finite-element methods for second-order hyperbolic equations. In: Numerical methods and applications, pp. 155–220. CRC, Boca Raton, FL (1994)

A Mixed Dimensional Model for the Interaction of a Well with a Poroelastic Material

Daniele Cerroni, Florin Radu, and Paolo Zunino

Abstract We develop a mathematical model for the interaction of the mechanics of a three-dimensional permeable reservoir or aquifer with the flow through wells. We apply a model reduction technique that represents the wells as one-dimensional channels with arbitrary configuration in the space and we introduce proper coupling conditions to account for the interaction of the wells with the bulk region. The resulting problem consists of coupled partial differential equations defined on manifolds with heterogeneous dimensionality. To highlight the potential of this modeling approach in the description of realistic scenarios, we combine it with a suitable discretization method and we discuss the results of preliminary simulations on an idealized test case containing two wells.

1 Introduction

In the late 70s Paceman proposed a mathematical method to account for a well in a reservoir simulation based on equally spaced grids [6–8]. Since then, improvement about well models has been scattered and rather scarce. We believe there is a genuine need of advanced well/reservoir interaction models that are general enough to be applied in the context of modern multiscale and multiphysics reservoir simulations, see for example [2].

With the objective to develop advanced computational models for the interaction of reservoirs with wells, we look for an approach that is appealing for industrial applications, involving realistic geological models and real configurations of multiple wells. To mitigate the difficulties of generating the reservoir model including wells and the corresponding computational cost of simulations, we propose to a use

D. Cerroni · P. Zunino (✉)
MOX, Department of Mathematics, Politecnico di Milano, Milano, Italy
e-mail: paolo.zunino@polimi.it

F. Radu
Applied Mathematics, University of Bergen, Bergen, Norway

F. J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*, Lecture Notes in Computational Science and Engineering 139, https://doi.org/10.1007/978-3-030-55874-1_123

mathematical model based on 3D description of the reservoir and a 1D description of the well, following the *embedded multiscale* model reduction strategy, originally proposed in [4] and lately refined in [1, 5]. In particular, we address here the case of a porous material that can deform under the action of pore pressure. More precisely, we consider the interaction of the flow through a well with a poroelastic medium [3].

After presenting the model, we discretize the equations using the finite element method. The approach proposed here facilitates this task, because it does not require conformity between the computational mesh of the reservoir and the one of the wells. We use the computational method to perform numerical simulations that illustrate the interaction between two neighboring wells with a poroelastic slab that deforms under the action of its weight. Interesting effects as reversal of the flow in the wells and inversion of the pressure drop between the well and the bulk appear.

2 The Mathematical Model

2.1 Three-Dimensional Model of a Well in a Deformable Material

The domain of interest is denoted as Ω and composed by two parts, Ω_w and $\Omega_p = \Omega \setminus \Omega_w$. We assume that Ω_w is the well and Ω_p the surrounding reservoir. More precisely, let Ω_w be a cylinder swept by a circle of radius ρ along a curve. Let $\boldsymbol{\lambda}(s) = [\xi(s), \nu(s), \zeta(s)],\ s \in (0, S)$ be a C^2-regular curve in the three-dimensional space. Let $\Lambda = \{\boldsymbol{\lambda}(s),\ s \in (0, S)\}$ be the centerline of the cylinder. For simplicity, let us assume that $\|\boldsymbol{\lambda}'(s)\| = 1$ such that the arc-length and the coordinate s coincide. Let $\boldsymbol{T}, \boldsymbol{N}, \boldsymbol{B}$ be the Frenet frame related to the curve. Let $\mathcal{D} = \{[r\cos\theta, r\sin\theta] : [0, \rho) \times [0, 2\pi)\}$ be a parametrization of the cross section. Let us also define the boundary of the cross section as $\partial\mathcal{D} = \{[\rho\sin\theta, \rho\cos\theta] : [0, 2\pi)\}$. Then, the cylinder Ω_w can be defined as follows

$$\Omega_w = \{\boldsymbol{\lambda}(s) + r\cos\theta\boldsymbol{N}(s) + r\sin\theta\boldsymbol{B}(s),\ r \in [0, \rho),\ s \in (0, S),\ \theta \in [0, 2\pi)\},$$

and the lateral boundary of it, denoted with Γ is,

$$\Gamma = \{\boldsymbol{\lambda}(s) + \rho\cos\theta\boldsymbol{N}(s) + \rho\sin\theta\boldsymbol{B}(s),\ s \in (0, S),\ \theta \in [0, 2\pi)\}.$$

We notice that Ω_w has *top* and *bottom* boundaries, which are $\partial\Omega_w \setminus \Gamma = \{\boldsymbol{\lambda}(0)+\mathcal{D}\} \cup \{\boldsymbol{\lambda}(S) + \mathcal{D}\}$. To model wells, without loss of generality, we assume that $\{\boldsymbol{\lambda}(0) + \mathcal{D}\}$ is the injection section of the well and for this reason it belongs to the external boundary, namely $\partial\Omega_p$. The other end point is the well tip, $\{\boldsymbol{\lambda}(S) + \mathcal{D}\}$ and it may be embedded into the reservoir.

We assume that the surrounding reservoir is described as an isotropic deformable porous medium filled with an isothermal single-phase fluid while we model the well by means of pressure-driven flow. More precisely, we assume that in the well, the pressure gradient is the main driving force for the flow motion. Concerning the reservoir deformable model, we assume that the material is subject to small deformations so that we can identify the initial spatial configuration with the current configuration of the system. This hypothesis implies that the variation of the material properties such as porosity and fluid density are small and can be evaluated as constants whenever required. In this framework the pressure and the displacement in the domain Ω is described by the following system of equations (namely the Biot model for small deformation):

$$-\nabla \cdot \sigma(\boldsymbol{u}_p) + \alpha \nabla p_p = \mathbf{f} \qquad \text{in } \Omega_p\,, \tag{1}$$

$$\partial_t \left(\frac{p_p}{M} + \alpha \nabla \cdot \boldsymbol{u}_p \right) - \nabla \cdot \boldsymbol{K}_p \nabla p_p = 0 \qquad \text{in } \Omega_p\,, \tag{2}$$

