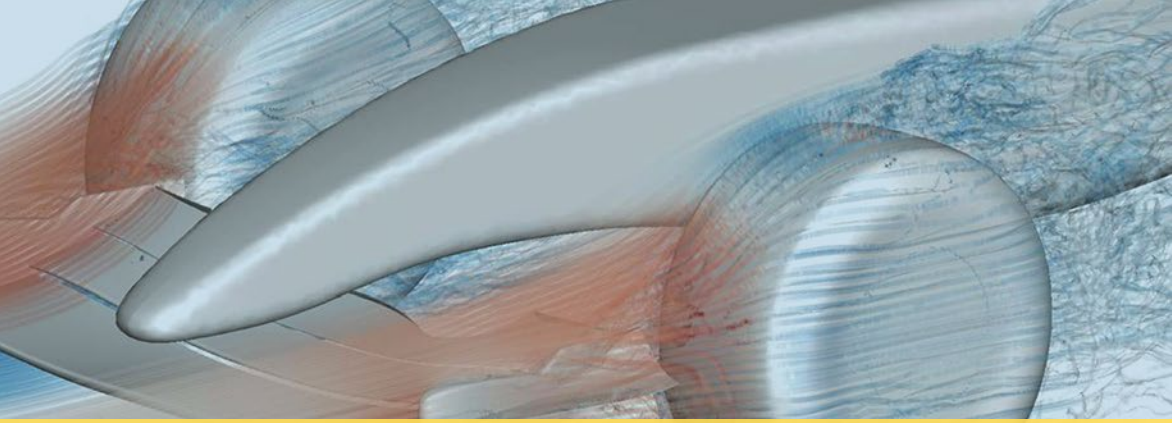


LECTURE NOTES IN COMPUTATIONAL SCIENCE AND ENGINEERING 106

Robert M. Kirby · Martin Berzins
Jan S. Hesthaven *Editors*

Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014

Springer

Lecture Notes in Computational Science and Engineering 106

Editors:

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

Robert M. Kirby • Martin Berzins • Jan S. Hesthaven
Editors

Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014

Selected papers from the ICOSAHOM conference, June 23-27, 2014, Salt Lake City, Utah, USA

Springer

Editors
Robert M. Kirby
School of Computing
University of Utah
Salt Lake City, Utah
USA

Martin Berzins
School of Computing
University of Utah
Salt Lake City, Utah
USA

Jan S. Hesthaven
EPFL-SB-MATHICSE-MCSS
Ecole Polytechnique Fédérale de Lausanne
Lausanne, Switzerland

ISSN 1439-7358 ISSN 2197-7100 (electronic)
Lecture Notes in Computational Science and Engineering
ISBN 978-3-319-19799-9 ISBN 978-3-319-19800-2 (eBook)
DOI 10.1007/978-3-319-19800-2

Library of Congress Control Number: 2015956608

Mathematics Subject Classification (2010): Primary: 65M70; 65N35; 65N30; 74S25; 76M10; 76M22; 78M10; 78M22
Secondary: 33Cxx; 41Axx; 65Cxx; 65Dxx; 65Lxx; 65Mxx; 65Nxx; 74Jxx; 76Mxx; 78Mxx; 80Mxx76

Springer Cham Heidelberg New York Dordrecht London

Cover image: Streamlines of velocity, coloured by pressure, as flow evolves over the front wing of a Formula 1 car at Re = 50;000. The flow solution was obtained using Nektar++ (courtesy of Prof. Spencer Sherwin, Imperial College London).

Printed on acid-free paper

Springer International Publishing AG Switzerland is part of Springer Science+Business Media (www.springer.com)

Preface

This volume presents selected papers from the tenth International Conference on Spectral and High-Order Methods (ICOSAHOM'14) that was held in Salt Lake City, UT, USA during the week of June 23–27, 2014. These selected papers were refereed by members of the scientific committee of ICOSAHOM as well as by other leading scientists.

The first ICOSAHOM conference was held in Como, Italy, in 1989 and marked the beginning of an international conference series in Montpelier, France (1992); Houston, TX, USA (1995); Tel Aviv, Israel (1998); Uppsala, Sweden (2001); Providence, RI, USA (2004); Beijing, China (2007); Trondheim, Norway (2009); and Gammarth, Tunisia (2012).

ICOSAHOM has established itself as the main meeting place for researchers with interests in the theoretical, applied, and computational aspects of high-order methods for the numerical solution of partial differential equations.

With over 300 participants, ICOSAHOM'14 was the largest conference devoted to high-order methods to date. The program consisted of eight invited lectures spread out through the week, 19 mini-symposia hosting approximately 192 talks, and 80 contributed talks.

The content of this proceedings is organized as follows. First, contributions from the invited speakers are included, listed in alphabetical order according to the name of the invited speaker. The remainder of the volume consists of refereed selected papers highlighting the broad spectrum of topics presented at ICOSAHOM'14.

The success of the meeting was ensured through the generous financial support given by the US National Science Foundation, the US Office of Naval Research (under the guidance of Dr. Reza Malek-Madani), the US Air Force Office of Sponsored Research (under the guidance of Dr. Fariba Fahroo), the US Army Research Office (under the guidance of Dr. Mike Coyle and Dr. Joe Myers), and the SCI Institute at the University of Utah.

Special thanks goes to our local organizing committee Yekaterina Epshteyn, Anne Gelb, Rodrigo Platte, Rosie Renaut, and Dongbin Xiu. They did an amazing job organizing and executing the event. Individual thanks also goes to Mrs. Deb

Zemek and Mr. Nathan Galli. Deb and Nathan were the ‘on the ground’ individuals who kept everything moving smoothly.

Salt Lake City, UT, USA — Robert M. Kirby
Salt Lake City, UT, USA — Martin Berzins
Lausanne, Switzerland — Jan S. Hesthaven

Contents

Part I
Invited Papers

C^0 Interior Penalty Galerkin Method for Biharmonic Eigenvalue Problems

Susanne C. Brenner, Peter Monk, and Jiguang Sun

Abstract We consider the C^0 interior penalty Galerkin method for biharmonic eigenvalue problems with the boundary conditions of the clamped plate, the simply supported plate and the Cahn-Hilliard type. We establish the convergence of the method and present numerical results to illustrate its performance. We also compare it with the Argyris C^1 finite element method, the Ciarlet-Raviart mixed finite element method, and the Morley nonconforming finite element method.

1 Introduction

We consider the numerical solution of several eigenvalue problems for the biharmonic operator by the C^0 interior penalty Galerkin (C^0 IPG) method. These eigenvalue problems appear for example in mechanics (vibration and buckling of plates).

The C^0 IPG method, developed in the last decade [8, 16], is a discontinuous Galerkin method for fourth order problems based on standard continuous finite element spaces for second order elliptic problems. The lowest order methods in this approach are almost as simple as classical nonconforming finite element methods [1, 21] and are much simpler than finite element methods using globally

S.C. Brenner (✉)
Department of Mathematics and Center for Computation & Technology, Louisiana State University, Baton Rouge, LA 70803, USA
e-mail: brenner@math.lsu.edu

P. Monk
Department of Mathematical Sciences, University of Delaware, Newark, DE 19716, USA
e-mail: monk@math.udel.edu

J. Sun
Department of Mathematical Sciences, Michigan Technological University, Houghton, MI 49931, USA
e-mail: jiguangs@mtu.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_1

C^1 functions [2, 23]. Unlike classical nonconforming finite element methods, higher order finite elements can be used in this approach to capture smooth solutions efficiently. Furthermore, the C^0 IPG method converges for the biharmonic source problem with boundary conditions of the clamped plate, the simply supported plate and the Cahn-Hilliard type that appears in mathematical models for phase separation phenomena [14]. It also preserves the symmetric positive-definiteness of the continuous problems. This last property is very attractive for eigenvalue problems since it means that the convergence for the eigenvalue problem can be derived from the convergence for the source problem through the classical spectral approximation theory. In contrast, the convergence of a mixed finite element method for the source problem does not necessarily lead to the convergence of the method for the eigenvalue problem unless the mixed method is chosen carefully [7].

In this paper we extend the C^0 IPG method to biharmonic eigenvalue problems (cf. Sect. 2). We show that the method converges for all three types of boundary conditions (cf. Sect. 3), and we present numerical results that validate the theory (cf. Sect. 4). We also compare the performance of the C^0 IPG method, the Argyris C^1 finite element method, the Ciarlet-Raviart mixed finite element method [15], and the Morley nonconforming finite element method (cf. Sect. 5). We end the paper with some concluding remarks in Sect. 6.

We note that numerical results for a related C^0 discontinuous Galerkin method were presented in [24] for the plate vibration and buckling problems on a square with the boundary conditions of simply supported plates. However the convergence of the method for the eigenvalue problem was not addressed in that paper.

Throughout the paper we will use C to denote a generic positive constant that is independent of the mesh size h.

2 Biharmonic Eigenvalue Problems

Let Ω denote a bounded polygonal domain in $\mathbb{R}^2$ with boundary $\partial\Omega$, and let n denote the unit outward normal. We consider biharmonic eigenvalue problems for plate vibration and plate buckling with three types of boundary conditions:

Clamped Plate (CP)

$$u = \frac{\partial u}{\partial n} = 0 \qquad \text{on } \partial\Omega \tag{1}$$

Simply Supported Plate (SSP)

$$u = \triangle u = 0 \qquad \text{on } \partial\Omega \tag{2}$$

Cahn-Hilliard Type (CH)

$$\frac{\partial u}{\partial n} = \frac{\partial \triangle u}{\partial n} = 0 \qquad \text{on } \partial\Omega \tag{3}$$

Let the bilinear forms $a(\cdot,\cdot)$ and $b(\cdot,\cdot)$ be defined by

$$a(u,v) = \int_\Omega D^2u : D^2v \, dx, \tag{4}$$

where $D^2u : D^2v = \sum_{i,j=1}^{2} u_{x_ix_j} v_{x_ix_j}$ is the Frobenius inner product of the Hessian matrices of u and v, and

$$b(u,v) = \begin{cases} (u,v) = \displaystyle\int_\Omega uv \, dx & \text{for plate vibration,} \\ (\nabla u, \nabla v) = \displaystyle\int_\Omega \nabla u \cdot \nabla v \, dx & \text{for plate buckling.} \end{cases} \tag{5}$$

The weak formulation of the biharmonic eigenvalue problem is to seek $(u,\lambda) \in V \times \mathbb{R}$ such that $u \neq 0$ and

$$a(u,v) = \lambda b(u,v) \qquad \forall\, v \in V, \tag{6}$$

where

$$V = H_0^2(\Omega) \qquad \text{for } \mathbf{CP}, \tag{7}$$

$$V = H^2(\Omega) \cap H_0^1(\Omega) \qquad \text{for } \mathbf{SSP}, \tag{8}$$

$$V = \{v \in H^2(\Omega) : \partial v/\partial n = 0 \text{ on } \partial\Omega \text{ and } (v,1) = 0\} \qquad \text{for } \mathbf{CH}. \tag{9}$$

Remark 1 Since the bilinear form $a(\cdot,\cdot)$ is symmetric positive-definite on V for all three types of boundary conditions, the biharmonic eigenvalues being considered are positive. Note that we have excluded the trivial eigenvalue 0 from the **CH** problem by imposing the zero mean constraint.

We will refer to the eigenvalue problem for plate vibration (where $b(\cdot,\cdot) = (\cdot,\cdot)$) with the three types of boundary conditions as the **V-CP**, **V-SSP** and **V-CH** problems, and the eigenvalue problem for plate buckling (where $b(\cdot,\cdot) = (\nabla\cdot, \nabla\cdot)$) with the three types of boundary conditions as the **B-CP**, **B-SSP** and **B-CH** problems.

3 The C^0 IPG Method for Biharmonic Eigenvalue Problems

Let $\mathscr{T}_h$ be a regular triangulation of Ω with mesh size h and $\tilde{V}_h \subset H^1(\Omega)$ be the $\mathbb{P}_k$ Lagrange finite element space ($k \geq 2$) associated with $\mathscr{T}_h$. Let $\mathscr{E}_h$ be the set of the edges in $\mathscr{T}_h$. For an edge $e \in \mathscr{E}_h$ that is the common edge of two adjacent triangles $T_\pm \in \mathscr{T}_h$ and for $v \in \tilde{V}_h$, we define the jump of the flux to be

$$[\![\partial v/\partial n_e]\!] = \frac{\partial v_{T_+}}{\partial n_e}\bigg|_e - \frac{\partial v_{T_-}}{\partial n_e}\bigg|_e,$$

where n_e is the unit normal pointing from T_- to T_+. We let

$$\frac{\partial^2 v}{\partial n_e^2} = n_e \cdot (D^2 v) n_e$$

and define the average normal-normal derivative to be

$$\left\{\!\!\left\{ \frac{\partial^2 v}{\partial n_e^2} \right\}\!\!\right\} = \frac{1}{2}\left(\frac{\partial^2 v_{T_+}}{\partial n_e^2} + \frac{\partial^2 v_{T_-}}{\partial n_e^2} \right).$$

For $e \in \partial\Omega$, we take n_e to be the unit outward normal and define

$$[\![\partial v/\partial n_e]\!] = -\frac{\partial v}{\partial n_e} \quad \text{and} \quad \left\{\!\!\left\{ \frac{\partial^2 v}{\partial n_e^2} \right\}\!\!\right\} = \frac{\partial^2 v}{\partial n_e^2}.$$

Let $\mathbb{R}_+$ be the set of positive real numbers. The C^0 IPG method for the biharmonic eigenvalue problem is to find $(u_h, \lambda_h) \in V_h \times \mathbb{R}_+$ such that $u_h \neq 0$ and

$$a_h(u_h, v) = \lambda_h b(u_h, v) \qquad \forall\, v \in V_h, \tag{10}$$

where the choices of V_h and $a_h(\cdot,\cdot)$ depend on the boundary conditions.

CP For this boundary condition the choices for V_h and $a_h(\cdot,\cdot)$ are given by

$$V_h = \tilde{V}_h \cap H_0^1(\Omega), \tag{11}$$

$$\begin{aligned} a_h(w, v) = {} & \sum_{T\in\mathscr{T}_h} \int_T D^2 w : D^2 v \, dx + \sum_{e\in\mathscr{E}_h} \int_e \left\{\!\!\left\{ \frac{\partial^2 w}{\partial n_e^2} \right\}\!\!\right\} \left[\!\!\left[\frac{\partial v}{\partial n_e} \right]\!\!\right] + \left\{\!\!\left\{ \frac{\partial^2 v}{\partial n_e^2} \right\}\!\!\right\} \left[\!\!\left[\frac{\partial w}{\partial n_e} \right]\!\!\right] ds \\ & + \sigma \sum_{e\in\mathscr{E}_h} \frac{1}{|e|} \int_e \left[\!\!\left[\frac{\partial w}{\partial n_e} \right]\!\!\right] \left[\!\!\left[\frac{\partial v}{\partial n_e} \right]\!\!\right] ds, \end{aligned} \tag{12}$$

where $\sigma > 0$ is a (sufficiently large) penalty parameter.

SSP For this boundary condition we use the same V_h in (11) and the bilinear form

$$a_h(w,v) = \sum_{T\in\mathscr{T}_h}\int_T D^2w : D^2v\,dx + \sum_{e\in\mathscr{E}_h^i}\int_e \left\{\frac{\partial^2 w}{\partial n_e^2}\right\}\left[\!\left[\frac{\partial v}{\partial n_e}\right]\!\right] + \left\{\frac{\partial^2 v}{\partial n_e^2}\right\}\left[\!\left[\frac{\partial w}{\partial n_e}\right]\!\right] ds$$
$$+ \sigma\sum_{e\in\mathscr{E}_h^i}\frac{1}{|e|}\int_e \left[\!\left[\frac{\partial w}{\partial n_e}\right]\!\right]\left[\!\left[\frac{\partial v}{\partial n_e}\right]\!\right] ds, \tag{13}$$

where $\mathscr{E}_h^i$ is the set of the edges interior to Ω.

CH For this boundary condition we use the same bilinear form $a_h(\cdot,\cdot)$ defined in (12) and take

$$V_h = \left\{v \in \tilde{V}_h : (v,1) = 0\right\}. \tag{14}$$

The convergence of the C^0 IPG method for these eigenvalue problems is based on the convergence of the C^0 IPG method for the corresponding source problems.

Let W be the space $L^2(\Omega)$ for the plate vibration problems, the space $H_0^1(\Omega)$ for the **B-CP** and **B-SSP** problems, and the space $\{v \in H^1(\Omega) : (v,1) = 0\}$ for the **B-CH** problem. We will denote by $\|f\|_b$ the norm induced by the bilinear form $b(\cdot,\cdot)$ defined in (5), i.e.,

$$\|f\|_b^2 = b(f,f).$$

Given $f \in W$, the weak formulation for the source problem is to find $u \in V$ such that

$$a(u,v) = b(f,v) \qquad \forall\, v \in V, \tag{15}$$

where the bilinear form $a(\cdot,\cdot)$ is defined in (4). For the **V-CH** source problem, we also assume that f satisfies the constraint $(f,1) = 0$.

The corresponding C^0 IPG method for (15) is to find $u_h \in V_h$ such that

$$a_h(u_h,v) = b(f,v) \qquad \forall\, v \in V_h, \tag{16}$$

where V_h and $a_h(\cdot,\cdot)$ are defined by

1. Equations (11) and (12) respectively for the **CP** boundary conditions,
2. Equations (11) and (13) respectively for the **SSP** boundary conditions, and
3. Equations (14) and (12) respectively for the **CH** boundary conditions.

The following lemma summarizes the results for the source problems obtained in [9, 10, 12].

Lemma 1 *The biharmonic source problem* (15) *and the discrete source problem* (16) *are uniquely solvable for the boundary conditions of* ***CP***, ***SSP*** *and* ***CH***. *In addition there exists* $\beta > 0$ *such that*

$$\|u - u_h\|_h \leq Ch^{\beta}\|f\|_b \quad \text{and} \quad \|u - u_h\|_b \leq Ch^{2\beta}\|f\|_b, \tag{17}$$

where $u \in V$ *(resp.* $u_h \in V_h$*) is the solution of* (15) *[resp.* (16)*], and the mesh-dependent energy norm* $\|\cdot\|_h$ *is defined by*

$$\|v\|_h^2 = \sum_{T\in\mathscr{T}_h} |v|_{H^2(T)}^2 + \sum_{e\in\mathscr{E}_h} |e|^{-1}\|[\![\partial v/\partial n_e]\!]\|_{L^2(e)}^2 \tag{18}$$

for the boundary conditions of ***CP*** *and* ***CH***, *and*

$$\|v\|_h^2 = \sum_{T\in\mathscr{T}_h} |v|_{H^2(T)}^2 + \sum_{e\in\mathscr{E}_h^i} |e|^{-1}\|[\![\partial v/\partial n_e]\!]\|_{L^2(e)}^2 \tag{19}$$

for the boundary conditions of ***SSP***.

Remark 2 Let V be the Sobolev space for the biharmonic problem defined in (7), (8) or (9) and V_h be the corresponding finite element space. In all three cases we have a Poincaré-Friedrichs inequality [11]

$$\|v\|_b \leq C\|v\|_h \qquad \forall\, v \in V + V_h. \tag{20}$$

Remark 3 The exponent β in (17) is given by $\beta = \min(\alpha, k-1)$, where α is index of elliptic regularity that appears in the elliptic regularity estimate [6]

$$\|u\|_{H^{2+\alpha}(\Omega)} \leq C_{\Omega,\alpha}\|f\|_b$$

for the solution u of the source problem (15). It is determined by the angles at the corners of Ω and the boundary conditions. For the **CP** boundary conditions (1), α belongs to $(\frac{1}{2}, 2]$ and $\alpha > 1$ if Ω is convex. For the **SSP** boundary conditions (2) and the **CH** boundary conditions (3), α belongs to $(0, 2]$ in general, $\alpha = 2$ for a rectangular domain, and α is any number strictly less than $1/3$ for an L-shaped domain.

The convergence analysis of the C^0 IPG method for the biharmonic eigenvalue problems involves two (bounded) solution operators $T : W \longrightarrow V\,(\subset W)$ and $T_h : W \longrightarrow V_h\,(\subset W)$ on the Hilbert space $\big(W, b(\cdot,\cdot)\big)$, which are defined by

$$a(Tf, v) = b(f, v) \quad \forall\, v \in V \quad \text{and} \quad a_h(T_h f, v) = b(f, v) \quad \forall\, v \in V_h.$$

Note that (6) is equivalent to $Tu = (1/\lambda)u$, (10) is equivalent to $T_h u_h = (1/\lambda_h)u_h$, and the estimates (17) can be rewritten as

$$\|(T - T_h)f\|_h \le Ch^{\beta}\|f\|_b \quad \text{and} \quad \|(T - T_h)f\|_b \le Ch^{2\beta}\|f\|_b \quad \forall f \in W. \tag{21}$$

Due to the compact embedding of V into W, the operator T is symmetric, positive-definite and compact. Therefore the spectrum of T consists of a sequence of positive eigenvalues $\mu_1 \ge \mu_2 \ge \dots$ decreasing to zero, and the numbers $\lambda_j = 1/\mu_j$ are the biharmonic eigenvalues that increase to infinity.

The theorem below on the convergence of the C^0 IPG method for the biharmonic eigenvalue problems follows from (20), (21) and the classical spectral approximation theory that can be found for example in [3, Sect. 2.7].

Theorem 1 *Let $0 < \lambda_1 \le \lambda_2 \le \dots$ be the biharmonic eigenvalues, $\lambda = \lambda_j = \dots = \lambda_{j+m-1}$ be a biharmonic eigenvalue with multiplicity m, and $0 < \lambda_{h,1} \le \lambda_{h,2} \le \dots$ be the discrete eigenvalues obtained by the C^0 IPG method. Then we have, as $h \to 0$,*

$$|\lambda_{h,l} - \lambda| \le Ch^{2\beta}, \quad l = j, j+1, \dots, j+m-1.$$

In addition, if $V_\lambda \subset V$ is the space spanned by the eigenfunctions corresponding to the biharmonic eigenvalues $\lambda_j, \dots, \lambda_{j+m-1}$, $V_{h,\lambda} \subset V_h$ is the space spanned by the eigenfunctions corresponding to the discrete eigenvalues $\lambda_{h,j}, \dots, \lambda_{h,j+m-1}$, and $\delta(V_\lambda, V_{h,\lambda})$ is the gap between them, then we have, as $h \to 0$, $\delta(V_\lambda, V_{h,\lambda}) \le Ch^{\beta}$ in the norm $\|\cdot\|_h$ and $\delta(V_\lambda, V_{h,\lambda}) \le Ch^{2\beta}$ in the norm $\|\cdot\|_b$.

Remark 4 We can apply the classical theory because we use the Hilbert space $\big(W, b(\cdot,\cdot)\big)$ and V_h is a subspace of W. This would not be possible if we use the space V in (7)–(9).

Remark 5 The convergence of the method in [24] for eigenvalue problems can similarly be established by the classical spectral approximation theory.

4 Numerical Examples of the C^0 IPG Method

In this section we present numerical results of the quadratic C^0 interior penalty method. The penalty parameter σ is taken to be 50 in all the computations. The discrete eigenvalue problems are solved in MATLAB by using the eigs command.

We first consider the unit square. In Table 1 we display the first biharmonic eigenvalues for the plate vibration problems, computed by the C^0 IPG method on a series of structured meshes generated by uniform refinement. We note that the first **V-CP** eigenvalue obtained in [25] is 1294.93398. The first **V-SSP** eigenvalue

Table 1 The first biharmonic plate vibration eigenvalues for the unit square on uniformly refined meshes

h	1/10	1/20	1/40	1/80
V-CP(1)	1381.7409	1319.9044	1301.6876	1296.6904
V-SSP(1)	395.2823	391.2186	390.0615	389.7466
V-CH(1)	98.2432	97.6410	97.4711	97.4251

Table 2 Biharmonic plate vibration eigenvalues of the L-shaped domain on quasi-uniform meshes

h	1/10	1/20	1/40	1/80
V-CP(1)	7828.3034	7102.9564	6853.9181	6762.3442
V-SSP(1)	2767.1992	2686.5991	2657.3435	2641.3376
V-SSP(3)	6327.5449	6573.0063	6259.2682	6240.6958
V-CH(1)	202.1341	188.1774	181.0593	176.5303
V-CH(3)	1603.9472	1571.3380	1562.0031	1559.4471

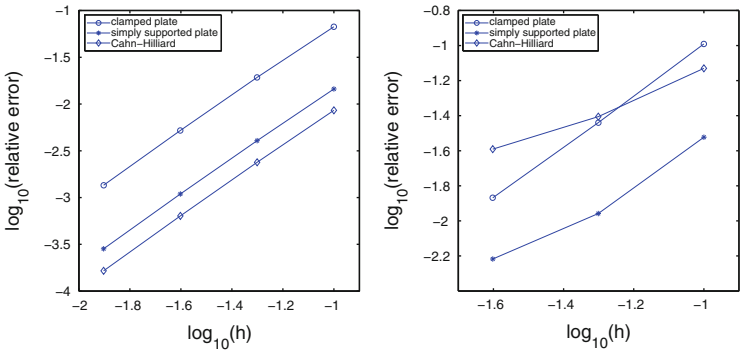


Fig. 1 Convergence history of the first biharmonic plate vibration eigenvalues. *Left*: the unit square. *Right*: the L-shaped domain

is $4\pi^4 \approx 389.6363$ and the first **V-CH** eigenvalue is $\pi^4 \approx 97.4091$. Therefore the C^0 IPG method provides good approximations in all three cases.

The second domain is the L-shaped domain. In Table 2 we present the first biharmonic plate vibration eigenvalues computed by the C^0 IPG method. We also include the results for the third eigenvalues of **V-SSP** and **V-CH**, whose exact values are $64\pi^4 \approx 6234.1818$ and $16\pi^4 \approx 1558.5455$, respectively. They are approximated correctly with less than 1 % relative error at the finest meshes.

In Fig. 1 we plot the convergence history of the C^0 IPG method. In the case of the unit square, the convergence rates are $O(h^2)$ as predicted by the theory in the previous section. In the case of the L-shaped domain, there is a decrease in the convergence rate due to the reentrant corner (1.6 for **V-CP**, 1.0 for **V-SSP** and 0.86 for **V-CH**), which is also consistent with the theoretical result.

Table 3 The first **B-CP**, **B-SSP** and **B-CH** eigenvalues for the unit square on quasi-uniform meshes

h	1/10	1/20	1/40	1/80
B-CP(1)	55.4016	53.2067	52.5757	52.4045
B-SSP(1)	20.0244	19.8193	19.7607	19.7448
B-CH(1)	9.9541	9.893	9.8758	9.8712

Next we present some numerical results for the **B-CP** problem, the **B-SSP** problem and the **B-CH** problem. The first eigenvalues on a series of uniformly refined meshes for the unit square are displayed in Table 3. The approximate eigenvalue for the **B-CP** on the unit square agree with the approximation obtained in [5], and the approximate eigenvalues for **B-SSP** (resp. **B- CH**) problem on the unit square also agrees with the exact eigenvalue $2\pi^2 \approx 19.73920880$ (resp. $\pi^2 \approx 9.869604401$).

The convergence history of the first eigenvalue for the plate buckling problem on the unit square (and the L-shaped domain) is similar to that of the plate vibration problem.

5 Comparison with Other Methods

In this section we compare the quadratic C^0 IPG method with the quintic Argyris C^1 finite element method [2], the Ciarlet-Raviart mixed finite element method [3, 15, 19], and the Morley nonconforming finite element method [21, 22].

The numerical results of the four methods for the plate vibration problem on the unit square, the L-shaped domain and with the three types of boundary conditions are presented in Tables 4, 5, 6, 7, 8, and 9. The mesh size used in the computations is $\approx$1/80. In addition to the first six biharmonic eigenvalues, we also display the number of degrees of freedom (DoF).

We observe that the numerical results for the quadratic C^0 IPG method and the Argyris C^1 finite element method are comparable in all six cases. In view of the high order of the finite element, the Argyris method provides very accurate approximation of the biharmonic eigenvalues corresponding to smooth eigenfunctions. Therefore the quadratic C^0 IPG method is also quite efficient. This can also be seen by comparing the approximate eigenvalues in Table 4 with the ones in [25].

From Tables 4, 6 and 8 we see that the Ciarlet-Raviart mixed finite element method converges on the unit square for all three types of boundary conditions. It is interesting to note that the eigenvalues computed by the Ciarlet-Raviart method are consistently larger than the corresponding eigenvalues computed by the C^0 IPG method, and the eigenvalues computed by the Argyris method are always between the other two with only one exception (the 4th eigenvalue in Table 4).

Table 4 The first six **V-CP** eigenvalues for the unit square using a uniform mesh

	DoF	1st	2nd	3rd	4th	5th	6th
C^0IPG	16129	0.1299e4	0.5411e4	0.5411e4	1.1811e4	1.7412e4	1.7584e4
Argyris	36226	0.1304e4	0.5427e4	0.5427e4	1.1798e4	1.7443e4	1.7608e4
Mixed	3969	0.1309e4	0.5451e4	0.5451e4	1.1877e4	1.7548e4	1.7714e4
Morley	16129	0.1290e4	0.5349e4	0.5349e4	1.1607e4	1.7113e4	1.7280e4

Table 5 The first six **V-CP** eigenvalues for the L-shaped domain using a quasi-uniform mesh

	DoF	1st	2nd	3rd	4th	5th	6th
C^0IPG	32705	0.6694e4	1.0815e4	1.4655e4	2.5862e4	3.3418e4	5.3545e4
Argyris	73502	0.6775e4	1.1122e4	1.4985e4	2.6274e4	3.3686e4	5.4003e4
Mixed	8097	0.6695e4	1.1063e4	1.4925e4	2.6201e4	3.3499e4	5.3713e4
Morley	32705	0.6630e4	1.1004e4	1.4842e4	2.6018e4	3.3164e4	5.3033e4

Table 6 The first six **V-SSP** eigenvalues for the unit square using a uniform mesh

	DoF	1st	2nd	3rd	4th	5th	6th
C^0IPG	16129	0.3896e3	2.4166e3	2.4166e3	6.1961e4	9.6768e3	9.6768e3
Argyris	36990	0.3896e3	2.4352e3	2.4352e3	6.2343e3	9.7409e3	9.7409e3
Mixed	3969	0.3900e3	2.4409e3	2.4409e3	6.2609e3	9.7806e3	9.7806e3
Morley	16385	0.3893e3	2.4295e3	2.4295e3	6.2143e4	9.6896e3	9.6896e3

Table 7 The first six **V-SSP** eigenvalues for the L-shaped domain using a quasi-uniform mesh

	DoF	1st	2nd	3rd	4th	5th	6th
C^0IPG	32705	0.2718e4	0.3743e4	0.6061e4	1.3666e4	1.9156e4	3.1027e4
Argyris	74454	0.2692e4	0.3765e4	0.6234e4	1.3972e4	1.9375e4	3.1281e4
Mixed	8097	0.1491e4	0.3699e4	0.6242e4	1.3969e4	1.6354e4	2.7617e4
Morley	33025	0.2414e4	0.3663e4	0.6225e4	1.3904e4	1.8642e4	3.0002e4

Table 8 The first six **V-CH** eigenvalues for the unit square using a uniform mesh

	DoF	1st	2nd	3rd	4th	5th	6th
C^0IPG	16641	0.0970e3	0.0970e3	0.3881e3	1.5524e3	1.5524e3	2.4277e3
Argyris	36994	0.0974e3	0.0974e3	0.3896e3	1.5585e3	1.5585e3	2.4352e3
Mixed	4225	0.0974e3	0.0974e3	0.3901e3	1.5606e3	1.5606e3	2.4409e3
Morley	16385	0.0974e3	0.0974e3	0.3893e3	1.5548e3	1.5548e3	2.4295e3

Table 9 The first six **V-CH** eigenvalues for the L-shaped domain using a quasi-uniform mesh

	DoF	1st	2nd	3rd	4th	5th	6th
C^0IPG	33345	0.1783e3	0.2089e3	1.5097e3	1.5138e3	2.0354e3	2.9839e3
Argyris	74462	0.1755e3	0.2068e3	1.5585e3	1.5585e3	2.0856e3	3.0373e3
Mixed	8417	0.0349e3	0.1998e3	1.5595e3	1.5595e3	2.0769e3	2.5333e3
Morley	33025	0.1498e3	0.1971e3	1.5575e3	1.5576e3	2.0701e3	2.9353e3

Table 10 The first eigenvalues of the plate buckling problems for the unit square

	B-CP	**B-SSP**	**B-CH**
C^0IPG	52.4045	19.7448	9.8712
Argyris	52.3469	19.7392	9.8695
Mixed	52.3671	19.7422	9.8704
Morley	52.3301	19.7383	9.8694

Table 11 The first eigenvalues of the plate buckling problems for the L-shaped domain

	B-CP	**B-SSP**	**B-CH**
C^0IPG	129.3580	61.6123	14.4305
Argyris	129.0132	61.9109	14.6288
Mixed	128.4905	38.6147	5.9099
Morley	127.7805	59.1396	13.9426

For the L-shaped domain, we observe from Table 5 that the Ciarlet-Raviart mixed finite element method also converges for the **V-CP** problem, and again the eigenvalues computed by the Ciarlet-Raviart mixed finite element method are consistently larger than the corresponding eigenvalues computed by the C^0 IPG method. For the boundary conditions of **SSP** and **CH**, the results in Tables 7 and 9 show spurious eigenvalues generated by the Ciarlet-Raviart mixed finite element method.

Comparing with the C^0 IPG method, the performance of the Morley finite element method is slightly better when the eigenfunction is very smooth and slightly worse when the eigenfunction is less smooth. The approximate eigenvalues generated by the Morley finite element method is consistently less than the approximations generated by the Argyris finite element method, which agrees with the discussion in [18].

Finally numerical results for the first eigenvalues of the plate buckling problems are presented in Tables 10 (unit square) and 11 (L-shaped domain). The mesh size h in the computations is $\approx 1/80$. For the unit square, the results from all four methods with respect to all three boundary conditions are consistent. For the L-shaped domain, the results from the C^0IPG method, the Argyris finite element method and the Morley finite element method are consistent for all three boundary conditions, whereas the Ciarlet-Raviart mixed finite element method is consistent with the other methods only for the **CP** boundary conditions and generates spurious eigenvalues for the other two boundary conditions.

6 Conclusion

We have demonstrated that the C^0 IPG method is a provably accurate scheme for approximating biharmonic eigenvalue problems. It is robust with respect to different boundary conditions, which is a significant advantage over the Ciarlet-Raviart mixed finite element method, because the latter produces spurious eigenvalues on

nonconvex domains for the boundary conditions of the simply supported plate and the Cahn-Hilliard type. Its performance is also comparable to the more complicated Argyris C^1 finite element method.

The results in this paper can be extended to three dimensions where the advantage over C^1 finite element methods would be even more obvious, and they can also be extended to domains with curved boundaries where the isoparametric version of the C^0 IPG method [9, 13] can be applied, while the constructions of C^1 finite element space for such domains are much more complicated.

From the numerical results in Sect. 5, we see that the Ciarlet-Raviart mixed finite element method converges on nonconvex domains for the boundary conditions of the clamped plate. As far as we know this method has only been analyzed for convex domains [4, 15, 17] or smooth domains [20] even for the source problem. It would be interesting to develop a convergence analysis of the Ciarlet-Raviart mixed finite element method on nonconvex domains for both the source problem and the eigenvalue problem.

Acknowledgements The work of the first author was supported in part by the NSF Grant DMS-1319172. The work of the second author is supported in part by the Air Force Office of Scientific Research Grant FA9550-13-1-0199 and by NSF Grant DMS-1216620. The work of the third author is supported in part by NSF Grants DMS-1521555 and DMS-132139.

References

1. A. Adini, R.W. Clough, Analysis of plate bending by the finite element method. NSF Report G.7337 (1961)
2. J.H. Argyris, I. Fried, D.W. Scharpf, The TUBA family of plate elements for the matrix displacement method. Aeronaut. J. R. Aeronaut. Soc. **72**, 701–709 (1968)
3. I. Babuška, J. Osborn, Eigenvalue problems, in *Handbook of Numerical Analysis II*, ed. by P.G. Ciarlet, J.L. Lions (North-Holland, Amsterdam, 1991), pp. 641–787
4. I. Babuška, J. Osborn, J. Pitkäranta, Analysis of mixed methods using mesh dependent norms. Math. Comput. **35**, 1039–1062 (1980)
5. P.E. Bjørstad, B.P. Tjøstheim, High precision solution of two fourth order eigenvalue problems. Computing **63**, 97–107 (1999)
6. H. Blum, R. Rannacher, On the boundary value problem of the biharmonic operator on domains with angular corners. Math. Methods Appl. Sci. **2**, 556–581 (1980)
7. D. Boffi, F. Brezzi, L. Gastaldi, On the problem of spurious eigenvalues in the approximation of linear elliptic problems in mixed form. Math. Comput. **69**, 121–140 (2000)
8. S.C. Brenner, C^0 interior penalty methods, in *Frontiers in Numerical Analysis - Durham 2010*. Lecture Notes in Computational Science and Engineering, vol. 85 (Springer, Berlin, 2012), pp. 79–147
9. S.C. Brenner, M. Neilan, A C^0 interior penalty method for a fourth order elliptic singular perturbation problem. SIAM J. Numer. Anal. **49**, 869–892 (2011)
10. S.C. Brenner, L. Sung, C^0 interior penalty methods for fourth order elliptic boundary value problems on polygonal domains. J. Sci. Comput. **22/23**, 83–118 (2005)
11. S.C. Brenner, K. Wang, J. Zhao, Poincaré-Friedrichs inequalities for piecewise H^2 functions. Numer. Funct. Anal. Optim. **25**, 463–478 (2004)

12. S.C. Brenner, S. Gu, T. Gudi, L.-Y. Sung, A quadratic C^0 interior penalty method for linear fourth order boundary value problems with boundary conditions of the Cahn-Hilliard type. SIAM J. Numer. Anal. **50**, 2088–2110 (2012)
13. S.C. Brenner, M. Neilan, L.-Y. Sung, Isoparametric C^0 interior penalty methods. Calcolo **49**, 35–67 (2013)
14. J.W. Cahn, J.E. Hilliard, Free energy of a nonuniform system-I: interfacial free energy. J. Chem. Phys. **28**, 258–267 (1958)
15. P.G. Ciarlet, P.-A. Raviart, A mixed finite element method for the biharmonic equation, in *Mathematical Aspects of Finite Elements in Partial Differential Equations* (Proc. Sympos., Math. Res. Center, University of Wisconsin, Madison, WI, 1974). (Academic Press, New York, 1974), pp. 125–145
16. G. Engel, K. Garikipati, T.J.R. Hughes, M.G. Larson, L. Mazzei, R.L. Taylor, Continuous/discontinuous finite element approximations of fourth order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity. Comput. Methods Appl. Mech. Eng. **191**, 3669–3750 (2002)
17. R. Falk, J. Osborn, Error estimates for mixed methods. RAIRO Anal. Numér. **14**, 249–277 (1980)
18. J. Hu, Y. Huang, Q. Shen, The lower/upper bound property of approximate eigenvalues by nonconforming finite element methods for elliptic operators. J. Sci. Comput. **58**, 574–591 (2014)
19. B. Mercier, J. Osborn, J. Rappaz, P.-A. Raviart, Eigenvalue approximation by mixed and hybrid methods. Math. Comput. **36**, 427–453 (1981)
20. P. Monk, A mixed finite element method for the biharmonic equation. SIAM J. Numer. Anal. **24**, 737–749 (1987)
21. L. Morley, The triangular equilibrium problem in the solution of plate bending problems. Aeronaut. Q. **19**, 149–169 (1968)
22. R. Rannacher, Nonconforming finite element methods for eigenvalue problems in linear plate theory. Numer. Math. **33**, 23–42 (1979)
23. J. Sun, A new family of high regularity elements. Numer. Methods Partial Differ. Equ. **28**, 1–16 (2012)
24. G.N. Wells, N.T. Dung, A C^0 discontinuous Galerkin formulation for Kirchhoff plates. Comput. Methods Appl. Mech. Eng. **196**, 3370–3380 (2007)
25. C. Wieners, Bounds for the N lowest eigenvalues of fourth-order boundary value problems. Computing **59**, 29–41 (1997)

Strong Stability Preserving Time Discretizations: A Review

Sigal Gottlieb

Abstract Strong stability preserving (SSP) high order time discretizations were developed to address the need for nonlinear stability properties in the numerical solution of hyperbolic partial differential equations with discontinuous solutions. These methods preserve the monotonicity properties (in any norm, seminorm or convex functional) of the spatial discretization coupled with first order Euler time stepping. This review paper describes the state of the art in SSP methods.

1 Overview

Explicit strong stability preserving (SSP) Runge–Kutta methods were developed [32, 33] for the time evolution of hyperbolic conservation laws $U_t + f(U)_x = 0$. Solving these methods numerically is complicated by the fact that the exact solutions may develop discontinuities. For this reason, significant effort has been expended on finding spatial discretizations that can handle discontinuities [9]. Once the spatial derivative is discretized, we obtain the system of ODEs

$$u_t = F(u), \tag{1}$$

where u is a vector of approximations to U: $u_j \approx U(x_j)$. This system of ODEs can then be evolved in time using standard methods. The spatial discretizations used to approximate $f(U)_x$ are carefully designed so that when (1) is evolved in time using the forward Euler method $u^{n+1} = u^n + \Delta t F(u^n)$ the solution satisfies the strong stability property

$$\|u^n + \Delta t F(u^n)\| \leq \|u^n\| \tag{2}$$

S. Gottlieb (✉)
Mathematics Department, University of Massachusetts Dartmouth, 285 Old Westport Rd, North Dartmouth, MA 02747, USA
e-mail: sgottlieb@umassd.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_2

under the time step restriction

$$\Delta t \leq \Delta t_{\text{FE}}. \tag{3}$$

The term $\|\cdot\|$ can represent any norm, semi-norm, or convex functional, as dictated by the design properties of the spatial discretization.

In practice, a higher order time discretization is needed for numerical simulations, but we want to ensure that the higher order discretization will preserve the strong stability properties of the spatial discretization coupled with forward Euler. To accomplish this, we attempt to re-write a higher order time discretization as a convex combination of forward Euler steps, so that any convex functional property that is satisfied by the forward Euler method will still be satisfied by the higher order time discretization, perhaps under a modified time step restriction

$$\Delta t \leq \mathscr{C} \Delta t_{\text{FE}}. \tag{4}$$

Methods that can be decomposed like this with $\mathscr{C} > 0$ are called strong stability preserving (SSP), and the $\mathscr{C}$ is known as the *SSP coefficient* of the method. SSP methods guarantee the strong stability of the numerical solution for any ODE and any convex functional provided *only* that the forward Euler condition (2) is satisfied under a time step restriction (3).

It is easy to see how a decomposition into convex combinations of forward Euler steps is a sufficient condition for strong stability preservation. It has also been shown [4, 5, 9, 12, 13] that this convex combination condition is *necessary* for strong stability preservation. If a method does not have a convex combination decomposition into forward Euler steps with a positive $\mathscr{C}$ we can always find some ODE with some initial condition such that the forward Euler condition is satisfied but the method does not satisfy the strong stability condition for any positive time-step [9].

Notice that there are two factors that play a role in determining the stable time step (4): the forward Euler time step Δt_{FE}, which depends on the spatial discretization alone, and the SSP coefficient $\mathscr{C}$, which depends only on the time discretization. For efficiency, we seek high order SSP Runge–Kutta methods that have the largest possible SSP coefficient $\mathscr{C}$ per function evaluation. The number of function evaluations is typically the number of stages s of a method, so we define the *effective SSP coefficient* $\mathscr{C}_{\text{eff}} = \frac{\mathscr{C}}{s}$ and aim to find methods that maximize this value. It has been shown [9] that all explicit general linear methods have an SSP bound $\mathscr{C} \leq s$, and therefore $\mathscr{C}_{\text{eff}} \leq 1$, but this upper bound is not always attained. In the following sections, we present some of the work done on SSP time discretization methods of several types, and provide a number of recommendations for the best SSP time discretizations based on the state-of-the-art in this field. Space constraints do not permit a detailed treatment of these topics, and many interesting methods such as additive methods, implicit-explicit methods, and multiderivative methods will not be addressed here, but the reader is encouraged to explore these as well.

2 SSP Runge–Kutta Methods

Runge–Kutta methods are typically written in the Butcher form, however these methods can also be written in the Shu-Osher form [32, 33], which is more convenient for our purpose:

$$u^{(i)} = v_i u^n + \sum_{j=1}^{s} \left(\alpha_{i,j} u^{(j)} + \Delta t \beta_{i,j} F(u^{(j)})\right) \quad \text{for} \quad 1 \le i \le s+1 \tag{5}$$
$$u^{n+1} = u^{(s+1)},$$

where we require $v_i + \sum_{j=1}^{s} \alpha_{i,j} = 1$ for consistency. If all the coefficients $v_i \ge 0$, $\alpha_{i,j} \ge 0$ and $\beta_{i,j} \ge 0$ then each stage of this Runge–Kutta method (5) can be rewritten as a convex combination of forward Euler steps, and we can bound any convex functional by

$$\begin{aligned} \|u^{(i)}\| &= \left\| v_i u^n + \sum_{j=1}^{s} \left(\alpha_{i,j} u^{(j)} + \Delta t \beta_{i,j} F(u^{(j)})\right) \right\| \\ &\le v_i \|u^n\| + \sum_{j=1}^{s} \alpha_{i,j} \left\| u^{(j)} + \Delta t \frac{\beta_{i,j}}{\alpha_{i,j}} F(u^{(j)}) \right\| \le \|u^n\|, \end{aligned}$$

where the final inequality is obtained by repeatedly using the forward Euler condition (2) with the requirement that $\frac{\beta_{i,j}}{\alpha_{i,j}} \Delta t \le \Delta t_{FE}$. This decomposition shows that if the forward Euler condition (2) holds under some time step restriction (3), and if $\alpha_{i,j}, \beta_{i,j} \ge 0$, then the solution obtained by the Runge–Kutta method (5) satisfies the strong stability bound $\|u^{n+1}\| \le \|u^n\|$ under the time step restriction (4) where $\mathscr{C} = \min_{i,j} \frac{\alpha_{i,j}}{\beta_{i,j}}$, and the ratio is understood as infinite if $\beta_{i,j} = 0$.

To facilitate finding methods with optimal SSP coefficients, Ketcheson formulated an optimization problem in [17]. This optimization problem was used extensively in [1, 11, 18, 21, 22], to generate optimal SSP methods. The optimization code in MATLAB is available at [23]. This code is the basis of many of the results described in this work. Table 1 (left) lists the effective SSP coefficients of the best known explicit and implicit SSP Runge–Kutta methods.