$$\partial_t p_w - \nabla \cdot \boldsymbol{K}_w \nabla p_w = 0 \qquad \text{in } \Omega_w\,, \tag{3}$$

where $\boldsymbol{u}_p$ is the solid matrix displacement vector, p_p and p_w are the variations of pore pressure from the hydrostatic load in Ω_p and Ω_w, respectively. We denote with $\boldsymbol{f}$ the gravity load, namely $(\rho_s - \rho_f)\mathbf{g}$ with ρ_s, ρ_f and $\mathbf{g}$ being the rock, the liquid density and the gravity force, respectively. The symbol ∂_t denotes the standard partial derivative with respect to time in the Eulerian framework while α, M, $\boldsymbol{K}_w$ and $\boldsymbol{K}_p$ are the Biot number the Biot modulus, the well and the reservoir permeability tensor, respectively. We also assume the linear elasticity behavior so that the stress tensor σ, appearing in (1), is defined by $\sigma(\boldsymbol{u}_p) := 2\mu\varepsilon(\boldsymbol{u}_p) + \lambda \nabla \cdot \boldsymbol{u}_p$, where μ and λ are the Lamé coefficients and $\varepsilon(\boldsymbol{u}_p)$ is the symmetric gradient of the displacement, defined as $\varepsilon(\boldsymbol{u}_p) := \frac{1}{2}(\nabla \boldsymbol{u}_p + \nabla \boldsymbol{u}_p^t)$. For further details on poromechanics, the interested reader is referred to e.g. [3]. Concerning the well model, we assume that the borehole is much more permeable to the fluid than the surrounding material. Such assumption implies that the dynamic of the pressure field in the well is faster than the evolution of the bulk pressure field, so that the term $\partial_t p_w$ can be neglected in the governing equation of the well. Finally, it is assumed that the interface Γ is permeable, namely it is crossed by a normal flux proportional to $K_\Gamma \left(p_p - p_w\right)$. The coefficient $K_\Gamma \geq 0$ denotes the permeability of the borehole lateral surface. For a well-posed problem we must also define appropriate boundary and initial conditions. For the boundary conditions we have prescribed that the pressure is fixed to values $p_{w,0}$, $p_{w,1}$ at the endpoints of the well. For the reservoir, we split the external boundary of Ω_p into complementary parts precisely $\Sigma_N \cup \Sigma_D = \partial\Omega_p \setminus \Gamma$ for the pressure and $\Sigma_{N_d} \cup \Sigma_{D_d} = \partial\Omega_p \setminus \Gamma$ for the displacement. On Σ_N we set Neuman boundary conditions for the bulk pressure. On Σ_D we set Dirichlet type condition, for a given bulk pressure value $p_{p,1}$. On Σ_{N_d} we set free normal stress conditions for the bulk displacement while on Σ_{D_d} we set vanishing Dirichlet type condition for the bulk displacement. Finally, concerning the initial condition, the following constraints are considered at $t = 0$: $\boldsymbol{u}_p = 0, \quad p_p = 0, \quad p_w = 0$ for any $\mathbf{x} \in \Omega(t = 0)$. As a result of these assumptions, we describe the interaction between the well and the reservoir by means of the following prototype problem,

$$\begin{cases} -\nabla\cdot\sigma(\boldsymbol{u}_p)+\alpha\nabla p_p=\mathbf{f} & \text{in } \Omega_p, \quad (4a)\\ \partial_t\left(\dfrac{p_p}{M}+\alpha\nabla\cdot\boldsymbol{u}_p\right)-\nabla\cdot\boldsymbol{K}_p\nabla p_p=0 & \text{in } \Omega_p, \quad (4b)\\ -\nabla\cdot\boldsymbol{K}_w\nabla p_w=0 & \text{in } \Omega_w, \quad (4c)\\ \boldsymbol{K}_p\nabla p_p\cdot\boldsymbol{n}_p=K_\Gamma\left(p_p-p_w\right) & \text{on } \Gamma, \quad (4d)\\ \boldsymbol{K}_w\nabla p_w\cdot\boldsymbol{n}_w=K_\Gamma\left(p_w-p_p\right) & \text{on } \Gamma, \quad (4e)\\ \nabla p_p\cdot\boldsymbol{n}_p=0 & \text{on } \Sigma_N, \quad (4f)\\ p_p=p_{p,1} & \text{on } \Sigma_D, \quad (4g)\\ \sigma\cdot\boldsymbol{n}_p=0 & \text{on } \Sigma_{N_d}, \quad (4h)\\ \boldsymbol{u}_p=0 & \text{on } \Sigma_{D_d}, \quad (4i)\\ p_w=p_{w,0} & \text{on } \{\boldsymbol{\lambda}(0)+\mathcal{D}\}, \quad (4j)\\ p_w=p_{w,1} & \text{on } \{\boldsymbol{\lambda}(S)+\mathcal{D}\}. \quad (4k) \end{cases}$$

Modeling a narrow borehole in three dimensions requires the resolution of the geometry, which in many real applications can be difficult to handle in the context of a reservoir model. Therefore we apply a topological model reduction, namely we go from a $3D$-$3D$ to a $3D$-$1D$ formulation following the approach proposed in [1].

2.2 Topological Model Reduction and Weak Formulation

The model reduction approach that we adopt is based on the following fundamental assumption. *The diameter of the well is small compared to the diameter of the reservoir* implying that the radius of the borehole $R=\rho/L$ is such that $0<R\ll 1$. As a consequence we also assume that the function p_w, together with the coefficients of the problem, have a uniform profile on each cross section $\mathcal{D}$, namely in cylindrical coordinates $u_w(r,s,\theta)=U(s)$. The permeability tensor in the borehole is isotropic, namely $\boldsymbol{K}_w=k_w\boldsymbol{I}$ and it is uniform on each cross section of the hole, that is $k_w(r,s,\theta)=k_w(s)$. The same restriction is enforced on the parameter $\boldsymbol{K}_\Gamma=\kappa\boldsymbol{I}$ on Γ, precisely $\kappa(\theta,s)=\kappa(s)$. Since the derivation of the reduced model is based on averaging we denote with $\overline{\overline{w}}$, $\overline{w}$ the following mean values respectively,

$$\overline{\overline{w}}(s)=(\pi R^2)^{-1}\int_{\mathcal{D}}wd\sigma\,,\quad \overline{w}(s)=(2\pi R)^{-1}\int_{\partial\mathcal{D}}wd\gamma\,,$$

and $d\omega=r\,d\theta\,dr\,ds$, $d\sigma=r\,d\theta\,dr$, $d\gamma=R\,d\theta$ represent volume, surface and curvilinear measures. We apply the averaging technique to Eq. (4c). Following the derivation in [1], to which we refer for details. From (4c) and (4e) we obtain the following averaged equation for the flow in the well

$$-\pi R^2 \partial_s (k_w \partial_s \overline{\overline{p}}_w) + 2\pi R\kappa(\overline{p}_w - \overline{p}_p) = 0 \quad \text{on } \Lambda\,. \tag{5}$$

Then, we introduce P_w, the one-dimensional approximation of the pressure in the well. As a results the weak form of Eq. (5) reads

$$\pi R^2 (k_w \partial_s P_w, \partial_s Q_w)_\Lambda + 2\pi R(\kappa P_w, Q_w)_\Lambda = 2\pi R(\kappa(\overline{p}_p - W), Q_w)_\Lambda \quad \forall Q_w \in H_0^1(\Lambda) \tag{6}$$

where $W = 1 - s$ denotes a suitable linear lifting of the Dirichlet boundary conditions of P_w on Λ. For the reservoir model we first extend the domain Ω_p to Ω. The coupling condition (4d) is enforced weakly after integrating by parts the term $\nabla \cdot \boldsymbol{K}_p \nabla p_p$ in Eq. (4b). Namely the weak formulation of the pressure problem in Ω reads

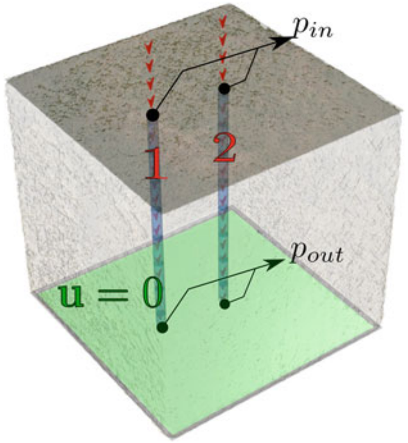
$$\frac{1}{M}(\partial_t p_p, q_p)_\Omega + \alpha(\nabla \cdot \partial_t \boldsymbol{u}, q_p)_\Omega + (\boldsymbol{k}_p \nabla p_p, \nabla q_p)_\Omega + (\kappa p_p, q_p)_\Gamma = (\kappa p_w, q_p)_\Gamma\,. \tag{7}$$

Recalling that the fluctuations of p_p and p_w on the cross section of Γ are small, see [1] for details, the last two terms of the previous equation become $2\pi R(\kappa \overline{p}_p, \overline{q}_p)_\Lambda$ and $2\pi R(\kappa P_w, \overline{q}_p)_\Lambda$. The problem of finding the displacement and the pressures $\boldsymbol{u}_p, p_p, P_w$ in Ω has been transformed into solving a $3D$ problem for $\boldsymbol{u}_p, p_p$ in Ω and a $1D$ problem for P_w in Λ. The weak formulation of the reduced 3D-1D model reads: for each $t \in (0, T]$, find $(\boldsymbol{u}_p, p_p, P_w)$ such that

$$\begin{cases} (2\mu\varepsilon(\boldsymbol{u}_p), \varepsilon(\boldsymbol{v}_p))_\Omega - \alpha(p_p, \nabla \cdot \boldsymbol{v}_p)_\Omega + \alpha(p_p, \boldsymbol{v} \cdot \mathbf{n}_p)_\Gamma = (\boldsymbol{f}, \boldsymbol{v})_\Omega & \mathbf{v} \in H^1_{\Sigma_{Dd}}(\Omega) \\ \dfrac{1}{M}(\partial_t p_p, q_p)_\Omega + \alpha(\nabla \cdot \partial_t \boldsymbol{u}, q_p)_\Omega + (\boldsymbol{k}_p \nabla p_p, \nabla q_p)_\Omega & \\ \qquad + 2\pi R(\kappa \overline{p}_p, \overline{q}_p)_\Lambda = 2\pi R(\kappa P_w, \overline{q}_p)_\Lambda\,, & \forall q_p \in H^1_{\Sigma_D}(\Omega) \\ \pi R^2 (k_w \partial_s P_w, \partial_s Q_w)_\Lambda & \\ \qquad + 2\pi R(\kappa P_w, Q_w)_\Lambda = 2\pi R(\kappa(\overline{p}_p - W), Q_w)_\Lambda, & \forall Q_w \in H_0^1(\Lambda)\,. \end{cases}$$

3 Numerical Experiments

We consider a cubic domain characterized by an edge of 1 km containing two vertical wells, with a radius of 1 m. A sketch of the domain is shown on the left part of Fig. 1 while on the right part the material properties are reported. The top surface



Young Modulus	E	10^{10}	Pa
Rock density	ρ_s	$2.2\,10^3$	kg/m^3
Water density	ρ_l	10^3	kg/m^3
Rock permeability	K_p	10^{-15}	m^2
Well permeability	K_w	10^{-12}	m^2
Borehole permeability	K_Γ	10^{-15}	m^2
Water viscosity	μ_l	10^{-3}	Pa s
Dimension	L	10^3	m
Radius	R	1	m
P_{in}	p_{w1}	0	MPa
P_{out}	p_{w2}	0.2	MPa

Fig. 1 Test 2: The geometry anld the boundary condition labels are shown on the left. The physical properties used in the simulation are reported on the right

of the basin is considered to be at the reference pressure ($p_p = 0$) while the bottom surface is fixed ($\boldsymbol{u}_p = 0$). Homogeneous Neuman conditions are considered in the remaining boundaries of the domain. Concerning the well problem, the boundary values of the pressure are fixed namely $p_{in} = 0$ and $p_{out} = 0.2$ MPa. Under this set up, we let the domain evolve towards a geostatic configuration namely the domain is subject to compaction due to its weight, and we investigate the evolution of the pressure and the displacement field for a temporal window $T \simeq 1day$ using a time step $\Delta t \simeq 1/10day$.