2.1 SSP Explicit Runge–Kutta Methods

The first two strong stability preserving Runge–Kutta methods were presented in [32, 33]. These were the explicit s stage SSP Runge–Kutta method of order $p = s = 2$

$$u^{(1)} = u^{(0)} + \Delta t F(u^{(0)}), \qquad u^{n+1} = \frac{1}{2} u^{(0)} + \frac{1}{2} \left(u^{(1)} + \Delta t F(u^{(1)})\right).$$

Table 1 Left: Effective SSP coefficients of best known explicit and implicit methods described in Sects. 1 and 2. Right: SSP coefficients of optimal explicit LNL methods described at the end of Sect. 3

	Explicit methods			Implicit methods				
$s\backslash p$	2	3	4	2	3	4	5	6
1	–	–	–	2	–	–	–	–
2	0.5	–	–	2	1.37	–	–	–
3	0.67	0.33	–	2	1.61	0.68	–	–
4	0.75	0.5	-	2	1.72	1.11	0.29	
5	0.8	0.53	0.30	2	1.78	1.21	0.64	
6	0.83	0.59	0.38	2	1.82	1.30	0.83	0.030
7	0.86	0.61	0.47	2	1.85	1.31	0.89	0.038
8	0.88	0.64	0.52	2	1.87	1.33	0.94	0.28
9	0.89	0.67	0.54	2	1.89	1.34	0.99	0.63
10	0.9	0.68	0.60	2	1.90	1.36	1.01	0.81
11	0.91	0.69	0.59	2	1.91	1.38	1.03	0.80

	Explicit LNL with $p = 4$ and higher linear order							
$s\backslash p_{lin}$	5	6	7	8	9	10	11	12
2	–	–	–	–	–	–	–	–
3	–	–	–	–	–	–	–	–
4	–	–	–	–	–	–	–	–
5	0.76	–	–	–	–	–	–	–
6	1.81	0.87	–	–	–	–	–	–
7	2.58	1.83	**1**	–	–	–	–	–
8	3.36	2.56	1.93	**1**	–	–	–	–
9	4.03	3.35	2.63	1.95	**1**	–	–	–
10	4.76	4.04	**3.37**	2.64	1.99	**1**	–	–
11	5.49	4.78	4.08	**3.37**	**2.65**	**2**	**1**	–
12	6.27	**5.52**	4.68	4.08	**3.37**	**2.65**	**2**	**1**

A dash indicates that SSP methods of this type cannot exist, a blank space indicates none were found

and the three stage third order method

$$u^{(1)} = u^n + \Delta t F(u^n), \qquad u^{(2)} = \tfrac{3}{4}u^n + \tfrac{1}{4}u^{(1)} + \tfrac{1}{4}\Delta t F(u^{(1)}),$$
$$u^{n+1} = \tfrac{1}{3}u^n + \tfrac{2}{3}u^{(2)} + \tfrac{2}{3}\Delta t F(u^{(2)}),$$

that have SSP coefficient $\mathscr{C} = 1$, and are optimal in the sense that there are no methods of that number of stages and order that have a larger SSP coefficient [7].

It was shown in [7] that no four stage fourth order explicit Runge–Kutta methods exist with positive SSP coefficient. By considering methods with $s > p$, fourth order methods with order $p = 4$ have been found. Notable among these is the $(s, p) = (5, 4)$ method with $\mathscr{C} = 1.508$ ($\mathscr{C}_{\text{eff}} = 0.302$) in [35], and the ten-stage fourth order method with $\mathscr{C} = 6$ ($\mathscr{C}_{\text{eff}} = 0.6$) in [17]

$$u^{(1)} = u^n + \frac{1}{6}\Delta t F(u^n), \qquad u^{(i+1)} = u^{(i)} + \frac{1}{6}\Delta t F(u^{(i)}) \quad i = 1, 2, 3$$
$$u^{(5)} = \frac{3}{5}u^n + \frac{2}{5}u^{(4)} + \frac{1}{15}\Delta t F(u^{(4)}), \qquad u^{(i+1)} = u^{(i)} + \frac{1}{6}\Delta t F(u^{(i)}) \quad i = 5, 6, 7, 8,$$
$$u^{n+1} = \frac{1}{25}u^n + \frac{9}{25}u^{(4)} + \frac{3}{5}u^{(9)} + \frac{3}{50}\Delta t F(u^{(4)}) + \frac{1}{10}\Delta t F(u^{(9)}).$$

It was shown [25, 31] that no methods of order $p \geq 5$ with positive SSP coefficients can exist. This means that explicit SSP Runge–Kutta methods have an order barrier of four and any higher order methods will not have a positive SSP coefficient. The three methods given in this section represent the state-of-the-art explicit SSP Runge–Kutta methods. They are all provably optimal and have nice low-storage properties.

2.2 SSP Implicit Runge–Kutta Methods

If a spatial discretization F satisfies the forward Euler condition (2) under some time step restriction (3) it will be *unconditionally* strongly stable, in the same norm, using the implicit (or "backward") Euler method [12, 16, 25]. Unfortunately, no methods of order $p > 1$ can be unconditionally SSP [9]. In fact, the general optimization method was used to investigate the SSP properties of fully implicit SSP Runge–Kutta methods [21], and found that all the methods with order $p \geq 2$ had effective SSP coefficients $\mathscr{C}_{\text{eff}} \leq 2$. This numerical bound holds for second order methods, which means that limiting ourselves to linear problems (see Sect. 2.3) will not alleviate this restriction.

It has been shown that implicit Runge–Kutta methods with positive SSP coefficient cannot exist for $p > 6$ [9]. This order barrier was shown to be sharp in [21] where SSP implicit Runge-Kutta methods of order up to and including six were found. It is interesting to note that in this work, all the optimal methods were

diagonally implicit even though the search was conducted over all fully implicit methods. Diagonally implicit methods have the property that each stage depends explicitly on the previous values and implicitly only on itself, so that each stage can be computed individually.

The SSP time step restriction is particularly desirable for implicit methods as it provides a guarantee of other properties. For example, the SSP condition guarantees a unique solution of the stage equations in implicit Runge–Kutta methods [25] and also ensures that the errors introduced in the solution of the stage equations due to numerical roundoff and errors in the implicit solver are not unduly amplified [25]. Unfortunately, the order barrier of $p \leq 6$ and the observed bounds on the SSP coefficient of implicit Runge–Kutta methods greatly limit their use in practice.

2.3 SSP Runge–Kutta Methods for Linear Problems

This restrictive order barriers on explicit SSP Runge–Kutta methods are partly a result of the nonlinearity of the ODEs. When dealing with linear autonomous ODE systems a smaller set of order conditions needs to be satisfied and we can find explicit SSP Runge–Kutta methods for arbitrarily high linear orders $p_{lin} > 4$ [6]. Such "linear" methods are interesting because their SSP coefficients serve as upper bounds for the usual methods, but they may also be useful in their own right, as the strong stability preserving property can be useful for linear problems involving Maxwell's equations and the equations of linear elasticity.

In [24], Kraaijevanger presented optimal *linear* methods for linear orders $1 \leq p_{lin} \leq s \leq 10$, and $p_{lin} \in \{1, 2, 3, 4, s-1, s-2, s-3, s-4\}$ for any s. For example, a family of provably optimal s-stage, linear order $p_{lin} = s - 1$ methods has $\mathscr{C} = 2$ and $\mathscr{C}_{\text{eff}} = \frac{2}{s}$:

$$
\begin{aligned}
u^{(i)} &= u^{(i-1)} + \frac{1}{2}\Delta t F\left(u^{(i-1)}\right), \qquad i = 1, \ldots, s-1 \\
u^{(s)} &= \sum_{j=0}^{s-2} \alpha_j^s u^{(j)} + \alpha_{s-1}^s \left(u^{(s-1)} + \frac{1}{2}\Delta t F\left(u^{(s-1)}\right)\right), \\
\mathbf{u}^{n+1} &= u^{(s)},
\end{aligned}
$$

where $u^{(0)} = u^n$ and the coefficients α_j^s of the final stage of the s-stage method are given iteratively by

$$
\alpha_j^s = \frac{2}{k}\alpha_{j-1}^{s-1} \quad \text{for} \quad j = 1, \ldots, s-2, \quad \alpha_{s-1}^s = \frac{2}{s}\alpha_{s-2}^{s-1}, \qquad \alpha_0^s = 1 - \sum_{j=1}^{s-1} \alpha_j^s,
$$

starting from the coefficients of the two-stage, first order method $\alpha_0^2 = 0$ and $\alpha_1^2 = 1$.

The linear and nonlinear order conditions are equivalent up to and including second order, so that the methods of order $p_{lin} > 2$ still have nonlinear order $p = 2$. In [11], we constructed explicit SSP Runge–Kutta methods that have nonlinear order $p = 3$ or $p = 4$ (which is optimal for SSP methods), and higher linear orders $p_{lin} > p$. We call these methods linear/nonlinear (or LNL) methods. The main observation is that the SSP coefficients of the methods that have higher nonlinear order is not significantly lower than those of the methods with nonlinear order $p = 2$. In fact, there is no difference at all between the SSP coefficient of methods with $p = 2$ and those with $p = 3$. If $p = 4$, the SSP coefficients are somewhat lower than the corresponding $p = 2$ methods for smaller s and p_{lin}, but as we increase the number of stages and the linear order these differences go away. Table 1 (right) gives the SSP coefficients of the $p = 4$ LNL methods up to $s = 12$ stages and linear order $p_{lin} = 12$. In boldface are the coefficients that equal those of the $p = 2$ methods. The conclusion we draw from these methods is that if one wants higher linear order without compromising nonlinear order, the cost in terms of SSP coefficient may be insignificant.

3 SSP Multistep Methods

The idea behind multistep methods is to use the solution at previous time-steps rather than intermediate stages to attain higher order. Compared to Runge–Kutta methods, multistep methods typically have larger storage requirements. However, the cost of computation per step is lower, and multistep methods are sometimes advantageous for certain typed of problems. Like Runge–Kutta methods, multistep methods can often be decomposed into convex combinations of forward Euler steps, and so may preserve the strong stability properties satisfied by the forward Euler method, perhaps under a different time-step restriction.

3.1 Explicit Multistep Methods

Explicit multistep methods have a unique form, which simplifies the study of their SSP properties: any explicit k step multistep method takes the form [32],

$$u^{n+1} = \sum_{i=1}^{k} \left(\alpha_i u^{n+1-i} + \Delta t \beta_i F(u^{n+1-i}) \right). \tag{6}$$

For this method to be consistent we require that $\sum_{i=1}^{k} \alpha_i = 1$. This method is of order p if the coefficients satisfy the consistency condition and the order conditions

$\sum_{i=1}^{k} i^q \alpha_i = q \sum_{i=1}^{k} i^{q-1} \beta_i$ for $q = 1, \ldots, p$. These order conditions are the only requirement for the method to be of order p whether the method is applied to linear or nonlinear problems.

It is easy to see that multistep methods can be written as convex combinations of forward Euler methods whenever the coefficients α_i and β_i are nonnegative and $\beta_i = 0$ whenever $\alpha_i = 0$:

$$u^{n+1} = \sum_{i=1}^{k} \left(\alpha_i u^{n+1-i} + \Delta t \beta_i F(u^{n+1-i}) \right) = \sum_{i=1}^{k} \alpha_i \left(u^{n+1-i} + \Delta t \frac{\beta_i}{\alpha_i} F(u^{n+1-i}) \right).$$

Under these conditions on the coefficients, these methods preserve any strong stability properties satisfied by the forward Euler time-stepping (2) for any time step (3), in the sense that we will have

$$\|u^{n+1}\| \leq \max\{\|u^n\|, \|u^{n-1}\|, \ldots, \|u^{n+1-k}\|\} \tag{7}$$

under the modified time-step restriction (4). where the SSP coefficient $\mathscr{C} = \min_i \frac{\alpha_i}{\beta_i}$. As before, our goal is to find multistep methods that are optimal in the sense of allowable time step, i.e. those that have the largest SSP coefficient.

As mentioned above, all explicit general linear methods have a bound on the SSP coefficient $\mathscr{C} \leq 1$ [9], and of course multistep methods are included in this bound. This bound is not tight: it has been shown in [26] that the SSP coefficient of an s step explicit linear multistep method of order $p > 1$ has $\mathscr{C} \leq \frac{k-p}{k-1}$. A class of methods that attains this bound is the family of optimal $k > 2$ step second order methods, given by the coefficients [26]

$$\alpha_1 = \frac{(k-1)^2 - 1}{(k-1)^2}, \qquad \alpha_k = \frac{1}{(k-1)^2}, \qquad \beta_1 = \frac{k}{k-1}$$

(any unlisted coefficients take the value zero). Furthermore, the third order method with $k = 5$

$$u^{n+1} = \frac{25}{32} u^n + \frac{25}{16} \Delta t F(u^n) + \frac{7}{32} u^{n-4} + \frac{5}{16} \Delta t F(u^{n-4}). \tag{8}$$

also attains the bound with SSP coefficient $\mathscr{C} = \frac{5-3}{5-1} = \frac{1}{2}$.

For low order and few steps, the SSP multistep methods have similar effective SSP coefficients as the corresponding low order and few stage Runge–Kutta methods. However, when we look at Runge–Kutta methods with many stages we observe that they have much better effective SSP coefficients than linear multistep methods with many steps. For example, the optimal six step fourth order method in [18, 19] has SSP coefficient $\mathscr{C} = \mathscr{C}_{\text{eff}} = 0.1648$ which is significantly less than the

bound of $\frac{2}{5}$, and compares very poorly to the five-stage fourth order SSP Runge–Kutta method with $\mathscr{C}_{\text{eff}} = 0.302$ and to the ten-stage fourth order SSP Runge–Kutta method with $\mathscr{C}_{\text{eff}} = 0.6$.

An advantage of explicit SSP multistep methods is that they do not suffer from the restrictive $p \leq 4$ order barrier that explicit SSP Runge–Kutta methods are subject to. However, these methods require a sufficient number of steps to attain the desired order: a multistep method of order $p > 1$ with positive SSP coefficient requires $k > p$ steps [9]. If one requires higher order accuracy and the SSP property, multistep methods may still be of interest. Ketcheson developed a fast algorithm for finding optimal SSP multistep of any number of steps and order [18, 19], and used this to obtain methods of up to 50 steps and order $p = 15$. The SSP coefficients of these methods are given in [9], and show that as we go to higher order, the SSP coefficients become much smaller than the bound, as seen in the fourth order method above. Also, the number of steps required increase dramatically: for instance, the minimum number of steps required for a tenth order SSP method is 22—far more than the minimum of 12 one would expect from the theory. For this reason, SSP multistep methods of order $p > 4$ are limited in their utility.

However, if specific starting procedures are used with the multistep methods and the strong stability property is relaxed, and one relaxes the SSP criteria and requires only boundedness, it is possible to create more useful multistep methods. Hundsdorfer et al. [16] examined the required step size for a boundedness property of the form

$$||u^n|| \leq M||u^0||$$

(where M depends only on the starting procedure) to hold for several multistep methods with particular starting procedures. In [15, 30], such boundedness preserving multistep methods of up to sixth order were given with reasonably large time step coefficients, for example a three step, third order method with $\mathscr{C} = 0.537$ and a four step, fourth order method with $\mathscr{C} = 0.458$.

3.2 Implicit Multistep Methods

Implicit multistep methods have the form

$$u^{n+1} = \sum_{i=1}^{k} \left(\alpha_i u^{n+1-i} + \Delta t \beta_i F(u^{n+1-i})\right) + \Delta t \beta_0 F(u^{n+1}). \tag{9}$$

If for $i \geq 1$ the coefficients $\alpha_i \geq 0$ and $\beta_i \geq 0$ where $\beta_i = 0$ whenever $\alpha_i = 0$ then this method will satisfy the SSP condition (7) for any time step that satisfies (4) with $\mathscr{C} = \min_{i=1,\ldots,k} \frac{\alpha_i}{\beta_i}$. Unfortunately, it has been proved that the maximal SSP

coefficient for implicit multistep methods of order $p > 1$ is no greater than two [15, 27]. This bound is in fact attained by the second order trapezoidal method, which uses only one function evaluation and has SSP coefficient $\mathscr{C} = 2$. If we compare the number of function evaluations with that of the two-stage second order explicit SSP Runge–Kutta method, which requires two function evaluations and has SSP coefficient $\mathscr{C} = 1$, we see that the explicit SSP Runge–Kutta method requires four relatively inexpensive explicit function evaluations per unit time while the trapezoid method requires one costly implicit solve of a system of equations. When we compare these costs, we see that the implicit method is not competitive, despite the seemingly larger SSP coefficient. Relaxing the strong stability condition and enforcing specific starting procedures do not help in overcoming the bound $\mathscr{C} \leq 2$ for $p > 1$ [15, 30] . For this reason, implicit SSP multistep methods are not as computationally efficient as explicit multistep methods: the bound $\mathscr{C} \leq 2$ is typically too restrictive to overcome the additional cost of solving an implicit system.

4 Explicit SSP Multistep Runge–Kutta Methods

As we saw above, SSP methods suffer from restrictive bounds on the size of the SSP coefficient and barriers on the order of the method. In particular, we observe that explicit Runge–Kutta methods with positive SSP coefficient cannot be more than fourth-order accurate [25, 31], while explicit SSP linear multistep methods of high-order accuracy require many steps, and therefore have large storage requirements [9, 26].

These constraints have inspired the investigation of explicit methods that have multiple steps and multiple stages in the hopes of attaining higher-order SSP methods with large effective SSP coefficients. These multistep Runge–Kutta (MSRK) methods have been considered in multiple works. In [8] Gottlieb et al. considered a class of two-step, two-stage methods. Spijker [34] developed a complete theory for strong stability preserving multi-step multi-stage methods and presented new second order and third order methods with optimal SSP coefficients. Constantinescu and Sandu [3] focused their search on MSRK methods with up to four stages and four steps of order $p \leq 4$. Huang [14] studied two-stage multistep methods, and found methods of up to seventh order with reasonable SSP coefficients. This work showed that the order barrier that Runge–Kutta methods suffer from, and the low SSP coefficients characteristic of higher order multistep methods, can both be alleviated by a combination of multiple steps and multiple stages. More recently, SSP MSRK methods with order as high as 12 have been developed in [29] and numerous similar works by the same authors, using sufficient conditions for monotonicity and focusing on a single set of parameters in each work.

In [1, 22], we studied MSRK methods with k steps and s stages:

$$
\begin{aligned}
y_1^n &= u^n \\
y_i^n &= \sum_{l=1}^{k} d_{il} u^{n-k+l} + \Delta t \sum_{l=1}^{k-1} \hat{a}_{il} F(u^{n-k+l}) + \Delta t \sum_{j=1}^{i-1} a_{ij} F(y_j^n) \quad 2 \le i \le s \\
u^{n+1} &= \sum_{l=1}^{k} \theta_l u^{n-k+l} + \Delta t \sum_{l=1}^{k-1} \hat{b}_l F(u^{n-k+l}) + \Delta t \sum_{j=1}^{s} b_j F(y_j^n).
\end{aligned}
$$

Here the values u^{n-k+j} denote the previous steps and y_j^n are intermediate stages used to compute the next solution value u^{n+1}. Spijker's theory (including necessary and sufficient conditions for monotonicity) was generalized to these MSRK methods and used to develop an optimization algorithm for explicit SSP methods of any number of steps k and stages s, and up to order $p = 10$. These works found optimized explicit MSRK methods of up to five steps, eight stages, and tenth order. The most useful methods fifth order are the $(s, k, p) = (3, 4, 5)$ method with $\mathscr{C}_{\text{eff}} = 0.33$ and the $(s, k, p) = (7, 2, 5)$ method with $\mathscr{C}_{\text{eff}} = 0.418$. For sixth order, the $(s, k, p) = (5, 3, 6)$ method with $\mathscr{C}_{\text{eff}} = 0.272$ is a good choice, or if one is willing to incur the additional storage cost of five steps the $(s, k, p) = (6, 5, 6)$ method with $\mathscr{C}_{\text{eff}} = 0.345$ is more efficient. The recommended seventh order methods are $(s, k, p) = (7, 3, 7)$ with $\mathscr{C}_{\text{eff}} = 0.243$ or, for the cost of an additional step the $(s, k, p) = (7, 4, 7)$ method with $\mathscr{C}_{\text{eff}} = 0.286$. The eighth order method $(s, k, p) = (8, 3, 8)$ is a good method, with $\mathscr{C}_{\text{eff}} = 0.1$, but increasing the number of stages by one and the number of steps by two yields a $(s, k, p) = (9, 5, 8)$ with more than double allowable time step, a $\mathscr{C}_{\text{eff}} = 0.229$. Finally, among the ninth and tenth order methods there are fewer to choose, with two good options being the $(s, k, p) = (9, 4, 9)$ method with $\mathscr{C}_{\text{eff}} = 0.1766$ and the $(s, k, p) = (20, 3, 10)$ method with $\mathscr{C}_{\text{eff}} = 0.0917$. The methods' SSP coefficients and coefficients (d_{il}, a_{il} etc.) can be found in [10].

The results of studies of explicit MSRK methods show that these methods allow higher order than explicit SSP Runge–Kutta methods while featuring larger SSP coefficients than the multistep methods of corresponding order. In [22], we proved an order barrier of eight for two-step methods and in [1] we showed that this barrier is broken for three-step methods. We also proved [1] an upper bound on the SSP coefficient of explicit MSRK methods of order two and above with $k \ge 2$ steps and $s \ge 1$ stages:

$$
\mathscr{C} \le \frac{(k-2)s + \sqrt{(k-2)^2 s^2 + 4s(s-1)(k-1)}}{2(k-1)}.
$$

We note that we also investigated implicit MSRK methods, but found (numerically) that even for second order we can obtain effective SSP coefficients no bigger

than two: $\mathscr{C}_{\text{eff}} \leq 2$. As mentioned above, this step size is not large enough to overcome the cost of the solver required for the implicit method.

5 Methods with Downwinding

In the initial presentation of SSP methods in [32, 33], it was suggested that the inclusion of downwinding in SSP time-stepping methods can be of benefit. Downwind operators approximate the same operator to the same order of accuracy as the upwind operator, the only difference is in the choice of upwinding direction. If F is a discretization of $-f(U)_x$ in the PDE that satisfies the forward Euler condition (2) under time-step restriction (3), then we can design a downwind spatial discretization, denoted by $-\tilde{F}$, which approximates $f(U)_x$ such that the backwards in time method $u^{n+1} = u^n - \Delta t \tilde{F}(u^n)$ is strongly stable $\|u^n - \Delta t \tilde{F}(u^n)\| \leq \|u^n\|$ under the same time step restriction (4). Typically, if the stable approximation F has a left-biased stencil, then the stable approximation $-\tilde{F}$ would have a right-biased stencil.

The idea of including a downwind operator is not new: it was used in the MacCormack scheme presented in 1969 [28]. In our case the inclusion of a downwind operator allows us to relax the restrictions on the coefficients of the SSP schemes and so alleviate some of the order barriers and time-step restrictions associated with SSP methods. If we allow the coefficients $\beta_{i,j}$ in Runge–Kutta methods (or β_j in multistep methods) to become negative, we can still have the SSP property hold with SSP coefficient $\tilde{C} = \min_{i,j} \frac{\alpha_{i,j}}{|\beta_{i,j}|}$ (where the ratio is understood as infinite if $\beta_{i,j} = 0$), as long as we replace the upwind operator F with the downwind spatial discretization $\tilde{F}$ whenever the coefficient is negative.

Allowing negative coefficients does not alleviate the bound $\mathscr{C}_{\text{eff}} \leq 1$ for explicit Runge–Kutta methods, but it does allow us to break the order barrier and obtain methods of order $p > 4$ [20]. Furthermore, implicit Runge–Kutta methods with downwinding break the (observed) bound $\mathscr{C}_{\text{eff}} \leq 2$. This is apparent in the method presented in [20]

$$y_1 = \frac{2}{r(r-2)} u^n + \frac{2}{r}\left(y_1 + \frac{\Delta t}{r} F(y_1)\right) + \frac{r^2 - 4r + 2}{r(r-2)}\left(y_2 - \frac{\Delta t}{r}\tilde{F}(y_2)\right)$$

$$y_2 = y_1 + \frac{\Delta t}{r} F(y_1), \qquad u^{n+1} = y_2 + \frac{\Delta t}{r} F(y_2)$$

that has an SSP coefficient $\mathscr{C} = r$ for any choice of r. Higher order methods ($p = 3, 4, 5$) that break the bound have also been found [2]. The downwind operators adds some diffusion to the numerical solution: a fact that can readily understood when looking at the difference between the upwind and downwind operators. However, for larger time-steps this diffusion is better than that of the backward Euler method, and is the typical price of generous strong stability properties.

6 Open Problems and Future Directions

Ongoing research in the field of SSP time stepping methods focuses on the search for higher order methods with largest allowable time-steps. Implicit methods with downwinding hold promise in this regard, and methods that break the SSP barrier and the order barrier of $p \leq 6$ are currently sought. Methods that attack different components of the ODE differently (e.g. additive methods and IMEX methods) are also a major area of interest. Furthermore, there is ongoing interest in determining the bounds and barriers of different methods theoretically: for example, a proof that for implicit nonlinear Runge–Kutta methods the effective SSP coefficient does not exceed two would be a major accomplishment. Finally, the search for a meaningful SSP definition outside the method-of-lines framework is an interesting new research area, that will expand the use of SSP time stepping methods in several directions.

Acknowledgements This work was supported by AFOSR grant FA-9550-12-1-0224.

References

1. C. Bresten, S. Gottlieb, Z. Grant, D. Higgs, D.I. Ketcheson, A. Nemeth, Explicit strong stability preserving multistep Runge–Kutta methods. Math. Comput. (Accepted)
2. S. Conde, S. Gottlieb, D. Ketcheson, Implicit SSP Runge–Kutta methods with downwind operators (in preparation)
3. E. Constantinescu, A. Sandu, Optimal explicit strong-stability-preserving general linear methods. SIAM J. Sci. Comput. **32**(5), 3130–3150 (2010)
4. L. Ferracina, M.N. Spijker, Stepsize restrictions for the total-variation-diminishing property in general Runge-Kutta methods. SIAM J. Numer. Anal. **42**, 1073–1093 (2004)
5. L. Ferracina, M.N. Spijker, An extension and analysis of the Shu-Osher representation of Runge-Kutta methods. Math. Comput. **249**, 201–219 (2005)
6. S. Gottlieb, L.J. Gottlieb, Strong stability preserving properties of Runge-Kutta time discretization methods for linear constant coefficient operators. J. Sci. Comput. **18**, 83–109 (2003)
7. S. Gottlieb, C.-W. Shu, Total variation diminishing Runge-Kutta schemes. Math. Comput. **67**, 73–85 (1998)
8. S. Gottlieb, C.-W. Shu, E. Tadmor, Strong stability preserving high-order time discretization methods. SIAM Rev. **43**, 89–112 (2001)
9. S. Gottlieb, D.I. Ketcheson, C.-W. Shu, *Strong Stability Preserving Runge–Kutta and Multistep Time Discretizations* (World Scientific Press, Singapore, 2011)
10. S. Gottlieb, D. Higgs, D.I. Ketcheson, Strong stability preserving site (2013). http://www.sspsite.org/msrk.html
11. S. Gottlieb, Z. Grant, D. Higgs, Optimal explicit strong stability preserving Runge–Kutta methods with high linear order and optimal nonlinear order. Math. Comput. **84**, 2743–2761 (2015)
12. I. Higueras, On strong stability preserving time discretization methods. J. Sci. Comput. **21**, 193–223 (2004)
13. I. Higueras, Representations of Runge-Kutta methods and strong stability preserving methods. SIAM J. Numer. Anal. **43**, 924–948 (2005)

14. C. Huang, Strong stability preserving hybrid methods. Appl. Numer. Math. **59**(5), 891–904 (2009)
15. W. Hundsdorfer, S.J. Ruuth, On monotonicity and boundedness properties of linear multistep methods. Math. Comput. **75**(254), 655–672 (2005)
16. W. Hundsdorfer, S.J. Ruuth, R.J. Spiteri, Monotonicity-preserving linear multistep methods. SIAM J. Numer. Anal. **41**, 605–623 (2003)
17. D.I. Ketcheson, Highly efficient strong stability preserving Runge-Kutta methods with low-storage implementations. SIAM J. Sci. Comput. **30**(4), 2113–2136 (2008)
18. D.I. Ketcheson, Computation of optimal monotonicity preserving general linear methods. Math. Comput. **78**, 1497–1513 (2009)
19. D.I. Ketcheson, High order strong stability preserving time integrators and numerical wave propagation methods for hyperbolic PDEs. Ph.D. Thesis, University of Washington, 2009
20. D.I. Ketcheson, Step sizes for strong stability preservation with downwind-biased operators. SIAM J. Numer. Anal. **49**(4), 1649–1660 (2011)
21. D.I. Ketcheson, C.B. Macdonald, S. Gottlieb, Optimal implicit strong stability preserving Runge-Kutta methods. Appl. Numer. Math. **52**(2), 373–392 (2009)
22. D.I. Ketcheson, S. Gottlieb, C.B. Macdonald, Strong stability preserving two-step Runge-Kutta methods. SIAM J. Numer. Anal. **49**, 2618–2639 (2012)
23. D.I. Ketcheson, M. Parsani, A.J. Ahmadia, Rk-opt: software for the design of Runge–Kutta methods, version 0.2. https://github.com/ketch/RK-opt
24. J.F.B.M. Kraaijevanger, Absolute monotonicity of polynomials occurring in the numerical solution of initial value problems. Numer. Math. **48**, 303–322 (1986)
25. J.F.B.M. Kraaijevanger, Contractivity of Runge-Kutta methods. BIT **31**, 482–528 (1991)
26. H.W.J. Lenferink, Contractivity-preserving explicit linear multistep methods. Numer. Math. **55**, 213–223 (1989)
27. H.W.J. Lenferink, Contractivity-preserving implicit linear multistep methods. Math. Comput. **56**, 177–199 (1991)
28. R.W. MacCormack, The effect of viscosity in hypervelocity impact cratering. AIAA Paper, pp. 69–354 (1969)
29. T. Nguyen-Ba, H. Nguyen-Thu, R. Vaillancourt, Strong-stability-preserving, k-step, 5-to 10-stage, Hermite-Birkhoff time-discretizations of order 12. Am. J. Comput. Math. **1**, 72–82 (2011)
30. S.J. Ruuth, W. Hundsdorfer, High-order linear multistep methods with general monotonicity and boundedness properties. J. Comput. Phys. **209**, 226–248 (2005)
31. S.J. Ruuth, R.J. Spiteri, Two barriers on strong-stability-preserving time discretization methods. J. Sci. Comput. **17**, 211–220 (2002)
32. C.-W. Shu, Total-variation diminishing time discretizations. SIAM J. Sci. Stat. Comput. **9**, 1073–1084 (1988)
33. C.-W. Shu, S. Osher, Efficient implementation of essentially non-oscillatory shock-capturing schemes. J. Comput. Phys. **77**, 439–471 (1988)
34. M.N. Spijker, Stepsize conditions for general monotonicity in numerical initial value problems. SIAM J. Numer. Anal. **45**, 1226–1245 (2007)
35. R.J. Spiteri, S.J. Ruuth, A new class of optimal high-order strong-stability-preserving time discretization methods. SIAM J. Numer. Anal. **40**, 469–491 (2002)

Solving PDEs with Hermite Interpolation

Thomas Hagstrom and Daniel Appelö

Abstract We examine the use of Hermite interpolation, that is interpolation using derivative data, in place of Lagrange interpolation to develop high-order PDE solvers. The fundamental properties of Hermite interpolation are recalled, with an emphasis on their smoothing effect and robust performance for nonsmooth functions. Examples from the CHIDES library are presented to illustrate the construction and performance of Hermite methods for basic wave propagation problems.

1 Introduction

Polynomials are the workhorse for approximating the solution to general PDE's—indeed, using Taylor expansions, it is clear that convergence of a method at high order with grid refinement is equivalent to it being at least approximately exact for polynomial solutions of high degree. Thus both high order finite difference methods and nodal spectral element methods are typically constructed using Lagrange interpolants. However, the two classes of method are obviously distinct in the way the polynomials are used—for difference methods they are implicitly reconstructed at each grid point via the difference formulas, while for element based approaches they are defined and used in a finite region. An advantage of the element-based interpolants is the possibility to directly use properties of the PDE to guarantee stability, as in the standard continuous and discontinuous Galerkin frameworks [11, 18], as well as the localization of much of the computational effort. A disadvantage, however, is the fact that high-degree polynomials can support

T. Hagstrom (✉)
Southern Methodist University, Dallas, TX, USA
e-mail: thagstrom@smu.edu

D. Appelö
The University of New Mexico, Albuquerque, NM, USA
e-mail: appelo@unm.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_3

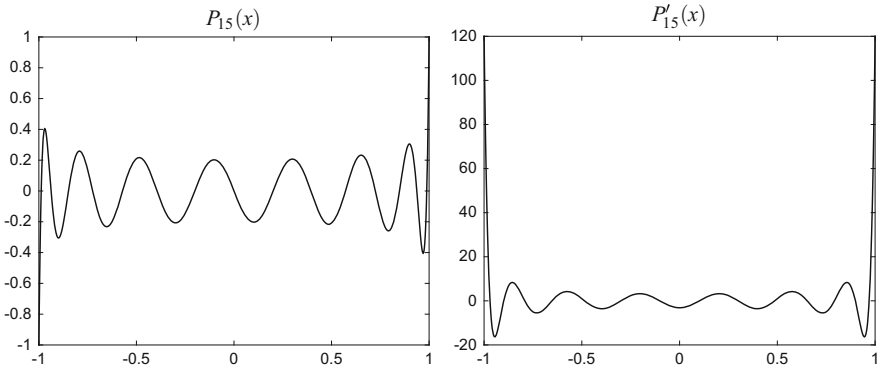


Fig. 1 Plot of the Legendre polynomial, $P_{15}(x)$ and its first derivative. Note that the maximum value is 1 and the maximum derivative value is 120—see (1)

boundary layers at element edges, as illustrated by the plot of the degree 15 Legendre polynomial and its derivative in Fig. 1.

This boundary layer phenomenon is encapsulated in the inequalities of Bernstein and Markov [5]:

Theorem 1 *Let $q(x)$ be a polynomial of degree n. Then for $-1 \le x \le 1$*

$$\left| \frac{dq}{dx} \right| \le \min\left(\frac{n}{\sqrt{1-x^2}}, n^2 \right) \|q\|_{L^\infty([-1,1])}. \tag{1}$$

The practical consequence of (1) and its generalization to multidimensional elements [14] is that differentiation matrices built from polynomials of degree n must have first derivative matrices whose norm scales like $\frac{n^2}{H}$, where H is the element width. Given that the element contains $n+1$ Lagrange nodes, this is a factor of n worse than the scaling of finite difference formulas with a comparable node density, leading to an artificially stiff semidiscretization. If second derivatives are present the situation becomes more extreme. Although approaches based on mappings (to produce nonpolynomial bases, e.g. [21]) or filtering [26] can be used, the fundamental fact is that a nonstiff polynomial differentiation matrix can only be based on differentiation near the element center.

Motivated by these facts, and in addition by the inherent stability properties detailed below, we propose the use of **Hermite interpolation** in place of Lagrange interpolation to construct high-order polynomial elements. That is, rather than using function values distributed throughout an element as the basic degrees-of-freedom, we use function and derivative values, or equivalently the coefficients of the Taylor polynomial, centered at an interior point. In many aspects the resulting methods enjoy the advantages of both finite difference and finite element discretizations:

1. Degree-independent stability constraints—with sufficiently accurate local time-stepping for hyperbolic problems all degrees-of-freedom in an element can be

updated independent of neighboring elements over a time step limited only by domain-of-dependence requirements.
2. Stability based on continuous energy estimates.
3. Highly localized evolution of many degrees-of-freedom.

Below, through simple examples, we will illustrate the basics of a PDE solver built on Hermite interpolation. The examples are implemented as Matlab programs and freely available as part of the CHIDES[1] library `chides.org`. Our initial release of CHIDES will contain, besides the Matlab implementation of the examples discussed here, various subroutines and drivers written in modern FORTRAN illustrating and enabling the construction of Hermite PDE solvers on structured meshes. We plan future releases including more complex capabilities such as coupling with DG methods on hybrid grids, as well as implementations on overset grids.

2 Hermite Interpolation

Theorem 2 (Hermite interpolation (Dahlquist and Björk [12])) *Let $\{x_i\}_{i=1}^s$ be s distinct points. Let $f(x)$ be a function defined and with derivatives up to order m_i at x_i. Then there exists a unique polynomial $p(x)$ of degree $\leq r-1$, where $r = \sum_{i=1}^s (m_i+1)$ solving the **Hermite interpolation problem**:*

$$\left.\frac{d^j p(x)}{dx_j}\right|_{x=x_i} = \left.\frac{d^j f(x)}{dx_j}\right|_{x=x_i}, \quad j=0,\ldots,m_i, \quad i=1,\ldots,s. \tag{2}$$

2.1 Piecewise Interpolation

Now consider the special form of Hermite interpolation used in CHIDES—namely piecewise interpolation using two nodes with m derivatives at each. Suppose $x_0 < x_1 < \ldots < x_N$. On an interval (x_{i-1}, x_i) we independently compute an interpolant, $p_i(x)$, of degree $2m+1$, satisfying (2) with $m_{i-1} = m_i = m$. The global piecewise interpolant we denote by:

$$\mathscr{I}_m f = p_i(x), \quad x \in (x_{i-1}, x_i). \tag{3}$$

Note that $\mathscr{I}_m f \in C^m$. We also employ piecewise degree $2m+1$ interpolation on a dual grid consisting of nodes $x_{i+1/2} = (x_i + x_{i+1})/2$ and define

$$\tilde{\mathscr{I}}_m f = p_{i+1/2}(x), \quad x \in (x_{i-1/2}, x_{i+1/2}), \tag{4}$$

[1] **C**harles **H**ermite **I**nterpolation **D**ifferential **E**quation **S**olver.

Table 1 A generalized Newton divided difference table

x_{i-1}	$f(x_{i-1})$					
x_{i-1}	$f(x_{i-1})$	$f^{(1)}(x_{i-1})/1!$	$f^{(2)}(x_{i-1})/2!$	$\star$		
x_{i-1}	$f(x_{i-1})$	$f^{(1)}(x_{i-1})/1!$	$\star$	$\star$	$\star$	
x_i	$f(x_i)$	$\star$	$\star$	$\star$	$\star$	$\star$
x_i	$f(x_i)$	$f^{(1)}(x_i)/1!$	$f^{(2)}(x_i)/2!$	$\star$		
x_i	$f(x_i)$	$f^{(1)}(x_i)/1!$				

with $p_{i+1/2}(x)$ being the solution to the Hermite interpolation problem with data consisting of derivatives through order m at $x_{i\pm 1/2}$.

2.2 Newton Form

The cellwise Hermite interpolation problem is solved repeatedly during each time step, and its cost is the dominant cost for linear systems with constant coefficients. An efficient way to solve to (2) is to form the generalized divided difference table used to find the interpolating Newton polynomial. We form the Newton table by first filling in $f^{(s)}(x_{i-1})/s!$ and $f^{(s)}(x_i)/s!$, $s = 0, \ldots, m$ as illustrated in Table 1. Next we fill in the missing positions (indicated by $\star$ in Table 1) one column at a time from left to right. The interpolating polynomial, $p_i(x)$, can then be found as

$$p_i(x) = a_0 + a_1(x - x_{i-1}) + \cdots + a_{m+1}(x - x_{i-1})^{m+1} + a_{m+2}(x - x_{i-1})^{m+1}(x - x_i) + \cdots + a_{2m+1}(x - x_{i-1})^{m+1}(x - x_i)^m,$$

where $a_j, j = 0, \ldots, 2m + 1$ are the coefficients on the upper diagonal in the table.

In our PDE solvers we work with monomial basis,

$$p_i(x) = \sum_{j=0}^{2m+1} c_j x^j. \tag{5}$$

The coefficients c_j can be obtained from a_j by a fast dual Vandermonde solve [12].

2.3 Error Estimates

Detailed formulas for the error in Hermite interpolation of a smooth function are given in [4]. Precisely, for $x \in (x_{i-1}, x_i)$, the Peano representation of the local error

can be easily derived by noting that $e = f - \mathscr{I}_m f$ solves the two point boundary value problem

$$\frac{d^{2m+2}e}{dx^{2m+2}} = \frac{d^{2m+2}f}{dx^{2m+2}}, \quad \frac{d^j e}{dx^j} = 0, \quad x = x_{i-1}, x_i, \quad j = 0, \ldots m. \tag{6}$$

Thus

$$f(x) - \mathscr{I}_m f(x) = \int_{x_{i-1}}^{x_i} K_i(x, s) \frac{d^{2m+2}f}{dx^{2m+2}}(s) ds, \tag{7}$$

where the kernel K_i is the Green's function for the two-point boundary value problem (6). Indeed, the local Hermite interpolant can be characterized as the unique solution of the inhomogeneous boundary value problem

$$\frac{d^{2m+2}p_i}{dx^{2m+2}} = 0, \quad \frac{d^j p_i}{dx^j} = \frac{d^j f}{dx^j}, \quad x = x_{i-1}, x_i, \quad j = 0, \ldots m. \tag{8}$$

Simple scaling arguments combined with the transformation $x = x_{i-1} + z h_i$ then show that $e = O(h_i^{2m+2})$ where $h_i = x_i - x_{i-1}$ is the element width. We also have the formula

$$f(x) - \mathscr{I}_m f(x) = \frac{(-1)^{m+1}}{(2m+2)!}(x - x_{i-1})^{m+1}(x_i - x)^{m+1} \frac{d^{2m+2}f}{dx^{2m+2}}(\eta). \tag{9}$$

These formulas show that the error is significantly smaller near the endpoints of the interval, and allows one to compute an accurate artificial dissipation coefficient in a modified equation approximation to the discrete evolution; see [2, 19] for details.

2.4 Smoothing Properties

A fundamental feature of piecewise Hermite interpolation is the following minimization property in the H^{m+1} seminorm,

$$|w|_{m+1}^2 \equiv \int_{x_0}^{x_N} \left(\frac{d^{m+1}w}{dx^{m+1}} \right)^2 dx. \tag{10}$$

Theorem 3 *Suppose g is any function in $H^{m+1}(x_0, x_N)$ satisfying $\frac{d^j g}{dx^j}(x_i) = \frac{d^j f}{dx^j}(x_i)$, $j = 0, \ldots, m$, $i = 0, \ldots, N$. Then $|\mathscr{I}_m f|_{m+1} \leq |g|_{m+1}$.*

This result holds locally on each interval and follows from the fact that $p_i(x)$ is **orthogonal** in the H^{m+1} semi-inner product to any function $w(x)$ satisfying $\frac{d^j w}{dx^j}(x_{i-1}) = \frac{d^j w}{dx^j}(x_i) = 0, j = 0, \ldots, m$. In fact by the Pythagorean Theorem

$$|f|^2_{m+1} = |\mathscr{I}_m f|^2_{m+1} + |f - \mathscr{I}_m f|^2_{m+1}. \tag{11}$$

These smoothing results are used to prove the stability of Hermite methods and establish optimal convergence results; see [16] and the discussion below.

2.5 Application to Nonsmooth Functions

The aforementioned smoothing properties of Hermite interpolation are also beneficial when dealing with nonsmooth functions. For example consider the canonical model of a shock wave, the step function $q(x) = -\text{sign}(x)$. Let $Q(x)$ be the Hermite interpolant of degree $2m + 1$ of $q(x)$ on $x \in [-1, 1]$. It is straightforward to prove (see [2]) that $Q(x)$ is monotone and thus the total variation of $Q(x)$ is identical to the total variation of $q(x)$. The first 20 Hermite interpolants are displayed in Fig. 2. A well-known result due to Bernstein is that the sequence of Lagrange interpolation polynomials for $|x|$ at equally spaced nodes in $x \in [-1, 1]$ diverges everywhere, except at zero and the end-points. As can be seen in Fig. 2 Hermite interpolation does considerably better. In fact, one can check that the degree $2m + 1$ Hermite interpolant for $|x|$ coincides with the polynomial

$$b(x) = \sum_{k=0}^{m} \binom{2k}{k} \frac{(-1)^{k+1}(x^2 - 1)^k}{2^{2k}(2k - 1)},$$

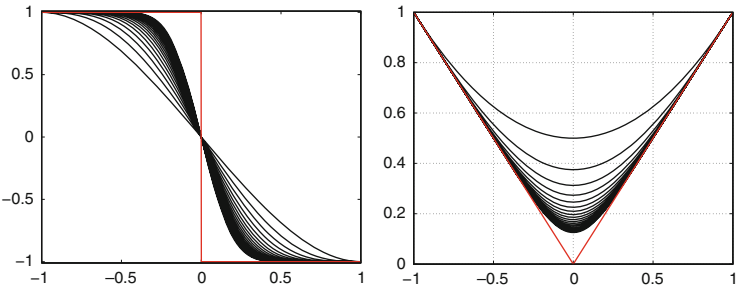


Fig. 2 Hermite interpolating polynomials of degree $2m + 1, m = 1, \ldots, 20$ of the step function $q(x) = -\text{sign}(x)$ (to the *left*) and the absolute value function $|x|$ (to the *right*)

which, in turn, is identical to the first terms of the generalized binomial expansion

$$(1+t)^{\frac{1}{2}} = \sum_{k=0}^{\infty} \binom{1/2}{k} t^k,$$

when we set $t = x^2 - 1$ (note that $|x| = (x^2)^{\frac{1}{2}}$). The sequence of Hermite interpolation polynomials thus do converge in $|x| < 1$.

3 A Hermite-Taylor Method for Solving $u_t + u_x = 0$

We now describe how the approximate solution of a PDE can be found using Hermite interpolation combined with Taylor series approximation in time. The algorithms are implemented in the Matlab files `Hermite_Taylor_1Ddriver.m`, `Advection1D_PDE.m` and `Advection1D_INIT.m` which can be downloaded from `chides.org`.

Consider the scalar advection equation with periodic boundary conditions:

$$\frac{\partial v(x,t)}{\partial t} + c\frac{\partial v(x,t)}{\partial x} = 0, \quad x \in [x_l, x_r], \quad t > 0, \tag{12}$$

$$v(x,0) = v_0(x), \quad v(x_l,t) = v(x_r,t). \tag{13}$$

The first step in our method (implemented in `Hermite_Taylor_1Ddriver.m`) is to define the primal and dual grids with $n_x + 1$ and n_x grid-points covering the computational domain

$$x_i = x_l + ih_x, \quad h_x = (x_r - x_l)/n_x, \tag{14}$$

with $i = 0, \ldots, n_x$ for the primal grid and $i = 1/2, \ldots, n_x - 1/2$ for the dual grid.

Next, we initialize the degrees-of-freedom used to describe the approximate solution, which are approximations to scaled derivatives of the solution of orders $0, \ldots, m$, or equivalently scaled coefficients of the degree-m Taylor polynomial. At $t = 0$ the piecewise degree-$2m+1$ Hermite interpolant $u(x,0)$ is determined by:

$$c_l = \frac{h_x^l}{l!} \left.\frac{d^l u(x,0)}{dx^l}\right|_{x=x_i} \approx \frac{h_x^l}{l!} \left.\frac{d^l v_0}{dx^l}\right|_{x=x_i}.$$

The data to be evolved is stored as an array of coefficients; `u(l,k,i)` holds the coefficient c_l of the kth field at the grid-point x_i. We obtain these basic degrees-of-freedom directly from the initial data. In the example in `Advection1D_INIT.m` we use $v_0(x) = \sin 20\pi x$ and may compute the coefficients directly, but in general we can find them by solving a local interpolation problem at each grid-point.

To evolve the approximate solution in time, we choose a time step Δt satisfying the CFL condition

$$c\Delta t < h_x. \tag{15}$$

Note that the degree m does not appear in this relation. We now form space-time polynomials centered at a grid-point on the dual grid and at time t_n (initially $t_n = 0$)

$$u^n_{i+\frac{1}{2}}(x,t) = \sum_{l=0}^{2m+1}\sum_{s=0}^{q} d_{ls}\left(\frac{x-x_{i+\frac{1}{2}}}{h_x}\right)^l\left(\frac{t-t_n}{\Delta t}\right)^s. \tag{16}$$

At time $t = t_n$ this expression reduces to

$$u^n_{i+\frac{1}{2}}(x,t_n) = \sum_{l=0}^{2m+1} d_{l0}\left(\frac{x-x_{i+\frac{1}{2}}}{h_x}\right)^l, \tag{17}$$

where the coefficients d_{l0} in (17) are determined so that (17) is the Hermite interpolant of the data at the adjacent primal nodes.