In Fig. 2 we show the pressure field together with the fluid flow vector field at $t_1 = \Delta t$ and $t_3 = T$ in the deformed domain. The pressure field in the bulk is mainly affected by two factors, the initial compaction and the interaction with the wells. The first factor is driven by the time derivative of the displacement field and it results in a time decreasing pressure field. Due to the imposed value of the pressure p_{in} and p_{out}, the wells act as source for the bulk pressure. The action of such source results in a logarithmic type singularity that produces a radially decreasing pressure field in the bulk. As shown in Fig. 2 reporting the pressure at times t_1 and t_3, the combination of these effects leads to a pressure field that increases with the depth in the first stages and decreases with the distance from the well as the steady state is reached.

In Fig. 3 the pressure field in the central slice of the bulk at t_1 and t_3 is shown. The contour lines of p_p put into evidence how the wells affect the pressure field in the bulk. More precisely, in Fig. 4 the pressure P_w in one well is compared at $t_1 = \Delta t$, $t_2 = T/2$ and $t_3 = T$ with the pressure $\overline{p}_p$ in the bulk. The difference of the two functions identifies regions where the net flow outgoing the well is positive (namely when $P_w > \overline{p}_p$) or negative (when $\overline{p}_p > P_w$). In the same plot we also report the pressure obtained in the uncoupled case (label nc), corresponding to $\boldsymbol{k}_\Gamma = 0$. We

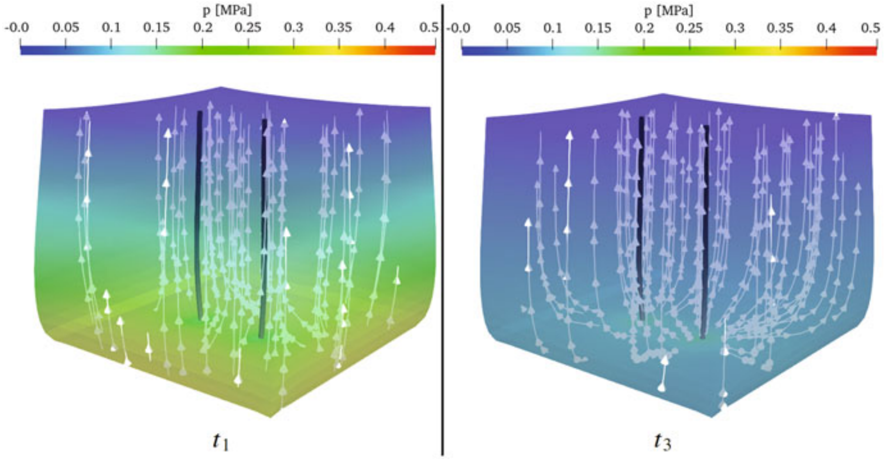


Fig. 2 The pressure field in the deformed domain at the initial (t_1) and the final time (t_3) is reported. Lines mark the direction of the flow in the porous medium. Deformations are amplified by a factor of 100 in order to be clearly visible

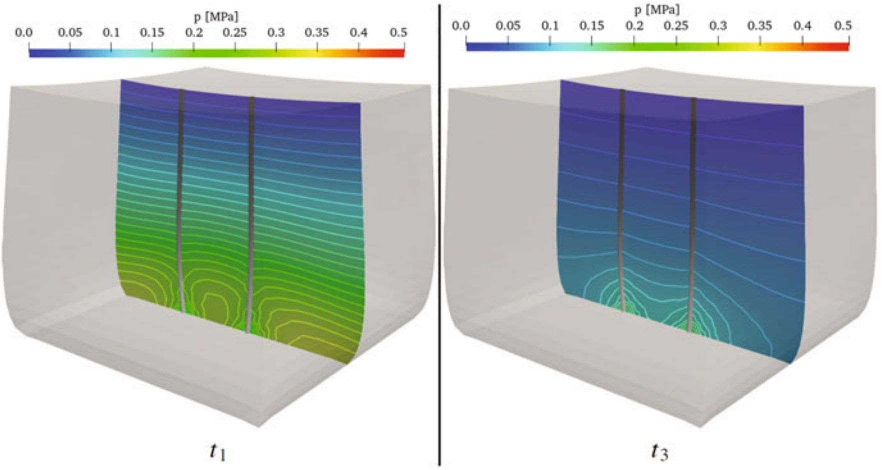


Fig. 3 The pressure field in the central slice of the domain at different at the initial (t_1) and the final time (t_3) is shown. Thick lines mark the contour of the bulk pressure, p_p

notice that in the first stages (curve t_1) the mechanical compaction dominates the bulk pressure variation. For this reason, we observe that also the pressure in the well increases at the bottom of the well, leading to flow reversal in the well. Moreover, in this region the scenario $\overline{p}_p > P_w$ takes place and the well partially acts as a sink removing fluid from the porous system. As the displacement becomes stationary and p_p reaches an equilibrium (curve t_2 , t_3) the flow in the well recovers the expected

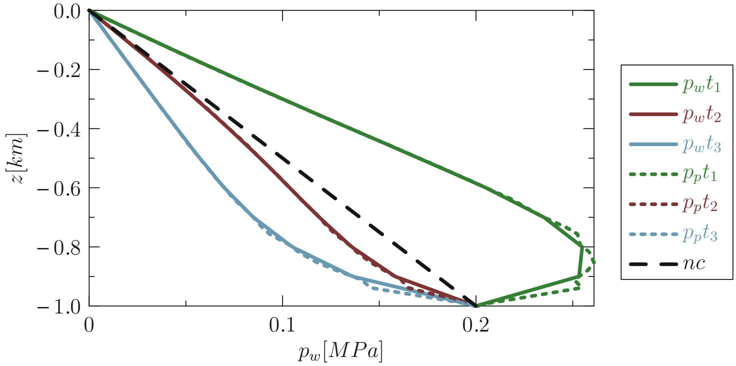


Fig. 4 The pressure field in the well is plotted at different times. The dashed line marks the pressure field obtained without taking into account the pressure coupling terms

unidirectional orientation and the pressure P_w slightly overcomes p_p, such that the fluid leaks form the well into the bulk.

In conclusion, the model confirms that the temporal dynamics of the well is faster than dynamics of the basin. Because of this effect, the variation of bulk pressure induced by mechanical compaction could lead to large and complex spatial and temporal patterns in the well pressure field.

References

1. Daniele Cerroni, Federica Laurino, and Paolo Zunino. Mathematical analysis, finite element approximation and numerical solvers for the interaction of 3d reservoirs with 1d wells. *GEM-International Journal on Geomathematics*, 10(4):1–28, 2019.
2. Daniele Cerroni, Mattia Penati, Giovanni Porta, Edie Miglio, Paolo Zunino, and Paolo Ruffo. Multiscale modeling of glacial loading by a 3d thermo-hydro-mechanical approach including erosion and isostasy. *Geosciences*, 9(11), 2019.
3. Olivier Coussy. *Poromechanics*. John Wiley & Sons, 2004.
4. C. D'Angelo and A. Quarteroni. On the coupling of 1d and 3d diffusion-reaction equations. application to tissue perfusion problems. *Mathematical Models and Methods in Applied Sciences*, 18(8):1481–1504, 2008.
5. Federica Laurino and Paolo Zunino. Derivation and analysis of coupled pdes on manifolds with high dimensionality gap arising from topological model reduction. *ESAIM: M2AN*, 53(6):2047–2080, 2019.
6. D.W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation. *Soc Pet Eng AIME J*, 18(3):183–194, 1978.
7. D.W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation with nonsquare grid blocks and anisotropic permeability. *Society of Petroleum Engineers journal*, 23(3):531–543, 1983.
8. D.W. Peaceman. Interpretation of well-block pressures in numerical reservoir simulation - part 3: Some additional well geometries. volume Sigma, pages 457–471, 1987.

Editorial Policy

1. Volumes in the following three categories will be published in LNCSE:

i) Research monographs
ii) Tutorials
iii) Conference proceedings

Those considering a book which might be suitable for the series are strongly advised to contact the publisher or the series editors at an early stage.

2. Categories i) and ii). Tutorials are lecture notes typically arising via summer schools or similar events, which are used to teach graduate students. These categories will be emphasized by Lecture Notes in Computational Science and Engineering. **Submissions by interdisciplinary teams of authors are encouraged.** The goal is to report new developments – quickly, informally, and in a way that will make them accessible to non-specialists. In the evaluation of submissions timeliness of the work is an important criterion. Texts should be well-rounded, well-written and reasonably self-contained. In most cases the work will contain results of others as well as those of the author(s). In each case the author(s) should provide sufficient motivation, examples, and applications. In this respect, Ph.D. theses will usually be deemed unsuitable for the Lecture Notes series. Proposals for volumes in these categories should be submitted either to one of the series editors or to Springer-Verlag, Heidelberg, and will be refereed. A provisional judgement on the acceptability of a project can be based on partial information about the work: a detailed outline describing the contents of each chapter, the estimated length, a bibliography, and one or two sample chapters – or a first draft. A final decision whether to accept will rest on an evaluation of the completed work which should include

- at least 100 pages of text;
- a table of contents;
- an informative introduction perhaps with some historical remarks which should be accessible to readers unfamiliar with the topic treated;
- a subject index.

3. Category iii). Conference proceedings will be considered for publication provided that they are both of exceptional interest and devoted to a single topic. One (or more) expert participants will act as the scientific editor(s) of the volume. They select the papers which are suitable for inclusion and have them individually refereed as for a journal. Papers not closely related to the central topic are to be excluded. Organizers should contact the Editor for CSE at Springer at the planning stage, see *Addresses* below.

In exceptional cases some other multi-author-volumes may be considered in this category.

4. Only works in English will be considered. For evaluation purposes, manuscripts may be submitted in print or electronic form, in the latter case, preferably as pdf- or zipped ps-files. Authors are requested to use the LaTeX style files available from Springer at http://www.springer.com/gp/authors-editors/book-authors-editors/manuscript-preparation/5636 (Click on LaTeX Template → monographs or contributed books).

For categories ii) and iii) we strongly recommend that all contributions in a volume be written in the same LaTeX version, preferably LaTeX2e. Electronic material can be included if appropriate. Please contact the publisher.

Careful preparation of the manuscripts will help keep production time short besides ensuring satisfactory appearance of the finished book in print and online.

5. The following terms and conditions hold. Categories i), ii) and iii):

Authors receive 50 free copies of their book. No royalty is paid.
Volume editors receive a total of 50 free copies of their volume to be shared with authors, but no royalties.

Authors and volume editors are entitled to a discount of 40 % on the price of Springer books purchased for their personal use, if ordering directly from Springer.

6. Springer secures the copyright for each volume.

Addresses:

Lecture Notes in Computational Science and Engineering

1. D. Funaro, *Spectral Elements for Transport-Dominated Equations.*

2. H.P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming.

3. W. Hackbusch, G. Wittum (eds.), *Multigrid Methods V.*

4. P. Deuflhard, J. Hermans, B. Leimkuhler, A.E. Mark, S. Reich, R.D. Skeel (eds.), *Computational Molecular Dynamics: Challenges, Methods, Ideas.*

5. D. Kröner, M. Ohlberger, C. Rohde (eds.), *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws.*

6. S. Turek, *Efficient Solvers for Incompressible Flow Problems.* An Algorithmic and Computational Approach.

7. R. von Schwerin, ***M**ulti **B**ody **S**ystem **SIM**ulation.* Numerical Methods, Algorithms, and Software.

8. H.-J. Bungartz, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.*

9. T.J. Barth, H. Deconinck (eds.), *High-Order Methods for Computational Physics.*

10. H.P. Langtangen, A.M. Bruaset, E. Quak (eds.), *Advances in Software Tools for Scientific Computing.*

11. B. Cockburn, G.E. Karniadakis, C.-W. Shu (eds.), *Discontinuous Galerkin Methods.* Theory, Computation and Applications.