To find the remaining coefficients d_{ls} we repeatedly differentiate (12) in space and time:

$$\frac{\partial^{l+s} v}{\partial x^l \partial t^s} = -c\frac{\partial^{l+s} v}{\partial x^{l+1}\partial t^{s-1}}, \tag{18}$$

and insist that our approximation u satisfy (18). In particular, note that at $(x_{i+\frac{1}{2}}, t_n)$ the following relation holds

$$d_{ls} = \frac{h_x^l}{l!}\frac{\Delta t^s}{s!}\left.\frac{\partial^l\partial^s u}{\partial x^l\partial t^s}\right|_{x=x_{i+\frac{1}{2}},t=t_n}, \tag{19}$$

which together with (18) yields the recursion

$$d_{ls} = -c\frac{l+1}{s}\frac{\Delta t}{h_x}d_{l+1\,s-1},\quad l = 0,\ldots,2m+1,\ s = 1,\ldots,q = 2m+2. \tag{20}$$

Thus, the coefficients d_{ls} are updated recursively. Once the "time-derivative-coefficients" are known we can simply update the approximation at the dual grid-point at the next half time level by evaluating

$$\frac{\partial^l u^n_{i+\frac{1}{2}}}{\partial x^l}(x, t_n + \Delta t/2),\quad l = 0,\ldots,m,$$

Table 2 Error data for the evolution of $v_0(x) = \sin 20\pi x$ for different methods and final times

Final time	m	n_x	# time steps	l_2-error	Final time	m	n_x	# time steps	l_2-error
1	1	2000	2222	1.92(−6)	1000	5	20	22,222	3.89(−3)
1	5	21	23	2.04(−6)	1000	15	5	5556	9.87(−8)
1	11	6	7	3.73(−7)	1000	25	4	4444	1.16(−9)

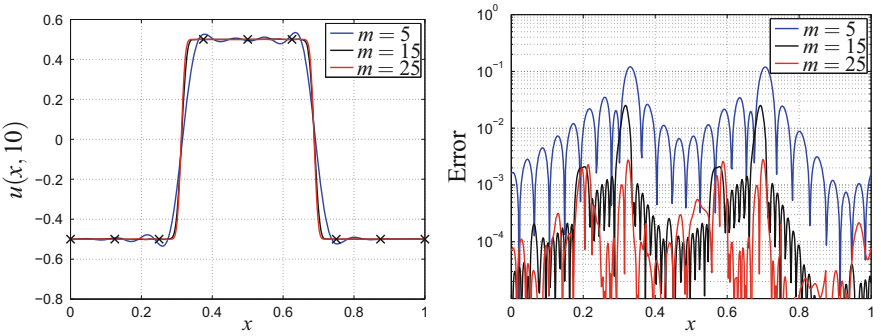


Fig. 3 *Left*: The square wave `(x>0.25).*(x<0.75)-0.5` at time 10 using $n_x = 8$ and m = 5, 15, 25. The *crosses* mark the location of the grid points. *Right*: The error at time 10 as a function of x for the three different choices of m

Repeating the procedure at the next half time level and using the periodic boundary conditions completes a full time step.

To demonstrate the method we run `Hermite_Taylor_1Ddriver.m` with the initial data $v_0(x) = \sin 20\pi x$. The computational domain is $x \in [0, 1]$; thus there are ten wavelengths inside the computational domain. We choose two different final times: 1 and 1000. This corresponds to waves traveling 10 and 10,000 wavelengths respectively. The results for some different combinations of m, n_x with the ratio $\frac{\Delta t}{h_x} = \frac{9}{10}$ are shown in Table 2. The table clearly demonstrates the benefits of using a **very high order method**; for example using a method of order 51 on a grid with 0.4 grid-points per wavelength to evolve the solution 10,000 wavelengths using only 4444 timesteps the error is $1.16 \cdot 10^{-9}$.

As a second example we evolve a square wave using $n_x = 8$ and $m = 5, 15, 25$, yielding l_2-errors: 2.99(−2), 4.66(−3) and 6.13(−4) at the final time 10. The approximation and errors are displayed in Fig. 3. As the solution is nonsmooth we cannot expect convergence at the full order of the method; see [2] where a discussion of the expected convergence behavior based on a modified equation is given. Despite this we still see a big improvement when a high order method is used.

3.1 Convergence Analysis

The analysis of convergence for the Hermite-Taylor method implemented above follows from the smoothing and convergence properties of Hermite interpolation combined with the observation that so long as (15) holds the updated Taylor polynomial at the cell center is in fact the Taylor expansion of the **exact solution** of the evolution problem over the half time step. Thus we may succinctly express the algorithm as:

$$u_h^{n+1/2} = \tilde{\mathscr{I}}_m S(\Delta t/2) u_h^n, \quad u_h^{n+1} = \mathscr{I}_m S(\Delta t/2) u_h^{n+1/2},$$

where S denotes the exact solution operator for the PDE—in this case simply translation by $c\Delta t/2$. Since S preserves the H^{m+1}-seminorm we immediately conclude from (11) that

$$|u_h^n|_{m+1} \leq |u_h^0|_{m+1}, \tag{21}$$

establishing stability.[2] We can then obtain a slightly suboptimal error estimate in the seminorm by combining (11) and the error bound obtained by taking $m+1$ derivatives of (7). Let $u(x,t)$ represent the true solution and $e^n = u - u_h^n$, $e^{n+1/2} = u - u_h^{n+1/2}$ represent the errors. Then

$$\begin{aligned} e^{n+1/2} &= S(\Delta t/2) u(\cdot, t_n) - \tilde{\mathscr{I}}_m S(\Delta t/2) u_h^n \qquad (22)\\ &= \tilde{\mathscr{I}}_m S(\Delta t/2) e^n + u(\cdot, t_{n+1/2}) - \tilde{\mathscr{I}}_m u(\cdot, t_{n+1/2}) \\ e^{n+1} &= S(\Delta t/2) u(\cdot, t_{n+1/2}) - \mathscr{I}_m S(\Delta t/2) u_h^{n+1/2} \qquad (23)\\ &= \mathscr{I}_m S(\Delta t/2) e^{n+1/2} + u(\cdot, t_{n+1}) - \mathscr{I}_m u(\cdot, t_{n+1}), \end{aligned}$$

which implies

$$|e^{n+1/2}|_{m+1}^2 \leq |e^n|_{m+1}^2 + O(h_x^{2m+2}), \quad |e^{n+1}|_{m+1}^2 \leq |e^{n+1/2}|_{m+1}^2 + O(h_x^{2m+2}).$$

Tracking these inequalities shows that $|e^n|_{m+1} = O(h_x^{m+1/2})$. In fact this argument can be refined to prove the optimal error estimate [16]:

Theorem 4 *There exists a constant, $C(T)$, independent of h_x and the initial data $u(x,0)$ such that for all $n \leq \frac{T}{\Delta t}$*

$$\|e^n\|_{L^2} \leq C h_x^{2m+1} \|u(\cdot, 0)\|_{2m+2}. \tag{24}$$

[2] We must also use the fact that the average value of the solution remains constant.

It is also shown in [16] that the result holds in general for constant coefficient symmetric hyperbolic systems in any number of space dimensions. It can also be generalized to variable coefficients and inexact time stepping so long as the local time stepping schemes are sufficiently accurate.

4 Incorporating Nonlinearity

For nonlinear PDEs it is often more efficient to use a one-step ODE solver than the Taylor series approach used above. In particular, using Taylor series requires the repeated differentiation in time of the PDE, spawning many new terms. In contrast a standard ODE solver just requires the computation of a single time derivative.

Assume we have found the Hermite interpolant at a dual grid-point $x_{i+\frac{1}{2}}$ but rather than expanding in time let the coefficients d_l be time dependent functions

$$u^n_{i+\frac{1}{2}}(x,t) = \sum_{l=0}^{2m+1} d_l(t) \left(\frac{x - x_{i+\frac{1}{2}}}{h_x} \right)^l. \tag{25}$$

For a PDE $v_t = f(v)$ we can insert (25):

$$\frac{\partial u^n_{i+\frac{1}{2}}(x,t)}{\partial t} = \sum_{l=0}^{2m+1} d'_l(t) \left(\frac{x - x_{i+\frac{1}{2}}}{h_x} \right)^l = f(u^n_{i+\frac{1}{2}}(x,t)). \tag{26}$$

As before we can differentiate in space and evaluate at $x = x_{i+\frac{1}{2}}$ to find

$$\frac{k!}{h_x^k} d'_k(t) = \frac{\partial^k}{\partial x^k} f(u^n_{i+\frac{1}{2}}(x,t)) \Big|_{x=x_{i+\frac{1}{2}}}. \tag{27}$$

To avoid the differentiation of the right hand side we first approximate $f(u^n_{i+\frac{1}{2}}(x,t))$ by a Taylor polynomial of degree $2m+1$

$$f(u^n_{i+\frac{1}{2}}(x,t)) \approx \sum_{l=0}^{2m+1} b_l(t) \left(\frac{x - x_{i+\frac{1}{2}}}{h_x} \right)^l, \tag{28}$$

for which differentiation is straightforward. With this approximation and after carrying out the differentiation in (27) we obtain the local system of ODEs

$$d'_k(t) = b_k(t), \quad k = 0, \ldots, 2m+1, \tag{29}$$

that can be solved to evolve our approximate solution. Of course, this requires us to first find the Taylor coefficients $b_k(t)$.

The precise way to compute $b_k(t)$ depends on the composition of f. For example, for the nonlinearity vv_x encountered, e.g., in Burgers' equation $v_t + vv_x = \varepsilon v_{xx}$, we may first compute the derivative

$$v_x \approx \sum_{l=1}^{2m+1} \frac{l}{h_x} d_l(t) \left(\frac{x - x_{i+\frac{1}{2}}}{h_x} \right)^l, \tag{30}$$

followed by a polynomial multiplication truncated to degree $2m+1$. This example is implemented in `Burgers1D_PDE.m` and discussed in detail below.

For more general non-linearities we can use techniques for finding recursions for Taylor series. Let $f(x)$, $w(x)$ and $u(x)$ have Taylor series around some base point with coefficients F_k, W_k and U_k. Then, for non-linearities which satisfy the differential equation $f'(x) = w(x)u'(x)$ (w is a function of f, u or both) we can directly compute the coefficients $F_k, k = 1, 2, \ldots$ using the formula [24]

$$F_k = W_0 U_k + \frac{1}{k} \sum_{j=0}^{k-1} j U_j W_{k-j}. \tag{31}$$

For example, if $f = \exp(u)$ we have $f'(x) = f(x)u'(x)$ and thus $w = f$. We start the recursion with $F_0 = \exp(U_0)$.

Thus, for general conservation laws in the form $v_t + (f(v))_x$ we may first use (31) to find a truncated Taylor series followed by differentiation [by the formula (30)].[3]

4.1 A Hermite-Runge–Kutta Solver for $v_t + vv_x = \varepsilon v_{xx}$

To make things concrete we now consider the approximate solution to viscous Burgers' equation using the approach outlined above. We evolve the local system of ODEs (29) using the classic fourth order Runge-Kutta method. The driver routine is called `Hermite_RK_1Ddriver.m` and the routines for the PDE and the initial data are `Burgers1D_PDE.m` and `Burgers1D_INIT.m`.

The nonlinearity in the PDE is handled as outlined above, we first differentiate and then perform a polynomial multiplication (in the code this is done using Matlab's built-in polynomial multiplication routine `conv`.)

The driver `Hermite_RK_1Ddriver.m` is nearly the same as the driver for the Hermite-Taylor method. As before the initial data is set up in a separate file, here in `Burgers1D_INIT.m`. As an example we choose the initial data to be $v(x, 0) = -sin(\pi x)$ on the domain $x \in [-1, 1]$ and $\varepsilon = 0.02$. This data develops into a

[3] Conservation can be enforced when we interpolate, but we have not yet experimented with this approach.

shock-like sharp transition around time 0.3 so we evaluate the error at 0.2, well before the formation time, and at 0.35, just after the shock forms. In order to maintain stability for this nonlinear problem we reduce $\frac{\Delta t}{h_x}$ to 0.1 and take a single Runge-Kutta substep. We vary the resolution using methods of order 7, 11 and 15; see the results in Table 3. The rate of convergence is not quite at the spatial design order, most likely due to the fourth order accurate time stepper.

4.2 p-Adaptivity

For problems with highly localized features it is often useful to employ adaptive methods. Methods based on Hermite interpolation can be enhanced with both p and H adaptivity and, in particular, incorporating p-adaptivity is quite straightforward. Noting that the above descriptions of the methods are local in the sense that we only require m derivatives at two adjacent nodes in order to evolve the solution a half time step, and also noting that Theorem 3 holds locally, we can allow m to vary spatially choosing $m_{\text{loc}} = \min(m_i, m_{i+1})$ when we form the Hermite interpolant at $x_{i+1/2}$. The driver routine `Padapt_Hermite_RK_1Ddriver.m` illustrates how natural it is to incorporate p-adaptivity.

Taking `mmax = 25` we compute the solution to the Burgers example using various tolerances. As can be seen from the results displayed in Table 4 the algorithm yields solutions with l_2-errors roughly at the level of the selected tolerance.

In Fig. 4 we display the solution at the end time 0.35 and the distribution of the number of derivatives used in the computation. Note that in order to meet the strict tolerance 10^{-12} we need to use $m = 25$, i.e. a method of order 51 around the shock.

5 Hermite-Taylor Methods for Systems in Multiple Dimensions

As a concrete example of a system of PDEs in multiple dimensions we consider Maxwell's equations in transverse magnetic form

$$\mu H_t^x = -E_y^z, \quad \mu H_t^y = E_x^z, \quad \varepsilon E_t^z = H_x^y - H_y^x, \tag{32}$$

on a rectangular domain $(x, y) \in [x_l, x_r] \times [y_b, y_t]$. To illustrate how simple boundary conditions can be imposed by a mirroring principle we consider the case where the boundary is a perfect electric conductor, i.e. $E_z = 0$. Then from (32) it is clear that $H^x = 0$ and $H_x^y = 0$ on $x = x_l, x_r$ and $H^y = 0$ and $H_y^x = 0$ on $y = y_b, y_t$.

Table 3 Errors at time 0.2 (left) and 0.35 (right) for Burgers equation for different order methods

nx	7	9	11	13	15
Error, $m = 3$	3.7(−3)	1.2(−3)	4.0(−4)	1.3(−4)	4.6(−5)
Rate		4.4	5.5	6.6	7.5
Error, $m = 5$	5.5(−4)	8.2(−5)	2.1(−5)	7.8(−6)	2.9(−6)
Rate		7.6	6.8	5.9	6.9
Error, $m = 7$	1.5(−4)	2.1(−5)	3.4(−6)	5.5(−7)	8.4(−8)
Rate		7.8	9.0	10.8	13.2

nx	15	35	55	75	95
$m = 3$	6.3(−2)	2.3(−2)	5.6(−3)	1.2(−3)	2.2(−4)
Rate		1.2	3.1	5.0	7.0
$m = 5$	9.6(−2)	1.2(−2)	8.5(−4)	2.1(−5)	1.5(−5)
Rate		2.4	5.9	12.0	1.4
$m = 7$	7.6(−2)	3.9(−3)	9.6(−5)	1.1(−5)	6.8(−7)
Rate		3.5	8.2	7.1	11.6

Table 4 Actual errors in the computed solutions for various tolerances for the adaptive Hermite-Runge-Kutta method applied to viscous Burgers' equation

Tol	1.0(−2)	1.0(−3)	1.0(−4)	1.0(−5)	1.0(−6)	1.0(−7)	1.0(−8)	1.0(−9)	1.0(−10)	1.0(−11)	1.0(−12)
l_2-error	1.1(−1)	8.2(−3)	1.5(−3)	1.5(−6)	1.7(−7)	1.6(−9)	3.3(−10)	1.8(−12)	4.2(−14)	1.1(−14)	1.0(−14)

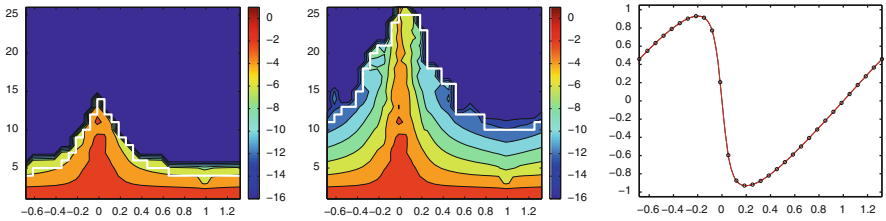


Fig. 4 In the *left* and *middle* figures the *white line* denotes the number of derivatives in the adaptive method that were used at the final time. The contour plot is the base 10 logarithm of the coefficients stored in `u`. The *left* figure corresponds to `tol = 1e-4` and the *middle* to `tol = 1e-12`. To the right the computed solution (*black line* behind the *red line*), the exact solution (*red*) and the solution at the 31 nodes (*black circles*) are displayed

The discretization of (32) is a direct generalization of the one dimensional Hermite-Taylor method on a staggered grid consisting of a primal grid:

$$(x_i, y_j) = (x_l + ih_x, y_b + jh_y), \quad (i,j) \in [0, n_x] \times [0, n_y], \quad h_x = (x_r - x_l)/n_x, \quad h_y = (y_t - y_b)/n_y,$$

and a dual grid

$$(x_{i+1/2}, y_{j+1/2}) = (x_l + (i+1/2)h_x, y_b + (j+1/2)h_y), \quad i = 0, \ldots, n_x - 1, \quad j = 0, \ldots, n_y - 1.$$

The method starts with the tensor product polynomials

$$u_{i,j,k_{\text{var}}}(x, y, t_0) = \sum_{l_x=0}^{m} \sum_{l_y=0}^{m} c_{l_x,l_y,k_{\text{var}}} \left(\frac{x - x_i}{h_x} \right)^{l_x} \left(\frac{y - y_j}{h_y} \right)^{l_y}, \tag{33}$$

where $u_{i,j,k_{\text{var}}}(x, y, t_0)$, $k_{\text{var}} = 1, 2, 3$ approximate H^x, H^y and E^z.

As in one dimension, the first step in the method is to form the Hermite interpolant at a dual node

$$u_{i+\frac{1}{2},j+\frac{1}{2},k_{\text{var}}}(x, y, t_0) = \sum_{l_x=0}^{2m+1} \sum_{l_y=0}^{2m+1} d_{l_x,l_y,k_{\text{var}}} \left(\frac{x - x_i}{h_x} \right)^{l_x} \left(\frac{y - y_j}{h_y} \right)^{l_y}. \tag{34}$$

Algorithmically, these polynomials are formed by applying the one dimensional interpolation to all y-derivatives at the bottom and top of a cell (see `get_tcofs1_2D`). and interpolating the resulting x-derivatives to the cell center.

The interpolated data is evolved by the Taylor series technique and the time derivative coefficients are computed in `Maxwell2D_PDE`. At the end of the first half time step the solution is known on all the dual nodes inside the boundary. Evolution of the approximate solution at the primal nodes inside the boundary is carried out as described above. At the primal nodes on the boundary we form the Hermite

Table 5 Errors and convergence rates for the Maxwell TM cavity problem with $\omega_x = 4\pi$, $\omega_y = 8\pi$

$m = 5$	h_x	1.0	6.7(−1)	5.0(−1)	4.0(−1)	$m = 10$	1.0	6.7(−1)	5.0(−1)	4.0(−1)
	l_2-error	2.1(0)	3.3(−1)	1.8(−2)	3.4(−3)		9.6(−4)	3.2(−7)	8.8(−10)	1.1(−11)
	Rate		4.5	10.1	7.5			19.8	20.5	19.7

interpolating polynomial by first extending the solution from interior dual nodes to ghost nodes just outside the boundary. The extension is done in such a way that the resulting interpolant is even or odd (depending on the boundary conditions, see `Maxwell2D_PDE`).

To demonstrate the method we set $\mu = 1$ and $\varepsilon = 1$, then in the cavity $(x, y) \in [-1, 1]^2$ a solution to the TM problem is

$$H^x = -\omega_y/\omega_t \sin(\omega_x x)\cos(\omega_y y)\sin(\omega_t t), \tag{35}$$

$$H^y = \omega_x/\omega_t \cos(\omega_x x)\sin(\omega_y y)\sin(\omega_t t), \tag{36}$$

$$E^z = \sin(\omega_x x)\sin(\omega_y y)\sin(\omega_t t), \tag{37}$$

with $\omega_t = \sqrt{\omega_x^2 + \omega_y^2}$.

We use this solution as initial data and evolve until time 3 and measure the error in the l_2-norm. As we choose $2\omega_x = \omega_y$ we also use $2h_x = h_y$ and set the time step as `dt = CFL*min(hx,hy)` with `CFL = 0.9`. The results, listed in Table 5, show a rate of convergence almost at the design rate.

6 Extensions and Other Work

Simulations of Compressible Flows: The first steps in constructing a compressible Navier-Stokes solver appear in the thesis of Dodson [15]. More recently we have been using the method to simulate compressible mixing layers, with an eye towards applications in aeroacoustics [1, 3, 17, 20].

Adaptive Implementations: The ease of incorporating p-adaptivity in Hermite methods is another of its attractive features, with the basic idea in one and two space dimensions explored in [7] and illustrated in the example above. We have also carried out preliminary studies of an h-adaptive version in [3]. Here we advocate quadtree/octree refinement of the Hermite cells with local time stepping. We believe that the stability of the resulting method follows directly from dissipativity of piecewise Hermite interpolation.

Dispersion and Dissipation: The dispersion and dissipation properties of Hermite methods in one and two space dimensions are studied in [2, 19, 20]. A conclusion is that the method is quite competitive in terms of cost with

other high-order structured grid discretizations, particularly if large time steps are taken. Note that the primary errors are dissipation errors which occur when the data is interpolated. Thus the method is most accurate (and most efficient) if the global time step is taken to be as large as possible while maintaining stability. Thus in the Runge-Kutta framework we suggest using as many substeps as needed to maintain accuracy and stability with a large global step.

Coupling with Other Methods: A drawback of Hermite methods is their reliance on structured grids and the need to utilize the PDE and geometry description to derive equations for normal derivatives at boundaries. To make the method more flexible we have implemented coupled Hermite-DG solvers on hybrid structured-unstructured grids [8]. Here the DG method obtains fluxes from the solution in neighboring Hermite cells, while Hermite cells bordering DG elements obtain data by interpolation. We adapt the local time stepping to the requirements of each method, so at high order we take many steps within the DG elements for each Hermite step. Using dissipative upwind DG schemes we have experimentally found the method to be quite robust.

Of course we are not the only researchers to have used Hermite interpolation to solve differential equations. Hermite-based finite element methods have been studied for quite some time, in particular for problems posed in spaces H^2 or higher [10]. Among the first applications to hyperbolic equations can be found in the work of Yabe and collaborators [27]. More recently, Nave, Rosales, and Seibold have used Hermite interpolation to solve advection problems, with a particular interest in using the Hermite-based advection solver in conjunction with level set methods [9, 23, 25]. They term the methods jet schemes borrowing terminology from differential geometry. These methods differ from the one presented here in that a staggered mesh is not employed.

Hermite interpolation has also been proposed by Butcher as a way to construct SDIRK methods for solving stiff systems of ordinary differential equations [6]. The methods are explicitly interpreted as collocation methods employing the Hermite interpolant by Mülthei [22].

To conclude we recall a quote from Davis [13]: "Hermite's formulas are rediscovered and republished every four years." We hope we have demonstrated to the reader the unique and useful properties of Hermite interpolants and their potential use for solving differential equations. We also wish to encourage the use of the codes in `chides.org` and invite any feedback for their improvement.

Acknowledgements Work of the first author was supported in part by ARO Grant W911NF-09-1-0344 and NSF Grants DMS-1418871, OCI-0904773. He also acknowledges the hospitality of the Courant Institute, where he was visiting during the preparation of the manuscript. Work of the second author was supported in part by NSF Grant DMS-1319054. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the Army Research Office or the National Science Foundation.

References

1. D. Appelö, T. Hagstrom, Experiments with Hermite methods for simulating compressible flows: Runge-Kutta time-stepping and absorbing layers, in *13th AIAA/CEAS Aeroacoustics Conference* (AIAA, 2007)
2. D. Appelö, T. Hagstrom, On advection by Hermite methods. Pac. J. Appl. Math. **4**, 125–139 (2012)
3. D. Appelö, T. Colonius, M. Inkman, T. Hagstrom, Recent progress on Hermite methods in aeroacoustics, in *17th AIAA/CEAS Aeroacoustics Conference* (AIAA, 2011)
4. G. Birkhoff, M. Schultz, R. Varga, Piecewise Hermite interpolation in one and two variables with applications to partial differential equations. Numer. Math. **11**, 232–256 (1968)
5. P. Borwein, T. Erdélyi, *Polynomials and Polynomial Inequalities* (Springer, New York, 1995)
6. J.C. Butcher, A generalization of singly-implicit formulas. BIT **21**, 175–189 (1981)
7. R. Chen, T. Hagstrom, P-adaptive Hermite methods for initial value problems. ESAIM Math. Model. Numer. Anal. **46**, 545–557 (2012)
8. R. Chen, D. Appelö, T. Hagstrom, A hybrid Hermite - discontinuous Galerkin method for hyperbolic systems with application to Maxwell's equations. J. Comput. Phys. **257**, 501–520 (2014)
9. P. Chidwagyai, J.-C. Nave, R. Rosales, B. Seibold, A comparative study of the efficiency of jet schemes. Int. J. Numer. Anal. Model.-B **3**, 297–306 (2012)
10. P. Ciarlet, *The Finite Element Method for Elliptic Problems*. Classics in Applied Mathematics, vol. 40 (SIAM, Philadelphia, 2002)
11. G. Cohen, *Higher-Order Numerical Methods for Transient Wave Equations* (Springer, New York, 2002)
12. G. Dahlquist, A. Björk, *Numerical Methods in Scientific Computing*, vol. I (SIAM, Philadelphia, 2008)
13. P. Davis, *Interpolation and Approximation*. (Dover Publications, New York, 1975)
14. Z. Ditzian, Multivariate Bernstein and Markov inequalities. J. Approx. Theory **70**, 273–283 (1992)
15. C. Dodson, A high-order hermite compressible Navier-Stokes solver. Master's thesis, The University of New Mexico, 2003
16. J. Goodrich, T. Hagstrom, J. Lorenz, Hermite methods for hyperbolic initial-boundary value problems. Math. Comput. **75**, 595–630 (2006)
17. T. Hagstrom, J. Goodrich, G. Zhu, A Hermite-Taylor algorithm for simulating subsonic shear flows, in *12th AIAA/CEAS Aeroacoustics Conference* (AIAA, 2006)
18. J. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods*. Texts in Applied Mathematics, vol. 54 (Springer, New York, 2008)
19. C.Y. Jang, T. Hagstrom, An analysis of the dispersion and dissipation properties of Hermite methods (2014, in preparation)
20. C.-Y. Jang, D. Appelö, T. Hagstrom, T. Colonius, An analysis of dispersion and dissipation properties of Hermite methods and its application to direct numerical simulation of jet noise, in *18th AIAA/CEAS Aeroacoustics Conference* (AIAA, 2012)
21. D. Kosloff, H. Tal-Ezer, A modified Chebyshev pseudospectral method with an $o(n^{-1})$ time step restriction. J. Comput. Phys. **104**, 457–469 (1993)
22. H.N. Mülthei, Maximale konvergenzordnung bei der numerischen lösung von anfangswert-problemen mit splines. Numer. Math. **39**, 449–463 (1982)
23. J.-C. Nave, R. Rosales, B. Seibold, A gradient-augmented level set method with an optimally local, coherent advection scheme. J. Comput. Phys. **229**, 3802–3827 (2010)
24. R. Neidinger, Efficient recurrence relations for univariate and multivariate Taylor series coefficients. J. Am. Inst. Math. Sci. **2013**, 587–596 (2013)
25. B. Seibold, R. Rosales, J.-C. Nave, Jet schemes for advection problems. Discrete Contin. Dyn. Syst. Ser. B **17**, 1229–1259 (2012)

26. T. Warburton, T. Hagstrom, Taming the CFL number for discontinuous Galerkin methods on structured meshes. SIAM J. Numer. Anal. **46**, 3151–3180 (2008)
27. T. Yabe, T. Ishikawa, P. Wang, T. Aoki, Y. Kadota, F. Ikeda, A universal solver for hyperbolic equations by cubic-polynomial interpolation. II. Two- and three-dimensional solvers. Comput. Phys. Commun. **66**, 233–242 (1991)

High-Order Adaptive Galerkin Methods

Claudio Canuto, Ricardo H. Nochetto, Rob Stevenson, and Marco Verani

Abstract We design adaptive high-order Galerkin methods for the solution of linear elliptic problems and study their performance. We first consider adaptive Fourier-Galerkin methods and Legendre-Galerkin methods, which offer unlimited approximation power only restricted by solution and data regularity. Their analysis of convergence and optimality properties reveals a sparsity degradation for Gevrey classes. We next turn our attention to the *hp*-version of the finite element method, design an adaptive scheme which hinges on a recent algorithm by P. Binev for adaptive *hp*-approximation, and discuss its optimality properties.

1 High-Order Adaptive Methods: Motivation

The advantages of high-order methods for problems with smooth solutions or solutions with localized singularities is documented well in the literature. The a priori error analysis started in the late 1970s whereas the a posteriori error analysis goes back to the late 1980s. We refer to the books [9, 29], to the survey paper [8] and the references therein for more details.

C. Canuto
Dipartimento di Scienze Matematiche, Politecnico di Torino, Corso Duca degli Abruzzi 24, I-10129 Torino, Italy
e-mail: claudio.canuto@polito.it

R.H. Nochetto (✉)
Department of Mathematics and Institute for Physical Science and Technology, University of Maryland, College Park, MD, USA
e-mail: rhn@math.umd.edu

R. Stevenson
Korteweg-de Vries Institute for Mathematics, University of Amsterdam, P.O. Box 94248, 1090 GE Amsterdam, The Netherlands
e-mail: r.p.stevenson@uva.nl

M. Verani
MOX-Dipartimento di Matematica, Politecnico di Milano, P.zza Leonardo Da Vinci 32, I-20133 Milano, Italy
e-mail: marco.verani@polimi.it

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_4

Despite the interest on these methods, *adaptivity* is much less developed than for the *h*-version of the FEM, for which a rather complete theory has been developed in the last decade [5, 16, 19, 25, 26, 30]. We mention [23], which proves a convergence rate (Theorem 2) under rather restrictive assumptions on the local estimator and local error; this is somewhat related to [17, 18, 27]. We also cite [1, 7, 20, 28], which prove convergence of the *hp*-FEM for practical estimators but do not derive rates of convergence. The latter is an outstanding open issue, supported by overwhelming computational evidence, but not yet accessible to analysis in full generality. The purpose of this survey is to shed light on the key issues at stake when the polynomial degree is unlimited. We devote Sects. 2–3 to the discussion of adaptive spectral methods, the so-called Fourier-Galerkin and Legendre-Galerkin methods, and next turn our attention to the *hp*-AFEM in Sects. 5–6. Our presentation is based on the recent papers [8, 10–12, 14] and on a key new idea by P. Binev for *hp*-adaptive tree approximation [2, 3], which we discuss in Sect. 4.

We highlight the difficulties associated with the analysis of adaptive high-order methods by considering a function u in a Banach space V and its best approximation with subspaces V_N of V of dimension $\leq N$:

$$E_N(u) := \inf_{\{V_N \subset V,\, \dim V_N \leq N\}} \inf_{U \in V_N} \|u - U\|.$$

We take N as a measure of the complexity of the approximation of u by N degrees of freedom; N provides information about the computational cost of constructing a best approximation $U \in V_N$ such that $E_N(u) = \|u - U\|$. Let's assume that an iterative procedure with counter $k \geq 1$ generates best approximations $U_k \in V_{N_k}$ with increasing N_k. We ask the following fundamental question:

> *How many degrees of freedom are necessary to reduce $E_{N_k}(u)$ by a fixed factor $\rho < 1$?*

It turns out that this is related to the decay of the *rearranged* coefficients of a suitable expansion of u. We assume that $E_{N_k}(u)$ exhibits a prescribed decay, either algebraic or exponential, in terms of N_k.

Algebraic Decay If $E_{N_k}(u) = AN_k^{-s}$, then a simple calculation yields

$$N_{k+1} = \rho^{-\frac{1}{s}} N_k \tag{1}$$

The new number of degrees of freedom N_{k+1} is proportional to the current one N_k, and the complexity of U_n is proportional to the last step:

$$\sum_{k=0}^{n} N_k = N_0 \sum_{k=0}^{n} \rho^{-\frac{k}{s}} \approx N_n.$$

This is what the *h*-theory of AFEM predicts [5, 16, 25, 26, 30].

Exponential Decay If $E_{N_k}(u) = Ae^{-\eta N_k}$, then a simple calculation reveals that

$$N_{k+1} - N_k = \eta^{-1} |\log \rho| \tag{2}$$

and *the number of degrees of freedom must only grow by an additive constant,* which is very hard to prove! Since $N_n = N_0 + n\eta^{-1}|\log \rho|$, the complexity of U_n is

$$\sum_{k=0}^{n} N_k = \sum_{k=0}^{n} \left(N_0 + k\eta^{-1}|\log \rho|\right) \approx nN_n.$$

Therefore, the complexity of the last step does not dominate the overall complexity, as in the algebraic case, which makes counting of degrees of freedom a very delicate matter. Even more delicate is the situation in which not all the rearranged components of u exhibit the ideal decay assumed in (2), as in the presence of *plateaux*, where a relevant number of expansion coefficients of u are constant. Then one can show that Dörfler marking adds many more frequencies, which poses further difficulties in the analysis of the exponential case. We explore these issues below.

2 Adaptive Spectral Methods

In this section we focus on the spectral analysis of elliptic PDE with emphasis on the Fourier analysis [10]. We briefly mention the Legendre approach [11]. Our goal is to describe an ideal adaptive Fourier algorithm, based on Dörfler marking, along with several more aggressive variants suitable for spectral analysis.

2.1 Elliptic PDE and Fourier Analysis

Let $d \geq 1$ and consider the following elliptic PDE in $\Omega = (0, 2\pi)^d$, with periodic boundary conditions,

$$Lu = -\nabla \cdot (\nu \nabla u) + \sigma u = f \quad \text{in } \Omega\ , \tag{3}$$

where ν and σ are sufficiently smooth coefficients satisfying $0 < \nu_* \leq \nu(x) \leq \nu^* < \infty$ and $0 < \sigma_* \leq \sigma(x) \leq \sigma^* < \infty$ in Ω; let us set $\alpha_* = \min(\nu_*, \sigma_*)$ and $\alpha^* = \max(\nu^*, \sigma^*)$. Its weak formulation reads

$$u \in H^1_p(\Omega): \quad a(u, v) = \langle f, v \rangle \qquad \forall v \in H^1_p(\Omega)\ , \tag{4}$$

where $H^1_p(\Omega)$ is the closure in $H^1(\Omega)$ of all smooth periodic functions, and the bilinear form is $a(u, v) = \int_\Omega \nu \nabla u \cdot \nabla v + \int_\Omega \sigma uv$. We denote by $|||v||| = \sqrt{a(v, v)}$

the energy norm of any $v \in H^1_p(\Omega)$, which satisfies

$$\sqrt{\alpha_*}\|v\|_{H^1_p(\Omega)} \le |||v||| \le \sqrt{\alpha^*}\|v\|_{H^1_p(\Omega)} \ . \tag{5}$$

Fourier Analysis We first introduce the trigonometric basis $\phi_k(x) = \frac{1}{(2\pi)^{d/2}} \mathrm{e}^{ik\cdot x}$ for any $k \in \mathbb{Z}^d$ and $x \in \mathbb{R}^d$. Any function $v \in L^2(\Omega)$ can be expanded in terms of $\{\phi_k\}_{k\in\mathbb{Z}^d}$ as follows:

$$v = \sum_k \hat{v}_k \phi_k \ , \qquad \hat{v}_k = \langle v, \phi_k \rangle \ , \qquad \|v\|^2_{L^2(\Omega)} = \sum_k |\hat{v}_k|^2 \ . \tag{6}$$

The space $H^1_p(\Omega)$ can now be easily characterized as the space of those $v \in L_2(\Omega)$ for which

$$\|v\|^2 = \|v\|^2_{H^1_p(\Omega)} = \sum_k |\hat{V}_k|^2 < \infty \qquad (\text{where } \hat{V}_k := c_k \hat{v}_k, \text{ with } c_k := (1+|k|^2)^{1/2}).$$

This induces an *isomorphism* between $H^1_p(\Omega)$ and $\ell^2(\mathbb{Z}^d)$: for each $v \in H^1_p(\Omega)$ let $\mathbf{v} = (\hat{V}_k)_k \in \ell^2(\mathbb{Z}^d)$ and note that $\|v\| = \|\mathbf{v}\|$. Likewise, the dual space $H^{-1}_p(\Omega) = (H^1_p(\Omega))'$ is characterized as the space of those functionals f for which $\|f\|^2 = \|f\|^2_{H^{-1}_p(\Omega)} = \sum_k |\hat{F}_k|^2$ with $\hat{F}_k := c_k^{-1}\hat{f}_k$. We also have an isomorphism between $H^{-1}_p(\Omega)$ and $\ell^2(\mathbb{Z}^d)$ upon setting $\mathbf{f} = (\hat{F}_k)_k$ for $f \in H^{-1}_p(\Omega)$ and realizing that $\|f\| = \|\mathbf{f}\|$.

We can now rewrite (4) in matrix-vector form upon introducing the bi-infinite matrix $\mathbf{A} = (a_{k,m})$, with $a_{k,m} := (c_k c_m)^{-1} a(\phi_m, \phi_k)$, that represents the bilinear form $a(\cdot,\cdot)$ in the basis $\{\phi_k\}_k$:

$$\mathbf{Au} = \mathbf{f}. \tag{7}$$

Fourier-Galerkin Approximation Given any finite set $\Lambda \subset \mathbb{Z}^d$ and corresponding subspace V_Λ, let

$$u_\Lambda \in V_\Lambda \ : \quad a(u_\Lambda, v_\Lambda) = \langle f, v_\Lambda \rangle \qquad \forall v_\Lambda \in V_\Lambda \ .$$

For any $w \in V_\Lambda$, we define the *residual*

$$r(w) = f - Lw = \sum_k \hat{r}_k(w)\phi_k \in H^{-1}_p(\Omega) \ ,$$

where $\hat{r}_k(w) = \langle f - Lw, \phi_k \rangle = \langle f, \phi_k \rangle - a(w, \phi_k)$, and let $\mathbf{r}(\mathbf{w}) = \mathbf{f} - \mathbf{Aw} \in \ell^2(\mathbb{Z}^d)$. The definition of u_Λ is equivalent to the condition

$$P^*_\Lambda r(u_\Lambda) = 0 \ , \qquad \text{i.e.,} \quad \hat{r}_k(u_\Lambda) = 0 \qquad \forall k \in \Lambda \ , \tag{8}$$

where P_Λ^* is the adjoint of the $H_p^1(\Omega)$-orthogonal projection onto Λ. This is called *Galerkin orthogonality.*

By the continuity and coercivity of the bilinear form $a(\cdot,\cdot)$, one has

$$\frac{1}{\alpha^*}\|r(u_\Lambda)\| \le \|u-u_\Lambda\| \le \frac{1}{\alpha_*}\|r(u_\Lambda)\|\,, \tag{9}$$

and also $\frac{1}{\sqrt{\alpha^*}}\|r(u_\Lambda)\| \le |||u-u_\Lambda||| \le \frac{1}{\sqrt{\alpha_*}}\|r(u_\Lambda)\|$, in light of (5). Therefore, the quantity

$$\|r(u_\Lambda)\| = \left(\sum_{k\notin\Lambda} |\hat{R}_k(u_\Lambda)|^2\right)^{1/2} = \|\mathbf{r}(\mathbf{u}_\Lambda)\|,$$

where $\hat{R}_k(u_n) := c_k^{-1}\hat{r}_k(u_n)$, is an *error estimator* from above and from below. However, this quantity is not computable because it involves infinitely many terms. We comment on Sect. 2.5 on a *feasible version*.

2.2 ADFOUR: Ideal Adaptive Fourier Algorithm

Fix any $\theta \in (0,1)$ and set $\Lambda_0 = \emptyset$, $u_{\Lambda_0} = 0$. For $n = 0, 1, \ldots,$ assume that Λ_n and $u_n := u_{\Lambda_n} \in V_{\Lambda_n}$ are already computed and choose $\Lambda_{n+1} := \Lambda_n \cup \partial\Lambda_n$ by *Dörfler's marking* (or *bulk chasing*) as

$$\|P^*_{\partial\Lambda_n} r(u_n)\| = \|P^*_{\Lambda_{n+1}} r(u_n)\| \ge \theta\|r(u_n)\| \quad \text{or} \quad \sum_{k\in\partial\Lambda_n} |\hat{R}_k(u_n)|^2 \ge \theta^2 \sum_{k\in\mathbb{Z}^d} |\hat{R}_k(u_n)|^2\,. \tag{10}$$

This can be implemented by rearranging the coefficients $\hat{R}_k(u_n)$ in decreasing order of modulus and picking the largest ones (*greedy approach*). However, this is only 'ideal' because the number of coefficients $\hat{R}_k(u_n)$ is infinite. The ideal algorithm thus reads:

ADFOUR(θ, tol)
set $r_0 := f$, $\Lambda_0 := \emptyset$, $n = -1$
do
  $n \leftarrow n+1$
  $\partial\Lambda_n :=$ **DÖRFLER**(r_n, θ)
  $\Lambda_{n+1} := \Lambda_n \cup \partial\Lambda_n$
  $u_{n+1} :=$ **GAL**(Λ_{n+1})
  $r_{n+1} :=$ **RES**(u_{n+1})
while $\|r_{n+1}\| >$ tol

The greedy approach gives sets $\partial\Lambda_n$ in (10) with minimal cardinality and is thus crucial below in the study of optimality of ADFOUR and its variants.

Theorem 1 (Contraction Property of ADFOUR) *Let* $\theta \in (0,1)$ *and let* $\{\Lambda_n, u_n\}_{n\geq 0}$ *be the sequence generated by ADFOUR. If* $\rho(\theta) = \sqrt{1-\frac{\alpha_*}{\alpha^*}\theta^2}$, *then*

$$|||u-u_{n+1}||| \leq \rho(\theta)|||u-u_n||| \ .$$

Proof Since the proof is simple but illuminating, we show it. We proceed in five steps.

- *Pythagoras orthogonality:* Since the spaces are nested $V_{\Lambda_n} \subset V_{\Lambda_{n+1}}$, the following holds

$$e_{n+1}^2 = e_n^2 - d_n^2,$$

  with $e_n := |||u-u_n|||$ and $d_n := |||u_{n+1}-u_n|||$.
- *Saturation property:* If $d_n^2 \geq \lambda e_n^2$ with $\lambda > 0$ independent of n, then $e_{n+1}^2 \leq (1-\lambda)e_n^2$. Since this is the assertion with $\rho^2 = 1-\lambda$, it remains to prove $d_n^2 \geq \lambda e_n^2$ with $\lambda = \frac{\alpha_*}{\alpha^*}\theta^2$.
- *Discrete efficiency:* In view of (9) and (8) we obtain

$$\alpha^* d_n^2 \geq \|L(u_{n+1}-u_n)\|^2 = \|r_{n+1}-r_n\|^2 \geq \|P^*_{\Lambda_{n+1}}(r_{n+1}-r_n)\|^2 = \|P^*_{\Lambda_{n+1}}r_n\|^2 \ .$$

- *Dörfler marking:* We recall (10), namely $\|P^*_{\Lambda_{n+1}}r_n\|^2 \geq \theta^2\|r_n\|^2$.
- *Upper bound:* We realize that $\|r_n\|^2 \geq \alpha_* e_n^2$ is a consequence of (9), and finally deduce $\lambda = \theta^2\frac{\alpha_*}{\alpha^*}$. □

2.3 Aggressive Versions of ADFOUR

The contraction factor ρ guaranteed for **ADFOUR** is bounded from below away of 0: $\rho(\theta) \geq \sqrt{1-\frac{\alpha_*}{\alpha^*}} > 0$. This is overly pessimistic in the context of smooth solutions, since a Fourier method allows for an exponential decay of the error as the number of (properly selected) active degrees of freedom is increased.

2.3.1 Quasi-Sparsity of A and $\mathbf{A}^{-1}$

The key to a contraction constant ρ arbitrarily close to 0 relies on the sparsity patterns of the bi-infinite Hermitian positive definite matrix $\mathbf{A}$, defined in (7), and its inverse $\mathbf{A}^{-1}$. The decay of the entries away from the diagonal depends on the

regularity of the coefficients ν and σ of L. In view of the orthogonality properties of $\{\phi_k\}_k$, only an operator L with constant coefficients ν and σ leads to a diagonal matrix $\mathbf{A}$. If ν and σ are real analytic in a neighborhood of Ω, then $a_{k,m}$ decays exponentially away from the diagonal [10]: there exist parameters $c_L, \eta_L > 0$ such that $|a_{k,m}| \leq c_L \exp(-\eta_L |k-m|)$ as $|k-m| \to \infty$. This justifies the *symmetric truncation* $\mathbf{A}_J$ of $\mathbf{A}$ with *parameter* J, defined as $(\mathbf{A}_J)_{\ell,k} = a_{\ell,k}$ if $|\ell - k| \leq J$ and $(\mathbf{A}_J)_{\ell,k} = 0$ otherwise, which satisfies

$$\|\mathbf{A} - \mathbf{A}_J\| \leq C_{\mathbf{A}} (J+1)^{d-1} \mathrm{e}^{-\eta_L J} .$$

Most notably, the inverse matrix $\mathbf{A}^{-1}$ is also quasi-sparse [10, Property 2.3]: if $c_L < \frac{1}{2}(\mathrm{e}^{\eta_L} - 1) \min_\ell a_{\ell,\ell}$, then there exist explicit constants $C_{\mathbf{A}^{-1}}$ and $\bar{\eta}_L \in (0, \eta_L]$ such that the symmetric truncation $(\mathbf{A}^{-1})_J$ of $\mathbf{A}^{-1}$ satisfies

$$\|\mathbf{A}^{-1} - (\mathbf{A}^{-1})_J\| \leq C_{\mathbf{A}^{-1}} (J+1)^{d-1} \mathrm{e}^{-\bar{\eta}_L J} .$$

2.3.2 A-ADFOUR: An Aggressive Version of ADFOUR

We can exploit the preceding quasi-sparsity of $\mathbf{A}$ and $\mathbf{A}^{-1}$ to enrich the set $\partial\Lambda_n$ obtained by Dörfler marking by a neighborhood in $\mathbb{Z}^d$ of radius $J \approx \left|\log(1-\theta^2)\right|$ around each point of $\partial\Lambda_n$. This leads to the procedure **E-DÖRFLER**(r_n, θ, J) and ensuing algorithm **A-ADFOUR**(θ, tol), instead of **ADFOUR**(θ, tol).

Theorem 2 (Contraction Property of A-ADFOUR [10]) *Let $\theta \in (0,1)$, $J = J(\theta)$, and let $\{\Lambda_n, u_n\}_{n\geq 0}$ be the sequence generated by **A-ADFOUR**. Then,*

$$|||u - u_{n+1}||| \leq \sqrt{\frac{\alpha_*}{\alpha^*}} \sqrt{1-\theta^2}\, |||u - u_n||| .$$

Note that the contraction factor $\rho(\theta) = \sqrt{\frac{\alpha_*}{\alpha^*}} \sqrt{1-\theta^2}$ can now be made arbitrarily close to 0, as desired.

2.4 Super-Aggressive Version of ADFOUR

The preceding enrichment process built in **E-DÖRFLER** can be further enhanced upon making a *dynamic* choice of parameters $\theta = \theta_n$ and $J = J_n$ for **E-DÖRFLER**. This is summarized as follows.