12. U. van Rienen, *Numerical Methods in Computational Electrodynamics.* Linear Systems in Practical Applications.

13. B. Engquist, L. Johnsson, M. Hammill, F. Short (eds.), *Simulation and Visualization on the Grid.*

14. E. Dick, K. Riemslagh, J. Vierendeels (eds.), *Multigrid Methods VI.*

15. A. Frommer, T. Lippert, B. Medeke, K. Schilling (eds.), *Numerical Challenges in Lattice Quantum Chromodynamics.*

16. J. Lang, *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems.* Theory, Algorithm, and Applications.

17. B.I. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition.*

18. U. van Rienen, M. Günther, D. Hecht (eds.), *Scientific Computing in Electrical Engineering.*

19. I. Babuška, P.G. Ciarlet, T. Miyoshi (eds.), *Mathematical Modeling and Numerical Simulation in Continuum Mechanics.*

20. T.J. Barth, T. Chan, R. Haimes (eds.), *Multiscale and Multiresolution Methods.* Theory and Applications.

21. M. Breuer, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.*

22. K. Urban, *Wavelets in Numerical Simulation.* Problem Adapted Construction and Applications.

23. L.F. Pavarino, A. Toselli (eds.), *Recent Developments in Domain Decomposition Methods.*

24. T. Schlick, H.H. Gan (eds.), *Computational Methods for Macromolecules: Challenges and Applications.*

25. T.J. Barth, H. Deconinck (eds.), *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics*.

26. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations*.

27. S. Müller, *Adaptive Multiscale Schemes for Conservation Laws*.

28. C. Carstensen, S. Funken, W. Hackbusch, R.H.W. Hoppe, P. Monk (eds.), *Computational Electromagnetics*.

29. M.A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations*.

30. T. Biegler, O. Ghattas, M. Heinkenschloss, B. van Bloemen Waanders (eds.), *Large-Scale PDE-Constrained Optimization*.

31. M. Ainsworth, P. Davies, D. Duncan, P. Martin, B. Rynne (eds.), *Topics in Computational Wave Propagation*. Direct and Inverse Problems.

32. H. Emmerich, B. Nestler, M. Schreckenberg (eds.), *Interface and Transport Dynamics.* Computational Modelling.

33. H.P. Langtangen, A. Tveito (eds.), *Advanced Topics in Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming.

34. V. John, *Large Eddy Simulation of Turbulent Incompressible Flows.* Analytical and Numerical Results for a Class of LES Models.

35. E. Bänsch (ed.), *Challenges in Scientific Computing - CISC 2002.*

36. B.N. Khoromskij, G. Wittum, *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface.*

37. A. Iske, *Multiresolution Methods in Scattered Data Modelling.*

38. S.-I. Niculescu, K. Gu (eds.), *Advances in Time-Delay Systems.*

39. S. Attinger, P. Koumoutsakos (eds.), *Multiscale Modelling and Simulation.*

40. R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Wildlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering.*

41. T. Plewa, T. Linde, V.G. Weirs (eds.), *Adaptive Mesh Refinement – Theory and Applications.*

42. A. Schmidt, K.G. Siebert, *Design of Adaptive Finite Element Software.* The Finite Element Toolbox ALBERTA.

43. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations II.*

44. B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Methods in Science and Engineering.*

45. P. Benner, V. Mehrmann, D.C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems.*

46. D. Kressner, *Numerical Methods for General and Structured Eigenvalue Problems.*

47. A. Boriçi, A. Frommer, B. Joó, A. Kennedy, B. Pendleton (eds.), *QCD and Numerical Analysis III.*

48. F. Graziani (ed.), *Computational Methods in Transport.*

49. B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, R. Skeel (eds.), *New Algorithms for Macromolecular Simulation.*

50. M. Bücker, G. Corliss, P. Hovland, U. Naumann, B. Norris (eds.), *Automatic Differentiation: Applications, Theory, and Implementations.*

51. A.M. Bruaset, A. Tveito (eds.), *Numerical Solution of Partial Differential Equations on Parallel Computers.*

52. K.H. Hoffmann, A. Meyer (eds.), *Parallel Algorithms and Cluster Computing.*

53. H.-J. Bungartz, M. Schäfer (eds.), *Fluid-Structure Interaction.*

54. J. Behrens, *Adaptive Atmospheric Modeling.*

55. O. Widlund, D. Keyes (eds.), *Domain Decomposition Methods in Science and Engineering XVI.*

56. S. Kassinos, C. Langer, G. Iaccarino, P. Moin (eds.), *Complex Effects in Large Eddy Simulations.*

57. M. Griebel, M.A Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations III.*

58. A.N. Gorban, B. Kégl, D.C. Wunsch, A. Zinovyev (eds.), *Principal Manifolds for Data Visualization and Dimension Reduction.*

59. H. Ammari (ed.), *Modeling and Computations in Electromagnetics: A Volume Dedicated to Jean-Claude Nédélec.*

60. U. Langer, M. Discacciati, D. Keyes, O. Widlund, W. Zulehner (eds.), *Domain Decomposition Methods in Science and Engineering XVII.*

61. T. Mathew, *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations.*

62. F. Graziani (ed.), *Computational Methods in Transport: Verification and Validation.*

63. M. Bebendorf, *Hierarchical Matrices.* A Means to Efficiently Solve Elliptic Boundary Value Problems.

64. C.H. Bischof, H.M. Bücker, P. Hovland, U. Naumann, J. Utke (eds.), *Advances in Automatic Differentiation.*

65. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations IV.*

66. B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Modeling and Simulation in Science.*

67. I.H. Tuncer, Ü. Gülcat, D.R. Emerson, K. Matsuno (eds.), *Parallel Computational Fluid Dynamics 2007.*

68. S. Yip, T. Diaz de la Rubia (eds.), *Scientific Modeling and Simulations*.

69. A. Hegarty, N. Kopteva, E. O'Riordan, M. Stynes (eds.), *BAIL* 2008 – *Boundary and Interior Layers.*

70. M. Bercovier, M.J. Gander, R. Kornhuber, O. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XVIII.*

71. B. Koren, C. Vuik (eds.), *Advanced Computational Methods in Science and Engineering.*

72. M. Peters (ed.), *Computational Fluid Dynamics for Sport Simulation.*

73. H.-J. Bungartz, M. Mehl, M. Schäfer (eds.), *Fluid Structure Interaction II - Modelling, Simulation, Optimization.*

74. D. Tromeur-Dervout, G. Brenner, D.R. Emerson, J. Erhel (eds.), *Parallel Computational Fluid Dynamics 2008.*

75. A.N. Gorban, D. Roose (eds.), *Coping with Complexity: Model Reduction and Data Analysis.*

76. J.S. Hesthaven, E.M. Rønquist (eds.), *Spectral and High Order Methods for Partial Differential Equations.*

77. M. Holtz, *Sparse Grid Quadrature in High Dimensions with Applications in Finance and Insurance*.

78. Y. Huang, R. Kornhuber, O.Widlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering XIX*.

79. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations V*.

80. P.H. Lauritzen, C. Jablonowski, M.A. Taylor, R.D. Nair (eds.), *Numerical Techniques for Global Atmospheric Models*.

81. C. Clavero, J.L. Gracia, F.J. Lisbona (eds.), *BAIL 2010 – Boundary and Interior Layers, Computational and Asymptotic Methods*.

82. B. Engquist, O. Runborg, Y.R. Tsai (eds.), *Numerical Analysis and Multiscale Computations*.

83. I.G. Graham, T.Y. Hou, O. Lakkis, R. Scheichl (eds.), *Numerical Analysis of Multiscale Problems*.

84. A. Logg, K.-A. Mardal, G. Wells (eds.), *Automated Solution of Differential Equations by the Finite Element Method*.

85. J. Blowey, M. Jensen (eds.), *Frontiers in Numerical Analysis - Durham 2010*.

86. O. Kolditz, U.-J. Gorke, H. Shao, W. Wang (eds.), *Thermo-Hydro-Mechanical-Chemical Processes in Fractured Porous Media - Benchmarks and Examples*.

87. S. Forth, P. Hovland, E. Phipps, J. Utke, A. Walther (eds.), *Recent Advances in Algorithmic Differentiation*.

88. J. Garcke, M. Griebel (eds.), *Sparse Grids and Applications*.

89. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VI*.

90. C. Pechstein, *Finite and Boundary Element Tearing and Interconnecting Solvers for Multiscale Problems*.

91. R. Bank, M. Holst, O. Widlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering XX*.

92. H. Bijl, D. Lucor, S. Mishra, C. Schwab (eds.), *Uncertainty Quantification in Computational Fluid Dynamics*.

93. M. Bader, H.-J. Bungartz, T. Weinzierl (eds.), *Advanced Computing*.

94. M. Ehrhardt, T. Koprucki (eds.), *Advanced Mathematical Models and Numerical Techniques for Multi-Band Effective Mass Approximations*.

95. M. Azaïez, H. El Fekih, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2012*.

96. F. Graziani, M.P. Desjarlais, R. Redmer, S.B. Trickey (eds.), *Frontiers and Challenges in Warm Dense Matter*.

97. J. Garcke, D. Pflüger (eds.), *Sparse Grids and Applications – Munich 2012*.

98. J. Erhel, M. Gander, L. Halpern, G. Pichot, T. Sassi, O. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XXI*.

99. R. Abgrall, H. Beaugendre, P.M. Congedo, C. Dobrzynski, V. Perrier, M. Ricchiuto (eds.), *High Order Nonlinear Numerical Methods for Evolutionary PDEs - HONOM 2013*.

100. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VII*.

101. R. Hoppe (ed.), *Optimization with PDE Constraints - OPTPDE 2014*.

102. S. Dahlke, W. Dahmen, M. Griebel, W. Hackbusch, K. Ritter, R. Schneider, C. Schwab, H. Yserentant (eds.), *Extraction of Quantifiable Information from Complex Systems*.

103. A. Abdulle, S. Deparis, D. Kressner, F. Nobile, M. Picasso (eds.), *Numerical Mathematics and Advanced Applications - ENUMATH 2013*.

104. T. Dickopf, M.J. Gander, L. Halpern, R. Krause, L.F. Pavarino (eds.), *Domain Decomposition Methods in Science and Engineering XXII*.

105. M. Mehl, M. Bischoff, M. Schäfer (eds.), *Recent Trends in Computational Engineering - CE2014*. Optimization, Uncertainty, Parallel Algorithms, Coupled and Complex Problems.

106. R.M. Kirby, M. Berzins, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM'14*.

107. B. Jüttler, B. Simeon (eds.), *Isogeometric Analysis and Applications 2014*.

108. P. Knobloch (ed.), *Boundary and Interior Layers, Computational and Asymptotic Methods – BAIL 2014*.

109. J. Garcke, D. Pflüger (eds.), *Sparse Grids and Applications – Stuttgart 2014*.

110. H. P. Langtangen, *Finite Difference Computing with Exponential Decay Models*.

111. A. Tveito, G.T. Lines, *Computing Characterizations of Drugs for Ion Channels and Receptors Using Markov Models*.

112. B. Karazösen, M. Manguoğlu, M. Tezer-Sezgin, S. Göktepe, Ö. Uğur (eds.), *Numerical Mathematics and Advanced Applications - ENUMATH 2015*.

113. H.-J. Bungartz, P. Neumann, W.E. Nagel (eds.), *Software for Exascale Computing - SPPEXA 2013-2015*.

114. G.R. Barrenechea, F. Brezzi, A. Cangiani, E.H. Georgoulis (eds.), *Building Bridges: Connections and Challenges in Modern Approaches to Numerical Partial Differential Equations*.

115. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations VIII*.

116. C.-O. Lee, X.-C. Cai, D.E. Keyes, H.H. Kim, A. Klawonn, E.-J. Park, O.B. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XXIII*.