Theorem 3 (Quadratic Convergence [13]) *The algorithm **A-ADFOUR** with a dynamic choice of parameters θ and J, according to $\sqrt{1-\theta_n^2} \simeq \|r_n\|$ and $J_n \simeq |\log \|r_n\||$, converges quadratically $\|u - u_{n+1}\| \lesssim \|u - u_n\|^2$.*

This theorem has three important consequences. The quadratic rate is consistent with exponential convergence. Second, the cardinalities $|\partial\Lambda_n|$ grow at a geometric rate [compare with (2)]:

$$E(U_n) = Ae^{-\eta|\Lambda_n|} \quad \Rightarrow \quad |\partial\Lambda_n| = |\Lambda_{n+1}| - |\Lambda_n| = |\Lambda_n| - \eta^{-1}\log A;$$

The third issue is that the computational cost of **A-ADFOUR** *scales linearly* with $|\Lambda_n|$. This is an optimal result and a rare instance in the theory of high-order methods.

2.5 Variants of ADFOUR

The ideal situation described above deals with the error estimator $\|r_n\|$, which is *not computable* because r_n has in general ∞-many components. We now introduce a *feasible* version of **ADFOUR**: given $0 < \gamma < 1$, let $\tilde{r}_n$ be a truncated residual so that $\|r_n - \tilde{r}_n\| \leq \gamma\|\tilde{r}_n\|$. We use $\|\tilde{r}_n\|$ as error estimator and apply Dörfler marking on $\tilde{r}_n$. We replace the module **RES** by a feasible version **F-RES** which hinges on two procedures $f_\varepsilon = \textbf{F-RHS}(f,\varepsilon)$ and $w_\varepsilon = \textbf{F-APPLY}(v,\varepsilon)$. They compute a finite expansion f_ε of f and a finite truncation w_ε of Lv so that $\|f - f_\varepsilon\| \leq \varepsilon$ and $\|Lv - w_\varepsilon\| \leq \varepsilon$. The *cardinalities of* f_ε *and* w_ε depend on the sparsity class of $f = Lu$ and Lv, which happens to be different from that of u and v, for the exponential class. This surprising issue is described in Sect. 3.

A second fundamental variant of **ADFOUR**, relevant for the p-version of the FEM, is *adaptive Legendre-Galerkin methods*. In case $d = 1$, with L_k being the Legendre polynomial of degree k normalized so that $L_k(1) = 1$, we replace the trigonometric basis by the Babuška-Shen basis

$$\eta_k(x) = \sqrt{k - \tfrac{1}{2}} \int_x^1 L_{k-1}(s)ds = \frac{1}{\sqrt{4k-2}}\big(L_{k-2}(x) - L_k(x)\big) \quad k \geq 2,$$

which is orthogonal in $H_0^1(0,1)$, equipped with $|\cdot|_{H^1(0,1)}$. The methodology developed for Fourier-Galerkin methods extends to Legendre-Galerkin methods for $d = 1$ [11]. The case $d > 1$ is, however, much harder because the tensorized Babuška-Shen basis is not orthogonal in $H_0^1(\Omega)$. Despite the stiffness matrix $\mathbf{S}$ for the Laplace operator not being spectrally equivalent to a diagonal matrix, a computationally feasible change of basis gives a matrix $\mathbf{S}$ with such a property [12, 15].

3 Sparsity Classes

The notion of best N-term approximation in nonlinear approximation theory is related to the sparsity pattern of the Fourier decomposition (6). In this section we explore connections between this concept and the asymptotic decay rate of **ADFOUR** and its variants.

3.1 Best N-Term Approximation and Rearrangement

We start with an arbitrary Hilbert space V equipped with an orthonormal basis $\{\psi_\lambda : \lambda \in \mathcal{M}\}$ (a generalisation to a Riesz basis causes no difficulties). Given any finite index set $\Lambda \subset \mathcal{M}$, we define the subspace $V_\Lambda = \operatorname{span}\{\phi_\lambda \,|\, \lambda \in \Lambda\}$ of V and set $|\Lambda| = \operatorname{card}\Lambda$, so that $\dim V_\Lambda = |\Lambda|$. If $v \in V$ admits an expansion $v = \sum_\lambda \hat{v}_\lambda \psi_\lambda$, then we define its projection $P_\Lambda v$ onto V_Λ by setting $P_\Lambda v = \sum_{\lambda\in\Lambda} \hat{v}_\lambda \psi_\lambda$. The *best N-term approximation error* for $v \in V$ is

$$E_N(v) := \inf_{|\Lambda|=N} \inf_{s_\Lambda \in V_\Lambda} \|v - s_\Lambda\| \,. \tag{11}$$

We classify v according to the decay of $E_N(v)$ as follows. Given a strictly decreasing function $\phi : \mathbb{N} \to \mathbb{R}_+$ such that $\phi(N) \to 0$ as $N \to \infty$, we define the *sparsity class* $\mathcal{A}_\phi$ by setting

$$\mathcal{A}_\phi := \{v \in V \,:\, |v|_{\mathcal{A}_\phi} := \sup_N \frac{E_N(v)}{\phi(N)} < +\infty\} \,.$$

So for $v \in \mathcal{A}_\phi$ and $\varepsilon > 0$, the number of degrees of freedom that is sufficient to achieve a target tolerance ε with a best N-term approximation is $N = N_\varepsilon = \lceil \phi^{-1}(\varepsilon/|v|_{\mathcal{A}_\phi})\rceil$.

Since $\|v - P_\Lambda v\|^2 = \sum_{\lambda\notin\Lambda} |\hat{v}_\lambda|^2$, a best N-term approximation is obtained by *rearranging* the coefficients of v in decreasing order of magnitude $|v_j^*| \geq |v_{j+1}^*|$ with $v_j^* := \hat{v}_{\lambda_j}$ for $j \geq 1$, and picking up the N largest ones to define $\Lambda = \Lambda_N$ (greedy approach). Since functions $v = \sum_\lambda \hat{v}_\lambda \psi_\lambda \in V$ and vectors $\mathbf{v} = (\hat{v}_\lambda) \in \ell^2(\mathcal{M})$ are isomorphically related, the sparsity class $\mathcal{A}_\phi$ has a natural counterpart ℓ_{ϕ^*}, related to any non-increasing rearrangement $\mathbf{v}^* = (v_j^*) \in \ell^2(\mathbb{N})$ of $\mathbf{v} \in \ell^2(\mathcal{M})$: let $\phi^* : \mathbb{N} \to \mathbb{R}_+$ be a strictly decreasing function such that $\phi^*(j) \to 0$ when $j \to \infty$, and set

$$\ell_{\phi^*} = \left\{\mathbf{v} \in \ell^2(\mathcal{M}) \,:\, |\mathbf{v}^*|_{\ell_{\phi^*}} := \sup_{j\geq 1} \frac{|v_j^*|}{\phi^*(j)} < +\infty\right\} \,.$$

In the next two examples, we assume $V = H^1(\Omega)$ with Ω a domain in $\mathbb{R}^d$, and $s, \eta, \tau > 0$ are parameters.

Example 1 (Algebraic Case) For $\phi(N) = N^{-s/d}$ and $\phi^*(N) = N^{-\frac{s}{d}-\frac{1}{2}}$, we write $\mathcal{A}_\phi$ as $\mathcal{A}_B^s$ and ℓ_{ϕ^*} as ℓ_B^s. We thus have

$$E_N(v) \le N^{-s/d}|v|_{\mathcal{A}_B^s} \quad \text{i.e.,} \quad N_\varepsilon \le \left\lceil \left(\frac{|v|_{\mathcal{A}_B^s}}{\varepsilon} \right)^{d/s} \right\rceil \qquad \forall v \in \mathcal{A}_B^s.$$

The subscript B stands for *Besov*, because of the connection with Besov regularity: for sufficiently smooth ψ_λ, $v \in B_\tau^{s+1}(L^\tau(\Omega))$ if and only if $\mathbf{v}^* \in \ell^\tau(\mathbb{N})$ where $\tau = (\frac{1}{2} + \frac{s}{d})^{-1}$, the latter being slightly stronger property than $\mathbf{v} \in \ell_B^s$. If a tree constraint is imposed, as with FEM, then $v \in B_\tau^{s+1}(L^\tau(\Omega))$ with $\frac{1}{\tau} < \frac{s}{d} + \frac{1}{2}$ implies $v \in \mathcal{A}_B^s$.

Example 2 (Exponential Case) Let $\{\psi_\lambda\}$ be the trigonometric basis of Sect. 2.1. For $\phi(N) = \mathrm{e}^{-\eta N^\tau}$ and $\phi^*(N) = N^{\frac{\tau-1}{2}}\mathrm{e}^{-\eta N^\tau}$, we write $\mathcal{A}^s$ as $\mathcal{A}_G^{\eta,\tau}$ and ℓ_{ϕ^*} as $\ell_G^{\eta,\tau}$. We thus have

$$E_N(v) \le \mathrm{e}^{-\eta N^\tau}|v|_{\mathcal{A}_G^{\eta,\tau}} \quad \text{i.e.,} \quad N_\varepsilon \le \left\lceil \left(\frac{1}{\eta} \log \frac{|v|_{\mathcal{A}_G^{\eta,\tau}}}{\varepsilon} \right)^{1/\tau} \right\rceil \qquad \forall v \in \mathcal{A}_G^{\eta,\tau}.$$

The subscript G stands for *Gevrey* because of the relation with Gevrey regularity ($\tau = t/d$): if $t < 1$ function v is C^∞ and if $t \ge 1$ function v is analytic.

3.2 Sparsity of the Range of Operator L

The algebraic class $\mathcal{A}_B^s$, or equivalently the class of sequences ℓ_B^s, is a vector space. However, the exponential class $\ell_G^{\eta,\tau}(\mathbb{N})$ is not as the following counterexample for $d = 1$ reveals [10]: the sequences in $\ell^2(\mathbb{N})$

$$\mathbf{u} = (u_k) = (e^{-\eta}, 0, e^{-2\eta}, 0, e^{-3\eta}, 0, e^{-4\eta}, 0, \cdots),$$
$$\mathbf{v} = (v_k) = (0, e^{-\eta}, 0, e^{-2\eta}, 0, e^{-3\eta}, 0, e^{-4\eta}, \cdots)$$

are in $\ell_G^{\eta,1}$ because $\mathbf{u}^* = (e^{-\eta}, e^{-2\eta}, e^{-3\eta}, e^{-4\eta}, \cdots)$ and $\mathbf{v}^* = (e^{-\eta}, e^{-2\eta}, e^{-3\eta}, e^{-4\eta}, \cdots)$, but their sum $\mathbf{u} + \mathbf{v} = (e^{-\eta}, e^{-\eta}, e^{-2\eta}, e^{-2\eta}, e^{-3\eta}, e^{-3\eta}, e^{-4\eta}, e^{-4\eta}, \cdots)$ belongs to $\ell_G^{\eta/2,1} \setminus \ell_G^{\eta,1}$. Therefore, the sparsity class for $\mathbf{u} + \mathbf{v}$ *deteriorates* from $\ell_G^{\eta,1}$ to $\ell_G^{\eta/2,1}$ (the decay rate is slower and dictated by $\eta/2$ instead of η). This indicates that we cannot expect the sparsity of the range and domain of operator L to be the same.

We confirm this observation with a counterexample from [10] for $d = 1$: let $\mathbf{v} = \{v_k\}_{k\in\mathbb{Z}}$ be defined by $v_k = e^{-\eta n}$ if $k = 2p(n-1)$ and $v_k = 0$ otherwise, for $p \geq 2$ and $n \geq 1$; since $v_n^* = e^{-\eta n}$ we deduce $\mathbf{v} \in \ell_G^{\eta,1}$. Let $\mathbf{A} = (a_{ij})_{i,j=1}^{\infty}$ be the Toeplitz matrix $a_{ij} = 1$ if $|i-j| \leq q$ and $a_{ij} = 0$ otherwise. To compute $\mathbf{Av}$ we observe that for $q < p$ consecutive frequencies of $\mathbf{v}$ do not interact, and

$$(\mathbf{Av})_i = e^{-\eta n} \quad \text{if} \quad \big|i - 2p(n-1)\big| \leq q \quad \text{for some} \quad n \geq 1$$

and otherwise $(\mathbf{Av})_i = 0$. This implies that the non-increasing rearrangement of $\mathbf{Av}$ reads $(\mathbf{Av})_m^* = e^{-\eta m}$ if $(2q+1)(n-1)+1 \leq m \leq (2q+1)n$, thereby exhibiting a sequence of plateaux of size $2q$. We thus conclude that the sparsity class of $\mathbf{Av}$ is

$$\mathbf{Av} \in \ell_G^{\bar\eta,1} \quad \text{with} \quad \bar\eta = \frac{\eta}{2q+1} < \eta.$$

The situation can be worse in the sense that $\mathbf{v} \in \ell_G^{\eta,\tau}$ may yield $\mathbf{Av} \in \ell_G^{\bar\eta,\bar\tau}$ with $\bar\eta < \eta$ and $\bar\tau < \tau$ [10]. This in turn affects the cost of the feasible version **F-ADFOUR** of **ADFOUR** for the exponential case: since $\|Lv\|_{\mathcal{A}_G^{\bar\eta,\bar\tau}} \lesssim \|v\|_{\mathcal{A}_G^{\eta,\tau}}$, symmetric truncation $\mathbf{A}_J$ of the bi-infinite matrix $\mathbf{A}$ allows for a finite truncation $\tilde r_n$ of $r_n = Lu_n$ satisfying the accuracy properties of **F-APPLY** but with cardinality dictated by the pair $(\bar\eta, \bar\tau)$ rather than (η, τ). This is an essential difficulty which does not occur for the algebraic class [10]. We can partially compensate by adding a *coarsening* step $\Lambda := \mathbf{COARSE}(w, \varepsilon)$: given $u \in \mathcal{A}_G^{\eta,\tau}$ and a function $w \in V$, which is known to satisfy $\|u - w\| \leq \varepsilon$, the output Λ is a set of *minimal cardinality* such that

$$\|w - P_\Lambda w\| \leq 2\varepsilon\,, \qquad |\Lambda| \leq \Big(\frac{1}{\eta}\log\frac{|u|_{\mathcal{A}_G^{\eta,\tau}}}{\varepsilon}\Big)^{1/\tau} + 1.$$

This yields **AC-ADFOUR**, an aggressive version of **ADFOUR** with coarsening, which satisfies [10]:

Theorem 4 (Contraction Property for AC-ADFOUR) *Let $\theta \in (0,1)$ be as close to 1 as desired and let $\{\Lambda_n, u_n\}_{n\geq 0}$ be the sequence generated by **AC-ADFOUR**. Then, there exists $C = C(\alpha^*/\alpha_*, \gamma) > 0$ such that*

$$|||u - u_{n+1}||| \leq C\sqrt{1-\theta^2}\,|||u - u_n|||\,.$$

Theorem 5 (Cardinality of AC-ADFOUR) *There exist constants $\tilde\eta$, $\hat\eta < \eta$ and $\bar\tau < \tau$ such that the cardinalities of the feasible residual $\tilde r(u_n)$ and intermediate sets $\hat\Lambda_n$ are suboptimal and given by*

$$|\mathrm{supp}\,\tilde r(u_n)| \leq \left(\frac{1}{\tilde\eta}\log\frac{C_*|u|_{\mathcal{A}_G^{\eta,\tau}}}{\|u-u_n\|}\right)^{1/\bar\tau}, \qquad |\hat\Lambda_{n+1}| \leq \left(\frac{1}{\hat\eta}\log\frac{C_*|u|_{\mathcal{A}_G^{\eta,\tau}}}{\|u-u_{n+1}\|}\right)^{1/\bar\tau},$$

but the final sets Λ_{n+1} exhibit optimal cardinality dictated by η and τ

$$|\Lambda_{n+1}| \leq \left(\frac{1}{\eta} \log \frac{C_* |u|_{\mathcal{A}_G^{\eta,\tau}(\Omega)}}{\|u - u_{n+1}\|} \right)^{1/\tau}.$$

4 *hp*-Adaptive Tree Approximation

To explain the challenges of *hp*-approximation, we observe that the exponential rate of convergence for the canonical example $u(x) = x^\alpha$ with $\alpha < 1$ on $I_0 = [0, 1]$ hinges on a *non-degeneracy* assumption for the interpolation error $E(I, p)$ with polynomials of degree $\leq p$ on any interval $I \subset [0, 1]$ [17, 18, 22, 23, 27]: there exist constants C_1, C_2 independent of I and p such that $C_2 \leq \frac{E(I,p+1)}{E(I,p)} \leq C_1$. This assumption is not valid for highly oscillatory functions, either global polynomials of high degree or piecewise linear on a very fine mesh. An optimal *hp*-adaptive selection strategy should account for these degenerate cases and avoid getting stuck for too long with a wrong choice. We now present one such strategy developed by P. Binev [2, 3].

4.1 Near-Best h-Adaptive Tree Approximation

This strategy is due to P. Binev and R. DeVore for *h*-refinement and finds a quasi-optimal tree with linear complexity [4]. We start with a couple of definitions. A tree $\mathcal{T}$ is a finite collection of elements with a root, and every element has two successors or none (leaves). The collection of leaves, denoted by $\mathcal{L}(\mathcal{T})$, defines a partition of the underlying domain Ω. Given a function v, a *local h-error functional* is a subadditive quantity $e(v, K)$, i.e. $e(v, K) \geq e(v, K') + e(v, K'')$, available for every $K \in \mathcal{T}$, where K' and K'' denote the children of K. For instance, if $v \in L^2(\Omega)$, then $e(v, K)$ is simply the square of the best L^2-error in approximating v on K by polynomials of fixed degree. The *global h-error functional* is given by $\mathcal{E}(v, \mathcal{T}) = \sum_{K \in \mathcal{L}(\mathcal{T})} e(v, K)$ and the *best h-approximation* of v is defined by $\sigma_N(v) := \inf_{\#\mathcal{L}(\mathcal{T}) \leq N} \mathcal{E}(v, \mathcal{T})$. However, computing a mesh which realizes $\sigma_N(v)$ has exponential complexity.

The key idea of P. Binev and R. DeVore is to penalize the lack of success in reducing the error. They achieve this upon modifying $e(v, K)$ to $\tilde{e}(v, K)$ for all $K \in \mathcal{T}$ as follows:

$$\tilde{e}(v, K) := e(v, K) \text{ if } K \text{ is a root;}$$
$$\frac{1}{\tilde{e}(v, K)} := \frac{1}{e(v, K)} + \frac{1}{\tilde{e}(v, K^*)} \text{ where } K^* \text{ is the parent of } K.$$

They apply a *greedy algorithm* to $\{\tilde{e}(v,K)\}_{K\in\mathcal{T}}$: given a tree $\mathcal{T}_N$, with $\#\mathcal{L}(\mathcal{T}_N) = N$, construct $\mathcal{T}_{N+1}$ by bisecting the leaf $K \in \mathcal{L}(\mathcal{T}_N)$ with largest $\tilde{e}(v,K)$. The sequence of trees so generated gives a near-best h-adaptive approximation in the sense $\mathcal{E}(v,\mathcal{T}_N) \le \frac{2N}{N-n+1}\sigma_n(v)$ for any integer $n \le N$, with complexity $\mathcal{O}(N)$. Therefore, taking n to be the floor of $N/2$, there exist universal constants $C_2 < 1 < C_1$ so that

$$\mathcal{E}(v,\mathcal{T}_N) \le C_1\sigma_{C_2N}(v). \tag{12}$$

4.2 Adaptive Strategy for hp-Refinements

This strategy, designed by P. Binev [2, 3], builds two trees: a *ghost h-tree* $\mathcal{T}$, similar to that in Sect. 4.1 but with degree dependent error and modified error functionals, and a *subordinate hp-tree* $\mathcal{P}$. The second tree is obtained by trimming the first one and increasing the polynomial degree as follows. Given $K \in \mathcal{T}$, we denote by $\mathcal{T}(K)$ the *subtree* of $\mathcal{T}$ emanating from K, and let $d(K,\mathcal{T})$ be the dimension of the *admissible polynomial space* on K (the number of leaves of $\mathcal{T}(K)$)

$$d(K,\mathcal{T}) = \#\mathcal{L}(\mathcal{T}(K)) \tag{13}$$

and $p(K,\mathcal{T})$ be the admissible polynomial degree (the largest p satisfying $\frac{(p+d)!}{p!d!} \le d(K,\mathcal{T})$ with d the space dimension). Let $e_p(v,K)$ denote the approximation error of function v on K with polynomials of degree p. The *local hp-error functionals* $E_{\mathcal{T}}(K) = E_{\mathcal{T}}(v,K)$ for $K \in \mathcal{T}$ are defined by

- $E_{\mathcal{T}}(K) = e_0(v,K)$ provided $K \in \mathcal{L}(\mathcal{T})$, i.e., when K is a leaf of $\mathcal{T}$;
- $E_{\mathcal{T}}(K) = \min\{E_{\mathcal{T}}(K') + E_{\mathcal{T}}(K''), e_{p(K,\mathcal{T})}(v,K)\}$ otherwise.

The subordinate *hp*-tree $\mathcal{P}$ is obtained from $\mathcal{T}$ employing a bottom-top approach, depicted in Fig. 1 for spatial dimension $d = 1$, that eliminates a node $K \in \mathcal{T}$ and corresponding subtree $\mathcal{T}(K)$ whenever $E_{\mathcal{T}}(K) = e_{p(K,\mathcal{T})}(v,K)$. The *hp*-tree $\mathcal{P}$ gives rise to an *hp*-partition $\mathcal{D}$, being a collection of *hp*-elements $D = (K_D, d_D)$, defined by $\{K_D : D \in \mathcal{D}\} = \mathcal{L}(\mathcal{P})$ and $d_D = d(K_D,\mathcal{T})$, with $p_D = p(K_D,\mathcal{T})$. The cardinality of $\mathcal{D}$ and corresponding *hp*-error functional for a given function v are defined by

$$\#\mathcal{D} = \sum_{D\in\mathcal{D}} d_D = \#\mathcal{L}(\mathcal{T}), \qquad \mathcal{E}(v,\mathcal{D}) = \sum_{D\in\mathcal{D}} e_{p_D}(v,K_D).$$

Theorem 6 (Instance Optimality [3]) *For any function v and tolerance τ, the algorithm sketched above, and detailed in [3], constructs a hp-tree $\mathcal{P}$ subordinate to an h-tree $\mathcal{T}$ such the hp-error functional and cardinality of the resulting hp-partition $\mathcal{D}$ satisfy*

$$\mathcal{E}(v,\mathcal{D}) \le \tau, \qquad \#\mathcal{D} \le 2\#\tilde{\mathcal{D}},$$

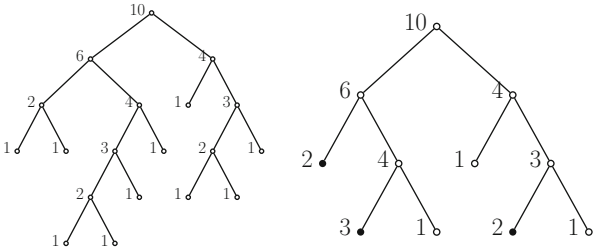


Fig. 1 Ghost *h*-tree $\mathcal{T}$ (*left*) with ten leaves ($\#\mathcal{L}(\mathcal{T}) = 10$) and $d = 1$. The label of each node K is $d(K, \mathcal{T}) = p(K, \mathcal{T}) + 1$; the root K_0 thus has an admissible polynomial degree 9. Subordinate *hp*-tree $\mathcal{P}$ (*right*) resulting from $\mathcal{T}$ upon trimming three subtrees and raising the polynomial degrees of the interior nodes of $\mathcal{T}$, now leaves of $\mathcal{P}$, from 0 to 1, 2, and 1 respectively

for any hp-partition $\tilde{\mathcal{D}}$ *with* $\mathcal{E}(v, \tilde{\mathcal{D}}) \leq \frac{1}{4}\tau$. *The cost of building* $\mathcal{P}$ *is bounded by* $\mathcal{O}\big(\sum_{K\in\mathcal{T}} d(K, \mathcal{T})\big)$, *and varies from* $\mathcal{O}(\#\mathcal{D} \log \#\mathcal{D})$ *for well balanced trees to* $\mathcal{O}((\#\mathcal{D})^2)$ *for highly unbalanced trees.*

5 hp-AFEM: The One Dimensional Case

In this section we discuss our *hp*-AFEM in dimension $d = 1$ along with our convergence and optimality theory [14]. We stress that, in contrast to [23, 27], we make no special assumptions on the error and estimator.

Let $\Omega := (0, 1)$ be the domain and $\mathbf{d} := (f, g, \nu, \sigma)$ be the data, where $f, g \in L^2(\Omega)$ are forcing functions and $\nu, \sigma \in H^1(\Omega)$ are the coefficients, which satisfy $0 < \nu_* \leq \nu \leq \nu^* < \infty$ and $0 \leq \sigma \leq \sigma^* < \infty$. Given that $H^1(\Omega)$ is compactly embedded in $L^\infty(\Omega)$ for $d = 1$, the regularity of the coefficients ν and σ is made for convenience to handle the approximation of ν and σ together with that of u, f, g within Binev's algorithm. Let $u \in H^1_0(\Omega)$ be the solution of the variational problem

$$a(u, v) = \int_\Omega \nu u'v' + \sigma uv = \int_\Omega fv - gv' \qquad \forall v \in H^1_0(\Omega), \tag{14}$$

where the bilinear form $a(\cdot, \cdot)$ satisfies the following bounds provided $\alpha_* = \nu_*$ and $\alpha^* = \nu^* + \frac{\sigma^*}{2}$

$$\alpha_* |v|^2_{H^1(\Omega)} \leq a(v, v) = |\!|\!| v |\!|\!|^2 \leq \alpha^* |v|^2_{H^1(\Omega)}.$$

For any interval $K \subset \Omega$, we equip $H^1_0(K)$ with $|\cdot|_{H^1(K)}$, and define $H^{-1}(K)$ as $(H^1_0(K))'$.

We now describe the *hp-discretization*. Let us consider an *hp*-partition $\mathcal{D}$ as defined in Sect. 4.2, which we write here as $\mathcal{D} = \{D = (K_D, p_D)\}$ in light of the simple relation $p_D = d_D - 1$ between the polynomial space dimension d_D and the polynomial degree p_D. The collection $\{K_D\}_{D\in\mathcal{D}}$ forms a partition of Ω that is created by a sequence of dyadic refinements of subintervals in the partition, starting from the initial partition $\{(0,1)\}$. Let $h_D = \operatorname{diam} K_D$ be the *diameter* of K_D. Let $\mathbb{V}_\mathcal{D}$ be the *hp*-finite element space associated with $\mathcal{D}$

$$\mathbb{V}_\mathcal{D} := \{v \in H_0^1(\Omega) : v_{|K_D} \in \mathbb{P}_{p_D}(K_D)\ \forall D \in \mathcal{D}\},$$

and let $\mathbb{V}_\mathcal{D}^L$ be the corresponding space of continuous piecewise linear functions w.r.t. $\{K_D\}_{D\in\mathcal{D}}$ vanishing at $x = 0, 1$. The natural orthogonal decomposition of $H_0^1(\Omega) = \mathbb{V}_\mathcal{D}^L \oplus \bigoplus_{D\in\mathcal{D}} H_0^1(K_D)$ has the discrete counterpart

$$\mathbb{V}_\mathcal{D} = \mathbb{V}_\mathcal{D}^L \oplus \bigoplus_{D\in\mathcal{D}} \mathbb{P}_{p_D}^0(K_D),$$

where $\mathbb{P}_{p_D}^0(K_D) := \mathbb{P}_{p_D}(K_D) \cap H_0^1(K_D)$. The *Galerkin solution* $U_\mathcal{D} \in \mathbb{V}_\mathcal{D}$ satisfies $a(U_\mathcal{D}, V) = \langle f, V\rangle - \langle g, V'\rangle$ for all $V \in \mathbb{V}_\mathcal{D}$. The *residual* $r = r(U_\mathcal{D}) \in H^{-1}(\Omega)$ is given by

$$\begin{aligned}\langle r(U_\mathcal{D}), v\rangle &= \langle f, v\rangle - \langle g, v'\rangle - a(U_\mathcal{D}, v)\\ &= \sum_{D\in\mathcal{D}} \int_{K_D} \underbrace{\left(f + \left(\nu U_\mathcal{D}'\right)' - \sigma U_\mathcal{D}\right)}_{=r_{1,D}} (v - I_D v) \underbrace{-g}_{=r_{2,D}} (v - I_D v)'\end{aligned}$$

for all $v \in H_0^1(\Omega)$; note that there are no jumps because the linear interpolant $I_D v$ equals v at the nodes. The *a posteriori error estimator* $\eta(U_\mathcal{D}; \mathcal{D})$ and local error indicator $\eta(U_\mathcal{D}; D)$ read

$$\eta^2(U_\mathcal{D}; \mathcal{D}) := \|r(U_\mathcal{D})\|_{H^{-1}(\Omega)}^2 = \sum_{D\in\mathcal{D}} \eta^2(U_\mathcal{D}; D)\,, \qquad \eta(U_\mathcal{D}; D) := \|r(U_\mathcal{D})\|_{H^{-1}(K_D)}.$$

Since we evaluate $r(U_\mathcal{D})$ in $H^{-1}(\Omega)$, we thus have the following upper and lower a posteriori error estimates

$$\frac{1}{\sqrt{\alpha^*}}\eta(U_\mathcal{D}; \mathcal{D}) \le |\!|\!| u - U_\mathcal{D} |\!|\!| \le \frac{1}{\sqrt{\alpha_*}}\eta(U_\mathcal{D}; \mathcal{D}).$$

5.1 Computable Residual and Saturation Property

We now describe a simple procedure to compute $\|r(U_{\mathcal{D}})\|_{H^{-1}(K_D)}$ for any $D \in \mathcal{D}$. We start with the H^1_0-representation of the element residual: let $e_{K_D} \in H^1_0(K_D)$ satisfy

$$\langle e'_{K_D}, v' \rangle_{L_2(K_D)} = \langle r(U_{\mathcal{D}}), v \rangle \quad \forall v \in H^1_0(K_D); \tag{15}$$

Exploiting the explicit expression of the Green's function on $K_D = (a, b)$

$$G(x, y) := \frac{(a-x)(b-y)}{b-a} \quad \text{if } x < y \qquad G(x, y) := \frac{(a-y)(b-x)}{b-a} \quad \text{if } x > y.$$

we obtain a simple integral representation of $e_{K_D}(x) = \int_a^b G(x, y) r_1(y) dy + \int_a^b \frac{\partial G(x,y)}{\partial y} r_2(y) dy$. This in turn yields the computable expression $\eta(U_{\mathcal{D}}; D) = |e_{K_D}|_{H^1(K_D)}$.

Piecewise polynomial data turns out to be useful to allow for exact computation of the stiffness matrix and Galerkin solution and to guarantee the *saturation property* below. We say that data $\mathbf{d} = (f, g, \nu, \sigma)$ satisfies the assumption of *no data oscillation w.r.t.* $\mathcal{D}$ if for any $D \in \mathcal{D}$ the data $\mathbf{d}$ is polynomial of degree $\bar{p}_D$ on K_D, with $\bar{p}_D \leq p_D$. If $U_{\mathcal{D}} \in \mathbb{V}_{\mathcal{D}}$ is the corresponding Galerkin solution, then the residuals $r_{1,D} = f + \left(\nu U'_{\mathcal{D}}\right)' - \sigma U_{\mathcal{D}}$ and $r_{2,D} = -g$ are also polynomials of degree $p_1 = \bar{p}_D + p_D$ and $p_2 = \bar{p}_D$ on K_D, whence the integral residual representation e_{K_D} is a polynomial of degree $\bar{p}_D + p_D + 2$ which can be computed exactly. With this observation in mind, we enrich the local bubble subspaces $\mathbb{P}^0_{p_D}(K_D)$ corresponding to marked elements $D \in \mathcal{M}$:

$$p^*_D := \begin{cases} \bar{p}_D + p_D + 2 & \text{when } D \in \mathcal{M} \\ p_D & \text{when } D \in \mathcal{D} \setminus \mathcal{M} \end{cases} \tag{16}$$

We consider the *hp*-decomposition $\mathcal{D}^* = \{D^* = (K_D, p^*_D)\} : \forall D \in \mathcal{D}\}$ and the *enriched finite element space*

$$\mathbb{V}_{\mathcal{D}^*} = \mathbb{V}^L_{\mathcal{D}} \oplus \bigoplus_{D \in \mathcal{D}^*} \mathbb{P}^0_{p^*_D}(K_D). \tag{17}$$

Lemma 1 (Saturation Property) *Let the assumption of no data oscillation w.r.t.* $\mathcal{D}$ *be valid. Let* $U = U_{\mathcal{D}} \in \mathbb{V}_{\mathcal{D}}$ *and* $U^* = U_{\mathcal{D}^*} \in \mathbb{V}_{\mathcal{D}^*}$ *be the Galerkin solutions in* $\mathbb{V}_{\mathcal{D}}$ *and the enriched space* $\mathbb{V}_{\mathcal{D}^*}$ *given by* (16)–(17)*. If* $0 < \theta < 1$ *is the parameter of Dörfler marking, then*

$$|||U - U^*|||^2 \geq \theta^2 \frac{\alpha_*}{\alpha^*} |||u - U|||^2. \tag{18}$$

Proof Since $H_0^1(\Omega) = \mathbb{V}_{\mathcal{D}}^L \oplus \bigoplus_{D\in\mathcal{D}} H_0^1(K_D)$ and both residuals $r(U)$ and $r(U^*)$ vanish against functions in $\mathbb{V}_{\mathcal{D}}^L$, the norm in $H^{-1}(\Omega)$ of $r(U) - r(U^*)$ localizes to contributions over elements $D \in \mathcal{D}$:

$$\begin{aligned}\alpha^* |\!|\!| U - U^* |\!|\!|^2 \geq \|r(U) - r(U^*)\|_{H^{-1}(\Omega)}^2 &= \sum_{D\in\mathcal{D}} \|r(U) - r(U^*)\|_{H^{-1}(K_D)}^2 \\ &\geq \sum_{D\in\mathcal{M}} \|r(U) - r(U^*)\|_{H^{-1}(K_D)}^2.\end{aligned}$$

Since U^* is the Galerkin solution w.r.t. the enriched finite element space and the degree of e_{K_D} is p_D^*, according to (15), we deduce

$$\sup_{v\in\mathbb{P}^0_{p_D^*}(K_D)} \frac{|\langle r(U) - r(U^*), v\rangle|^2}{|v|^2_{H^1(K_D)}} = \sup_{v\in\mathbb{P}^0_{p_D^*}(K_D)} \frac{|\langle r(U), v\rangle|^2}{|v|^2_{H^1(K_D)}} = \|r(U)\|^2_{H^{-1}(K_D)} \qquad \forall D \in \mathcal{M}.$$

Adding over $D \in \mathcal{M}$ we show *discrete efficiency*: $\alpha^* |\!|\!| U - U^* |\!|\!|^2 \geq \eta^2(U;\mathcal{M})$. Combining Dörfler marking $\eta(U;\mathcal{M}) \geq \theta\eta(U;\mathcal{D})$ with the upper bound $\alpha_* |\!|\!| u - U |\!|\!|^2 \leq \eta(U;\mathcal{D})^2$ we obtain the assertion. □

5.2 Contraction Property of Module PDE

Module PDE Given a *hp*-partition $\hat{\mathcal{D}}$, we suppose that data $\hat{\mathbf{d}}$ satisfies a no data oscillation assumption w.r.t. $\hat{\mathcal{D}}$, and let $\hat{u} \in H_0^1(\Omega)$ be the corresponding solution of (14). If θ denotes the Dörfler parameter and ε the error tolerance, the module **PDE** computes an *hp*-discretization $\mathcal{D}$ finer than $\hat{\mathcal{D}}$ and Galerkin solution $U_{\mathcal{D}}$ such that $|\hat{u} - U_{\mathcal{D}}|_{H_0^1(\Omega)} \leq \varepsilon$:

$[\mathcal{D}, U_{\mathcal{D}}] =$ **PDE** $(\hat{\mathcal{D}}, \hat{\mathbf{d}}, \varepsilon)$
set $\ell = 0$; $\mathcal{D}^{(0)} = \hat{\mathcal{D}}$;
do
  $U_{\mathcal{D}^{(\ell)}} =$ **SOLVE**$(\hat{\mathbf{d}}, \mathcal{D}^{(\ell)})$
  $\{\eta(U_{\mathcal{D}^{(\ell)}}; D),\ D \in \mathcal{D}^{(\ell)}\} =$ **ESTIMATE**$(U_{\mathcal{D}^{(\ell)}}, \mathcal{D}^{(\ell)})$
  $\mathcal{M}^{(\ell)} =$ **MARK**$(\mathcal{D}^{(\ell)}, \eta(U_{\mathcal{D}^{(\ell)}}; \mathcal{D}^{(\ell)}), \theta)$
  $\mathcal{D}^{(\ell+1)} =$ **ENRICH**$(\mathcal{M}^i, \mathcal{D}^{(\ell)})$
  update $\ell \leftarrow \ell + 1$
while $(\eta(U_{\mathcal{D}^{(\ell)}}; \mathcal{D}^{(\ell)}) > \alpha_*\varepsilon)$
$\mathcal{D} := \mathcal{D}^{(\ell)}$; $U_{\mathcal{D}} := U_{\mathcal{D}^{(\ell)}}$

We point out that data $\hat{\mathbf{d}}$ is piecewise polynomial of degree $\bar{p}_D$ for all $D \in \hat{\mathcal{D}}$, comes from an outer iteration, and does not change within **PDE**. This allows **SOLVE** and **ESTIMATE** to compute stiffness matrices, Galerkin solutions, and estimators exactly. The module **MARK** uses a greedy approach to select estimators. Finally,

ENRICH increases the local polynomial degree p_D of $\mathbb{V}_{\mathcal{D}^{(\ell)}}$ according to (16) and thus guarantees the saturation property (18) as well as the following analogue of Theorem 1, with a similar proof.

Theorem 7 (Contraction Property of PDE) *If* $\rho = \left(1 - \theta^2 \frac{\alpha_*}{\alpha^*}\right)^{1/2}$, *then the iterates* $U^{(\ell)} = U_{\mathcal{D}^{(\ell)}}$ *for* $\ell \geq 0$ *satisfy* $|||u - U^{(\ell+1)}||| \leq \rho |||u - U^{(\ell)}|||$. *Moreover, the estimate* $|\hat{u} - U_{\mathcal{D}}|_{H^1(\Omega)} \leq \epsilon$ *is valid upon termination.*

Given a reduction parameter $\gamma < 1$, this theorem implies that the number of iterations within **PDE** to reduce the energy error from the value δ to $\gamma\delta$ is uniform in p, h, and δ, but depends on $\gamma < 1$.

Perturbation Analysis We must now account for the piecewise polynomial approximation $\hat{\mathbf{d}} = (\hat{f}, \hat{g}, \hat{\nu}, \hat{\sigma})$ of general data $\mathbf{d} = (f, g, \nu, \sigma)$. We first enforce the following bounds for the coefficients $\hat{\nu}, \hat{\sigma}$

$$0 < \hat{\nu}_* = \frac{\nu_*}{2} \leq \hat{\nu} \leq \nu^* + \frac{\nu_*}{2} = \hat{\nu}^*, \qquad -\frac{\nu_*}{2} \leq \hat{\sigma} \leq \sigma^* + \frac{\nu_*}{2} = \hat{\sigma}^*;$$

Since $\|v\|_{L^\infty(\Omega)} \leq |v|_{H^1(\Omega)}$ and $\|v\|_{L^2(\Omega)} \leq \frac{1}{\sqrt{2}}|v|_{H^1(\Omega)}$, we have the following *perturbation estimate* for the solution $\hat{u}$ of $\hat{L}\hat{u} := -(\hat{\nu}\hat{u}')' + \hat{\sigma}\hat{u} = \hat{f} + \hat{g}'$: if $M = \frac{4}{\nu_*}\left(\|f\|_{H^{-1}(\Omega)} + \|g\|_{L^2(\Omega)}\right)$, then $|\hat{u}|_{H^1(\Omega)} \leq M$ and

$$\nu_* |u - \hat{u}|_{H^1(\Omega)} \leq \|f - \hat{f}\|_{H^{-1}(\Omega)} + \|g - \hat{g}\|_{L^2(\Omega)} + \frac{M}{2}\|\sigma - \hat{\sigma}\|_{H^1(\Omega)} + M\|\nu - \hat{\nu}\|_{H^1(\Omega)}. \tag{19}$$

5.3 Module hp-NEARBEST and Quasi-optimality

The procedure $[\hat{\mathcal{D}}, \hat{V}, \hat{\mathbf{d}}] =$ **hp-NEARBEST**$(v, \mathbf{d}, \varepsilon)$ is the algorithm of Sect. 4.2 applied with $\tau = \epsilon^2$ to a specific local error functional which, for a given function $v \in H^1_0(\Omega)$ and data $\mathbf{d}$, outputs an *hp*-discretization $\hat{\mathcal{D}}$ and *hp*-quasi best approximations $\hat{V}$ and $\hat{\mathbf{d}}$. The latter are simply the local L^2-projections $\hat{f} = P^0_{\hat{\mathcal{D}}} f$ and $\hat{g} = P^0_{\hat{\mathcal{D}}} g$ onto $\mathbb{V}_{\hat{\mathcal{D}}}$ of f and g, or the local H^1-projections $\hat{\nu} = P^1_{\hat{\mathcal{D}}} \nu$ and $\hat{\sigma} = P^1_{\hat{\mathcal{D}}} \sigma$ onto $\mathbb{V}_{\hat{\mathcal{D}}}$ of ν and σ; note that $\hat{V}$ and $\hat{\mathbf{d}}$ are globally discontinuous. We thus define *data oscillation* to be $\mathrm{osc}^2(\mathbf{d}, \hat{\mathcal{D}}) := \sum_{D \in \hat{\mathcal{D}}} \mathrm{osc}^2(\mathbf{d}, D)$ with

$$\mathrm{osc}^2(\mathbf{d}, D) = \|p_D^{-1} h_D (f - \hat{f})\|^2_{L^2(K_D)} + \|g - \hat{g}\|^2_{L^2(K_D)} + \|\sigma - \hat{\sigma}\|^2_{H^1(K_D)} + \|\nu - \hat{\nu}\|^2_{H^1(K_D)} \quad \forall D \in \hat{\mathcal{D}}.$$

In light of (19), we realize that $|u-\hat{u}|_{H^1(\Omega)} \leq \frac{2M_*}{\nu_*}\mathrm{osc}(\mathbf{d},\hat{\mathcal{D}})$ with $M_* := \max\{1, M\}$. Given a suitable parameter $\omega \leq 1$, to be chosen later, we let the *local error functional* be

$$\mathrm{e}(v,\mathbf{d},D) := |v - P^1_{\hat{\mathcal{D}}} v|^2_{H^1(K_D)} + \frac{1}{\omega^2}\mathrm{osc}^2(\mathbf{d},D) \qquad \forall D \in \hat{\mathcal{D}}, \tag{20}$$

set $\hat{V} := P^1_{\hat{\mathcal{D}}} v$, and define the *global error functional* to be $E(v,\mathbf{d},\hat{\mathcal{D}}) := \sum_{D\in\hat{\mathcal{D}}} \mathrm{e}(v,\mathbf{d},D)$. In view of Theorem 6, the output $[\hat{\mathcal{D}},\hat{V},\hat{\mathbf{d}}]$ = **hp-NEARBEST** $(v,\mathbf{d},\epsilon)$ is *instance optimal*:

$$E(v,\mathbf{d},\mathcal{D}) \leq \varepsilon^2, \qquad \#\hat{\mathcal{D}} \leq 2\,\#\tilde{\mathcal{D}}, \tag{21}$$

for any admissible *hp*-mesh $\tilde{\mathcal{D}}$ with $E(v,\mathbf{d},\tilde{\mathcal{D}}) \leq \frac{1}{4}\varepsilon^2$. This result is promising but unfortunately cannot be applied to the solution $u \in H^1_0(\Omega)$ of (14) because u is not directly accessible. However, we still achieve a *near-best hp-approximation* provided there is an approximation $v \in H^1_0(\Omega)$ of u so that $|u-v|_{H^1(\Omega)} \leq \varepsilon$: it is easily seen that the *hp*-mesh $\hat{\mathcal{D}}$ generated by **hp-NEARBEST**$(v,\mathbf{d},3\varepsilon)$ with tolerance 3ϵ satisfies

$$E(u,\mathbf{d},\hat{\mathcal{D}}) \leq 16\varepsilon^2, \qquad \#\hat{\mathcal{D}} \leq 2\,\#\tilde{\mathcal{D}},$$

for any admissible *hp*-mesh $\tilde{\mathcal{D}}$ with $E(u,\mathbf{d},\tilde{\mathcal{D}}) \leq \frac{1}{4}\varepsilon^2$.

5.4 The hp-AFEM

Given data $\mathbf{d} = (f,g,\nu,\sigma)$ and parameters $\gamma,\omega < 1$ and tol, the following algorithm **hp-AFEM** generates an *hp*-partition $\mathcal{D}$ and Galerkin solution U over $\mathcal{D}$ so that $|u-U|_{H^1_0(\Omega)} \leq$ tol :

$[\mathcal{D},U]$ = **hp-AFEM**$(\mathbf{d},\mathrm{tol})$
let $n=0$ and initialize $\mathcal{D}_0$, $\mathbf{d}_0$, and ε_0 with $|u-U_0|_{H^1(\Omega)} \leq 2\epsilon_0$;
do
  $n \leftarrow n+1$; $\varepsilon_n = \gamma\varepsilon_{n-1}$
  $[\hat{\mathcal{D}}_n,\hat{U}_n,\hat{\mathbf{d}}_n]$ = **hp-NEARBEST**$(U_{n-1},\mathbf{d},6\varepsilon_{n-1})$
  $[\mathcal{D}_n,U_n] = \mathbf{PDE}(\hat{\mathcal{D}}_n,\hat{\mathbf{d}}_n,\varepsilon_n)$
while $\varepsilon_n > \frac{1}{2}$tol
$\mathcal{D} := \mathcal{D}_n$; $U = U_n$

The presence of the parameter ω in (20) penalizes data approximations and prepares data $\mathbf{d}_{n+1}$ for the next level of approximation within the outer loop in **hp-AFEM**. This is critical for the next result.

Theorem 8 (Convergence and Optimality of hp-AFEM) *Let* $\omega \leq \frac{\gamma \nu_*}{12 M_*}$. *Then* ***hp-AFEM*** *terminates in finite number of steps* N *and* $|u - U|_{H^1(\Omega)} \leq tol$. *The last output* $(\hat{\mathcal{D}}_N, \hat{U}_N, \hat{\mathbf{d}}_N)$ *of* ***hp-NEARBEST*** *satisfies*

$$|u - \hat{U}_N|_{H^1(\Omega)} \leq \frac{4}{\gamma}\, tol, \qquad \mathrm{osc}(\mathbf{d}, \hat{\mathcal{D}}_N) \leq \frac{\nu_*}{4 M_*}\, tol, \qquad \hat{\mathcal{D}}_N \leq 2\, \#\tilde{\mathcal{D}},$$

for any admissible hp-mesh $\tilde{\mathcal{D}}$ *with* $E(u, \mathbf{d}, \tilde{\mathcal{D}})^{\frac{1}{2}} \leq \frac{1}{2}\, tol$. *The number of inner loops of* ***PDE*** *is independent of* n, *whence the complexity of* ***PDE*** *scales linearly with* $\#\mathcal{D}_n$. *The complexity of* ***hp-NEARBEST*** *scales from linear to quadratic in terms of* $\#\hat{\mathcal{D}}_n$ *depending on the tree structure.*

6 *hp*-AFEM: The Two Dimensional Case

We finally give a brief overview of the case $d = 2$ treated in [14]. We consider a polygonal domain $\Omega \subset \mathbb{R}^2$ and data $\mathbf{d} = f$ with $f \in L^2(\Omega)$, and let $u \in H^1_0(\Omega)$ be the solution of the model problem $-\Delta u = f$. Since the operator has constant coefficients, the notion of *data oscillation* is simpler than in Sect. 5, namely $\mathrm{osc}^2(\mathbf{d}, D) = \|p_D^{-1} h_D (f - \hat{f})\|^2_{L^2(K_D)}$. However, the *local error functional* $e_D(v, \mathbf{d})$ and module **hp-AFEM** remain the same.

Our theory of **hp-AFEM** for $d = 2$ hinges on the following three issues [14]:

- **Saturation property (SP):** Existence of constant $\lambda > 0$, independent of p, so that $|||U - U_*||| \geq \lambda |||u - U|||$. This is illustrated in Theorem 1 for **ADFOUR** and Lemma 1 for **hp-AFEM** with $d = 1$. The existing a posteriori error estimators for $d = 2$ are not completely satisfactory. The residual estimator of J. Melenk and B. Wohlmuth [24] is p-sensitive, whereas the hypercircle estimators of D. Braess, V. Pillwein and J. Schöberl [6], and A. Ern and M. Vohralík [21] are p-insensitive but not known to satisfy (SP). In [14] we use [1] and thus get a p-sensitive module **PDE**, but we are currently constructing a p-insensitive a posteriori estimator for polynomial data $\mathbf{d}$ which satisfies (SP).
- **Local implies global approximation:** The hp-approximation generated by **hp-NEARBEST** is local. We claim that local H^1-approximation w.r.t. a conforming mesh is equivalent to global H^1-approximation for any polynomial degree p with a logarithmic dependence on p [14]. This result extends recent ones of A. Veeser [32] for fixed polynomial degree and is fully documented in [14].
- **Completion for *hp*-refinement:** The algorithm **hp-NEARBEST** developed by P. Binev may output a nonconforming mesh with abrupt and unlimited transitions of polynomial degrees between adjacent elements [2, 3]; we exhibit a pathological example in [14]. This is undesirable and in fact difficult to handle in practice for $d \geq 2$ and may deteriorate the complexity of **hp-AFEM**, but not its instance

optimality. The module **hp-NEARBEST** is to be modified to ensure a smooth transition of polynomial degrees or to output (nearly) conforming meshes, a process called *p*-completion. A related *h*-completion process for bisection is instead well understood [5, 26, 31].

Acknowledgements The authors "Claudio Canuto" and "Marco Verani" were partially supported by the Italian national grant PRIN 2012HBLYE4. The author "Ricardo H. Nochetto" was partially supported by NSF grants DMS-1109325 and DMS-1411808.

References

1. R. Bank, A. Parsania, S. Sauter, Saturation estimates for hp-finite element methods. Technical. Report 03, ETH-Zurich (2014)
2. P. Binev, Instance optimality for *hp*-type approximation. Oberwolfach Rep. **39**, 14–16 (2013)
3. P. Binev, Tree approximation for *hp*-adaptivity. In preparation
4. P. Binev, R. DeVore. Fast computation in adaptive tree approximation. Numer. Math. **97**(2), 193–217 (2004)
5. P. Binev, W. Dahmen, R. DeVore, Adaptive finite element methods with convergence rates. Numer. Math. **97**(2), 219–268 (2004)
6. D. Braess, V. Pillwein, J. Schöberl, Equilibrated residual error estimates are *p*-robust. Comput. Methods Appl. Mech. Eng. **198**(13–14), 1189–1197 (2009)
7. M. Bürg, W. Dörfler, Convergence of an adaptive *hp* finite element strategy in higher space-dimensions. Appl. Numer. Math. **61**(11), 1132–1146 (2011)
8. C. Canuto, M. Verani, On the numerical analysis of adaptive spectral/*hp* methods for elliptic problems, in *Analysis and Numerics of Partial Differential Equations*. Springer INdAM Series, vol. 4 (Springer, Milan, 2013), pp. 165–192
9. C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods. Fundamentals in Single Domains*. Scientific Computation (Springer, Berlin, 2006)
10. C. Canuto, R.H. Nochetto, M. Verani, Adaptive Fourier-Galerkin methods. Math. Comput. **83**(288), 1645–1687 (2014)
11. C. Canuto, R.H. Nochetto, M. Verani, Contraction and optimality properties of adaptive Legendre-Galerkin methods: the one-dimensional case. Comput. Math. Appl. **67**(4), 752–770 (2014)
12. C. Canuto, V. Simoncini, M. Verani, On the decay of the inverse of matrices that are sum of Kronecker products. Linear Algebra Appl. **452**, 21–39 (2014)
13. C. Canuto, R.H. Nochetto, R. Stevenson, M. Verani, A feasible super-aggressive Galerkin-Fourier method. In preparation
14. C. Canuto, R.H. Nochetto, R. Stevenson, M. Verani, Convergence and Optimality of *hp*-AFEM (2015). arXiv:1503.03996
15. C. Canuto, V. Simoncini, M. Verani, Contraction and optimality properties of an adaptive Legendre-Galerkin method: the multi-dimensional case. J. Sci. Comput. **63**(3), 769–798 (2015)
16. J.M. Cascón, C. Kreuzer, R.H. Nochetto, K.G. Siebert, Quasi-optimal convergence rate for an adaptive finite element method. SIAM J. Numer. Anal. **46**(5), 2524–2550 (2008)
17. W. Dahmen, K. Scherer, Best approximation by piecewise polynomials with variable knots and degrees. J. Approx. Theory **26**(1), 1–13 (1979)
18. R. DeVore, K. Scherer, Variable knot, variable degree spline approximation to x^β, in *Quantitative Approximation (Proc. Internat. Sympos., Bonn, 1979)* (Academic, New York, 1980), pp. 121–131
19. W. Dörfler, A convergent adaptive algorithm for Poisson's equation. SIAM J. Numer. Anal. **33**(3), 1106–1124 (1996)

20. W. Dörfler, V. Heuveline, Convergence of an adaptive *hp* finite element strategy in one space dimension. Appl. Numer. Math. **57**(10), 1108–1124 (2007)
21. A. Ern, M. Vohralík, Polynomial-degree-robust a posteriori estimates in a unified setting for conforming, nonconforming, discontinuous Galerkin, and mixed discretizations. INRIA Preprint (2014)
22. W. Gui, I. Babuška, The *h*, *p* and *h-p* versions of the finite element method in 1 dimension. II. The error analysis of the *h*- and *h-p* versions. Numer. Math. **49**(6), 613–657 (1986)
23. W. Gui, I. Babuška, The *h*, *p* and *h-p* versions of the finite element method in 1 dimension. III. The adaptive *h-p* version. Numer. Math. 49(6), 659–683 (1986)
24. J.M. Melenk, B.I. Wohlmuth, On residual-based a posteriori error estimation in *hp*-FEM. Adv. Comput. Math. **15**(1–4), 311–331 (2001). A posteriori error estimation and adaptive computational methods
25. P. Morin, R.H. Nochetto, K.G. Siebert, Data oscillation and convergence of adaptive FEM. SIAM J. Numer. Anal. **38**(2), 466–488 (electronic) (2000)
26. R.H. Nochetto, K.G. Siebert, A. Veeser, Theory of adaptive finite element methods: an introduction, in *Multiscale, Nonlinear and Adaptive Approximation* (Springer, Berlin, 2009), pp. 409–542
27. K. Scherer, On optimal global error bounds obtained by scaled local error estimates. Numer. Math. **36**(2), 151–176 (1980)
28. A. Schmidt, K.G. Siebert, A posteriori estimators for the *h-p* version of the finite element method in 1D. Appl. Numer. Math. **35**(1), 43–66 (2000)
29. Ch. Schwab, *p*- and *hp*-finite element methods, in *Numerical Mathematics and Scientific Computation* (The Clarendon Press, Oxford University Press, New York, 1998). Theory and applications in solid and fluid mechanics
30. R. Stevenson, Optimality of a standard adaptive finite element method. Found. Comput. Math. **7**(2), 245–269 (2007)
31. R. Stevenson, The completion of locally refined simplicial partitions created by bisection. Math. Comput. **77**, 227–241 (2008)
32. A. Veeser, Approximating gradients with continuous piecewise polynomial functions. Technical report, Dipartimento di Matematica 'F. Enriques', Università degli Studi di Milano (2012)

Nonlinear Elasticity for Mesh Deformation with High-Order Discontinuous Galerkin Methods for the Navier-Stokes Equations on Deforming Domains

Bradley Froehle and Per-Olof Persson

Abstract We present a numerical framework for simulation of the compressible Navier-Stokes equations on problems with deforming domains where the boundary motion is prescribed by moving meshes. Our goal is a high-order accurate, efficient, robust, and general purpose simulation tool. To obtain this, we use a discontinuous Galerkin space discretization, diagonally implicit Runge-Kutta time integrators, and fully unstructured meshes of triangles and tetrahedra. To handle the moving boundaries, a mapping function is produced by first deforming the mesh using a neo-Hookean elasticity model and a high-order continuous Galerkin FEM method. The resulting nonlinear equations are solved using Newton's method and a robust homotopy approach. From the deformed mesh, we compute grid velocities and deformations that are consistent with the time integration scheme. These are used in a mapping-based arbitrary Lagrangian-Eulerian formulation, with numerically computed mapping Jacobians which satisfy the geometric conservation law. We demonstrate our methods on a number of problems, ranging from model problems that confirm the high-order accuracy to the flow in domains with complex deformations.

1 Introduction

Over the last decade, high-order accurate methods such as discontinuous Galerkin (DG) methods [2, 6] have become increasingly popular for computational fluid dynamics simulations [15]. One of the main reasons for this popularity is that the schemes produce stable discretizations of conservation laws on fully unstructured meshes of tetrahedral elements, with arbitrary orders of accuracy. More recently, they have also been applied to problems with moving boundaries and deforming domains [13], for applications such as flapping flight simulations [16].

B. Froehle • P.-O. Persson (✉)
Department of Mathematics, University of California, Berkeley, Berkeley, CA 94720-3840, USA
e-mail: brad.froehle@gmail.com; persson@berkeley.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_5

A popular technique for handling the deforming domains is the Arbitrary Lagrangian Eulerian (ALE) method [4, 8, 14], which allows for a deforming grid by using a discretization which accounts for the grid motion. While usually formulated in a moving grid framework, in [13] it was demonstrated how these schemes can be used in a DG setting with a mapping-based formulation and a fixed reference domain, to easily obtain high-order accuracy in both space and time.

For complex geometries and deformations, this domain mapping has to be solved for numerically using some type of mesh deformation scheme. In this work, we show how to do this using a quasi-static nonlinear elasticity approach, similar to the one used for high-order curved mesh generation in [12]. We show how to use the resulting deformed meshes in a DG-based ALE scheme, and how to derive discretely consistent grid velocities for diagonally implicit Runge-Kutta methods. Using a non-trivial test problem we can demonstrate optimal order convergence. We also show that a lower-order element-wise mapping is preferable to a full isoparametric mapping, which is convenient in the case of rigid body motions. Finally we show how the scheme has been applied to two complex flapping flight applications.

2 Governing Equations

The fluid flow is governed by the compressible Navier-Stokes equations, which can be written in conservation form as:

$$\frac{\partial}{\partial t}(\rho) + \frac{\partial}{\partial x_j}(\rho u_j) = 0 \tag{1}$$

$$\frac{\partial}{\partial t}(\rho u_i) + \frac{\partial}{\partial x_j}(\rho u_i u_j + p\delta_{ij}) = \frac{\partial}{\partial x_j}\tau_{ij} \qquad \text{for } i = 1, 2, 3 \tag{2}$$

$$\frac{\partial}{\partial t}(\rho E) + \frac{\partial}{\partial x_j}(\rho u_j E + u_j p) = \frac{\partial}{\partial x_j}(-q_j + u_i \tau_{ij}) \tag{3}$$

where the conserved variables are the fluid density ρ, momentum in the jth spatial coordinate direction ρu_j, and total energy ρE. The viscous stress tensor and heat flux are given by

$$\tau_{ij} = \mu\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i} - \frac{2}{3}\frac{\partial u_k}{\partial x_j}\delta_{ij}\right) \quad \text{and} \quad q_j = -\frac{\mu}{\text{Pr}}\frac{\partial}{\partial x_j}\left(E + \frac{p}{\rho} - \frac{1}{2}u_k u_k\right). \tag{4}$$

Here, μ is the viscosity coefficient and $\text{Pr} = 0.72$ is the Prandtl number which we assume to be constant. For an ideal gas, the pressure p has the form $p = (\gamma - 1)\rho\,(E - u_k u_k/2)$, where γ is the adiabatic gas constant. We write the

system of conservation laws (1)–(3) in vector form as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{f}(\boldsymbol{u}, \nabla \boldsymbol{u}) = \boldsymbol{0}, \tag{5}$$

where $\boldsymbol{u} = [\rho, \rho u_1, \rho u_2, \rho u_3, \rho E]$ is the vector of conserved quantities and $\boldsymbol{f}$ is the corresponding flux function. In our examples we impose two types of boundary conditions—free-stream conditions and adiabatic no-slip wall conditions.

The deformable domains are handled through an Arbitrary Lagrangian Eulerian (ALE) formulation. A point $\boldsymbol{X}$ in a fixed reference domain V is mapped to $\boldsymbol{x}(\boldsymbol{X}, t)$ in a time-varying domain $v(t)$. The deformation gradient $\boldsymbol{G}$, mapping (or mesh) velocity $\boldsymbol{v}$, and mapping Jacobian g are defined as

$$\boldsymbol{G} = \nabla_{\boldsymbol{X}} \boldsymbol{x}, \qquad \boldsymbol{v} = \frac{\partial \boldsymbol{x}}{\partial t}, \qquad g = \det \boldsymbol{G} \tag{6}$$

The system (5) in the physical domain $(\boldsymbol{x}, t)$ can then be rewritten as a system of conservation laws in the reference domain $(\boldsymbol{X}, t)$

$$\frac{\partial \boldsymbol{U}}{\partial t} + \nabla_{\boldsymbol{X}} \cdot \boldsymbol{F}(\boldsymbol{U}, \nabla_{\boldsymbol{X}} \boldsymbol{U}) = \boldsymbol{0} \tag{7}$$

where the conserved quantities in reference space are $\boldsymbol{U} = g\boldsymbol{u}$ with the fluxes $\boldsymbol{F} = g\boldsymbol{G}^{-1}\boldsymbol{f} - \boldsymbol{u}\boldsymbol{G}^{-1}\boldsymbol{v}$, and the gradient of the solution is given by

$$\nabla \boldsymbol{u} = (\nabla_{\boldsymbol{X}}(g^{-1}\boldsymbol{U}))\boldsymbol{G}^{-T} = (g^{-1}\nabla_{\boldsymbol{X}}\boldsymbol{U} - \boldsymbol{U}\nabla_{\boldsymbol{X}}(g^{-1}))\boldsymbol{G}^{-T}. \tag{8}$$

For more details, including a convenient method for satisfying the Geometric Conservation Law (GCL) by introducing an additional set of ODEs, see [13].

3 Numerical Schemes

3.1 Discretization of the Navier-Stokes Equations

Our 3DG flow solver is based on the high-order Discontinuous Galerkin (DG) method with tetrahedral mesh elements and nodal basis functions. For simplicity, we change the notation and use lower-case symbols for the solution $\boldsymbol{u}$, and we omit the subscripts on the derivative operators. We also split the fluxes into an inviscid component $\boldsymbol{F}^i(\boldsymbol{u})$ and a viscous component $\boldsymbol{F}^v(\boldsymbol{u}, \nabla \boldsymbol{u})$. The ALE system (7) can then be written in a split form as

$$\frac{\partial \boldsymbol{u}}{\partial t} + \nabla \cdot \boldsymbol{F}^i(\boldsymbol{u}) - \nabla \cdot \boldsymbol{F}^v(\boldsymbol{u}, \boldsymbol{q}) = \boldsymbol{0}, \tag{9}$$

$$\nabla \boldsymbol{u} = \boldsymbol{q}. \tag{10}$$

Next, we introduce a computational mesh $\mathscr{T}_h = \{K\}$ of the reference domain Ω, and the finite element spaces $\mathscr{V}_h^p$ and Σ_h^p:

$$\mathscr{V}_h^p = \{\boldsymbol{v} \in [L^2(\Omega)]^5 \mid \boldsymbol{v}|_K \in [\mathscr{P}_p(K)]^5 \ \forall K \in \mathscr{T}_h\}, \tag{11}$$

$$\Sigma_h^p = \{\boldsymbol{\tau} \in [L^2(\Omega)]^{5\times 3} \mid \boldsymbol{\tau}|_K \in [\mathscr{P}_p(K)]^{5\times 3} \ \forall K \in \mathscr{T}_h\}, \tag{12}$$

where $\mathscr{P}_p(K)$ is the space of polynomial functions of degree at most $p \geq 1$ on K, and 3 and 5 refer to the dimension and number of solution components of the Navier-Stokes equations in three dimensions. We multiply the system of Eqs. (9)–(10) by test functions $\boldsymbol{v}$, $\boldsymbol{\tau}$ and integrate by parts. Our semi-discrete DG formulation is then expressed as: find $\boldsymbol{u}_h \in \mathscr{V}_h^p$ and $\boldsymbol{q}_h \in \Sigma_h^p$ such that for all $K \in \mathscr{T}_h$, we have

$$\int_K \frac{\partial \boldsymbol{u}_h}{\partial t} \cdot \boldsymbol{v}\, dx + \int_K \left(\boldsymbol{F}^i(\boldsymbol{u}_h) - \boldsymbol{F}^v(\boldsymbol{u}_h, \boldsymbol{q}_h)\right) : \nabla \boldsymbol{v}\, dx - \oint_{\partial K} \left(\widehat{\boldsymbol{F}^i(\boldsymbol{u}_h)} - \boldsymbol{F}^v(\widehat{\boldsymbol{u}_h, \boldsymbol{q}_h})\right) \cdot \boldsymbol{v}\, ds = \boldsymbol{0}, \forall \boldsymbol{v} \in [\mathscr{P}_p(K)]^5 \tag{13}$$

$$\int_K \boldsymbol{q}_h : \boldsymbol{\tau}\, dx + \int_K \boldsymbol{u}_h \cdot (\nabla \cdot \boldsymbol{\tau})\, dx - \oint_{\partial K} (\hat{\boldsymbol{u}}_h \otimes \boldsymbol{n}) : \boldsymbol{\tau}\, ds = \boldsymbol{0}, \forall \boldsymbol{\tau} \in [\mathscr{P}_p(K)]^{5\times 3} \tag{14}$$

To complete the description we need to specify the numerical fluxes for all element boundaries ∂K. The inviscid fluxes $\widehat{\boldsymbol{F}^i(\boldsymbol{u}_h)}$ are computed using a standard approximate Riemann solver and the modification for our ALE formulation described in [13]. For the viscous fluxes $\widehat{\boldsymbol{F}_h^v}$, $\hat{\boldsymbol{u}}_h$, we use a formulation based on the Compact DG (CDG) method [10]. At a boundary face, we impose either far field or no-slip conditions weakly through the fluxes.

Using a standard finite element procedure, we obtain the semi-discrete form of our equations:

$$M\frac{d\bar{\boldsymbol{u}}}{dt} = \bar{\boldsymbol{r}}(\bar{\boldsymbol{u}}), \tag{15}$$

for discrete solution vector $\bar{\boldsymbol{u}}$, mass matrix $\boldsymbol{M}$, and residual function $\bar{\boldsymbol{r}}(\bar{\boldsymbol{u}})$. We integrate this system of ODEs in time using Diagonally Implicit Runge-Kutta (DIRK) methods [1], where the solution is advanced from time t_n to t_{n+1} by:

$$M\bar{\boldsymbol{k}}_i = \bar{\boldsymbol{r}}\left(t_n + c_i \Delta t, \bar{\boldsymbol{u}}_n + \Delta t \sum_{j=1}^{s} a_{ij}\bar{\boldsymbol{k}}_j\right), \quad i = 1, \ldots, s \tag{16}$$

$$\bar{\boldsymbol{u}}_{n+1} = \bar{\boldsymbol{u}}_n + \Delta t \sum_{j=1}^{s} b_j \bar{\boldsymbol{k}}_j. \tag{17}$$

We consider a variety of DIRK schemes, but in particular the 2- and 3-stage L-stable schemes presented in [1]. Note that the implicit scheme requires inversion of matrices of the form $M - a_{ii}\Delta t d\bar{\boldsymbol{r}}/d\bar{\boldsymbol{u}}$. This is accomplished by using a preconditioned parallel Newton-Krylov solver, see [11] for details.

3.2 Computation of Gradients and Mesh Velocities

The ALE equations (7) require the mesh deformation gradient $\boldsymbol{G}$, which is computed as the gradient of the mesh position $\boldsymbol{x}$. The $\nabla_X g^{-1}$ term is computed as

$$\nabla_X g^{-1} = \frac{-1}{g^2}\nabla_X g = \frac{-1}{g^2}\nabla_X \det \boldsymbol{G} \tag{18}$$

where the gradient of $\det \boldsymbol{G}$ is computed component-wise using the formula

$$\frac{d \det(\boldsymbol{G})}{dX_i} = \det(\boldsymbol{G}) \operatorname{tr}\left(\boldsymbol{G}^{-1}\frac{d\boldsymbol{G}}{dX_i}\right) \tag{19}$$

with $d\boldsymbol{G}/dX_i$ computed numerically.

Next we consider the computation of the mesh velocity $\boldsymbol{v} = \partial \boldsymbol{x}/\partial t$. Depending on the specifics of the problem there are a few different ways in which the mesh velocities may be calculated. In the simplest case the mesh motion may be given as an analytic function of time, in which case we may simply take the derivative to compute the mesh velocity. For example, if the mesh position is given by an interpolation of a deformation of the boundary using radial basis functions [3], it is often natural to use the same interpolation process to interpolate boundary deformation velocities into mesh velocities.

However, if only numerical values of the mesh position are available we must resort to a numerical differentiation procedure to compute the mesh velocity. It is desirable to use a definition which uses specific details of the time integrator used for the time integration. We say a method of calculating mesh velocities is consistent if, when integrated using the numerical method they recover the numerical mesh positions. This was done in, for example, [9] for several explicit multistep and Runge-Kutta methods. Here we show an extension of this idea to the case of diagonally implicit Runge-Kutta methods.

Given the mesh position $\boldsymbol{x}_i$ at stages $i = 1, \ldots, s$, we say the mesh velocities $\boldsymbol{v}_i$ at stages $i = 1, \ldots, s$ are *stage consistent* if

$$\boldsymbol{x}_i = \boldsymbol{x}_0 + \Delta t \sum_{j=1}^{s} a_{ij}\boldsymbol{v}_j, \quad i = 1, \ldots, s \tag{20}$$

where $\boldsymbol{x}_0$ is the initial mesh position and a_{ij} are the Runge-Kutta coefficients.

In the case when A is of full rank, i.e., a fully implicit Runge-Kutta or diagonally implicit Runge-Kutta method, some algebraic manipulation allows us to write the stage mesh velocity as a linear combination of the mesh positions

$$\boldsymbol{v}_i = \sum_{j=1}^{s} (A^{-1})_{ij} \frac{\boldsymbol{x}_j - \boldsymbol{x_0}}{\Delta t}, \quad i = 1, \ldots, s. \tag{21}$$

For a diagonally implicit Runge-Kutta method A^{-1} is lower triangular so each stage mesh velocity may be calculated using only mesh positions from that and previous stages. This preserves an obvious time dependency relationship and may be desirable, especially in cases when the stage mesh position is calculated on-the-fly from current stage variables as in the case of a fluid-structure interaction problem [5].

If the first stage of the Runge-Kutta scheme is explicit, say in an ESDIRK method, the coefficient matrix A will not be invertible and thus a different approach is required. In fact, it is clear that the stage mesh velocities $\boldsymbol{v}_i$ are not even uniquely defined in terms of the stage mesh positions $\boldsymbol{x}_i$. In this case it is natural to require an initial mesh velocity $\boldsymbol{v}_0$. The first (explicit) stage mesh velocity $\boldsymbol{v}_1$ is set to this value and mesh velocities at later stages are then uniquely given by Eq. (20). The ESDIRK schemes we have considered all have the first same as last property, that is, the final stage coefficients are the same as the weights, and so it is natural to use the mesh velocity at the final stage $\boldsymbol{v}_s$ of one timestep as the initial mesh velocity in the following timestep.

4 Mesh Deformation

For the mesh deformation, we use a quasi-static hyperelastic neo-Hookean formulation [7]. The deformation is given by a mapping $\boldsymbol{x}(\boldsymbol{X})$ which maps a point $\boldsymbol{X}$ in the unstretched reference configuration Ω to its location $\boldsymbol{x}$ in the deformed configuration. We differentiate $\boldsymbol{x}$ with respect to space to obtain the *deformation gradient tensor* $\boldsymbol{G}$ as $\boldsymbol{G} = \nabla_{\boldsymbol{X}} \boldsymbol{x}(\boldsymbol{X})$. The governing equations are then given by

$$-\nabla \cdot \boldsymbol{P}(\boldsymbol{G}) = \boldsymbol{b} \qquad \text{in } \Omega, \tag{22}$$

$$\boldsymbol{x} = \boldsymbol{x}_D \qquad \text{on } \Gamma, \tag{23}$$

where $\boldsymbol{P}$ is the first Piola-Kirchhoff stress tensor and $\boldsymbol{b}$ is an external body force per unit reference volume, which we typically assume is zero. On the boundary of the domain $\Gamma = \partial\Omega$ we have assumed Dirichlet boundary conditions, i.e., specified material positions $\boldsymbol{x}_D$.

In this work we use a compressible neo-Hookean material model, with first Piola-Kirchhoff stress tensor given by [7]

$$\boldsymbol{P}(\boldsymbol{G}) = \frac{\partial W}{\partial \boldsymbol{G}} = \mu J^{-2/3}\left(\boldsymbol{G} - \frac{1}{3}\operatorname{tr}(\boldsymbol{G}\boldsymbol{G}^T)\boldsymbol{G}^{-T}\right) + \kappa(J-1)J\boldsymbol{G}^{-T}, \tag{24}$$

where the constants μ and κ are the shear and bulk modulus of the material. For two-dimensional problems we use a plane strain formulation in which we treat the stretching in the third dimension as constant.

To develop a finite element formulation for (22)–(23), we define the space of continuous piecewise polynomials of degree p:

$$\mathscr{V}_h^p = \left\{\boldsymbol{v} \in [\mathscr{C}_0(\Omega)]^3 \;\middle|\; \boldsymbol{v}|_K \in [\mathscr{P}_p(K)]^3 \;\forall K \in \mathscr{T}_h\right\}, \tag{25}$$

where the domain Ω is divided into elements $\mathscr{T}_h = \{K\}$, and $\mathscr{P}_p(K)$ is the space of polynomial functions of degree at most $p \geq 1$ on K. Furthermore, we define the subspaces of functions in $\mathscr{V}_h^p$ that satisfy the non-homogeneous as well as the homogeneous Dirichlet boundary conditions:

$$\mathscr{V}_{h,D}^p = \left\{\boldsymbol{v} \in \mathscr{V}_h^p \;\middle|\; \boldsymbol{v}|_{\partial V} = \boldsymbol{x}_D^p\right\}, \tag{26}$$

$$\mathscr{V}_{h,0}^p = \left\{\boldsymbol{v} \in \mathscr{V}_h^p \;\middle|\; \boldsymbol{v}|_{\partial V} = 0\right\}. \tag{27}$$

Here, $\boldsymbol{x}_D^p$ is a suitable projection of $\boldsymbol{x}_D$ onto the space of piecewise polynomials of order p defined over ∂V. By multiplying (22) by an arbitrary test function $\boldsymbol{z} \in \mathscr{V}_{h,0}^p$, integrating over the domain V, and integrating by parts, we obtain our finite element formulation: find $\boldsymbol{x}_h \in \mathscr{V}_{h,D}^p$ such that for all $\boldsymbol{z} \in \mathscr{V}_{h,0}^p$,

$$\int_V \boldsymbol{P}(\boldsymbol{G}(\boldsymbol{x}_h)) : \nabla \boldsymbol{z}\, dV = \int_V \boldsymbol{b} \cdot \boldsymbol{z}\, dV. \tag{28}$$

This system of equations is generated using standard finite element techniques. Using nodal basis functions, the computed elemental residuals are assembled into a global discrete system of equations $\bar{\boldsymbol{r}}(\bar{\boldsymbol{x}}) = \boldsymbol{0}$. We solve this system using a standard Newton method, which involves the Jacobian matrix $\boldsymbol{K} = \partial\bar{\boldsymbol{r}}/\partial\bar{\boldsymbol{x}}$ which is evaluated for each element and assembled into a global matrix. The prescribed displacement at the boundary nodes is imposed by elimination of the corresponding variables from the system of equations. The linear systems that arise are solved using a direct sparse solver. For problems with complex deformations, we use the simple homotopy approach described in [12] to obtain global convergence.

5 Results

5.1 Deformed Mesh Quality

As a test problem to demonstrate the quality of the nonlinear elasticity based mesh deformation, we consider a square with a smaller square removed from the center:

$$\Omega = [0.0, 1.0]^2 \setminus [0.4, 0.6]^2. \tag{29}$$

The domain is triangulated in a structured fashion using isoparametric elements of polynomial degree 2. We fix the outer boundary of the domain and rotate the inner boundary about the center, $[0.5, 0.5]^T$, by an angle θ. Clearly, for increasing θ any deformation strategy will eventually fail and produce invalid elements. However, for moderate angles this is a good test case for comparing different methods.

We first perform the mesh deformation using the commonly used radial basis function interpolation [3]. Here we seek an interpolant giving the deformed mesh position $\boldsymbol{x}$ as function of the position $\boldsymbol{X}$ in the reference mesh, of the form

$$\boldsymbol{x}(\boldsymbol{X}) = \sum_{j=1}^{n} \boldsymbol{\alpha}_j \phi_j(\|\boldsymbol{X} - \boldsymbol{X}_j\|_2 / r_j) + \boldsymbol{p}(\boldsymbol{X}) \tag{30}$$

where $\boldsymbol{X}_j$ are a set of control points, ϕ_j radial basis functions, r_j characteristic radii, and $\boldsymbol{p}$ a linear polynomial. The coefficients $\boldsymbol{\alpha}_j$ and coefficients of the polynomial $\boldsymbol{p}$ are found by imposing the value of $\boldsymbol{x}$ at the control points $\boldsymbol{X}_j$, and additionally requiring that the function preserves polynomial deformations of degree less than or equal to the degree of $\boldsymbol{p}$. We solve the resulting linear system using a direct solver.

There are many choices of radial basis functions, but on the recommendation of [3] we use a C^2 compactly supported function

$$\phi(r) = \begin{cases} (1-r)^4(4r+1) & \text{if } 0 \le r \le 1 \\ 0 & \text{if } 1 \le r. \end{cases} \tag{31}$$

with characteristic radius 1, which gave the best results for several different RBF interpolants and radii examined. The resulting deformed mesh for rotations of 30°, 60°, 90°, and 120° are shown in Fig. 1 (top). Here we see that this mesh deformation method does a very good job with the small deformation (30°), but has some difficulty with larger deformations. In particular, some elements have already inverted (i.e., the determinant of the local Jacobian mapping is negative) by 90°.

In addition one can easily show that a non-inverting deformation of a 180° rotation of the inner square is not possible using this technique for any choice of radial basis function interpolant. To see this, recall that the deformed position of any node depends linearly on the positions of the boundary nodes. Since a $+180°$ and $-180°$ rotation of the inner square would lead to the same locations of the boundary

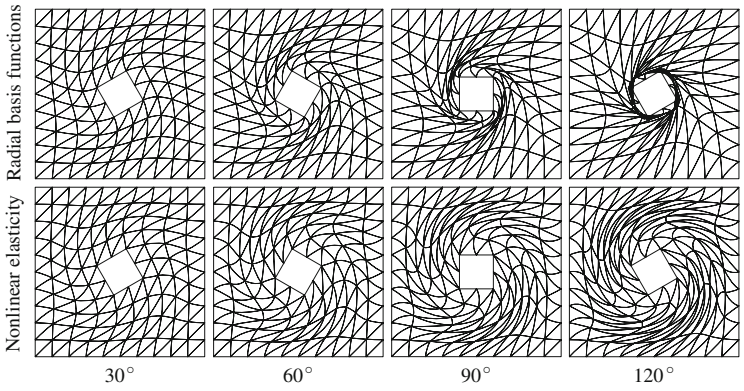


Fig. 1 Mesh deformation using radial basis function interpolation (*top*) and using the quasi-static nonlinear elasticity method (*bottom*)

nodes, the RBF interpolant is unable to distinguish between these two cases. In particular, a curve connecting the left outer boundary to the left inner boundary in the undeformed mesh would have to pass both under the square in the $+180°$ rotation and under the square in the $-180°$ rotation, which is not possible.

In the bottom plots of Fig. 1 we repeat the same experiment, this time using the nonlinear elasticity deformation method. Here we set $\nu = 0.40$ and a spatially varying E according to

$$E(x) = 1 + \frac{100}{1 + (d(x)/d_0)^2} \tag{32}$$

where $d_0 = 0.05$ and $d(x) = \max\{0.0, \min(\text{dist}(x, \Gamma_{in}) - d_0, \text{dist}(x, \Gamma_{out}) + 2d_0)\}$. Here, Γ_{in} and Γ_{out} are the inner and outer boundaries. This expression for E was chosen to cause more deformation to occur in the intermediate region between the inner and the outer boundaries, which is desirable. As the figure shows, the resulting mesh still has not inverted, even at a rotation of 120°, although the element quality does become quite poor for the larger rotations. These results are significantly better than what we could achieve even with the best possible parameters for the RBF deformation.

Because the deformation equations are nonlinear, the system may exhibit multiple solutions for a given configuration of the boundary. In particular, the zero that we find is going to be dependent upon the initial approximation in the Newton solver. In particular this means that we are in principle able to construct deformed meshes corresponding to $+180°$ and $-180°$ rotations of the inner boundary using essentially a homotopy of intermediate rotations.

5.2 Convergence Test, Expanding Pressure Wave

To study the accuracy of the Arbitrary Lagrangian-Eulerian formulation, we consider a case with a specified analytic mesh deformation and compare the spatial convergence for several deformation strategies. As a non-trivial test problem, we consider a viscous flow problem with a small Gaussian perturbation in the density and the pressure of an otherwise constant state.

As the domain we choose $\Omega = [0, 1]^2$ with far-field boundary conditions on the left, bottom, and right walls and an adiabatic no-slip condition on the top wall. The momentum is initialized as $\rho \boldsymbol{u} = 0$, and the spatially varying initial density and pressure are $\rho = \rho_\infty \varphi(x)$ and $p = p_\infty \varphi(x)$, respectively, where

$$\varphi(x) = 1 + d_0 \exp(\|x - x_0\|_2^2 / r_0^2) \tag{33}$$

and the non-dimensionalized far-field density $\rho_\infty = 1$. The far-field pressure p_∞ is calculated using the non-dimensionalized sound speed $a_\infty = 5$. The perturbation parameters where chosen as $d_0 = 0.1$, $r_0 = 0.1$, and $x_0 = [0.5, 0.7]^T$.

The fluid is modeled using the compressible Navier-Stokes equations (1)–(3), with dynamic viscosity $\mu = 1/1000$. The background mesh was deformed using an analytic mapping

$$x(X, Y, t) = X + A \sin(2\pi X) \sin(2\pi Y) \sin(2\pi f t), \tag{34}$$

$$y(X, Y, t) = Y + A \sin(2\pi X) \sin(2\pi Y) \sin(4\pi f t), \tag{35}$$

with amplitude $A = 0.05$ and frequency $f = 20$.

For the time-integration, we use an explicit RK4 scheme with a sufficiently small Δt so that the spatial errors are dominating. We integrate until a final time of $T = 1/20$, which is one entire period of the mesh deformation so that the mesh starts in an undeformed configuration at time $t = 0$ and returns to an undeformed configuration at time $t = T$. This allows us to measure the accuracy of the ALE mapping by comparing the numerical solution of the problem at $t = T$ to one obtained on a non-deforming mesh.

The domain Ω is discretized using a regular grid of triangles with element size h, and we use polynomial degrees $p = 1$ through 5 within each element. Numerically the mesh deformation is represented on each element using either a linear $p = 1$ representation or an isoparametric representation. A time series of the solution on two meshes is shown in Fig. 2.

We observe that both deformation strategies are able to accurately capture the radiating pressure wave. Notice that when we represent the mesh deformation using $p = 1$ elements the resulting map $\boldsymbol{x}(\boldsymbol{X}, t)$ is piecewise linear and hence the ALE formulation in Sect. 2 simplifies significantly as the deformation gradient $\boldsymbol{G}$ and mapping determinant g are both constant. This also simplifies the calculation of the viscous derivative as an entire term $\nabla_{\boldsymbol{X}}(g^{-1})$ vanishes. However, a $p = 1$

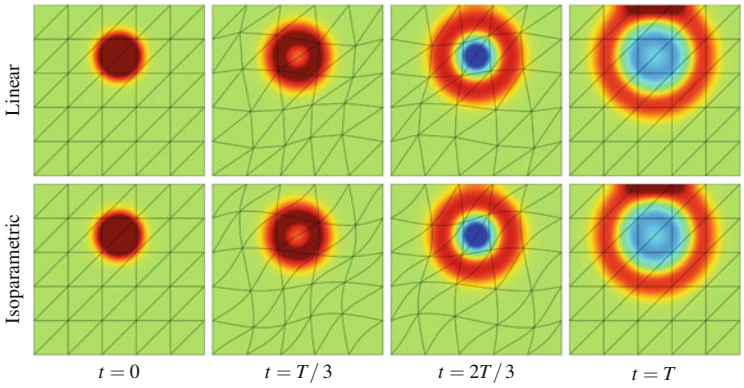


Fig. 2 An expanding pressure wave on a deforming mesh using a linear deformation (*top*) and an isoparametric deformation (*bottom*), for polynomial degrees $p = 5$. (Pressure)

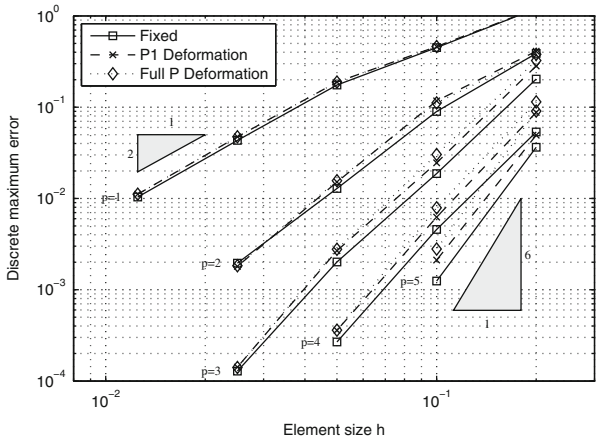


Fig. 3 Spatial convergence in the discrete maximum error at the final simulation time for an expanding pressure wave for meshes of polynomial degree $p = 1$ to 5. The deformation is either linear (P1) or isoparametric (Full P)

mesh deformation representation is likely not able to capture complicated boundary motions as accurately as the isoparametric $p = 5$ representation.

The relative accuracy of using a $p = 1$ deformation instead of an isoparametric can be discussed. We would expect the linear $p = 1$ mapping to produce slightly better results because it introduces less variations in the solution fields. This intuition is reflected in a numerical convergence plot which is shown in Fig. 3. Here we measure the error in the solution at $t = T$ in the discrete maximum norm for a non-deforming fixed mesh, a $p = 1$ deformation, and an isoparametric deformation ('Full P') for elements of order $p = 1$ through 5. In general we observe convergence orders at the expected $p + 1$ rate for all the cases. For the lower p the difference in

accuracy between the three methods is difficult to ascertain. However, for higher p there is a notable difference in accuracy between the three methods, with the fixed mesh being the most accurate and the isoparametric deformation being the least accurate.

From this experiment we can generally recommend using a linear representation of the mesh deformation if possible. If not, the isoparametric deformation gives adequate results and is able to represent a much larger class of deformations. Mixed approaches should be feasible and represent a possible compromise.

5.3 *Flapping Wing Applications*

As two final examples, we show how our methods have been successfully applied to flapping flight problems. In Fig. 4, the simulation of a pair of flapping bat wings is shown. A representative surface mesh frame was chosen for the reference domain, and a high-quality tetrahedral mesh of the domain was generated (left plot). This mesh was then deformed for each subsequent time frame using the nonlinear elasticity approach (middle plot), and a preliminary simulation at a low Reynolds number was performed (right plot).

The second example is from [16], where several energetically optimal flapping wing designs were computed using a multi-fidelity approach. These designs were simulated using the high-fidelity DG framework presented here. Figure 5 (top) shows the mesh deformation, and the bottom plots show flow fields from a sample design.

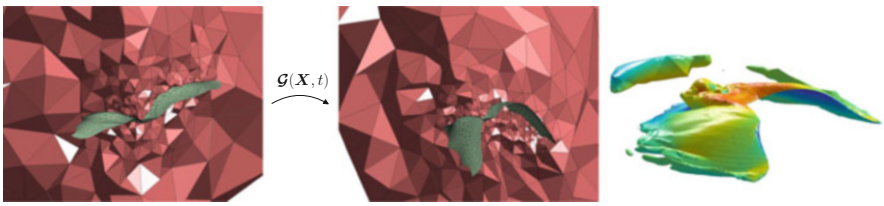


Fig. 4 A large deformation example of the flapping flight of a bat. The reference mesh (*left*) is deformed in time using the nonlinear elasticity approach which maintains the high-quality of the elements (*middle*). The *right plot* shows a sample solution field

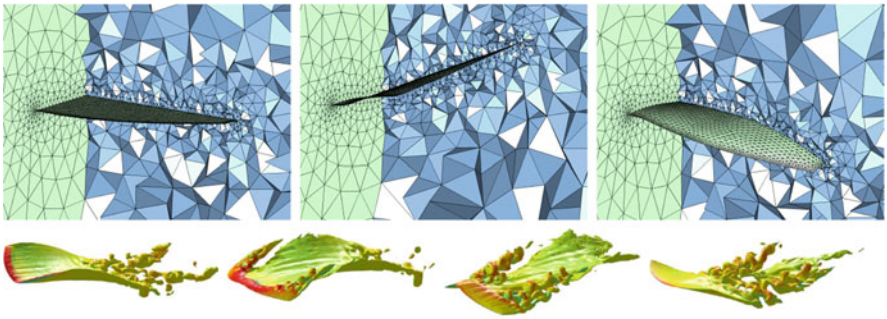


Fig. 5 High-order simulation of energetically optimal flapping wings (from [16]). The figures show a reference mesh, two deformed meshes, and some flow fields for a sample design

References

1. R. Alexander, Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. SIAM J. Numer. Anal. **14**(6), 1006–1021 (1977)
2. B. Cockburn, C.W. Shu, Runge-Kutta discontinuous Galerkin methods for convection-dominated problems. J. Sci. Comput. **16**(3), 173–261 (2001)
3. A. de Boer, M.S. van der Schoot, H. Bijl, Mesh deformation based on radial basis function interpolation. Comput. Struct. **85**, 784–795 (2007)
4. C. Farhat, P. Geuzaine, Design and analysis of robust ALE time-integrators for the solution of unsteady flow problems on moving grids. Comput. Methods Appl. Mech. Eng. **193**(39–41), 4073–4095 (2004)
5. B. Froehle, P.O. Persson, A high-order discontinuous Galerkin method for fluid-structure interaction with efficient implicit-explicit time stepping. J. Comput. Phys. **272**, 455–470 (2014)
6. J.S. Hesthaven, T. Warburton, Nodal discontinuous Galerkin methods, in *Texts in Applied Mathematics*, vol. 54 (Springer, New York, 2008). Algorithms, analysis, and applications
7. G.A. Holzapfel, *Nonlinear Solid Mechanics* (Wiley, Chichester, 2000). A continuum approach for engineering
8. I. Lomtev, R.M. Kirby, G.E. Karniadakis, A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. J. Comput. Phys. **155**(1), 128–159 (1999)
9. C.A.A. Minoli, D.A. Kopriva, Discontinuous Galerkin spectral element approximations on moving meshes. J. Comput. Phys. **230**(5), 1876–1902 (2011)
10. J. Peraire, P.O. Persson, The compact discontinuous Galerkin (CDG) method for elliptic problems. SIAM J. Sci. Comput. **30**(4), 1806–1824 (2008)
11. P.O. Persson, J. Peraire, Newton-GMRES preconditioning for discontinuous Galerkin discretizations of the Navier-Stokes equations. SIAM J. Sci. Comput. **30**(6), 2709–2733 (2008)
12. P.O. Persson, J. Peraire, Curved mesh generation and mesh refinement using Lagrangian solid mechanics, in *47th AIAA* (ASME, Orlando, 2009). AIAA-2009-949
13. P.O. Persson, J. Bonet, J. Peraire, Discontinuous Galerkin solution of the Navier-Stokes equations on deformable domains. Comput. Methods Appl. Mech. Eng. **198**(17–20), 1585–1595 (2009)
14. C.S. Venkatasubban, A new finite element formulation for ALE (arbitrary Lagrangian Eulerian) compressible fluid mechanics. Int. J. Eng. Sci. **33**(12), 1743–1762 (1995)
15. Z. Wang et al., High-order CFD methods: current status and perspective. Int. J. Numer. Methods Fluids **72**(8), 811–845 (2013)
16. D.J. Willis, P.O. Persson, Multiple-fidelity computational framework for the design of efficient flapping wings. AIAA J. **52**(12), 2840–2854 (2014)

Exploiting Superconvergence Through Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering

Jennifer K. Ryan

Abstract There has been much work in the area of superconvergent error analysis for finite element and discontinuous Galerkin (DG) methods. The property of superconvergence leads to the question of how to exploit this information in a useful manner, mainly through superconvergence extraction. There are many methods used for superconvergence extraction such as projection, interpolation, patch recovery and B-spline convolution filters. This last method falls under the class of Smoothness-Increasing Accuracy-Conserving (SIAC) filters. It has the advantage of improving both smoothness and accuracy of the approximation. Specifically, for linear hyperbolic equations it can improve the order of accuracy of a DG approximation from $k+1$ to $2k+1$, where k is the highest degree polynomial used in the approximation, and can increase the smoothness to $k-1$. In this article, we discuss the importance of overcoming the mathematical barriers in making superconvergence extraction techniques useful for applications, specifically focusing on SIAC filtering.

1 Introduction

Many numerical methods experience a phenomenon known as superconvergence. **Superconvergence** is higher than theoretical predicted convergence:

$$|(u-u_h)(\xi)| \leq Ch^{r+\sigma},$$

where r is the expected convergence and $\sigma > 0$ [22]. So-called "natural" Superconvergence occurs when the function is evaluated at a point and compared with the exact solution. We can create globally superconvergent solutions through post-processing the approximation. In this article we focus on a specific post-processing technique that uses B-spline convolution to obtain a superconvergent

J.K. Ryan (✉)
University of East Anglia, Norwich, UK
e-mail: Jennifer.Ryan@uea.ac.uk

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_6

approximation. Specifically, we concentrate on SIAC filters, which have their roots in work by Bramble and Schatz [2] and Cockburn, Luskin, Shu and Süli [6].

2 Motivation and Background

We frame our discussion in the context of a linear hyperbolic equation with smooth initial data,

$$u_t + \sum_{i=1}^{d} A_i \frac{\partial}{\partial x_i} u + A_0 u = 0, \qquad \mathbf{x} \in \Omega \times [0, T], \tag{1}$$

$$u(\mathbf{x}, 0) = u_0(\mathbf{x}), \qquad \mathbf{x} \in \Omega. \tag{2}$$

We also assume periodic boundary conditions for simplicity. For these types of equations, the superconvergence property is straight-forward to prove in both the pointwise setting and in terms of the negative-order norm.