117. T. Sakurai, S.-L. Zhang, T. Imamura, Y. Yamamoto, Y. Kuramashi, T. Hoshi (eds.), *Eigenvalue Problems: Algorithms, Software and Applications in Petascale Computing*. EPASA 2015, Tsukuba, Japan, September 2015.

118. T. Richter (ed.), *Fluid-structure Interactions*. Models, Analysis and Finite Elements.

119. M.L. Bittencourt, N.A. Dumont, J.S. Hesthaven (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2016.* Selected Papers from the ICOSAHOM Conference, June 27-July 1, 2016, Rio de Janeiro, Brazil.

120. Z. Huang, M. Stynes, Z. Zhang (eds.), *Boundary and Interior Layers, Computational and Asymptotic Methods BAIL 2016.*

121. S.P.A. Bordas, E.N. Burman, M.G. Larson, M.A. Olshanskii (eds.), *Geometrically Unfitted Finite Element Methods and Applications*. Proceedings of the UCL Workshop 2016.

122. A. Gerisch, R. Penta, J. Lang (eds.), *Multiscale Models in Mechano and Tumor Biology*. Modeling, Homogenization, and Applications.

123. J. Garcke, D. Pflüger, C.G. Webster, G. Zhang (eds.), *Sparse Grids and Applications - Miami 2016*.

124. M. Schäfer, M. Behr, M. Mehl, B. Wohlmuth (eds.), *Recent Advances in Computational Engineering*. Proceedings of the 4th International Conference on Computational Engineering (ICCE 2017) in Darmstadt.

125. P.E. Bjørstad, S.C. Brenner, L. Halpern, R. Kornhuber, H.H. Kim, T. Rahman, O.B. Widlund (eds.), *Domain Decomposition Methods in Science and Engineering XXIV*. 24th International Conference on Domain Decomposition Methods, Svalbard, Norway, February 6–10, 2017.

126. F.A. Radu, K. Kumar, I. Berre, J.M. Nordbotten, I.S. Pop (eds.), *Numerical Mathematics and Advanced Applications – ENUMATH 2017*.

127. X. Roca, A. Loseille (eds.), *27th International Meshing Roundtable*.

128. Th. Apel, U. Langer, A. Meyer, O. Steinbach (eds.), *Advanced Finite Element Methods with Applications*. Selected Papers from the 30th Chemnitz Finite Element Symposium 2017.

129. M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differencial Equations IX*.

130. S. Weißer, BEM-based Finite Element *Approaches on Polytopal Meshes*.

131. V.A. Garanzha, L. Kamenski, H. Si (eds.), *Numerical Geometry, Grid Generation and Scientific Computing*. Proceedings of the 9th International Conference, NUMGRID2018/Voronoi 150, Celebrating the 150th Anniversary of G. F. Voronoi, Moscow, Russia, December 2018.

132. H. van Brummelen, A. Corsini, S. Perotto, G. Rozza (eds.), *Numerical Methods for Flows*.

133. —–

134. S.J. Sherwin, D. Moxey, J. Peiro, P.E. Vincent, C. Schwab (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2018*.

135. G.R. Barrenechea, J. Mackenzie (eds.), *Boundary and Interior Layers, Computational and Asymptotic Methods BAIL 2018*.

136. H.-J. Bungartz, S. Reiz, B. Uekermann, P. Neumann, W.E. Nagel (eds.), *Software for Exascale Computing - SPPEXA 2016–2019*.

137. M. D'Elia, M. Gunzburger, G. Rozza (eds.), *Quantification of Uncertainty: Improving Efficiency and Technology*.

138. —–

139. F.J. Vermolen, C. Vuik (eds.), *Numerical Mathematics and Advanced Applications ENUMATH 2019*.

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/3527

Monographs in Computational Science and Engineering

1. J. Sundnes, G.T. Lines, X. Cai, B.F. Nielsen, K.-A. Mardal, A. Tveito, *Computing the Electrical Activity in the Heart.*

For further information on this book, please have a look at our mathematics catalogue at the following URL: www.springer.com/series/7417

Texts in Computational Science and Engineering

1. H. P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 2nd Edition
2. A. Quarteroni, F. Saleri, P. Gervasio, *Scientific Computing with MATLAB and Octave.* 4th Edition
3. H. P. Langtangen, *Python Scripting for Computational Science*. 3rd Edition
4. H. Gardner, G. Manduchi, *Design Patterns for e-Science*.
5. M. Griebel, S. Knapek, G. Zumbusch, *Numerical Simulation in Molecular Dynamics*.
6. H. P. Langtangen, *A Primer on Scientific Programming with Python.* 5th Edition
7. A. Tveito, H. P. Langtangen, B. F. Nielsen, X. Cai, *Elements of Scientific Computing.*
8. B. Gustafsson, *Fundamentals of Scientific Computing.*
9. M. Bader, *Space-Filling Curves.*
10. M. Larson, F. Bengzon, *The Finite Element Method: Theory, Implementation and Applications.*
11. W. Gander, M. Gander, F. Kwok, *Scientific Computing: An Introduction using Maple and MATLAB.*
12. P. Deuflhard, S. Röblitz, *A Guide to Numerical Modelling in Systems Biology*.
13. M. H. Holmes, *Introduction to Scientific Computing and Data Analysis*.
14. S. Linge, H. P. Langtangen, *Programming for Computations* - A Gentle Introduction to Numerical Simulations with MATLAB/Octave.
15. S. Linge, H. P. Langtangen, *Programming for Computations* - A Gentle Introduction to Numerical Simulations with Python.
16. H.P. Langtangen, S. Linge, *Finite Difference Computing with PDEs* - A Modern Software Approach.
17. B. Gustafsson, *Scientific Computing from a Historical Perspective*.
18. J. A. Trangenstein, *Scientific Computing*. Volume I - Linear and Nonlinear Equations.

19. J. A. Trangenstein, *Scientific Computing*. Volume II - Eigenvalues and Optimization.

20. J. A. Trangenstein, *Scientific Computing*. Volume III - Approximation and Integration.

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/5151