2.1 Discontinuous Galerkin Methods

The important components that aid in creating a superconvergent approximation from a discontinuous Galerkin solution are that

1. The approximation space consists of piecewise polynomials of degree $\leq k$:

$$V_h^k = \{v \in L^2(\Omega) : v \in \mathbb{P}^k(\tau_e), j = 1, \ldots, N\} \tag{3}$$

   where τ_e are the elements in the associated mesh and $\Omega = \cup_e \tau_e$.
2. The variational formulation of the discontinuous Galerkin scheme:

$$\int_{\tau_e} (u_h)_t v_h(x)\, dx - \sum_{i=1}^{d} \int_{\tau_e} A_i u_h(\mathbf{x}, t)(v_h)_{x_i}(x)\, dx + \int_{\tau_e} A_0 u_h(\mathbf{x}, t) v_h\, d\mathbf{x}$$

$$+ \sum_{i=1}^{d} \int_{\partial \tau_e} \widehat{A_i u_h} \hat{n}_i\, v_h, ds = 0$$

3. The weak continuity at the element interfaces that are enforced through the choice of the fluxes in the discontinuous Galerkin scheme.

The reader is advised to consult [5] for a more detailed discussion of the discontinuous Galerkin method.

2.2 Error Estimates: Convergence and Superconvergence

Assuming the initial condition is regular enough, the errors in L^2 for the DG approximation are given by

$$\|u - u_h\|_0 \leq C\, h^{k+1} |u_0|_{H^{k+2}} \tag{4}$$

[5]. However, Adjerid et al. noted that the approximation has the property of pointwise superconvergence [1]. That is, the *local* error at the"outflow" edge converges at twice the usual convergence rate,

$$(u - u_h)(x^-_{j+1/2}) = \alpha_{k+1} \frac{(-a)^{k+1} k!}{2k+1} h^{2k+2} + \mathcal{O}(h^{2k+3}) \tag{5}$$

for linear equations such as $u' - au = 0$. This occurs at the roots of the right Radau polynomial.

3 Extracting Superconvergence

We would like to turn the local superconvergence property into a globally superconvergent solution. There are many different options for this to be accomplished. A few are to interpolate using superconvergent fluxes [4, 15], elementwise post-processing [3], or convolution kernel post-processing [2, 6]. We focus on the latter, specifically the Smoothness-Increasing Accuracy-Conserving filter [10, 18, 21]. This last technique allows for *global superconvergence* and *smoothness*.

3.1 Smoothness-Increasing Accuracy-Conserving (SIAC) Filtering

The SIAC filter has its roots in an accuracy-enhancing post-processor. Motivated by the work of Mock and Lax [14], Bramble and Schatz introduced a central B-spline kernel to post-process finite element approximations to elliptic equations [2]. This was also explored from a Fourier perspective and for derivative filtering by Thomeé [20]. Cockburn, Luskin, Shu and Süli then extended it to discontinuous Galerkin approximations to linear hyperbolic equations [6]. It was further extended to a broader class of problems in [7, 8, 11].

The basic idea of the original post-processor, $u^*(x)$, is to convolve the numerical approximation with a B-spline kernel,

$$u^*(x) = (K_H^{2(k+1),k+1} * u_h(\cdot, T))(x). \tag{6}$$

This allows us to achieve $u - u_h \sim \mathcal{O}(h^{2k+1})$ in L^2 as shown in [6]. A more general form of the B-spline kernel, $K_H^{2(k+1),k+1}(x)$, will be discussed in Sect. 3.2.

The post-processor is useful for removing the highly oscillatory errors in the discontinuous Galerkin approximation. The result is a solution that has increased smoothness and accuracy.

3.2 The SIAC Kernel

The SIAC kernel is a more general form of the B-spline kernel above. It is a linear combination of suitably scaled B-spline translates,

$$K_H^{(r+1,\ell)}(x) = \frac{1}{H}\sum_{\gamma=0}^{r} c_\gamma^{(r+1,\ell)} \psi^{(\ell)}\left(\frac{x}{H} - x_\gamma\right), \tag{7}$$

where $r+1$ is the number of B-splines in the kernel and ℓ is the order of the B-splines. In (7), c_γ are weights of the B-splines, $\psi^{(\ell)}(x)$, and are determined by reproducing polynomials of degree less than or equal to r. For the original kernel $r = 2k$, $\ell = k+1$ and $x_\gamma = -k + \gamma$ as given in (6). In the more general SIAC filter, x_γ depends on the point being evaluated and we have more flexibility both in the number of B-splines and order of the B-splines.

Central B-splines are defined as $\psi^{(1)} = \chi_{[-\frac{1}{2},\frac{1}{2}]}$, $\psi^{(\ell)} = \psi^{(\ell-1)} * \chi_{[-\frac{1}{2},\frac{1}{2}]}, \ell \geq 2$. Here, χ is equal to one on $[-\frac{1}{2},\frac{1}{2}]$ and otherwise is zero. The central B-splines that form the post-processed solution are chosen because of their compact support of $\text{supp}\{\psi^{(\ell)}\} = [-\frac{\ell}{2},\frac{\ell}{2}]$. Further, they are easy to compute through a recurrence relation. Lastly, there is a natural relation between their derivatives and divided differences: $D^\alpha \psi^{(\ell)} = \partial_H^\alpha \psi^{(\ell-\alpha)}$. In Fig. 1 a plot of the B-splines making up the convolution kernel as well as the convolution kernel is shown for $k = 2$.

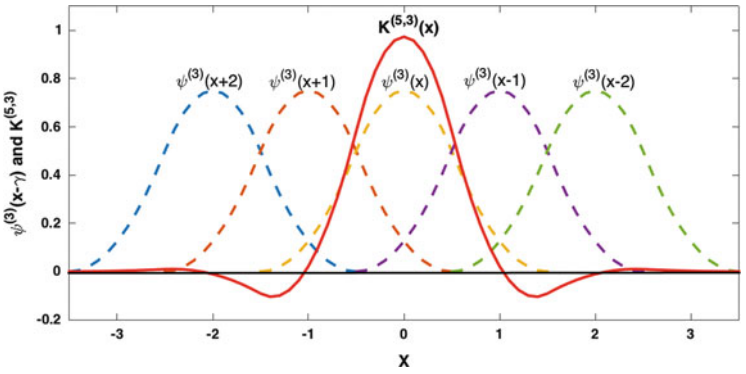


Fig. 1 *Dashed lines*: the B-splines, $\psi^{(3)}(x+k-\gamma)$, $\gamma = 0,\ldots,4$ used in the $k=2$ kernel. *Solid line*: the kernel, $K_H^{5,3}(x)$ for $k=2$

The convolution coefficients that weigh the B-spline translates are found by using the property of polynomial reproduction.

As an example, we give the original symmetric B-spline kernel for the second order approximation, $k = 1$. The kernel coefficients are found by using $K_h^{4,2} * p = p$ for $p = 1, x, x^2$. This creates the kernel

$$K^{4,2}(x) = \frac{-1}{12}\psi^{(2)}(x-1) + \frac{7}{6}\psi^{(2)}(x) - \frac{1}{12}\psi^{(2)}(x+1). \tag{8}$$

To summarise, the convolution kernel is designed to extract higher order accuracy through polynomial reproduction. It induces smoothness of $\mathcal{C}^{\ell-2}$ through the convolution with the B-splines and uses a local stencil of size $(r+\ell)H$. The kernel is a polynomial of degree $\ell - 1$, making the post-processed solution a polynomial of degree $\ell + k$. It has theoretical and numerical convergence of $\mathcal{O}(h^s)$, $s = \min\{r+1, 2k+1\}$ in both L^2- and L^∞-norms for linear hyperbolic equations over uniform meshes.

3.3 Implementing the Post-Processor

Assuming the one-dimensional discontinuous Galerkin approximation can be written as

$$u_h(x,t) = \sum_{n=0}^{k} u_e^{(n)}(t)\phi_e^{(n)}(x), \qquad x \in \tau_e, \tag{9}$$

where $\phi_e^{(n)}(x)$ are the basis functions for the DG approximation. Using this modal form of the DG approximation, the post-processed solution can be written as

$$u^*(x) = \sum_{j=-p'}^{p'} \sum_{n=0}^{k} C(j,n,k,x)\, u_{e+j}^{(n)} \tag{10}$$

where $p' = \lceil \frac{r+\ell}{2} \rceil$ and

$$C(j,n,k,x) = \frac{1}{h}\sum_{\gamma=0}^{r} c_\gamma^{r+1,\ell} \underbrace{\int_{I_{e+j}} \psi^{(\ell)}\left(\frac{y-x}{h} - \gamma\right)\phi_{e+j}^{(n)}(y)\,dy}_{\in \mathbb{P}^{\ell+k}}. \tag{11}$$

The multi-dimensional kernel is a tensor product of the one-dimensional kernel. For example, in two-dimensions,

$$K_H(x,y) = \frac{1}{H_x H_y} \sum_{\gamma_x=0}^{r} \sum_{\gamma_y=0}^{r} c_{\gamma_x}^{r+1,\ell} c_{\gamma_y}^{r+1,\ell} \psi^{(\ell)} \left(\frac{x}{H_x} - x_{\gamma_x} \right) \psi^{(\ell)} \left(\frac{y}{H_y} - y_{\gamma_y} \right). \tag{12}$$

It is expected that the kernel can be applied to $\mathbb{Q}^k$-polynomial approximations, but it is also effective for $\mathbb{P}^k$-polynomial approximations.

3.4 Convergence of the SIAC Filtered Solution

Let $u_h^*(x,T) = K_H * u_h$ be the post-processed DG approximation at the *final time.* Then the errors for the post-processed solution are given by

$$\|u - u_h^*(x,T)\|_{0,\Omega} \leq \underbrace{\|u - u^*\|_{0,\Omega}}_{\text{Exact filtered}} + \underbrace{\|(u-u_h)^*\|_{0,\Omega}}_{\text{DG errors}}. \tag{13}$$

The estimate for the first term comes about from the ability of the kernel to reproduce polynomials of degree r. Then, using a Taylor expansion we obtain $\|u - K_h * u\|_\Omega \leq Ch^{r+1}$ [6, 9]. The second term can be bounded by the negative-order norm [2, 6]. If we can show the negative-order norm is of higher order, then we can demonstrate superconvergence of the filtered solution.

In Fig. 2 a comparison of the convergence rates and errors between the discontinuous Galerkin approximation and the SIAC filtered approximation is given. If

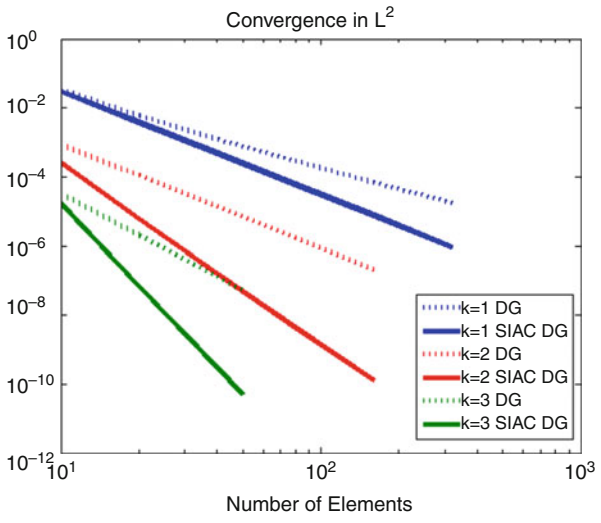


Fig. 2 A comparison of the convergence rates and errors between the discontinuous Galerkin approximation and the SIAC filtered DG approximation

we consider the $k = 1$ filtered approximation and compare it with the $k = 2$ DG approximation, we can see that although they have the same convergence rate, the errors for the $k = 2$ approximation are better.

3.5 Applications

Currently, the applications of SIAC filtering include extracting accuracy out of existing code [18] and visualization filtering [19]. However, there is promising relations to image processing [13, 23] as well as potential in LES filtering [7, 8].

3.6 Interesting Challenges

The challenge in making SIAC filtering applicable to broader areas of applications include: A negative-order norm estimate that depends upon the PDE, the ability to extract derivative information, filtering near a boundary, and most importantly mesh geometry. In the following sections we discuss the challenges in extending SIAC filtering to a range of applications.

4 The Error Estimate

Recall that in Eq. (13) the SIAC filtered error estimate is controlled by our ability to prove superconvergence in the negative-order norm, where the negative-order norm is given by

$$\|\partial_H^\alpha (u - u_h)\|_{-(k+1),\Omega} = \sup_{\phi \in \mathcal{C}_0^\infty(\Omega)} \frac{(\partial_H^\alpha (u - u_h), \phi)_\Omega}{\|\phi\|_{k+1,\Omega}} \leq C h^{2k+1} \|u_0\|_{k+1,\mathcal{D}\Omega_1} \tag{14}$$

if $H = h$ [6].

For the negative-order norm, we actually only need to consider the numerator in Eq. (14). In general, the estimate depends on defining a suitable dual equation and we are able to prove $\|u - u_h^*\| \leq Ch^{2k+m}$. Details of the existing estimates for various equations are provided in Table 1.

Table 1 SIAC Filter error estimates for various types of equations

m	$s = 2k + m$	Equation
2	$2k + 2$	Elliptic (FEM) [2]
1	$2k + 1$	Linear Hyperbolic (DG) [6]
$1 \leq m \leq 2$	$2k + 1 \leq s \leq 2k + 2$	Convection-diffusion (DG) [7]
1	$2k + 1$	Variable-Coefficient Hyperbolic (DG) [11]
$0, \frac{1}{2}, 2$	$2k \leq s \leq 2k + 2$	Nonlinear hyperbolic (DG) [8]

5 Derivative SIAC Filtering

Another interesting aspect of SIAC filtering is that it allows us to create a superconvergent approximation to derivatives. In general, the approximation obtained via a DG method will give $\|\partial^\alpha (u - u_h)\| \leq Ch^{k+1-\alpha}$ for the derivatives. This makes it impossible to obtain a good second order derivative approximation for $k = 1$. However, using SIAC filtering makes it possible to obtain higher order derivatives even for a piecewise linear approximation. In order to obtain a superconvergent derivative approximation, there are two options: accept a reduction in order of accuracy by taking the derivative of the filtered solution, or forming a kernel that uses higher-order B-splines whose errors do not reduce in order with differentiation. Each method has its advantages and disadvantages and both will give a superconvergent derivative approximation.

In the first method, we compute the derivative of the SIAC filtered solution directly. This gives

$$\frac{d^\alpha}{dx^\alpha}\left(K_h^{r+1,\ell} * u_h(\cdot, T)\right)(x) = \frac{d^\alpha}{dx^\alpha}\left(\frac{1}{H}\int_{\mathbb{R}} K_H^{(r+1,\ell)}\left(\frac{x-y}{h}\right) u_h(y,t)\, dy\right). \tag{15}$$

Recall that the post-processed approximation induces smoothness of $\mathcal{C}^{\ell-2}$ and is up to $2k + 1$th-order accurate. If we calculate the derivative of the post-processing polynomial directly we would then have $\sim \mathcal{O}(h^{\min\{2k+2,r+2\}-\alpha})$, for $\alpha \leq \ell - 1$, which would give a reduced order of accuracy with each successive derivative. Further, the oscillations in the error increase [18]. This method may be more advantageous if only a first or second derivative is needed.

There is an alternative that allows us to obtain the same superconvergent approximation to the derivatives. That is, we can obtain a $2k + 1$ order accuracy approximation to the αth-derivative using higher order splines in our kernel [16, 20]. This gives a derivative approximation whose order or convergence is independent of α. The derivative kernel is defined as

$$K_H^{r+1,\alpha,\ell}(x) = \frac{1}{H}\sum_{\gamma=0}^{r} d_\gamma^{r+1,\alpha,\ell}\psi^{(\ell+\alpha)}\left(\frac{x}{H} - x_\gamma\right). \tag{16}$$

The difference to the kernel in Equation (7) is that it uses *smoother* B-splines. Note that smoother B-splines give an increased support size. Further, computing the αth derivative only requires computing the convolution of translations of the B-spline $\psi^{(\ell)}$ with u_h. This allows us to obtain the error estimate:

Theorem 1 (Ryan and Cockburn [16]) *Let u_h be the approximate solution given by the DG method for the model problem $u_t + (au)_x = 0, \quad (x,t) \in \mathbb{R} \times (0,T)$. Assume that the initial data u_o is very smooth. Then*

$$\left\| \frac{d^\alpha}{dx^\alpha} u(x,T) - \frac{d^\alpha}{dx^\alpha} \left(K_h^{r+1,\alpha,\ell} * u_h(\cdot,T) \right)(x). \right\|_{0,\Omega_0} \leq \mathsf{C}\, h^s,$$

where $s = \min\{r+1, 2k+1\}$ and C depends upon the smoothness of the solution.

In Fig. 3 and Table 2, we can see how these two methods of obtaining a derivative approximation compare by considering a variable coefficient equation taken from [16]. If we take the derivative of the SIAC filtered approximation, we can still obtain

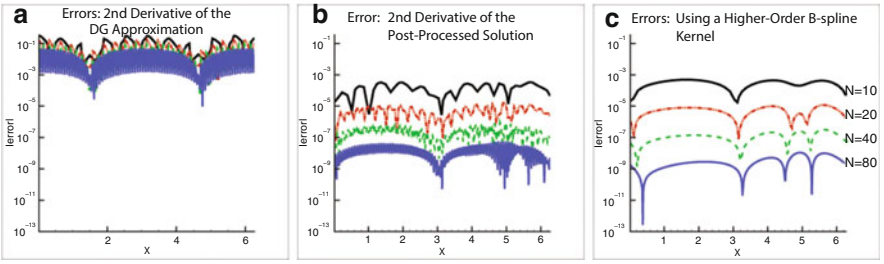


Fig. 3 Pointwise errors in log scale for the second derivatives for the DG approximation together with the SIAC filtered solutions. (**a**) $\partial^2(u - u_h)$. (**b**) $\partial^2(u - K_h * u_h)$. (**c**) $\partial^2(u - \tilde{K}_h * u_h)$

Table 2 L^2-errors and orders the first and second derivatives for the DG approximation together with the SIAC filtered solutions

$\mathbb{P}^2$						
N	$\partial_x^\alpha u_h$		$\partial_x^\alpha (K * u_h)$		$\tilde{K} * \partial_h^\alpha u_h$	
	L^2 error	Order	L^2 error	Order	L^2 error	Order
	1st derivatives					
40	8.7240E−04	–	5.5069E−08	–	2.4411E−06	–
60	3.8775E−04	2.00	6.9067E−08	5.12	3.2245E−06	4.99
80	2.1811E−04	2.00	1.6903E−09	5.03	7.6554E−08	4.99
100	1.3959E−04	2.00	5.8972E−09	4.72	2.5074E−09	5.00
	2nd derivatives					
40	3.3923E−02	–	3.2544E−07	–	1.4294E−07	–
60	2.2619E−02	1.00	6.1855E−08	4.10	1.7735E−08	5.15
80	1.6966E−02	1.00	1.9310E−08	4.05	4.2872E−09	4.94
100	1.3573E−02	1.00	7.8612E−09	4.03	1.4798E−09	4.77

$2k+1$ order accuracy for the first derivative, but each successive derivative looses an order. However, if we use smooth B-splines of higher order, we can maintain $2k+1$ order accuracy for higher derivatives as well.

6 Filtering Near a Boundary

The next question that would be useful to answer is how to filter near a boundary or discontinuity. This requires modifying the filter [17, 21]. To do so, we first use B-splines that depend continuously on the evaluation point through the shift function $\lambda(\bar{x})$:

$$x_\gamma = -\frac{r}{2} + \gamma + \lambda(x), \quad \lambda(x) = \begin{cases} \min\left\{0, -\frac{r+\ell}{2} + \frac{\bar{x}-x_L-\frac{\epsilon h}{2}}{h}\right\}, & x \in [x_L, \frac{x_L+x_R}{2}], \\ \max\left\{0, \frac{r+\ell}{2} + \frac{\bar{x}-x_R+\frac{\epsilon h}{2}}{h}\right\}, & x \in (\frac{x_L+x_R}{2}, x_R], \end{cases} \tag{17}$$

where the one-dimensional domain is defined as $\Omega = [x_L, x_R]$.

The accuracy is improved by using extra B-splines near a boundary so that the post-processed solution is

$$u_h^*(\bar{x}) = \underbrace{\theta(\bar{x}) \underbrace{u_{h,2k+1}^*(\bar{x})}_{\text{filtering with } 2k+1 \text{ B-splines}} + (1-\theta(\bar{x})) \underbrace{u_{h,4k+1}^*(\bar{x})}_{\text{filtering with } 4k+1 \text{ B-splines}}}_{\text{smooth convex combination}}. \tag{18}$$

In this example, θ is chosen such that $\theta(\bar{x}) = 1$ in the interior (giving the symmetric filter); $\theta(\bar{x}) = 0$ near the boundary (to obtain extra accuracy from extra B-splines); θ is smooth in the transition regions between symmetric and boundary filtering.

As an example, we consider the linear equation $u_t + u_x = 0$ with Dirichlet boundary conditions. Plots of the errors are given in Fig. 4 and errors are given in Table 3. We can see from these that we have an improved convergence rate as well as reduction in errors. This occurs even near the boundary and for non-periodic boundary conditions.

Adapting the kernel to handle filtering near boundaries allows us to obtain the following L^∞-error estimate:

Theorem 2 (Ji et al. [9]) *Let u_h be a DG approximation to an exact solution u for a linear hyperbolic equation. Construct u_h^* by applying the position-dependent SIAC filter to u_h, $k \geq 1$. Then,*

$$\|u - u_h^*\|_{\infty,\Omega} \leq C\|u_0\|_{2k+3+[d/2],\Omega} h^s,$$

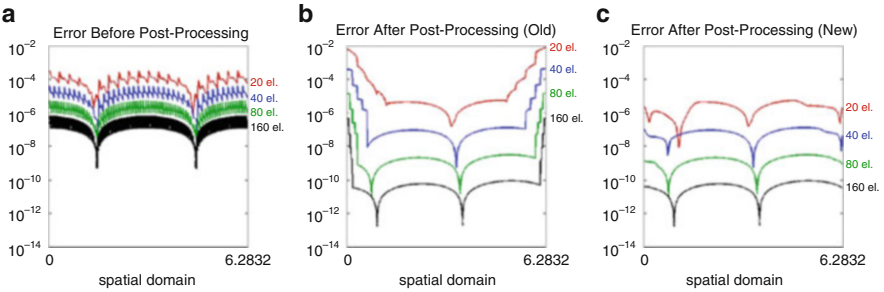


Fig. 4 Pointwise errors in log scale for before and after post-processing for a linear hyperbolic equation with Dirichlet boundary conditions. A comparison of using $2k+1$ (*middle*) versus $4k+1$ (*right*) central B-splines when near a boundary. (**a**) DG errors. (**b**) SIAC DG ($2k+1$). (**c**) SIAC DG ($4k+1$)

Table 3 L^2-errors and order for before and after post-processing for a linear hyperbolic equation with Dirichlet boundary conditions. A comparison of using $2k+1$ (middle column) versus $4k+1$ (right column) central B-splines when near a boundary

	Before		After ($2k+1$)		After ($4k+1$)	
Mesh	L^2-error	Order	L^2-error	Order	L^2-error	Order
	$\mathbb{P}^2$					
20	2.681e−04	–	4.003e−03	–	6.984e−06	–
40	3.352e−05	3.00	2.108e−04	4.25	1.850e−07	5.24
80	4.190e−06	3.00	5.464e−06	5.27	4.798e−09	5.27
160	5.238e−07	3.00	1.254e−07	5.45	1.498e−10	5.00
	$\mathbb{P}^3$					
20	5.176e−06	–	1.304e−04	–	3.751e−07	–
40	3.236e−07	4.00	4.712e−06	4.79	6.396e−10	9.20
80	2.023e−08	4.00	3.406e−08	7.11	2.867e−12	7.80
160	1.264e−09	4.00	1.999e−10	7.41	3.079e−14	6.54

and

$$\|u - u_h^*\|_{0,\Omega} \leq C\|u_0\|_{2k+2,\Omega} h^{2k+1},$$

where $s = \min\{2k+1, 2k+2-\frac{d}{2}\}$ *and C is a constant, dependent on the* L^1*-norm of the kernel coefficients but independent of the mesh.*

However, there are still limitations to overcome. For example, using extra B-splines at the boundaries is good for lower-order approximations, but not for higher-order approximations due to the excessive support size and increased condition number of the matrices involved. Further, the added support does not aid in creating a better approximation for non-uniform meshes.

7 Mesh Geometry

Until now, the assumptions on the applicability of the SIAC filter have required a uniform mesh. A logical question to then ask is whether it can work for nonuniform meshes. The challenges that are incurred when attempting to extend the SIAC filter to a nonuniform mesh is that it requires $\mathcal{O}(h^{2k+1})$ convergence in the negative-order norm for both the approximation as well as the divided difference of the approximation. This requires defining a suitable dual equation and a DG scheme for the divided differences. If the mesh is translation invariant, it is easy to show appropriate convergence for the divided differences [10]. However, let us investigate further the actual requirements of the scaling parameter.

Recall that our error estimate is $\|u - K_H * u_h\|_\Omega \leq CH^{2k+1}$, where H is the kernel scaling parameter. The translation invariance property requires that $T_H^\ell v(x) = v(x+H\ell)$. Thus the mesh is translation invariant for a scaling of mH, $m \in \mathbb{Z}$ as well. This is illustrated in Fig. 5. In this figure, a kernel scaling of $H = mh$ is used for the convolution kernel in the SIAC filter for a discontinuous Galerkin approximation over a uniform mesh designated by h. We can see that error reduction actually occurs even when $H < h$. Superconvergent order starts to occur around $H = h$ and errors start to increase for $H > h$. The sweetspot of reduced errors and superconvergence seems to occur around $H = h$.

Although the typical meshes tested involve some type of translation invariance, the SIAC filter has also been tested over unstructured triangular meshes with promising results [10, 12]. For example Fig. 6 shows the difference in the pointwise errors for the DG approximation versus the SIAC filtered DG approximation. The

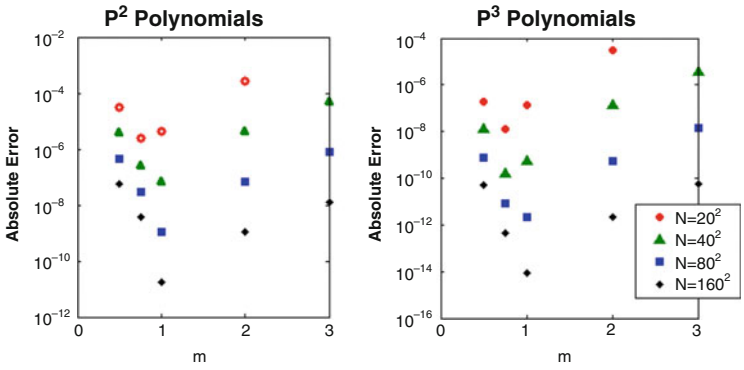


Fig. 5 Effect of H scaling for $\mathbb{P}^2$ and $\mathbb{P}^3$ polynomials for a uniform mesh

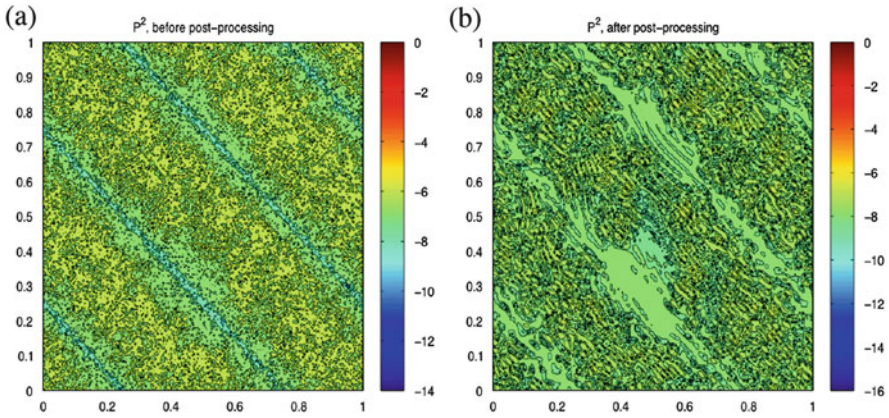


Fig. 6 Typical pointwise error plots for SIAC filtering over a Delaunay Mesh with element splitting. (**a**) DG errors. (**b**) SIAC filtered errors

Table 4 Typical errors for SIAC filtering over a Delaunay Mesh with element splitting

	$m = 0.5$		$m = 1$		$m = 2$	
Mesh	L^2-error	Order	L^2-error	Order	L^2-error	Order
	$\mathbb{P}^2$					
776	7.08E−05	–	1.25E−04	–	x	–
3104	7.84E−06	3.17	6.45E−06	4.27	x	–
12,416	8.24E−07	3.25	5.02E−07	3.68	1.98E−06	–
49,664	1.09E−07	2.20	5.97E−08	3.07	8.11E−08	4.60
	$\mathbb{P}^3$					
776	9.88E−07	–	8.52E−06	–	x	–
3104	2.71E−08	5.18	1.30E−07	6.03	x	–
12,416	3.28E−09	6.02	1.99E−09	6.02	4.58E−08	–
49,664	2.34E−10	3.80	5.85E−11	5.08	6.20E−10	6.20

L^2-errors are given in Table 4. Figure 7 displays the effect of different scalings, when h is taken to be the longest element edge and the kernel is scaled by $H = mh$. Clearly, one can achieve error reduction.

With SIAC filtering we can usually improve the DG convergence rate from order $k+1$ to order $2k+1$ but we have to be careful with kernel scaling [10]. Table 5 gives a list of some of the meshes that SIAC filtering has been tested over and whether reduced errors, improved order or increased smoothness occurs.

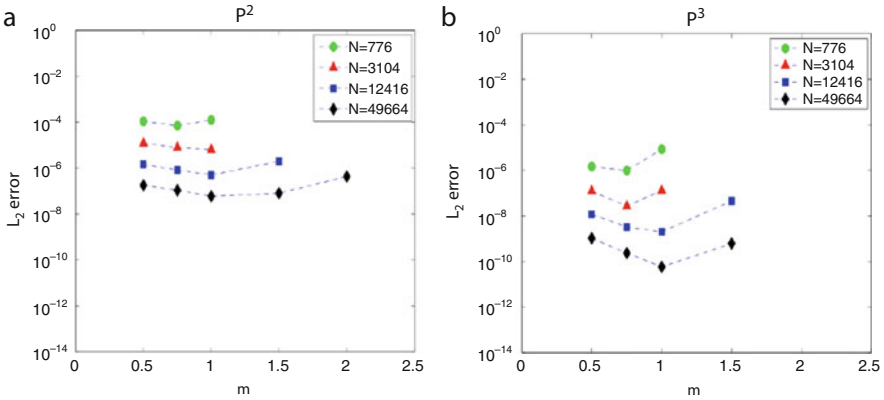


Fig. 7 The effect of the kernel scaling for SIAC filtering over a Delaunay Mesh with element splitting. (**a**) $\mathbb{P}^2$-polynomials. (**b**) $\mathbb{P}^3$-polynomials

Table 5 Some typical meshes over which SIAC filtering has been tested. Listed is the mesh type along with whether SIAC filtering will aid in error reduction, improved convergence order or increasing the smoothness of the solution

Mesh type	Reduced errors	Improved order	Increased smoothness
Uniform quadrilateral	✓	✓	✓
Variable cross quadrilateral	✓	✓	✓
Uniform structured triangle	✓	✓	✓
Structured variable triangle	✓	✓	✓
Delaunay mesh	✓	?	?

8 Summary

We can make superconvergence useful through accuracy extraction techniques. SIAC filtering is one technique that uses a B-spline convolution kernel that induces smoothness on the DG field and enhances accuracy. In general, we can obtain order improvement from $\mathcal{O}(h^{k+1})$ to $\mathcal{O}(h^s)$ where $s = \min\{r+1, 2k+1\}$. However, the expected order improvement relies on higher-order estimates in the negative-order norm for the approximation as well as the divided differences. Once we are able to prove these estimates we can concentrate on other issues in SIAC filtering such as modifying the filter for higher-order derivative information or boundary filtering. For the scaling of the kernel, we must exploit information about the mesh geometry in order to have a reduction in the errors.

Acknowledgements Portions of this research are sponsored by the European Office of Aerospace Research and Development (EOARD) under the U.S. Air Force Office of Scientific Research (AFOSR) under grant number FA 8655-13-1-3017.

References

1. S. Adjerid, K.D. Devine, J.E. Flaherty, L. Krivodonova, A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems. Comput. Methods Appl. Mech. Eng. **191**, 1097–1112 (2002)
2. J.H. Bramble, A.H. Schatz, Higher order local accuracy by averaging in the finite element method. Math. Comput. **31**, 94–111 (1977)
3. F. Celiker, B. Cockburn, H.K. Stolarski, Locking-free optimal discontinuous Galerkin methods for Timoshenko Beams. SIAM J. Numer. Anal. **44**, 2297–2325 (2006)
4. B. Cockburn, R. Ichikawa, Adjoint recovery of superconvergent linear functionals from Galerkin approximations: the one-dimensional case. J. Sci. Comput. **232**, 201–232 (2007)
5. B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*. Lecture Notes in Mathematics, vol. 1697 (Springer, Heidelberg, 1998)
6. B. Cockburn, M. Luskin, C.-W. Shu, E. Süli, Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. Math. Comput. **72**, 577–606 (2003)
7. L. Ji, Y. Xu, J.K. Ryan, Accuracy enhancement of the linear convection-diffusion equation in multiple dimensions. Math. Comput. **81**, 1929–1950 (2012)
8. L. Ji, Y. Xu, J.K. Ryan, Negative-order norm estimates for nonlinear hyperbolic conservation laws. J. Sci. Comput. **54**, 269–310 (2013)
9. L. Ji, P. van Slingerland, J.K. Ryan, C.W. Vuik, Superconvergent error estimates for a position-dependent smoothness-increasing accuracy-conserving filter for DG solutions. Math. Comput. **83**, 2239–2262 (2014)
10. J. King, H. Mirzaee, J.K. Ryan, R.M. Kirby, Smoothness-increasing accuracy-conserving (SIAC) filtering for discontinuous Galerkin solutions: improved errors versus higher-order accuracy. J. Sci. Comput. **53**, 129–149 (2012)
11. H. Mirzaee, L. Ji, J.K. Ryan, R.M. Kirby, Smoothness-increasing accuracy-conserving (SIAC) post-processing for discontinuous Galerkin solutions over structured triangular meshes. SIAM J. Numer. Anal. **49**, 1899–1920 (2011)
12. H. Mirzaee, J. King, J.K. Ryan, R.M. Kirby, Smoothness-increasing accuracy-conserving (SIAC) filters for discontinuous Galerkin solutions over unstructured triangular meshes. SIAM J. Sci. Comput. **35**, A212–A230 (2013)
13. M. Mirzargar, J.K. Ryan, R.M. Kirby, Smoothness-increasing accuracy-conserving (SIAC) filtering and quasi-interpolation: a unified view. J. Sci. Comput. 1–25 (2015). doi:10.1007/s10915-015-0081-9
14. M.S. Mock, P.D. Lax, The computation of discontinuous solutions of linear hyperbolic equations. Commun. Pure Appl. Math. **31**, 423–430 (1978)
15. K. Mustapha, J.K.Ryan, Post-processing discontinuous Galerkin solutions to Volterra integro-differential equations: analysis and simulations. J. Comput. Appl. Math. **253**, 89–103 (2013)
16. J.K. Ryan, B. Cockburn, Local derivative post-processing for the discontinuous Galerkin method. J. Comput. Phys. **228**, 8642–8664 (2009)
17. J.K. Ryan, C.-W. Shu, One-sided post-processing for the discontinuous Galerkin method. Methods Appl. Anal. **10**, 295–307 (2003)
18. J.K. Ryan, C.-W. Shu, H. Atkins, Extension of a post-processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem. SIAM J. Sci. Comput. **26**, 821–843 (2005)
19. M. Steffan, S. Curtis, R.M. Kirby, J.K. Ryan, Investigation of smoothness enhancing accuracy-conserving filters for improving streamline integration through discontinuous fields. IEEE Trans. Vis. Comput. Graph. **14**, 680–692 (2008)

20. V. Thomée, High order local approximations to derivatives in the finite element method. Math. Comput. **31**, 652–660 (1977)
21. P. van Slingerland, J.K. Ryan, C. Vuik, Position-dependent smoothness-increasing accuracy-conserving (SIAC) filtering for accuracy for improving discontinuous Galerkin solutions. SIAM J. Sci. Comput. **33**, 802–825 (2011)
22. L.B. Wahlbin, *Superconvergence in Galerkin Finite Element Methods* (Springer, Heidelberg, 1995)
23. G. Wasserman, R. Archibald, A. Gelb, Image reconstruction from fourier data using sparsity of edges polynomial annihilation sparsifying transform. J. Sci. Comput. (2014)

Computational Comparison of Continuous and Discontinuous Galerkin Time-Stepping Methods for Nonlinear Initial Value Problems

Bärbel Janssen and Thomas P. Wihler

Abstract This article centers on the computational performance of the continuous and discontinuous Galerkin time stepping schemes for general first-order initial value problems in $\mathbb{R}^n$, with continuous nonlinearities. We briefly review a recent existence result for discrete solutions from Janssen and Wihler (Existence results for the continuous and discontinuous Galerkin time stepping methods for nonlinear initial value problems, 2014, Submitted), and provide a numerical comparison of the two time discretization methods.

1 Introduction

In this paper we focus on (possibly high-order) continuous and discontinuous Galerkin (cG and dG, respectively) time stepping discretizations as applied to initial value problems of the form

$$u'(t) = \mathscr{F}(t, u(t)), \qquad t \in (0, T), \tag{1}$$

$$u(0) = u_0. \tag{2}$$

Here, $u : (0, T) \to \mathbb{R}^n$, for some $n \in \mathbb{N}$ and $T > 0$, is an unknown solution. The initial vector $u_0 \in \mathbb{R}^n$ prescribes the solution u at the start-up time $t = 0$, and $\mathscr{F} : [0, T] \times \mathbb{R}^n \to \mathbb{R}^n$ is a possibly nonlinear, continuous operator. We will usually omit to explicitly write the dependence on the first argument t.

Galerkin-type time stepping methods for initial-value problems are based on weak formulations. For both the cG and the dG schemes, the test spaces constitute

B. Janssen
Department of High Performance Computing and Visualization, School of Computer Science and Communication, KTH Royal Institute of Technology, Stockholm, Sweden
e-mail: barbel@kth.se

T.P. Wihler (✉)
Mathematics Institute, University of Bern, Bern, Switzerland
e-mail: wihler@math.unibe.ch

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_7

of polynomials that are discontinuous at the time nodes. In this way, the discrete Galerkin formulations decouple into local problems on each time step, and the discretizations can hence be understood as implicit one-step schemes. Galerkin time stepping methods have been analyzed for ordinary differential equations (ODEs), e.g., in [2–5, 7, 10, 13].

In the current article, we will start by reviewing the definitions of the cG and dG schemes of arbitrary order in Sect. 2. Furthermore, we will recall the recent work [6] which shows that the existence of discrete cG and dG solutions for continuous nonlinearities is independent of the approximation order and only requires the local time steps to be sufficiently small (and thereby generalizes the previous works [10, 13], where Lipschitz continuous nonlinearities where considered). The focus of this work is to provide a computational comparison of the two schemes in Sect. 4.

Throughout the paper, we shall use the following notation: For an interval $I = (a, b)$, $a < b$, the space $C^0(\bar{I})$ consists of all functions $u : \bar{I} \to \mathbb{R}^n$ that are continuous on $\bar{I}$. Moreover, introducing, for $1 \le p < \infty$, the norm

$$\|u\|_{L^p(I)} = \left(\int_I |u(t)|^p \, \mathrm{d}t \right)^{1/p},$$

and, for $p = \infty$, the norm $\|u\|_{L^\infty(I)} = \operatorname{ess\,sup}_{t \in I} |u(t)|$, we write $L^p(I)$ to signify the space of measurable functions $u : I \to \mathbb{R}^n$ so that the corresponding norm is bounded. We note that $L^2(I)$ is a Hilbert space with the inner product

$$(u, v)_{L^2(I)} = \int_I (u(t), v(t)) \, \mathrm{d}t.$$

Here, $(\cdot, \cdot)$ and $|\cdot|$ denote the standard dot product and Euclidean norm in $\mathbb{R}^n$, respectively.

2 Galerkin Time Stepping

On an interval $I = [0, T]$, consider time nodes $0 = t_0 < t_1 < \cdots < t_{M-1} < t_M = T$ which introduce a time partition $\mathscr{M} = \{I_m\}_{m=1}^M$ of I into M open time intervals $I_m = (t_{m-1}, t_m)$, $m = 1, \ldots, M$. The length $k_m = t_m - t_{m-1}$ of a time interval (which may vary locally) is called the mth time step. Furthermore, we let $r \ge 0$ to be a (global) polynomial degree, which takes the role of an approximation order. Then, given $s \in \mathbb{N}_0$, the set

$$\mathscr{P}^s(J) = \left\{ p \in C^0(\bar{J}) : p(t) = \sum_{i=0}^{s} x_i t^i, \, x_i \in \mathbb{R}^n \right\}$$

signifies the space of all polynomials of degree at most s on an interval $J \subset \mathbb{R}$ with values in $\mathbb{R}^n$.

The cG$(r+1)$ and dG(r) time marching methods on $\mathscr{M}$ will seek solutions that locally belong to the spaces $\mathscr{P}^{r+1}(I_m)$ and $\mathscr{P}^r(I_m)$, respectively. We emphasize that, for both schemes, the local test space is $\mathscr{P}^r(I_m)$.

2.1 The cG Method

With the notation above, the cG$(r+1)$ time marching scheme is iteratively given as follows: For a prescribed initial vector $U_{m-1} := U|_{I_{m-1}}(t_{m-1}) \in \mathbb{R}^n$ (with $U_0 := u_0$, where $u_0 \in \mathbb{R}^n$ is the initial vector from (2)), we find $U|_{I_m} \in \mathscr{P}^{r+1}(I_m)$ through the weak formulation

$$\begin{aligned} \int_{I_m} (U', V)\,\mathrm{d}t &= \int_{I_m} (\mathscr{F}(U), V)\,\mathrm{d}t \qquad \forall V \in \mathscr{P}^r(I_m), \\ U(t_{m-1}) &= U_{m-1}, \end{aligned} \tag{3}$$

for any $1 \le m \le M$. Notice that, in order to enforce the initial condition on each individual time step (and thereby to obtain a globally continuous solution U on $(0,T)$), the local trial space possesses one degree of freedom more than the local test space.

Introducing the (local) L^2-projection $\Pi_m^r : L^2(I_m) \to \mathscr{P}^r(I_m)$ onto $\mathscr{P}^r(I_m)$ given by

$$\int_{I_m} (v - \Pi_m^r v, w)\,\mathrm{d}t = 0 \qquad \forall w \in \mathscr{P}^r(I_m),$$

the following result is quite elementary to deduce:

Proposition 1 *A function $U \in \mathscr{P}^{r+1}(I_m)$ is a solution of (3) if and only if U satisfies the fixed point equation*

$$U(t) = U_{m-1} + \int_{t_{m-1}}^{t} \Pi_m^r \mathscr{F}(U)\,d\tau, \tag{4}$$

for any $t \in I_m$.

2.2 The dG Method

In order to define the discontinuous Galerkin scheme, some additional notation is required: We define the one-sided limits of a piecewise continuous function U at

each time node t_m by

$$U_m^+ := \lim_{s \searrow 0} U(t_m + s), \qquad U_m^- := \lim_{s \nearrow 0} U(t_m + s).$$

Then, the discontinuity jump of U at t_m, for $0 \le m \le M-1$, is defined by $[\![U]\!]_m = U_m^+ - U_m^-$; for $m = 0$ we set $U_0^- = u_0$, where u_0 is the initial vector from (2).

With these definitions the dG(r) time stepping method for (1)–(2) reads: Find $U|_{I_m} \in \mathscr{P}^r(I_m)$ such that

$$\int_{I_m} (U', V)\,\mathrm{d}t + ([\![U]\!]_{m-1}, V_{m-1}^+) = \int_{I_m} (\mathscr{F}(U), V)\,\mathrm{d}t \qquad \forall V \in \mathscr{P}^r(I_m), \tag{5}$$

for any $1 \le m \le M$. We underline that, in contrast to the continuous Galerkin formulation, the local trial and test spaces are the same for the discontinuous Galerkin scheme. This is due to the fact that the initial values are weakly imposed (by means of an upwind flux) on each time interval.

In order to derive a fixed-point formulation for the dG scheme as in (4), we revisit [11, Sect. 4.1] to define a lifting operator, for $1 \le m \le M$,

$$\mathsf{L}_m^r : \mathbb{R}^n \to \mathscr{P}^r(I_m),$$

by

$$\int_{I_m} (\mathsf{L}_m^r(z), V)\,\mathrm{d}t = (z, V_{m-1}^+) \qquad \forall V \in \mathscr{P}^r(I_m),\ z \in \mathbb{R}^n.$$

Then, looking at the discrete derivative operator

$$\chi : \mathscr{P}^r(I_m) \to \mathscr{P}^r(I_m), \qquad U \mapsto \chi(U) = U' + \mathsf{L}_m^r(U_{m-1}^+), \tag{6}$$

we recall the following result from [6].

Proposition 2 *The operator χ from (6) is an isomorphism, and satisfies the bound, for any $p \in [1, \infty]$,*

$$\|\chi^{-1}(U)\|_{L^\infty(I_m)} \le 2k_m^{1-1/p} \|U\|_{L^p(I_m)} \qquad \forall U \in \mathscr{P}^r(I_m).$$

Moreover, a function $U \in \mathscr{P}^r(I_m)$ is a solution of (5) if and only if the fixed point equation

$$U = U_{m-1}^- + \chi^{-1}\left(\Pi_m^r \mathscr{F}(U)\right) \tag{7}$$

is fulfilled.

Remark 1 We note that the discrete operator χ from (6) is closely related to the (parabolic) reconstruction operator as discussed in, e.g., [8].

3 Existence of Discrete Galerkin Solutions

The well-known Peano Theorem (see, e.g., [12]) guarantees the existence of C^1-solutions u of (1)–(2) within some limited time range, $t \in (0, T^\star)$, for some $T^\star > 0$. Notice that the existence interval for solutions may be arbitrarily small even for smooth $\mathscr{F}$: For instance, the initial value problem (1)–(2) may exhibit solutions that may become unbounded in finite time; to give an example, let us consider the initial value problem of finding a $\mathbb{R}$-valued function u which satisfies

$$u'(t) = |u(t)|^{\beta-1}u(t), \qquad u(0) = 1, \tag{8}$$

for a given constant $\beta > 1$. It is elementary to check that

$$u(t) = (1 - (\beta - 1)t)^{\frac{1}{1-\beta}}$$

is a solution of (8), and we see that there appears a blow-up as $t \nearrow T^\star := \frac{1}{\beta-1}$.

Based on the fixed point equations (4) and (7) for the cG and dG schemes, respectively, it is possible to prove the ensuing existence result for solutions, see [6]:

Theorem 1 *Let $1 \le m \le M$, and suppose that, for some $\kappa_m > 0$,*

$$K_m^{\kappa_m} := \sup_{(t,y)\in I_m \times B_{\kappa_m}} |\mathscr{F}(t, y)| < \infty,$$

where $B_{\kappa_m} = \{y \in \mathbb{R}^n : |y - U_{m-1}^-| \le \kappa_m\}$. Then, if the local time step is chosen such that

$$k_m \le \frac{\kappa_m}{C_{\mathrm{ex}} K_m^{\kappa_m}}, \tag{9}$$

where

$$C_{\mathrm{ex}} = \begin{cases} 1 & \text{for the cG}(r+1)\text{ scheme}, \\ 2 & \text{for the dG}(r)\text{ scheme}, \end{cases} \tag{10}$$

then the cG$(r + 1)$ and dG(r) methods from (3) and (5), respectively, on the time interval I_m each possess at least one solution in $M_m^{\kappa_m} := \{Y \in \mathscr{P}^{r+2-C_{\mathrm{ex}}}(I_m) : Y(t) \in B_{\kappa_m}\ \forall t \in \bar{I}_m\}$. In particular, the existence of discrete Galerkin solutions is independent of the polynomial degree r.

Remark 2 We note that Theorem 1 still holds true for varying polynomial degrees on each time interval.

4 Numerical Experiments

We will now compare the cG and dG discretizations by means of a few numerical tests. Specifically, we consider the initial value problem (8) for the linear case $\beta = 1$,

$$u'(t) = u(t), \quad t \geq 0, \qquad u(0) = 1,$$

as well as for the nonlinear case $\beta = 2$,

$$u'(t) = u(t)^2, \quad t \geq 0, \qquad u(0) = 1.$$

The former problem has an analytic exact solution which is given by $u(t) = \exp(t)$. For $\beta = 2$, the exact solution is $u(t) = (1-t)^{-1}$, and features a blow-up as $t \nearrow 1$.

The time meshes in our computations are based on the existence criterion from Theorem 1, i.e., the individual time steps are chosen according to (9) (independently of the polynomial degree r). For $\beta = 1$ and some $\kappa_m > 0$ there holds that

$$K_m^{\kappa_m} = \sup_{|y-U_{m-1}^-|\leq\kappa_m} |y| = \kappa_m + |U_{m-1}^-|.$$

Hence, for k_m in (9) we obtain

$$k_m \leq \frac{\kappa_m}{C_{\mathrm{ex}}(\kappa_m + |U_{m-1}^-|)} \to C_{ex}^{-1},$$

as $\kappa_m \to \infty$, where C_{ex} is the constant from (10). In our experiments we shall choose

$$k_m = \frac{1}{2C_{\mathrm{ex}}} \qquad (\beta = 1).$$

For $\beta = 2$, it has been shown in [6] that the maximal possible time step according to (9) is given by

$$k_m = \frac{1}{4C_{\mathrm{ex}}|U_{m-1}^-|} \qquad (\beta = 2).$$

Here, C_{ex} is again the constant from (10). Incidentally, while the time steps for $\beta = 1$ are chosen to be of constant size, the time mesh for $\beta = 2$ turns out to be geometrically refined towards the blow-up point at $T = 1$.

In order to deal with the nonlinearities, the Newton method will be applied. We note that, for $\beta = 2$ close to the blow-up, the Newton iterations may deteriorate or take a long time to converge. If the Newton method fails to converge, we simply stop the time iteration.

In Figs. 1, 2, 3, and 4 we compare the performance of the dG(r) and the cG($r+1$) time stepping methods as applied to our model problem (8); note that, for given $r \geq 0$, these methods feature the same number of degrees of freedom on each time step (as they are both based on the same test spaces). In each of the figures below we display the *ratio* of the cG($r+1$) and dG(r) errors for different error types, including the accumulated L^2 errors, the L^∞ errors, and the nodal end time errors, for different problem parameters. More precisely, we use the following notation:

- Accumulated L^2 error:

$$\frac{\|u - U^{\mathsf{cG}}\|_{L^2(0,t_m)}}{\|u - U^{\mathsf{dG}}\|_{L^2(0,t_m)}}, \qquad m \geq 1;$$

- L^∞ error:

$$\frac{\|u - U^{\mathsf{cG}}\|_{L^\infty(0,t_m)}}{\|u - U^{\mathsf{dG}}\|_{L^\infty(0,t_m)}}, \qquad m \geq 1;$$

- Nodal end time error:

$$\frac{|u(t_m) - U_m^{\mathsf{cG}}|}{|u(t_m) - (U^{\mathsf{dG}})_m^-|}, \qquad m \geq 1.$$

Here, u is the exact solution of (8) (for $\beta \in \{1, 2\}$), and U^{cG} and U^{dG} denote the corresponding cG($r+1$) and dG(r) solutions defined by (3) and (5), respectively. In our experiments we perform tests for both $C_{\mathrm{ex}} = 1$ (existence for the cG method) and $C_{\mathrm{ex}} = 2$ (existence for the dG method) for both schemes; cf. (10).

Discussion of the Results

In terms of the L^2 and L^∞ errors in the low-order context, the cG method seems to perform better than the dG scheme in both the smooth ($\beta = 1$) as well as in the blow-up ($\beta = 2$) case. For $\beta = 1$, however, the ratios tend to a limit just below 1 for increasing polynomial degrees r. This behavior is similar for $\beta = 2$ (away from the blow-up time $T = 1$), although here we observe that the ratios seem to stabilize slightly above 1 for higher r.

For the ratios of the nodal end time errors, we only show results for polynomial degrees $r = 0, 1, 2, 3$, and for $\beta = 1$; indeed, for higher polynomial degrees (and $\beta = 2$ away from the blow-up) the nodal end time errors become quickly

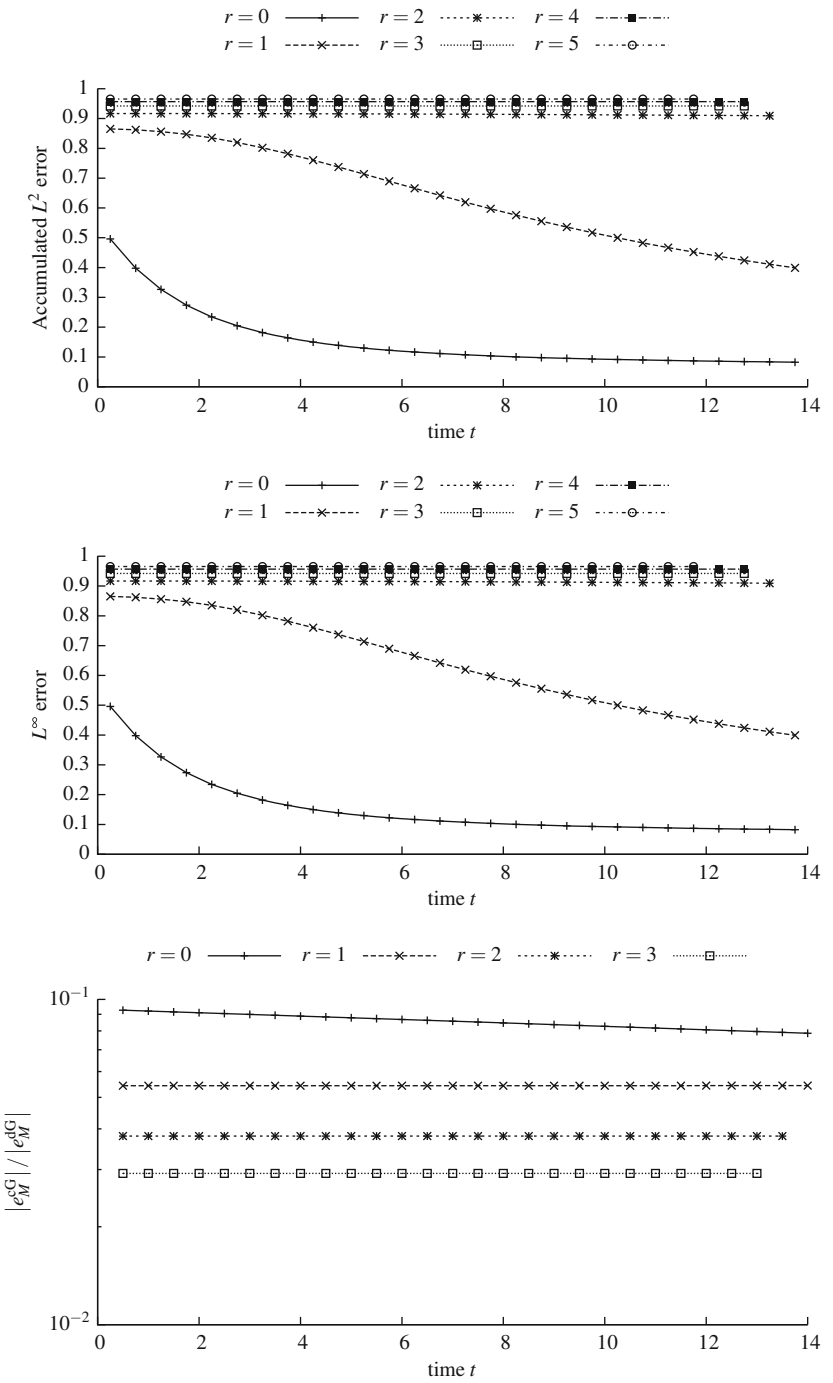


Fig. 1 Error ratios for $\beta = 1$ (smooth solution) and $C_{\text{ex}} = 1$

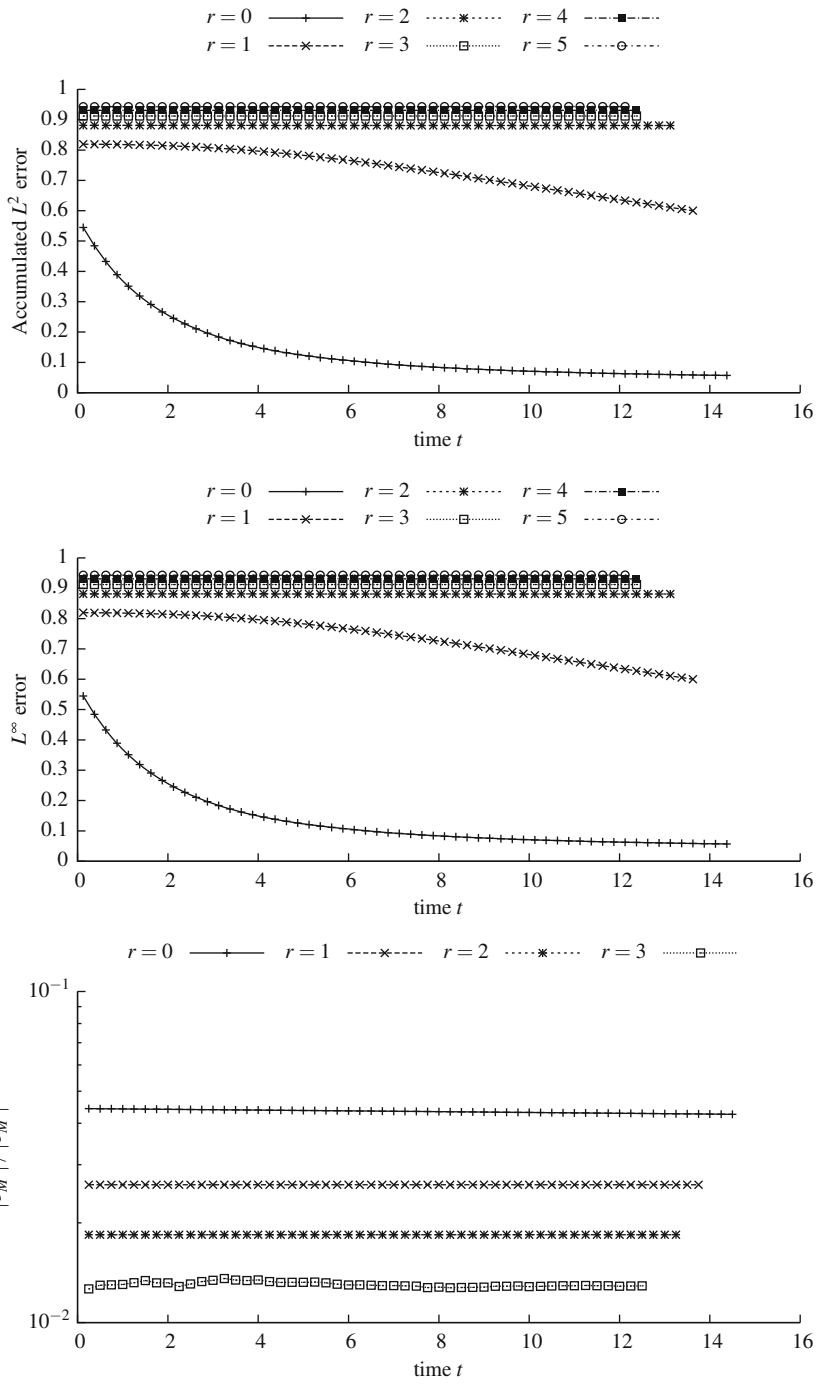


Fig. 2 Error ratios for $\beta = 1$ (smooth solution) and $C_{\text{ex}} = 2$

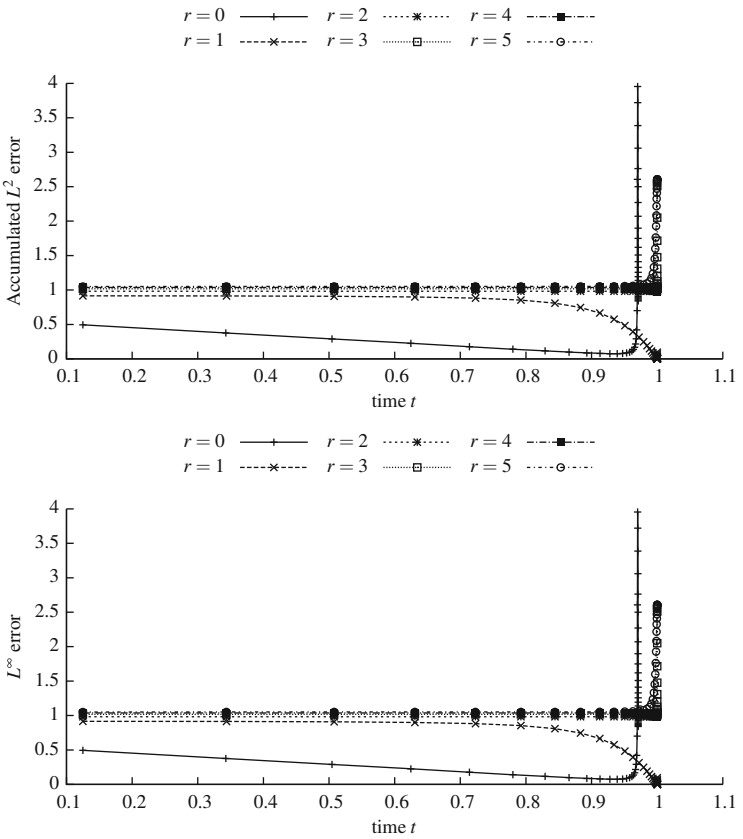


Fig. 3 Error ratios for $\beta = 2$ (blow-up solution) and $C_{\text{ex}} = 1$

close to machine precision due to well-known super convergence effects at nodes in Galerkin time stepping discretizations. We observe that the cG method performs again better than the dG method; for increasing polynomial degree, the dominance of the cG scheme over the dG scheme becomes even more pronounced. This behavior is not surprising since the super convergence regime of the cG scheme for smooth solutions is (at least theoretically) superior to the dG method (see the papers [1, 9] for related super convergence results for Galerkin methods).

In conclusion, both discretization schemes perform similarly in the high-order context, whereas the cG method seems a little more favorable in the low-order setting for the examples considered here.

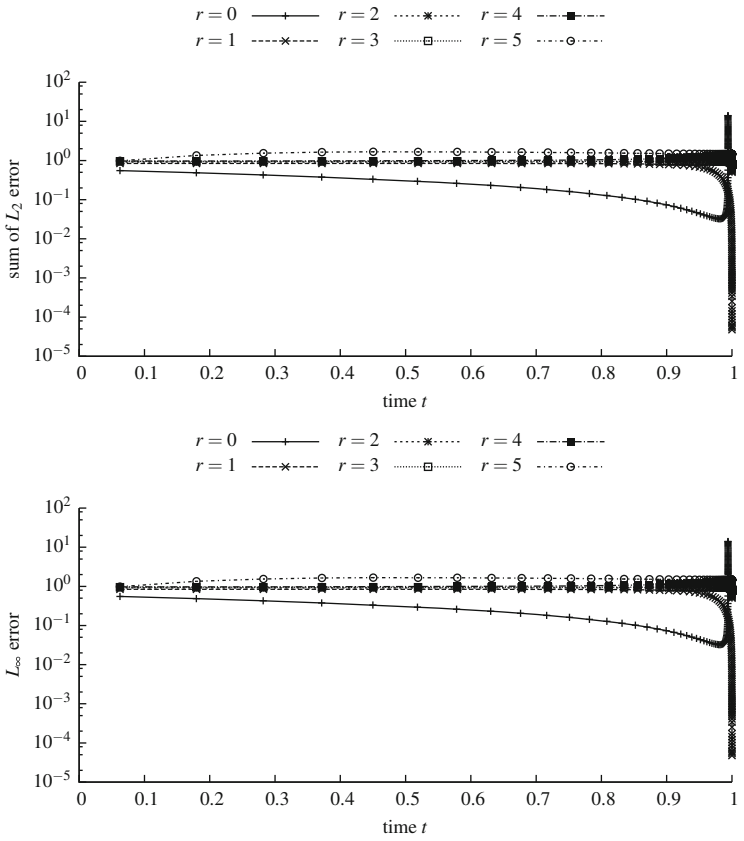


Fig. 4 Error ratios for $\beta = 2$ (blow-up solution) and $C_{\text{ex}} = 2$

Acknowledgements Thomas P. Wihler would like to thank the scientific committee and the local organizers of ICOSAHOM 2014 for the conference invitation to Salt Lake City. Furthermore, he acknowledges the financial support of the Swiss National Science Foundation.

References

1. S. Adjerid, K.D. Devine, J.E. Flaherty, L. Krivodonova, A posteriori error estimation for discontinuous Galerkin solutions of hyperbolic problems. Comput. Methods Appl. Mech. Eng. **191**(11–12), 1097–1112 (2002)
2. W. Bangerth, R. Rannacher, *Adaptive Finite Element Methods for Differential Equations*. Lectures in Mathematics, ETH Zürich (Birkhäuser, Basel, 2003)
3. M. Delfour, W. Hager, F. Trochu, Discontinuous Galerkin methods for ordinary differential equations. Math. Comput. **36**, 455–473 (1981)
4. D. Estep, A posteriori error bounds, global error control for approximation of ordinary differential equations. SIAM J. Numer. Anal. **32**, 1–48 (1995)

5. D. Estep, D. French, Global error control for the continuous Galerkin finite element method for ordinary differential equations. RAIRO Modél. Math. Anal. Numér. **28**, 815–852 (1994)
6. B. Janssen, T.P. Wihler, Existence results for the continuous and discontinuous Galerkin time stepping methods for nonlinear initial value problems. Report number 1407.5520 (2014)
7. C. Johnson, Error estimates and adaptive time-step control for a class of one-step methods for stiff ordinary differential equations. SIAM J. Numer. Anal. **25**, 908–926 (1988)
8. C. Makridakis, R.H. Nochetto, A posteriori error analysis for higher order dissipative methods for evolution problems. Numer. Math. **104**(4), 489–514 (2006)
9. K. Mustapha, W. McLean, Superconvergence of a discontinuous Galerkin method for fractional diffusion and wave equations. SIAM J. Numer. Anal. **51**(1), 491–515 (2013)
10. D. Schötzau, C. Schwab, An *hp* a-priori error analysis of the DG time-stepping method for initial value problems. Calcolo **37**, 207–232 (2000)
11. D. Schötzau, T.P. Wihler, A posteriori error estimation for *hp*-version time-stepping methods for parabolic partial differential equations. Numer. Math. **115**(3), 475–509 (2010). doi:10.1007/s00211-009-0285-8. http://dx.doi.org/10.1007/s00211-009-0285-8
12. G. Teschl, *Ordinary Differential Equations and Dynamical Systems*, vol. 140, 9th edn. (American Mathematical Society, Providence, 2012)
13. T.P. Wihler, An a-priori error analysis of the *hp*-version of the continuous Galerkin FEM for nonlinear initial value problems. J. Sci. Comput. **25**, 523–549 (2005)

Part II
Contributed Papers

Recovering Piecewise Smooth Functions from Nonuniform Fourier Measurements

Ben Adcock, Milana Gataric, and Anders C. Hansen

Abstract In this paper, we consider the problem of reconstructing piecewise smooth functions to high accuracy from nonuniform samples of their Fourier transform. We use the framework of nonuniform generalized sampling (NUGS) to do this, and to ensure high accuracy we employ reconstruction spaces consisting of splines or (piecewise) polynomials. We analyze the relation between the dimension of the reconstruction space and the bandwidth of the nonuniform samples, and show that it is linear for splines and piecewise polynomials of fixed degree, and quadratic for piecewise polynomials of varying degree.

1 Introduction

In a number of applications, including Magnetic Resonance Imaging (MRI), electron microscopy and Synthetic Aperture Radar (SAR), measurements are collected nonuniformly in the Fourier domain. The corresponding sampling patterns may be highly irregular; for example, one may sample more densely at low frequencies and more sparsely in high frequency regimes. Standard tools for reconstruction from such data such as gridding [14] seek to compute approximations to the harmonic Fourier modes, which can be then further postprocessed by conventional filtering and/or edge detection algorithms. However, gridding methods are low order, and lead to both physical (e.g. Gibbs phenomena) and unphysical artefacts [18].

In this paper we consider high-order, artefact-free methods for the reconstruction of one-dimensional piecewise smooth functions. To do this, we use the

B. Adcock (✉)
Department of Mathematics, Simon Fraser University, Burnaby, BC, Canada V5A 1S6
e-mail: ben_adcock@sfu.ca

M. Gataric
CCA, Centre for Mathematical Sciences, University of Cambridge, Cambridge CB3 0WA, UK
e-mail: m.gataric@maths.cam.ac.uk

A.C. Hansen
DAMTP, Centre for Mathematical Sciences, University of Cambridge, Cambridge CB3 0WA, UK
e-mail: ach70@cam.ac.uk

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_8

recently-introduced tool of nonuniform generalized sampling (NUGS) [6]. NUGS is reconstruction framework for arbitrary nonuniform samples which allows one to tailor the reconstruction space to suit the function to be approximated. Critically, in NUGS the dimension of the reconstruction space, which we denote by T, is allowed to vary in relation to the *bandwidth K* of the samples. By doing so, one obtains a reconstruction which is numerically stable and quasi-optimal. Hence, if T is chosen appropriately for the given function—for example, a polynomial or spline space for smooth functions, or a piecewise polynomial space for piecewise smooth functions—one obtains a rapidly-convergent approximation.

The key issue prior to implementation is to determine such scaling. In principle, this depends on both the nature of the nonuniform samples *and* the choice of reconstruction space. In this paper we provide a general analysis which allows one to simultaneously determine such scaling for all possible nonuniform sampling schemes by scrutinizing two intrinsic quantities ζ and γ of the reconstruction space T, related to the maximal uniform growth of functions in T and the maximal growth of derivatives in T respectively. Provided these are known (as is the case for many choices of T), one can immediately estimate this scaling. As a particular consequence, for trigonometric polynomials, splines and piecewise algebraic polynomials (with fixed polynomial degree), we can show that this scaling is linear, and for piecewise algebraic polynomials with varying degree we show that it is quadratic. The asymptotic order of such estimates is provably optimal.

2 Nonuniform Generalized Sampling

Throughout we work in the space $\mathrm{H} = \mathrm{L}^2(0,1)$ with its usual inner product $\langle \cdot, \cdot \rangle$ and norm $\|\cdot\|$. Define the Fourier transform by $\hat{f}(\omega) = \int_0^1 f(x)\mathrm{e}^{-2\pi \mathrm{i}\omega x}\,\mathrm{d}x$ for $\omega \in \mathbb{R}$. We let $\{\Omega_N\}_{N\in\mathbb{N}}$ be a sequence of ordered nonuniform sampling points, i.e. $\Omega_N = \{\omega_{n,N}\}_{n=1}^N \subseteq \mathbb{R}$ where $-\infty < \omega_{1,N} < \omega_{2,N} < \ldots < \omega_{N,N} < \infty$, and let $\{\mathrm{T}_M\}_{M\in\mathbb{N}}$ be a sequence of finite-dimensional subspaces of H. We make the natural assumption that the sequence of orthogonal projections $\mathscr{P}_M = \mathscr{P}_{\mathrm{T}_M} : \mathrm{H} \to \mathrm{T}_M$ converge strongly to the identity operator $\mathscr{I}$ on H. That is, any function $f \in \mathrm{H}$ can be approximated to arbitrary accuracy from T_M for sufficiently large M.

Our goal is the following: given the samples $\{\hat{f}(\omega_{n,N})\}_{n=1}^N$ compute an approximation $f_{N,M}$ to f from the subspace T_M. Proceeding as in [6], we do this via the following weighted least-squares:

$$f_{N,M} = \underset{g\in\mathrm{T_M}}{\operatorname{argmin}} \sum_{n=1}^{N} \mu_{n,N} \left|\hat{f}(\omega_{n,N}) - \hat{g}(\omega_{n,N})\right|^2, \tag{1}$$

where $\mu_{n,N} \geq 0$ are appropriate weights (see later). As discussed in [6], the key is to choose M suitably small for a given N (or equivalently N suitably large for a given M) so that the approximation $\{\hat{f}(\omega_{n,N})\}_{n=1}^N \mapsto f_{N,M} \in \mathrm{T}_M$ is numerically stable and

quasi-optimal. To this end, the following estimates were shown in [6]:

$$\|f - f_{N,M}\| \leq C(N,M) \inf_{g \in \mathrm{T}_M} \|f - g\|, \quad \|f_{N,M}\| \leq C(N,M)\|f\|, \quad \forall f \in \mathrm{H}, \tag{2}$$

where $C(N,M) = \sqrt{C_1(N)/C_1(N,M)}$ and $C_1(N,M)$ and $C_2(N)$ are the optimal constants in the inequalities

$$\sum_{n=1}^{N} \mu_{n,N} \left|\hat{f}(\omega_{n,N})\right|^2 \geq C_1(N,M)\|f\|^2, \quad \forall f \in \mathrm{T}_M,$$

$$\sum_{n=1}^{N} \mu_{n,N} \left|\hat{f}(\omega_{n,N})\right|^2 \leq C_2(N)\|f\|^2, \quad \forall f \in \mathrm{H}.$$

In particular, $f_{N,M}$ exists uniquely for any $f \in \mathrm{H}$ if and only if $C_1(N,M) > 0$.

Remark 1 Recently, a number of other works have investigated the problem of high-order reconstructions from nonuniform Fourier data. In [9, 18] spectral reprojection techniques were used for this task, and a frame-theoretic approach was introduced in [10]. Recovering the Fourier transform to high accuracy was studied in [16], and in [8, 15] the problem of high-order edge detection was addressed. We note that the methods we consider in this paper based on NUGS can be shown to achieve optimal convergence rates amongst all stable, convergent algorithms [3, 5]. However, a more detailed discussion is beyond the scope of this paper.

3 A Sufficient Condition for Stability and Quasi-Optimality

To ensure that $C(N,M)$ is small and finite, and hence guarantee stability and quasi-optimality via (2), we first need the following density assumption:

Definition 1 The sequence $\{\Omega_N\}_{N\in\mathbb{N}}$ is uniformly δ-dense for some $0 < \delta < 1$ if: (i) there exists a sequence $\{K_N\}_N \subseteq [0,\infty)$ with $K_N \to \infty$ as $N \to \infty$ such that $\Omega_N \subseteq [-K_N, K_N]$, and (ii) for each N, the density condition $\max_{n=0,\dots,N}\{\omega_{n+1,N} - \omega_{n,N}\} \leq \delta$ holds, where $\omega_{0,N} = \omega_{N,N} - 2K_N$ and $\omega_{N+1,N} = \omega_{1,N} + 2K_N$.

This condition ensures that the sample points spread to fill the whole real line whilst remaining sufficiently dense.[1] We will commonly refer to the numbers K_N as the sampling *bandwidths*. Note that the δ-dense sample points can have arbitrary locations. In particular, the points $\{\omega_{n,N}\}_{n=1}^{N}$ are allowed to cluster arbitrarily. To

[1] We remark in passing that the case of critical density $\delta = 1$ can also be addressed [6], but one cannot in general expect stable reconstruction for $\delta > 1$. See also [11, 12].

compensate for this, we choose the weights $\mu_{n,N}$ in the least-squares (2) as follows:

$$\mu_{n,N} = \tfrac{1}{2}\left(\omega_{n+1,N} - \omega_{n-1,N}\right), \quad n = 1, \ldots, N. \tag{3}$$

With this to hand, we next define the *z-residual* of a finite-dimensional space $\mathrm{T} \subseteq \mathrm{H}$:

$$E_{\mathrm{T}}(M, z) = \sup\left\{ \|\hat{f}\|_{\mathbb{R}\setminus(-z,z)} : f \in \mathrm{T}_M, \|f\| = 1 \right\}, \quad z \in (0, \infty).$$

Here $\|f\|_I = \sqrt{\int_I |f(x)|^2 \, \mathrm{d}x}$ denotes the Euclidean norm over a set I.

Theorem 1 ([6]) *Let $\{\Omega_N\}_{N\in\mathbb{N}}$ be uniformly δ-dense, $\{\mathrm{T}_M\}_{M\in\mathbb{N}}$ be a sequence of finite-dimensional subspaces and let $0 < \epsilon < 1 - \delta$. Let $M, N \in \mathbb{N}$ be such that*

$$E_{\mathrm{T}}(M, K_N - 1/2)^2 \leq \epsilon(2 - \epsilon), \tag{4}$$

then the reconstruction $f \mapsto f_{N,M}$ defined by (1) with weights given by (3) has constant $C(N, M)$ satisfying

$$C(N, M) \leq \frac{1 + \delta}{1 - \epsilon - \delta}. \tag{5}$$

This theorem reinterprets the required scaling of M and N in terms of the z-residual $E(M, K_N - 1/2)$. Note that this residual is independent of the geometry of the sampling points, and depends solely on bandwidths K_N. Hence, provided (4) holds, one ensures stable, quasi-optimal recovery for *any* sequence of sample points $\{\Omega_N\}_{N\in\mathbb{N}}$ with the same parameters K_N.

Unsurprisingly, the behaviour of the z-residual depends completely on the choice of subspaces $\{\mathrm{T}_M\}_{M\in\mathbb{N}}$. Whilst one can often derive estimates for this quantity using ad-hoc approaches for each particular choice of $\{\mathrm{T}_M\}_{M\in\mathbb{N}}$—for example, see [4, 6] for the case of wavelet spaces—it is useful to have a more unified technique to reduce the mathematical burden. We now present such an approach.

Definition 2 ([17]) Let U and V be closed subspaces of H with corresponding orthogonal projections $\mathscr{P}_{\mathrm{U}}$ and $\mathscr{P}_{\mathrm{V}}$ respectively. The gap between U and V is the quantity $G(\mathrm{U}, \mathrm{V}) = \|(\mathscr{I} - \mathscr{P}_{\mathrm{U}})\mathscr{P}_{\mathrm{V}}\|$, where $\mathscr{I} : \mathrm{H} \to \mathrm{H}$ is the identity.

Proposition 1 *Let $\{\mathrm{T}_M\}_{M\in\mathbb{N}}$ and $\{\mathrm{S}_L\}_{L\in\mathbb{N}}$ be sequences of finite-dimensional subspaces of* H. *Then $E_{\mathrm{T}}(M, z) \leq E_{\mathrm{S}}(L, z) + G(\mathrm{S}_L, \mathrm{T}_M)$ for every $M, L \in \mathbb{N}$.*

Proof Let $f \in \mathrm{T}_M$, $\|f\| = 1$. Then

$$\begin{aligned}\|\hat{f}\|_{\mathbb{R}\setminus(-z,z)} &\leq \|\widehat{\mathscr{P}_{\mathrm{S}_L} f}\|_{\mathbb{R}\setminus(-z,z)} + \|f - \mathscr{P}_{\mathrm{S}_L} f\| \\ &\leq E_{\mathrm{S}}(L, z)\|\mathscr{P}_{\mathrm{S}_L} f\| + G(\mathrm{S}_L, \mathrm{T}_M)\|f\| \leq E_{\mathrm{S}}(L, z) + G(\mathrm{S}_L, \mathrm{T}_M).\end{aligned}$$

This result implies the following: if the behaviour of z-residual $E_{\mathrm{S}}(L,z)$ and the gap $G(\mathrm{S}_L,\mathrm{T}_M)$ are known, then one can immediately determine the required scaling of M with z to ensure that $E_{\mathrm{T}}(M,z)$ satisfies (4). We now make the following choice for $\{\mathrm{S}_L\}_{L\in\mathbb{N}}$ to allow us to exploit this result:

$$\mathrm{S}_L = \left\{g \in \mathrm{H} : g|_{[l/L,(l+1)/L)} \in \mathbb{P}_0,\ l = 0,\dots,L-1\right\}. \tag{6}$$

Here $\mathbb{P}_0$ is space of polynomials of degree zero. In [6], it was shown that there exists a constant $c_0(\epsilon) > 0$ such that $E_{\mathrm{S}}(L,z) \leq \epsilon$ whenever $z \geq c_0(\epsilon)L$. Therefore, according to Proposition 1, to estimate $E_{\mathrm{T}}(M,z)$ we now only need to determine $G(\mathrm{S}_L,\mathrm{T}_M)$.

From now on, we let $0 < w_1 < \ldots < w_k < 1$ be a fixed sequence of nodes, and define the space $\mathrm{H}^1_w(0,1) = \{f : f|_{(w_j,w_{j+1})} \in \mathrm{H}^1(w_j,w_{j+1}),\ j = 0,\dots,k\}$ where $w_0 = 0$, $w_{k+1} = 1$ and $\mathrm{H}^1(I)$ is the usual Sobolev space of functions on an interval I. By convention, if $k = 0$ then $\mathrm{H}^1_w(0,1) = \mathrm{H}^1(0,1)$.

Proposition 2 *Suppose that* $\mathrm{T}_M \subseteq \mathrm{H}^1_w(0,1)$ *and let* S_L *be given by (6). If* $L^{-1} \leq \eta = \min_{j=0,\dots,k}\{w_{j+1} - w_j\}$ *then* $G(\mathrm{S}_L,\mathrm{T}_M) \leq \sqrt{\gamma_M^2/(\pi L)^2 + 4\zeta_M^2/L}$, *where*

$$\gamma_M = \max_{j=0,\dots,k} \sup\left\{\|f'\|_{(w_j,w_{j+1})} : f \in \mathrm{T}_M,\ \|f\|_{(w_j,w_{j+1})} = 1\right\},$$

$$\zeta_M = \max_{j=0,\dots,k} \sup\left\{\|f\|_{\infty,(w_j,w_{j+1})} : f \in \mathrm{T}_M,\ \|f\|_{(w_j,w_{j+1})} = 1\right\},$$

and, if I *is an interval,* $\|f\|_I^2 = \int_I |f(x)|^2\,\mathrm{d}x$ *and* $\|f\|_{\infty,I} = \operatorname{ess\,sup}_{x\in I}|f(x)|$. *Moreover, if* $k = 0$, *i.e.* $\mathrm{T}_M \subseteq \mathrm{H}^1(0,1)$, *then* $G(\mathrm{S}_L,\mathrm{T}_M) \leq \gamma_M/(\pi L)$.

Proof Since $L \geq 1/\eta$ there exist $l_j \in \mathbb{N}$ with $l_1 < l_2 < \ldots < l_k$ such that $0 \leq Lw_j - l_j < 1$ for $j = 1,\dots,k$. For an interval $I \subseteq \mathbb{R}$, let us now write $f_I = \frac{1}{|I|}\int_I f$. Then

$$\|f - \mathscr{P}_{\mathrm{S}_L} f\|^2 = \sum_{l=0}^{L-1}\int_{I_l}|f - f_{I_l}|^2 = \sum_{\substack{l=0\\ l\neq l_1,\dots,l_k}}^{L-1}\int_{I_l}|f - f_{I_l}|^2 + \sum_{j=1}^{k}\int_{I_{l_j}}\left|f - f_{I_{l_j}}\right|^2,$$

where $I_l = [l/L,(l+1)/L)$. Since $f \in \mathrm{H}^1(I_l)$ for $l \neq l_1,\dots,l_k$, an application of Poincaré's inequality gives that

$$\|f - \mathscr{P}_{\mathrm{S}_L} f\|^2 \leq \frac{1}{(L\pi)^2}\sum_{\substack{l=0\\ l\neq l_1,\dots,l_k}}^{L-1}\|f'\|_{I_l}^2 + \sum_{j=1}^{k}\int_{I_{l_j}}\left|f - f_{I_{l_j}}\right|^2. \tag{7}$$

We now consider the second term. Write $I_{l_j} = (l_j/L, w_j) \cup (w_j, (l_j+1)/L) = A_j \cup B_j$ and note that for an arbitrary interval I we have $\int_I |f - f_I|^2 = \|f\|_I^2 - |I||f_I|^2$. Hence

$$\begin{aligned}\int_{I_{l_j}} \left|f - f_{I_{l_j}}\right|^2 &= \int_{A_j} \left|f - f_{A_j}\right|^2 + \int_{B_j} \left|f - f_{B_j}\right|^2 + \frac{|A_j||B_j|}{|A_j| + |B_j|} \left|f_{A_j} - f_{B_j}\right|^2 \\ &\leq \frac{1}{(\pi L)^2} \left(\|f'\|_{A_j}^2 + \|f'\|_{B_j}^2\right) + \frac{2|A_j||B_j|}{|A_j| + |B_j|} \left(\|f\|_{\infty,A_j}^2 + \|f\|_{\infty,B_j}^2\right),\end{aligned}$$

where in the final step we use Poincaré's inequality once more and the fact that f is H^1 within A_j and B_j. Since $|A_j|, |B_j| \leq L^{-1}$ and $|A_j| + |B_j| = |I_{l_j}| = L^{-1}$ we now get

$$\sum_{j=1}^{k} \int_{I_{l_j}} \left|f - f_{I_{l_j}}\right|^2 \leq \frac{1}{(\pi L)^2} \sum_{j=1}^{k} \left(\|f'\|_{A_j}^2 + \|f'\|_{B_j}^2\right) + \frac{4}{L} \sum_{j=0}^{k} \|f\|_{\infty,(w_j,w_{j+1})}^2.$$

Combining this with (7) gives

$$\|f - \mathscr{P}_{\mathrm{S}_L} f\|^2 \leq \left(\frac{\gamma_M}{L\pi}\right)^2 \sum_{j=0}^{k} \|f\|_{(w_j,w_{j+1})}^2 + \frac{4\zeta_M^2}{L} \sum_{j=0}^{k} \|f\|_{(w_j,w_{j+1})}^2.$$

Since $\|f\|^2 = \sum_{j=0}^{k} \|f\|_{(w_j,w_{j+1})}^2$ the result now follows.

This proposition provides the main result of this paper. Using it, we deduce that for any $\{\mathrm{T}_M\}_{M\in\mathbb{N}}$, the question of stable reconstruction from any uniformly δ-dense samples now depends solely on the quantities γ_M and ζ_M, which are intrinsic properties of the subspaces completely unrelated to the sampling points.

4 Examples

To illustrate this result, we now present several examples.

Trigonometric Polynomials Functions f that are smooth and periodic can be approximated in finite-dimensional spaces of trigonometric polynomials $\mathrm{T}_M = \left\{\sum_{m=-M}^{M} a_m \mathrm{e}^{2\pi \mathrm{i} m x} : a_m \in \mathbb{C}\right\}$. If $f \in \mathrm{C}^\infty(\mathbb{T})$, where $\mathbb{T} = [0, 1)$ is the unit torus, then the projection error $\|f - \mathscr{P}_{\mathrm{T}_M} f\|$ decay superalgebraically fast in M; that is, faster than any power of M^{-1}. If f is also analytic then the error decays exponentially fast.

For this space, we have $\mathrm{T}_M \subseteq \mathrm{H}^1(0,1)$ and $\gamma_M \leq 2\pi M$ by Bernstein's inequality. Hence Theorem 1 and Propositions 1, 2 give that the reconstruction $f_{N,M}$ is stable and quasi-optimal provided M scales linearly with the sampling bandwidth K_N. This result extends a previous result of [3] to the case of arbitrary nonuniform samples.

Note that this is the best scaling possible up to a constant: for an arbitrary sequence $\{\mathrm{T}_M\}_{M\in\mathbb{N}}$ with $\dim(\mathrm{T}_M) = M$ the scaling of M with K_N is at best linear [6].

Algebraic Polynomials Functions that are smooth but nonperiodic can be approximated by algebraic polynomials. If $\mathrm{T}_M = \mathbb{P}_M$ is the space of algebraic polynomials of degree at most M, then the projection error $\|f - \mathscr{P}_{\mathrm{T}_M f}\|$ decays superalgebraically fast in M whenever $f \in \mathrm{C}^\infty[0,1]$, and exponentially fast when f is analytic.

The classical Markov inequality for this space gives that $\gamma_M \leq \sqrt{2}M^2$, $\forall M \in \mathbb{N}$ [7]. Hence we deduce stability and quasi-optimality of the reconstruction, but only with the square-root scaling $M = \mathscr{O}\left(\sqrt{K_N}\right)$, $N \to \infty$ (this result extends previous results [1, 2, 13] to the case of nonuniform Fourier samples). On the face of it, this scaling is unfortunate since it means the approximation accuracy of $f_{N,M}$ is limited to root-exponential in K_N, which is much slower than the exponential decay rate of the projection error. However, such scaling is the best possible: as shown in [5], any reconstruction algorithm (linear or nonlinear) that achieves faster than root-exponential accuracy for analytic functions must necessarily be unstable.

Piecewise Algebraic Polynomials There are two issues with the previous result. First, the space is not suitable for approximating piecewise smooth functions. Second, the scaling is severe. To mitigate both issues, we may consider spaces of piecewise polynomials on subintervals. In the first case, we fix the intervals corresponding to the discontinuities of the function, and vary the polynomial degree. In the second case, we vary the subinterval size whilst keeping the polynomial degree fixed.

Mathematically, both scenarios equate to considering the subspaces $\mathrm{T}_{w,M} = \{f \in \mathrm{H} : f|_{[w_j,w_{j+1})} \in \mathbb{P}_{M_j},\ j = 0,\dots,k\}$, where $w = \{w_1,\dots,w_k\}$ for $0 = w_0 < w_1 < \dots w_k < w_{k+1} = 1$ and $M = \{M_0,\dots,M_k\} \in \mathbb{N}^{k+1}$. If f is piecewise smooth with jump discontinuities at known locations $0 = w_0 < w_1 < \dots w_k < w_{k+1} = 1$ then the projection error decays superalgebraically fast in powers of $(M_{\min})^{-1}$ as $M_{\min}$ increases, where $M_{\min} = \min\{M_0,\dots,M_k\}$, and exponentially fast if f is piecewise analytic. Alternatively, if f is smooth and the points w are varied whilst the degrees M are fixed, then the error decays like $h^{-M_{\min}-1}$, where $h = \max_{j=0,\dots,k}|w_{j+1} - w_j|$ and $M_{\min} = \min\{M_0,\dots,M_k\}$.

For analysis, we need to determine γ_M and ζ_M. For the first we use the scaled Markov inequality $\|p'\|_I \leq \sqrt{2}M^2/|I|\|p\|_I$, $\forall p \in \mathbb{P}_M$, $M \in \mathbb{N}$, where $|I|$ denotes the length of I. Hence, if $\eta = \min_{j=0,\dots,k}\{w_{j+1} - w_j\}$ then $\gamma_M \leq \sqrt{2}M_{\max}^2/\eta$. For ζ_M, we recall the following inequality for polynomials $\|p\|_{\infty,I} \leq cM/\sqrt{|I|}\|p\|_I$, $\forall p \in \mathbb{P}_M$, $M \in \mathbb{N}$, where $c > 0$ is a constant. Hence $\zeta_M \leq cM_{\max}/\sqrt{\eta}$. We therefore deduce the following sufficient condition: $M_{\max}^2/\eta = \mathscr{O}(K_N)$ as $N \to \infty$. In the first scenario, where η is fixed and $M_{\max}$ is varied, we attain the same square-root-type scaling for piecewise smooth functions when approximated by piecewise polynomials as with the polynomial space of the previous example. In the second scenario, where $M_{\max}$ is fixed and η is varied, we see that this leads to a linear relation between K_N and η. Thus, by forfeiting the superalgebraic/exponential convergence of the polynomial space for only algebraic convergence, we obtain a better scaling with K_N. Note that

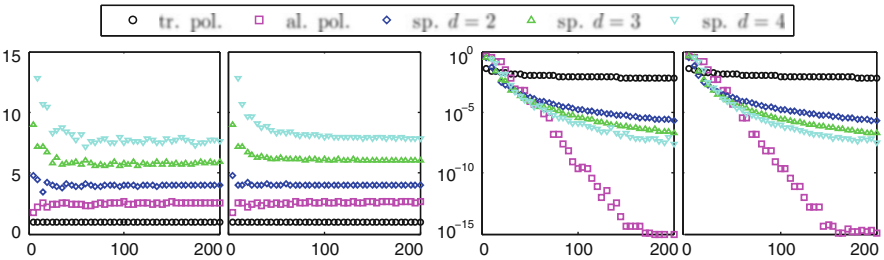


Fig. 1 In the first pair of panels, depending on the type of the reconstruction space, appropriate ratios are shown: M/K_N (for trigonometric polynomials), $M/\sqrt{K_N}$ (for algebraic polynomials) and Md^2/K_N (for splines of order d), where for a given $K_N \in [5, 200]$, we used $M = \max\{M \in \mathbb{N} : C(N, M) \leq 3\}$. In the second pair of panels, for such K_N and M, the error $\|f - f_{N,M}\|$ is plotted where $f(x) = x^2 + x\sin(4\pi x) - \exp(x/2)\cos(3\pi x)^2$. We used different sampling schemes Ω_N: jittered (for the first and third panel) and log (for the second and forth panel)

in some cases it may be desirable to approximate using functions that are themselves smooth (up to a finite order). In this case, we can replace $\mathrm{T}_{w,M}$ by the spline space $\tilde{\mathrm{T}}_{w,M_{\min}}$ of degree $M_{\min}$ on the knot sequence w. Since $\tilde{\mathrm{T}}_{w,M_{\min}} \subseteq \mathrm{T}_{w,M}$ we obtain the same linear scaling with K_N in this case as well.

Numerical Results We demonstrate our results using two common nonuniform sampling schemes; jittered and log sampling (see [6] for details). In the first two panels of Fig. 1, we illustrate the scaling for different spaces T_M between the sampling bandwidth K_N and space dimension M such that $C(N, M)$ is bounded. For such K_N and M, in the second pair of panels, we compute the L^2 error of the approximation $f_{N,M}$ for a continuous function f. The superiority of the spline spaces for small N is evident, with the polynomial space becoming better as N increases.

Acknowledgements Ben Adcock acknowledges support from the NSF DMS grant 1318894. Milana Gataric acknowledges support from the UK EPSRC grant EP/H023348/1 for the University of Cambridge Centre for Doctoral Training, the Cambridge Centre for Analysis. Anders C. Hansen acknowledges support from a Royal Society University Research Fellowship as well as the EPSRC grant EP/L003457/1.

References

1. B. Adcock, A.C. Hansen, Stable reconstructions in Hilbert spaces and the resolution of the Gibbs phenomenon. Appl. Comput. Harmon. Anal. **32**(3), 357–388 (2012)
2. B. Adcock, A.C. Hansen, Generalized sampling and the stable and accurate reconstruction of piecewise analytic functions from their Fourier coefficients. Math. Comp. **84**, 237–270 (2015)
3. B. Adcock, A.C. Hansen, C. Poon, Beyond consistent reconstructions: optimality and sharp bounds for generalized sampling, and application to the uniform resampling problem. SIAM J. Math. Anal. **45**(5), 3114–3131 (2013)

4. B. Adcock, A.C. Hansen, C. Poon, On optimal wavelet reconstructions from Fourier samples: linearity and universality of the stable sampling rate. Appl. Comput. Harmon. Anal. **36**(3), 387–415 (2014)
5. B. Adcock, A.C. Hansen, A. Shadrin, A stability barrier for reconstructions from Fourier samples. SIAM J. Numer. Anal. **52**(1), 125–139 (2014)
6. B. Adcock, M. Gataric, A.C. Hansen, On stable reconstructions from nonuniform Fourier measurements. SIAM J. Imaging Sci. **7**(3), 1690–1723 (2015)
7. A. Böttcher, P. Dörfler, Weighted Markov-type inequalities, norms of Volterra operators, and zeros of Bessel functions. Math. Nachr. **283**(1), 40–57 (2010)
8. A. Gelb, T. Hines, Detection of edges from nonuniform Fourier data. J. Fourier Anal. Appl. **17**, 1152–1179 (2011)
9. A. Gelb, T. Hines, Recovering exponential accuracy from non-harmonic Fourier data through spectral reprojection. J. Sci. Comput. **51**, 158–182 (2012)
10. A. Gelb, G. Song, A frame theoretic approach to the non-uniform fast Fourier transform. SIAM J. Numer. Anal. **52**(3), 1222–1242 (2014)
11. K. Gröchenig, Reconstruction algorithms in irregular sampling. Math. Comp. **59**, 181–194 (1992)
12. K. Gröchenig. Irregular sampling, Toeplitz matrices, and the approximation of entire functions of exponential type. Math. Comp. **68**(226), 749–765 (1999)
13. T. Hrycak, K. Gröchenig, Pseudospectral Fourier reconstruction with the modified inverse polynomial reconstruction method. J. Comput. Phys. **229**(3), 933–946 (2010)
14. J.I. Jackson, C.H. Meyer, D.G. Nishimura, A. Macovski, Selection of a convolution function for Fourier inversion using gridding. IEEE Trans. Med. Imaging **10**, 473–478 (1991)
15. A. Martinez, A. Gelb, A. Gutierrez, Edge detection from non-uniform Fourier data using the convolutional gridding algorithm. J. Sci. Comput. **61**, 490–512 (2014)
16. R. Platte, A.J. Gutierrez, A. Gelb, Fourier reconstruction of univariate piecewise-smooth functions from non-uniform spectral data with exponential convergence rates. Appl. Comput. Harm. Anal. **39**(3), 427–449 (2015)
17. D. Szyld, The many proofs of an identity on the norm of oblique projections. Numer. Algorithms **42**, 309–323 (2006)
18. A. Viswanathan, A. Gelb, D. Cochran, R. Renaut, On reconstructions from non-uniform spectral data. J. Sci. Comput. **45**(1–3), 487–513 (2010)

A Parallel-in-Time-and-Space HPC Framework for a Class of Fractional Evolution Equations

Ahmad Alyoubi and Mahadevan Ganesh

Abstract We develop a high performance computing (HPC) framework for efficient simulations of a class of fractional-order partial differential equations (FPDE), using high-order in time and space parallel algorithms. HPC systems provide a large number of processing cores with limitations on the amount of memory available per core. Such limitations impose severe constraints for resolving fine spatial structures that require large degrees of freedom (DoF). In this article, using several message passing interface (MPI) communicators, we develop and demonstrate an efficient hybrid framework that combines parallel in time and space tasks that facilitate careful balance between parallel performance within the memory constraint to simulate the FPDE model. We demonstrate the approach for a 3D fractional PDE using several million spatial DoF.

1 Introduction

In this work, we consider a fractional-order space-time evolution equation that is of recent interest to efficiently model (anomalous) physical processes that do not conform to standard modeling based on integer (local) time derivative operators. In particular, our focus is on an efficient high performance computing (HPC) approach to resolve fine spatial structures in the solution $u(\cdot, t),\ t \in (0, T]$ and perform long-time simulation (say, the final time is several hundred order higher compared to the standard simulation approach of taking $T = 1$). For example, in tumor growth (semi-linear) biological models [2] (with Turing space parameters), the main interest is in the long-time (steady state) behavior of a diffusion driven evolution process, resulting from perturbation of the homogeneous (diffusion less) steady state.

In such cases, a practical requirement is to analyze the behavior of the solution at various user specified *short* list of increasing time periods $t_k \in (0, T],\ k = 1, \ldots, m$ (with t_1 small and t_m relatively large) and, if required, augment the list with a few

A. Alyoubi (✉) • M. Ganesh
Department of Applied Mathematics and Statistics at Colorado School of Mines, Golden, CO, USA
e-mail: ahmad.alyoubi@yahoo.com; mganesh@mines.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_9

more periods until a desired property is reached. Depending on the behavior at t_k, one may also be interested in altering the input source term in the model at t_{k+1} and then probe the model, to explain properties such as phase transitions, vortex states and symmetry breaking that have been observed in various experiments [4–6].

For such practical requirements, traditional time-stepping (serial-in-time) algorithms lead to severe computational bottleneck. Particularly, if t_m is large and at each discrete time step, several million spatial of degrees of freedom (DoF) are needed to resolve fine scale structures. It is well known that the time-stepping requirement in linear models can be avoided by using the Laplace transform (LT) approach.

Standard LT algorithms require that the LT of the source term can be evaluated. This source term restriction is severe, especially for linearized version of the semi-linear models (with source terms depending on the unknown solution) and also for cases where one needs to dynamically modify the source term, depending on the behavior of the solution at the previous time period in the user specified list. In this article, we develop an efficient high-order HPC parallel-in-time-and-space framework that does not impose such severe restrictions on the source term.

The rate of convergence of the LT algorithm-based approximate solution, say $u_{M_z,h}$, depends crucially on the (inverse-LT) choice of a contour in the complex plane to evaluate the solution in the time domain using a contour integral and a quadrature rule to discretize the integral. Here, M_z denotes the number of quadrature points required to discretize the contour integral and h is the chosen spatial discretization parameter for simulating the frequency domain elliptic partial differential equation (PDE) model at each LT based quadrature point. The M_z elliptic PDEs are independent and hence, the LT based approach leads to a naturally parallel algorithm (NPA). The standard HPC approach for the NPA is to use only built-in message passing interface (MPI) communicator `MPI_COMM_WORLD` to distribute the tasks.

A major disadvantage of the standard parallel implementation of the LT algorithms is the requirement of large memory (as several elliptic PDEs need to be solved simultaneously). Distribution of the work is severely restricted by the availability of memory. Memory limitation (in each multi-core compute node of a cluster of HPC nodes) demands that substantial effort is required to achieve balanced scalability with respect to memory. Thus, parallelization in the LT and spatial variables are required.

The main contribution of this short (8-page) article is to describe and demonstrate such HPC technical details (based on multiple MPI communicators) for a model problem for a range of processing cores P (that depends on the data size parameters M_z, h). We demonstrate our approach for 2D and 3D models using high-order discretization in the LT variable and high/low-order finite-element/finite-difference methods (FEM/FDM). Many industrial standard codes for 3D elliptic PDEs use low-order approximations such as second-order FDM or equivalent piecewise linear FEM. However, it is efficient to develop high-order FEM in 2D models.

High-order approximations in the LT variable and evaluation of the solution even for very small time can be achieved using the smooth contour proposed and analyzed in [3, 7] (that depends on certain properties of the spatial differential operator). Our

approach is based on the method developed and analyzed in [8] (that utilizes various techniques in [3, 7] and references therein). Implementation of the method in [3, 7, 8] were carried out only in serial (using one processor) for some zero (no spatial) and two dimensional spatial examples with only a few thousand DoF using low-order piecewise-linear FEM. The present work is a practical HPC counterpart of the mathematical methods and analysis developed over the last decade. Our long-time simulations include high-order FEM discretization and millions of DoF that are typically required to resolve fine spatial structures in the solution of the model.

2 A Model Problem: Ansatz and Discretization

We consider the following model problem based on non-local (fractional) time and local spatial derivative operators: Compute approximations to the solution $u(\mathbf{x}, t)$, for $\mathbf{x} \in \Omega \subset \mathbb{R}^n$, $n = 2, 3$ and at $t \in S_m = \{t_k : k = 1, \ldots, m\} \subset (t_0, T]$ such that

$$\frac{\partial u}{\partial t} - \frac{\partial^{-\alpha}}{\partial t^{-\alpha}} [\nabla \cdot (\sigma \nabla u)] = f(\mathbf{x}, t), \qquad \text{in } \Omega \times (t_0, T], \quad u(\mathbf{x}, t_0) = u_0(\mathbf{x}), \quad \text{in } \Omega, \tag{1}$$

and $u(\mathbf{x}, t) = 0$ on $\partial\Omega$ for $t \in [t_0, T]$. We refer to [8] and references therein for further details of the PDE. In (1), $\sigma > 0$ is independent of time and chosen so that $A = -\nabla \cdot (\sigma \nabla)$ satisfies technical conditions required in [8] for the inverse-LT contour representation and analysis. For example, $\sigma = 1$ on Ω satisfies such conditions.

The model problem (1) with $\alpha = 0$ is the standard parabolic evolution initial boundary value problem. For non-zero $\alpha \in (-1, 1)$, we use the Riemann–Liouville definition of the non-local fractional-order operator: If $-1 < \alpha < 0$, $\frac{\partial imath^{-\alpha}}{\partial imath t^{-\alpha}} u(t) := \frac{\partial imath}{\partial imath t} \int_0^t \frac{(t-s)^{\alpha}}{\Gamma(1+\alpha)} u(s) \mathrm{d}s$, and if $0 < \alpha < 1$, $\frac{\partial imath^{-\alpha}}{\partial imath t^{-\alpha}} u(t) := \int_0^t \frac{(t-s)^{\alpha-1}}{\Gamma(\alpha)} u(s) \mathrm{d}s$, where Γ is the *Euler Gamma function*. The case of $-1 < \alpha < 0$ in (1) is for anomalous sub-diffusion models, and $0 < \alpha < 1$ case is suitable for viscoelastic applications [8].

Let $\widetilde{\Gamma}$ be a contour in the complex plane $\mathbb{C}$ chosen as in [8] (and references therein). Throughout the article, we use $z \in \mathbb{C}$ to denote points on $\widetilde{\Gamma}$. For $t \in S_m$, we represent the solution using a contour integral on $\widetilde{\Gamma}$. For a given function g defined on $\widetilde{\Gamma} \times \Omega \times S_m \cup \{t_0\}$, we define g_k on $\widetilde{\Gamma} \times \Omega$, $k = 0, \ldots, m$, as $g_k(z, \mathbf{x}) = g(z, \mathbf{x}, t_k)$.

For a chosen data g_k, it is convenient to consider a PDE solution operator $S^{\alpha}(z)$ defined as $[S^{\alpha}(z) g_k](\mathbf{x}) = w_k(\mathbf{x}, z)$, for $\mathbf{x} \in \Omega$, where w_k is the solution of the PDE

$$z^{1+\alpha} w_k(\mathbf{x}, z) - \nabla \cdot (\sigma \nabla w_k(\mathbf{x}, z)) = z^{\alpha} g_k(z, \mathbf{x}), \quad \mathbf{x} \in \Omega, \quad z \in \widetilde{\Gamma}, \quad k = 1, \ldots, m, \tag{2}$$

satisfying the homogeneous Dirichlet boundary condition on $\partial\Omega$. For example, if we take $f = 0$ in (1) and then apply the LT in (1), we obtain (2) with $g_k(\mathbf{x}, z) = u_0(\mathbf{x})$ and w_k being the LT of u [8]. In this work, we avoid the requirement that the LT of the forcing function f in (2) to exist or is known. Hence, for each t_k, we consider $S^\alpha(z)g_k$ to represent the solution of (1) using Duhamel's formula based data g_k, depending on u_0, f and a set of quadrature points z (to approximate the inverse-LT).

We simulate the PDE (2) using a high-order FEM or low-order FDM with a mesh parameter h and denote the corresponding solution as $w_k^h(\cdot, z)$. As described below, we choose $2N + 1$ points $z_j \in \widetilde{\Gamma}$, $j = -N, \ldots, N$. The computational complexity is dominated by the need to solve $m(2N + 1)$ independent PDEs (2) with $z = z_j$. More precisely, for our solution representation, we need to compute $w_{j,k}^h(\mathbf{x}) = w_k^h(\mathbf{x}, z_j)$ for $j = -N, \ldots, N$, $k = 1, \ldots m$. The algorithm provides a convenient framework to add more points in S_m and changes in the source term at different time steps.

Following [8], using Duhamel's formula, the forward and inverse LT techniques, we represent the desired solution $u(\mathbf{x}, t_k)$ of (1) for $k = 1, \ldots, m$ as

$$u(\mathbf{x}, t_k) = u_0(\mathbf{x}) + \int_0^{t_k} f(\mathbf{x}, s)\mathrm{d}s + \frac{1}{2\pi \mathrm{i}} \int_{\widetilde{\Gamma}} \left[S^\alpha(z) g_k(z, \mathbf{x}) - z^{-1} g_k(z, \mathbf{x})\right] \mathrm{d}z, \tag{3}$$

where g_k is defined using the initial and source data in the model:

$$g(z, \mathbf{x}, t) = e^{zt} u_0(x) + \int_0^t e^{z(t-s)} f(\mathbf{x}, s)\mathrm{d}s, \qquad g_k(z, \mathbf{x}) = g(z, \mathbf{x}, t_k). \tag{4}$$

For a chosen LT variable discretization parameter N, we use a quadrature approximation [8] of the contour integral in (3) (by mapping the contour to the real line, accounting for the Jacobian term, and choosing $2N + 1$ equally spaced points). Using the quadrature points $z_j \in \widetilde{\Gamma}$, $j = -N. \ldots, N$ and a weight parameter q_N (for example $q_N = 1/\sqrt{N}$) and denoting the Jacobian function evaluated at z_j as z_j', the computable approximation to $u(\mathbf{x}, t_k)$, for $\mathbf{x} \in \Omega$ and $k = 1, \ldots, m$, is defined as

$$u_{N,h}(\mathbf{x}, t_k) = u_0(\mathbf{x}) + \int_0^{t_k} f(\mathbf{x}, s)\mathrm{d}s + \frac{q_N}{2\pi \mathrm{i}} \sum_{j=-N}^{N} \left[w_{j,k}^h(\mathbf{x}) - z_j^{-1} g_k(z_j, \mathbf{x})\right] z_j'. \tag{5}$$

The LT based time discretization approach has been proven to be high-order accurate [8]. It is ideal to choose N to match the spatial discretization accuracy. The standard piecewise linear FEM approach leads to $\mathcal{O}(h^2)$ convergence and this can be improved to $\mathcal{O}(h^{d+1})$, in the L^2 norm using the spline degree $d \geq 2$, provided the solution of the continuous model is sufficiently smooth. In this work, we demonstrate the approach using both the high-order FEM (for a smooth test solution case) and use a $\mathcal{O}(h^2)$ method to simulate a 3D fractional PDE (FPDE) model on a domain with edges and corners with unknown solution that is in general non-smooth.

3 Parallel Implementation Techniques

For each $k = 1, \ldots, m$, computation of the solution in (5) requires solving $(2N + 1)$ independent PDEs (2) with $z = z_j$, for $j = -N, \ldots, N$. For simple model problems, it is straightforward to parallelize with respect to the LT variable using the MPI by distributing the discrete LT points among all cores determined by the built-in communicator `MPI_COMM_WORLD`. The simplicity is based on the assumption that the FEM/FDM based discretization of each PDE requires only a few thousand DoF and that the memory accessible by each core is sufficient to accommodate the complexity. In this work, we are interested in solving model problems with fine scale features and hence, it requires tens of thousands to several million DoF. In such cases, parallelization in spatial variable is also necessary, by distributing the entries of the mass and stiffness matrices of the discrete system. In this section, we describe efficient coupling of the parallel-in-time algorithm (described in the last section) and a general class of parallel-in-space algorithms using multiple MPI communicators.

Cores and Load Distribution We distinctly identify each processing core in two local classes of communicators containing subsets of cores in the `MPI_COMM_WORLD` communicator. We call the two classes of communicators, obtained by reshaping the total number of cores in a rank two matrix form, as row and column communicators. We use row communicators to distribute the LT discretization points among them. For each row communicator, the FEM/FDM matrices and vectors are distributed among its cores. We use the column communicators to gather the partial solutions from all row communicators and then use these to obtain the sum in (5).

More precisely, let N_t (a non-prime integer number) be the total number of cores in `MPI_COMM_WORLD` and let $N_t = N_r\, N_c$. We create N_r row communicators and N_c column communicators to obtain additional $N_r + N_c$ communicators. The choice of N_r and N_c depends on the availability of memory for each core and the spatial DoF. We tag each core with three ranks: first being that in `MPI_COMM_WORLD` and the other two depend on the position in the matrix form of the local row-column communicators. Let $p_{\check{i},\check{j}}$ denote the rank of a core if it belongs to both the $\check{i}$-th row communicator and $\check{j}$-th column communicator, for $\check{i} = 0, \ldots, N_r - 1$ and $\check{j} = 0, \ldots, N_c - 1$. Ranks of cores in the `MPI_COMM_WORLD` communicator are denoted by $P_{\check{k}}$ where $\check{k} = 0, \ldots, N_t - 1$.

There are two kinds of parallel tasks required for our model problem: parallel tasks for the LT variable discretization and parallel tasks for spatial discretization. In this segment, these tasks are identified and distributed. Each core should have a subset from both kinds. Let $\tau_z = m(2N + 1)$ be the total number of parallel tasks described in the previous section for the parallel-in-time algorithm. We distribute these parallel tasks among the N_r row communicators. For each fixed k and for

$l = 0, \ldots, N_r - 1$, let $[d_0^l, d_1^l]$ be the sub-interval we use for distributing indices of the LT discretization points. The interval can be obtained for the l-th row communicator as follows:

$$d_0^l = l\theta + 1, \quad d_1^l = \begin{cases} d_0^l + \theta + \kappa - 1 & \text{if } l = N_r \text{ and } \kappa > 0, \\ d_0^l + \theta - 1 & \text{if } l < N_r \text{ or } \kappa = 0, \end{cases} \tag{6}$$

where $\theta = \lfloor (2N+1)/N_r \rfloor$ and $\kappa = (2N+1) \bmod N_r$. Since the LT discretization points are the same for all t_k, these can be computed once and used repeatedly to get the solution for t_k. Thus, for a given row communicator, there are $m(d_1^l - d_0^l + 1)$ parallel tasks.

The parallel tasks in the spatial variables are distributed among all cores in a given row communicator. For each k and $j = -N, \ldots, N$, the FDM/FEM discretization of the PDE (2) with $z = z_j$ yields a linear algebraic system of the form $Ax = b$ where A is a sparse matrix of size $M \times M$ and b is a load vector. We distribute these load vectors and the matrices equally among all cores in a given row communicator. The vectors and the matrices should be locally created. Let $s_{\check{j}}$ be the total number of rows of the local arrays in the $\check{j}$-th core in a row communicator. Thus, the local x and b vectors are of length $s_{\check{j}}$ whereas the local matrix A is of size $s_{\check{j}} \times M$. With $\eta = \lfloor M/N_c \rfloor$, $\chi = M \bmod N_c$ and $0 \le \check{j} \le N_c - 1$, we have $s_{\check{j}}$

$$s_{\check{j}} = \begin{cases} \eta + 1 & \text{if } \check{j} < \chi, \\ \eta & \text{if } \check{j} \ge \chi. \end{cases} \tag{7}$$

Additionally, it is necessary to compute the global range of local vectors and the rows of the local matrices. For $\check{j} = 0, \ldots, N_c - 1$, this range is represented by the interval $[r_0^{\check{j}}, r_1^{\check{j}}]$ where $r_0^{\check{j}}$ is the first index and $r_1^{\check{j}}$ is the last index of the first dimension of the local arrays for the $\check{j}$-th core in any row communicator. They should be locally computed as

$$r_0^{\check{j}} = \begin{cases} \check{j}\eta & \text{if } \chi = 0, \\ \check{j}(\eta+1) & \text{if } \check{j} < \chi \text{ and } \chi > 0, \\ \chi(\eta+1) + \left(\check{j} - \chi\right)\eta & \text{if } \check{j} \ge \chi \text{ and } \chi > 0, \end{cases} \quad r_1^{\check{j}} = \begin{cases} r_0^{\check{j}} + \eta & \text{if } \check{j} < \chi, \\ r_0^{\check{j}} + \eta - 1 & \text{if } \check{j} \ge \chi. \end{cases} \tag{8}$$

Further for efficient memory utilization, in addition to dynamic memory allocation, it is essential (especially for 3D models) to exploit the sparse structure of the linear system. This can be achieved in several ways. For example, in the PETSc

environment, we can utilize *shell matrix* technique (also known as matrix-free operations).

Computation of Approximation Solution For $l = 0, \ldots, N_r - 1$, let $u^l_{N,h}$ denotes the partial approximation solution that is owned by a single row communicator and determined by the representation in the third term in (5) and the full summation is replaced with partial summation from d^l_0 to d^l_1. At this point, there are N_r partial approximation solutions performed by N_r row communicators. These solutions are summed up using one of the MPI reduce subroutines using N_c column communicators. Then, the first two terms in (5) should be computed and added to the final solution.

4 Numerical Experiments

In this section, we demonstrate the efficiency of the approach described in the last two sections for an FPDE 3D model (1) with $\alpha = -0.5$, $\sigma = 1$, $t_0 = 0$, $T = 1000$, $m = 5$, $S_m = \{0.1, 1, 10, 100, 1000\}$, and $\Omega = (0, 4) \times (0, 4) \times (0, 4)$. The evolution process is induced by an initial condition u_0 with several scales, as shown in Fig. 1 for a fixed $x_3 \in (0, 4)$.

For demonstrating the accuracy of computed solutions, as the number of DoF increases, it is standard to test the code with some special choice of exact solution $u(\mathbf{x}, t)$ and derive the source function $f(\mathbf{x}, t)$ by substituting the test solution in the LHS of (1). Such an approach is practical for (1) only with $\alpha = 0$. Otherwise, a standard approach is to consider a computed fine grid solution as the reference solution and demonstrate the accuracy. It is important to make sure that the fine grid is chosen so that the reference solution is highly accurate.

In order to obtain some ideas about the fine grid for the 3D FPDE model, we first consider a 2D version of the model problem with $\alpha = 0$. Results in Table 1 demonstrate that for the simulation of a special test choice of $u(\mathbf{x}, t)$ with $\mathbf{x} \in (0, 4) \times (0, 4)$ (with $t = 0$ case having multiple scales) requires over half a million DoF (using a high-order method) even for about 0.3 % accuracy. This is the case despite

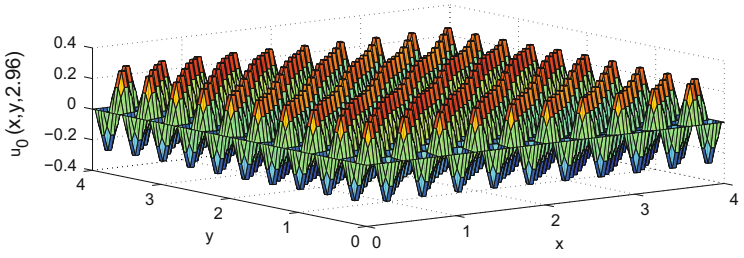


Fig. 1 Initial state of a fractional evolutionary process

Table 1 Results (L^2-error, EOC-estimated order of convergence, and CPU time using a compute node with 12 cores) for the 2D model problem at $T = 1000$, using a high-order FEM with quadratic ($d = 2$) and cubic ($d = 3$) splines various mesh sizes with coarse mesh $h = 0.125$ and $N = 640$

$d = 2$				
h_{min}	DoF	L^2 Error	*EOC*	Time (s)
h	1953	8.72E+0	–	3.9559E+0
$h/2$	8001	1.22E+0	2.84	1.7969E+1
$h/4$	32,385	2.36E−1	2.37	1.1146E+2
$h/8$	130,305	2.73E−2	3.11	9.9093E+2
$h/16$	522,753	3.21E−3	3.09	1.5492E+4

$d = 3$			
DoF	L^2 Error	*EOC*	Time (s)
4465	3.29E+0	–	9.9152E+0
18,145	4.61E−1	2.84	5.7653E+1
73,153	4.34E−2	3.41	5.7319E+2
293,761	2.70E−3	4.01	8.3841E+3

Table 2 Left table: Maximum nodal errors and the EOC for several grid sizes at $T = 1000$ with $N = 100$. Right table: Maximum nodal errors for simulating at all $t \in S_m$ using DoF=16,777,216 grid

DoF	h	Max. nodal error	EOC
4096	0.250000	1.59E−1	–
32,768	0.125000	3.98E−2	2.00
262,144	0.062500	9.86E−3	2.01
2,097,152	0.031250	2.35E−3	2.07
16,777,216	0.015625	4.69E−4	2.32

$t \in S_5$	Max. nodal error
0.1	1.03E−5
1.0	1.32E−5
10.0	4.93E−5
100.0	1.67E−4
1000.0	4.69E−4

using a high-order FEM with spline degree $d = 2$ and similar accuracy can be obtained with about half of the DoF by increasing the spline degree to $d = 3$ because of the high estimated order convergence (EOC) is $\mathcal{O}(h^{d+1})$ in the L^2 norm. For the 2D model problem, we chose the smooth solution to be

$$u(x, y, t) = \left(1 + \frac{t}{T}\right) e^{-\frac{t}{T}} \sin(20\,\pi x) \sin(10\,\pi y), \tag{9}$$

and hence with $t = 0$, it is easy to observe features of the solution as in Fig. 1.

The 2D model (smooth solution) simulation results in Table 1 clearly indicate that for the 3D model problem with unknown (non-smooth) solution, we need several millions of DoF to obtain the reference solution. For our 3D numerical experiments, we compute the reference solution with DoF= 134,217,728 using a fine $512 \times 512 \times 512$ grid on Ω. We apply a lower order FDM with EOC $\mathcal{O}(h^2)$ for the 3D model.

Results in Table 2 (with N_t, N_c, N_r as described in the last section) demonstrate that millions of DoF are required to simulate the FPDE model in 3D to achieve at least 0.1 % accuracy, even for the simple case $f(\mathbf{x}, t) = \exp(-t/(4T))$, for $\mathbf{x} \in \Omega$.

Next, we demonstrate the scalability of the practical HPC approach. We utilize the PETSc environment to implement spatial parallelization [1]. Our hybrid parallel-in-time-and-space approach provides a framework to efficiently utilize the naturally parallel aspect of the LT based time discretization and the amount of memory available. We simulated the 3D model using a Blue Gene Q HPC cluster with each compute node comprising a PowerPC A2 processor with 16 cores and

Table 3 Parallel performance for the 3D FPDE model to compute solution at $T = 1000$ with $N = 100$

$N_r \times N_c$	Time (h)	Speedup
1×8	9.71	
2×8	4.90	1.98
4×8	2.92	3.33
8×8	1.52	6.38
16×8	0.82	11.81

$N_r \times N_c$	Time (h.)	Speedup
1×16	29.66	
2×16	14.98	1.98
4×16	9.33	3.18
8×16	5.13	5.78

Left Table: DoF= 27,000,000 and N_c=8. Right Table: DoF= 134,217,728 and N_c=16

16 GB memory. Thus, memory available per core in the powerful machine is just 1 GB.

Results in Table 3 (with f depending on $\mathbf{x}$ and t) demonstrate the HPC performance of our hybrid parallelization with (1) 27, 000, 000 DoF using a $300 \times 300 \times 300$ grid; and (2) 134,217,728 DoF using a $512 \times 512 \times 512$ grid, for the 3D FPDE model problem. Based on the data size and memory limitation, several communicators (with a careful choice of N_r and N_c) are needed for efficient parallel implementation. For 27,000,000 and 134,217,728 cases, we choose N_c to be 8 and 16, respectively and achieve good parallel performance, as demonstrated in Table 3.

Acknowledgements The research of the first author was supported by Aramco and the second author was supported, in part, by grant DMS-1216889 from the NSF. Support of the Colorado Golden Energy Computing Organization is gratefully acknowledged.

References

1. S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W. Gropp, D. Kaushik, M. Knepley, L. McInnes, B. Smith, H. Zhang, PETSc users manual. Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory, 2013
2. M. Chaplain, M. Ganesh, I. Graham, Spatio-temporal pattern formation on spherical surfaces: numerical simulation and application to solid tumour growth. J. Math. Biol. **642**, 387–423 (2001)
3. I.P. Gavrilyuk, V.L. Makarov, Exponentially convergent algorithms for the operator exponential with applications to inhomogeneous problems in Banach spaces. SIAM J. Numer. Anal. **43**(5), 2144–2171 (2005)
4. N. Goldenfeld, *Lectures on Phase Transitions and the Renormalization Group*. Frontiers in Physics, vol. 85 (Addison-Wesley, Boston, 1992)
5. A. Kanda, B.J. Baelus, F.M. Peeters, K. Kadowaki, Y. Ootuka, Experimental evidence for giant vortex states in a mesoscopic superconducting disk. Phys. Rev. Lett. **93**, 257002 (2004)
6. M. Kryvohuz, J. Cao, Noise-induced dynamic symmetry breaking and stochastic transitions in ABA molecules: II. Symmetric-antisymmetric normal mode switching. Chem. Phy. **370**, 258–269 (2010)
7. M. Lopez-Fernandez, C. Palencia, A. Schadle, A spectral order method for inverting sectorial Laplace transforms. SIAM J. Numer. Anal. **44**(3), 1332 (2006)
8. W. McLean, V. Thomée, Numerical solution via Laplace transforms of a fractional order evolution equation. J. Integral Equ. Appl. **22**, 57–94 (2010)

High-Order Upwind Methods for Wave Equations on Curvilinear and Overlapping Grids

J.W. Banks and W.D. Henshaw

Abstract In this work we discuss a newly developed class of robust and high-order accurate upwind schemes for wave equations in second-order form on curvilinear and overlapping grids. The schemes are based on embedding d'Alembert's exact solution for a local Riemann-type problem directly into the discretization (Banks and Henshaw, J Comput Phys 231(17):5854–5889, 2012). High-order accuracy is obtained using a single-step space-time scheme. Overlapping grids are used to represent geometric complexity. The method of manufactured solutions is used to demonstrate that the dissipation introduced through upwinding is sufficient to stabilize the wave equation in the presence of overlapping grid interpolation.

1 Introduction

Upwind methods for first-order hyperbolic partial differential equations (PDEs) have been extremely effective at facilitating the simulation of a wide variety of physical problems. The success of upwind methods can largely be attributed to the incorporation of natural dissipation through the embedding of the characteristic wave-structure of the hyperbolic system into the discretization. Many well-known and powerful schemes have their roots in these ideas. A partial list includes the Courant-Isaacson-Rees scheme [9], flux-corrected transport [5], total-variation-diminishing methods [16] the piecewise-parabolic method (PPM) [8], essentially-non-oscillatory (ENO) schemes [12], discontinuous Galerkin (DG) approximations [7], and the weighted-essentially-non-oscillatory (WENO) class of methods [15].

In a recent paper [2], we extended these powerful ideas to wave equations written directly in second-order form without the need to recast governing equations as a system of first-order PDEs. There are numerous potential advantages of solving the second-order form directly such as fewer dependent variables and fewer constraint equations. The approach was based on incorporating the well-known d'Alembert

J.W. Banks (✉) • W.D. Henshaw
Department of Mathematical Sciences, Rensselaer Polytechnic Institute, New York, NY, USA
e-mail: banksj3@rpi.edu; henshw@rpi.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_10

solution into the discretization. Following the well established procedure for upwind treatments for the first-order form, a localized expression of the upwind flux was derived that enables easy application to a wide class of problems including multiple dimensions and variable coefficients. In this work we demonstrate the extension of upwind scheme for the wave equation in second-order form to the cases of curvilinear grids and overlapping grids. As discussed in [1], dissipation free schemes may exhibit instabilities on overlapping grids due to perturbations from the interpolation formula; these instabilities were found to be naturally suppressed by upwind schemes for waves equations in first-order form. This property has permitted many stable overlapping grid capabilities for hyperbolic PDEs (e.g. [1, 3, 4, 14]). In the current work we demonstrate that upwind methods for the second-order system also appear to be naturally stable when used with overlapping grids.

2 Governing Equations and Overlapping Grids

Consider the discretization of the scalar wave equation on a domain Ω,

$$\frac{\partial^2 u}{\partial t^2} = Lu \equiv c^2 \Delta u, \quad \mathbf{x} \in \Omega, \tag{1}$$

where $\mathbf{x} \in R^d$, $u = u(\mathbf{x}, t)$, c is a constant wave speed, and Δu is the Laplacian operator in d space dimensions. Appropriate boundary and initial conditions are also applied. We will discretize (1) using an overlapping grid approach where the overall domain is covered by an overlapping grid G consisting of a set of component grids G_k that communicate through interpolation. Such a scenario is depicted in Fig. 1 which shows a domain consisting of an annular grid (green) and a rectangular grid (blue). In the region where these two grids overlap the solution is communicated from one grid to the other using interpolation. For further details on overlapping grids refer to [6] and the references therein.

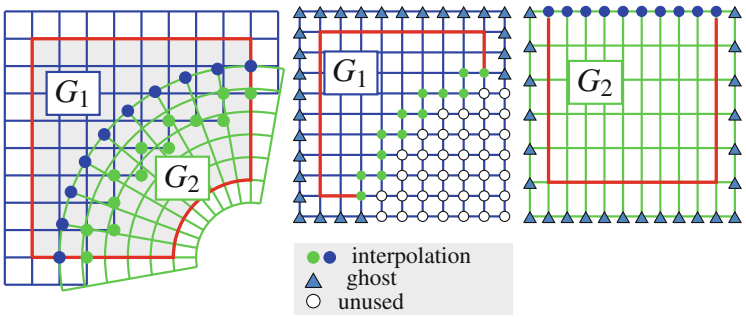


Fig. 1 *Left*: an overlapping grid consisting of two structured curvilinear component grids, $\mathbf{x} = G_1(\mathbf{r})$ and $\mathbf{x} = G_2(\mathbf{r})$. *Middle* and *right*: component grids for the square and annular grids in the unit square parameter space $\mathbf{r}$. Grid points are classified as discretization points, interpolation points or unused points. Ghost points are used to apply boundary conditions

For each component grid we define a smooth mapping $\mathbf{x} = \mathbf{G}(\mathbf{r})$ from physical space $\mathbf{x}$ to the unit square $\mathbf{r} \in [0,1]^d$ in parameter space. Following the notation in [13], in the parameter space coordinates the governing equation (1) can be written in conservative form as

$$L(u) = \frac{1}{J} \sum_{m=1}^{d} \sum_{n=1}^{d} \frac{\partial}{\partial r_m} \left(J A^{mn} \frac{\partial u}{\partial r_n} \right), \tag{2}$$

where

$$A^{mn} = c^2 \sum_{\mu=1}^{d} \frac{\partial r_m}{\partial x_\mu} \frac{\partial r_n}{\partial x_\mu},$$

and J is the determinant of the Jacobian matrix $[\partial x_i / \partial r_j]$. Note that in (2) the conserved quantity is Ju, and the metrics of the mapping enter the equation as variable coefficients. The form (2) is applicable to general curvilinear and non-orthogonal grids; an optimized scheme is used for Cartesian grids.

As discussed in [13], self-adjoint discretizations of (2) can be developed to arbitrary order for the case of a single component grid. These discretizations have a compact stencil and are free from numerical dissipation. However, when overlapping grids are used, the perturbations introduced by the interpolation between component grids can result in numerical instabilities. In [13] these instabilities were treated by adding a simple dissipation operator whose coefficients were chosen experimentally and with expert judgement. In [1], a proof was presented showing the presence of these unstable modes for overlapping grids. The analysis indicated a form of dissipation operator that would stabilize the schemes against overlapping grid interpolation. Centered (dissipation free) discretizations of the first-order system were also shown to exhibit similar instabilities associated with overlapping grid interpolation, but the dissipation inherent to standard upwind discretizations for the first-order system was shown to stabilize the system. The form of dissipation required to stabilize the wave equation in second-order form against overlapping grid interpolation has since been shown to be naturally present in "upwind" discretizations of the second-order system as described in [2]. For this reason, we will develop upwind discretizations of (2).

2.1 Upwind Discretization

Following the approach in [2] we introduce the time derivative of the field quantity (indicated using a dot as in $\dot{u} \equiv \frac{\partial u}{\partial t}$), and rewrite the equations as

$$\frac{\partial}{\partial t} \begin{bmatrix} u \\ \dot{u} \end{bmatrix} = \begin{bmatrix} \dot{u} \\ 0 \end{bmatrix} + \frac{1}{J} \sum_{m=1}^{d} \frac{\partial}{\partial r_m} \begin{bmatrix} 0 \\ \sum_{n=1}^{d} J A^{mn} \frac{\partial u}{\partial r_n} \end{bmatrix}. \tag{3}$$

As in [2] we integrate in time over a time step Δt, and produce the exact differential-difference equations

$$\dot{u}(\mathbf{x},t^{n+1}) = \dot{u}(\mathbf{x},t^n) + \frac{\Delta t}{J}\sum_{m=1}^{d} D_{+r_m}\mathscr{F}^{\dot{u}}_{r_m}(\mathbf{x}-\frac{h_{r_m}}{2}\mathbf{e}_{r_m},t^n), \tag{4}$$

$$u(\mathbf{x},t^{n+1}) = u(\mathbf{x},t^n) + \Delta t\dot{u}(\mathbf{x},t^n) + \frac{\Delta t^2}{J}\sum_{m=1}^{d} D_{+r_m}\mathscr{F}^{u}_{r_m}(\mathbf{x}-\frac{h_{r_m}}{2}\mathbf{e}_{r_m},t^n). \tag{5}$$

Here r_m is the mth direction in index space, $\mathbf{e}_{r_m}$ is the unit vector in the r_m direction, D_{+r_m} is the forward divided difference operator in the r_m direction, h_{r_m} is the grid spacing in the r_m direction, and the integrals of the fluxes are defined as

$$\mathscr{F}^{\dot{u}}_{r_m}(\mathbf{x},t^n) = \frac{1}{\Delta t}\int_0^{\Delta t} \check{f}_{r_m}(\mathbf{x},t^n+\tau)\,d\tau, \tag{6}$$

$$\mathscr{F}^{u}_{r_m}(\mathbf{x},t^n) = \frac{1}{\Delta t^2}\int_0^{\Delta t}\int_0^{\tau} \check{f}_{r_m}(\mathbf{x},t^n+\tau')\,d\tau'\,d\tau, \tag{7}$$

where the upwind flux functions are given by

$$\begin{aligned}\check{f}_{r_m}(\mathbf{x}+\frac{h_{r_m}}{2}\mathbf{e}_{r_m},t^n+\tau) \equiv{}& \sum_{\mu=1}^{d}\mathscr{A}_{r_m}JA^{m\mu}\frac{\partial u}{\partial r_\mu}(\mathbf{x}+\frac{h_{r_m}}{2}\mathbf{e}_{r_m},t^n+\tau) \\ &+ \mathscr{A}_{r_m}\frac{J\sqrt{A^{mm}}}{2}\left(\dot{u}^{r_m^+}(\mathbf{x}+\frac{h_{r_m}}{2}\mathbf{e}_{r_m},t^n+\tau)-\dot{u}^{r_m^-}(\mathbf{x}+\frac{h_{r_m}}{2}\mathbf{e}_{r_m},t^n+\tau)\right).\end{aligned} \tag{8}$$

In (8) we have introduced the operator $\mathscr{A}_{r_m}$ which is defined to satisfy the identity $\frac{\partial w}{\partial r_m}(\mathbf{x}) = D_{+r_m}\left(\mathscr{A}_{r_m}w(\mathbf{x}-\frac{h_{r_m}}{2}\mathbf{e}_{r_m})\right)$ for any sufficiently smooth function w and is given by the expansion $\mathscr{A}_{r_m}w(\mathbf{x},t) = \sum_{j=0}^{\infty}\alpha_j h_{r_m}^{2j}\frac{\partial^{2j}w}{\partial r_m^{2j}}(\mathbf{x},t)$. The coefficients α_j can be computed from the identity $\zeta/2 = \sinh(\zeta/2)\sum_{j=0}^{\infty}\alpha_j\zeta^{2j}$ following the approach described in [10, 11]. Values for the first few coefficients are $\alpha_0 = 1$, $\alpha_1 = -\frac{1}{24}$, $\alpha_2 = \frac{7}{5760}$, $\alpha_3 = \frac{31}{967680}$. As in the description in [2] we use m-point Gaussian quadrature to evaluate the integrals in (4) and (5). Taylor expansions in space and time are used to define the quantities in (8) to the desired order. The final result is a single-step scheme of the desired accuracy. Such a time integration technique is often referred to as a modified-equation, Cauchy–Kovalevskaya, or Lax-Wendroff time-stepper.

The maximal stable time step of the upwind schemes for each component grid can be computed exactly assuming constant coefficients (i.e. rectangular grids). See [2] for details. This bound is applied locally as an estimate for the maximal

stable time step for curvilinear component grids. Such a procedure is similar to the use of a linearized estimate to determine the time step for computations of the Navier–Stokes equations. The time step for the overall simulation is then taken to be the smallest of the time steps computed over all component grids. The exact form of the discrete stability bound in multiple dimensions is found to be quite complex. In addition, the time step assuming constant coefficients is often an overly optimistic estimate for curvilinear grids. Therefore, we fit a simplified bound (which also gives a simple explicit expression for Δt) of the form

$$\sum_{m=1}^{d} \lambda_m^{\sigma} \leq \Lambda_{\max}^{\sigma}$$

where $\lambda_m = \max_{\mathbf{i}}(\frac{|A_{mm}|}{h_{r_m}})\Delta t$ and the maximum is taken over all grid points. The coefficients σ and $\Lambda_{\max}$ are determined for each discretization through a normal mode stability analysis of the linearized constant coefficient problem. Figure 2 gives the numerical values for these parameters as well as a plot of the bounds in two dimensions for discretization orders two, four, and six. Note that larger time-steps can be taken in the higher order schemes compared to the second-order scheme. Finally, we use an additional safety factor of 0.9 and so the final time step is only 90% of the value computed by taking the minimum allowable over all grids.

order	σ	$\Lambda_{\max}$
2	1.35	0.605
4	2.275	1.075
6	1.7	0.93

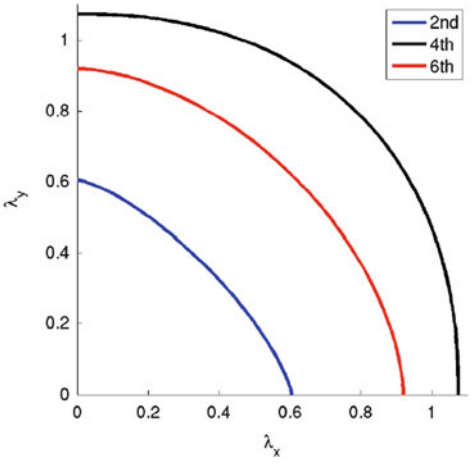


Fig. 2 At *left* are coefficients defining the simplified stability bound for the schemes of various orders. At *right* is a plot showing those stability bounds in two space dimensions. The discretizations are stable for parameters that lie to the lower left of the appropriate curve

3 Numerical Examples

In this section we present some initial results to demonstrate the accuracy of the overall approach as well as the stability of the upwind discertizations on overlapping grids. To this end we present convergence tests using twilight zone solutions, also known as the method of manufactured solutions, in both two- and three-dimensions. In this approach an exact solution u_e is posed, in this case we choose trigonometric functions in space and time, and a source term is applied to the governing equations so that a solution to the forced system is the presupposed exact solution u_e. This modified system reads

$$\frac{\partial^2 u}{\partial t^2} = Lu + \frac{\partial^2 u_e}{\partial t^2} - Lu_e, \quad \mathbf{x} \in \Omega. \tag{9}$$

For this study we take Dirichlet boundary conditions on physical boundaries with the exact solution being specified.

3.1 Twilight Zone in Two Space Dimensions

Here we investigate the discrete solution of the wave equation on a two-dimensional unit disk. The exact solution is chosen as $u_e = \frac{1}{2}\cos(2\pi x)\cos(2\pi y)\cos(2\pi t)$. The overlapping grid, shown in Fig. 3, uses a narrow boundary fitted grid near the edge of the disk and a large background Cartesian grid over the domain interior. Figure 3 also shows the solution at the final time $t = 0.5$. A convergence study

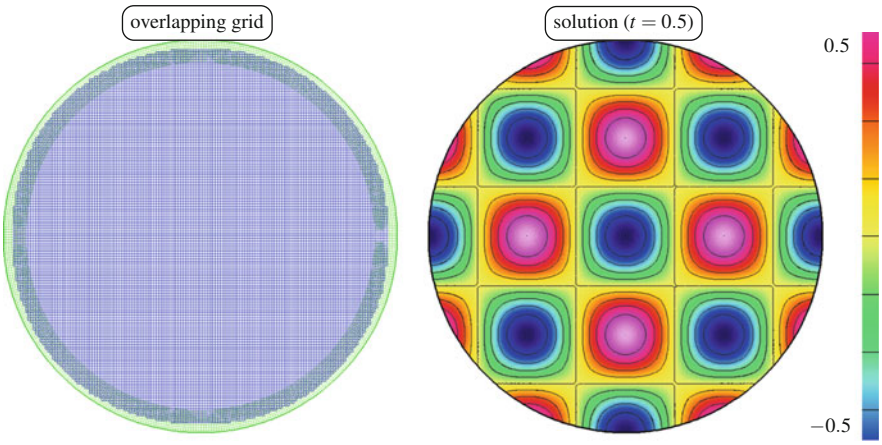


Fig. 3 *Left*: overlapping grid for the disk. *Right*: trigonometric twilight zone solution at $t = 0.5$

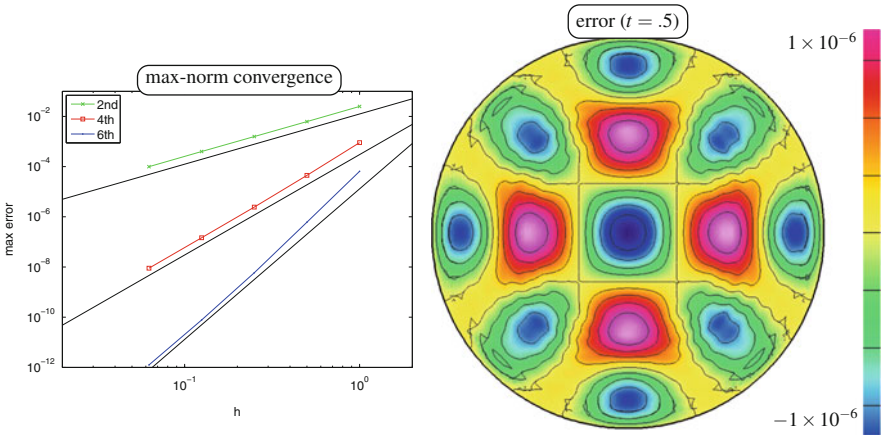


Fig. 4 *Left*: convergence results for the various schemes. *Right*: solution error at $t = 0.5$ for the fourth order scheme on the 3rd refinement grid

is performed on a series of grids of increasing resolution. As discussed in [1], it is more challenging from a stability perspective to refine the boundary fitted grids keeping the number of grid lines normal to the boundary fixed; the boundary grids thus become narrower as the grid is refined. Figure 4 presents results from this study showing max-norm errors at the final time for the second, fourth, and sixth-order methods. Convergence at the designed accuracy is demonstrated and there are no indications of instability. The error field for the fourth-order scheme and the 3rd refinement grid is also shown in Fig. 4. Note that the error magnitude is uniform across the grid overlap. Also note that due to the overlapping grid interpolation and the upwinding, there is no conserved discrete energy. Instead the discrete energy converges to the order of accuracy of the scheme.

3.2 Twilight Zone in Three Space Dimensions

For three dimensions we perform simulations for a domain consisting of the box $(x, y, z) \in [-2, 2] \times [-2, 2] \times [-2, 2]$ with a spherical cavity of radius 0.5 in the center. The exact solution is chosen as $u_e = \cos(2\pi x)\cos(2\pi y)\cos(2\pi z)\cos(2\pi t)$. Figure 5 shows the simulation geometry and the exact solution at the initial time. Also shown are results for a max-norm convergence study for the second-, fourth-, and sixth-order schemes. As in two dimensions, convergence at the designed accuracy is obtained and there is no evidence of instability associated with the overlapping grid interpolation.

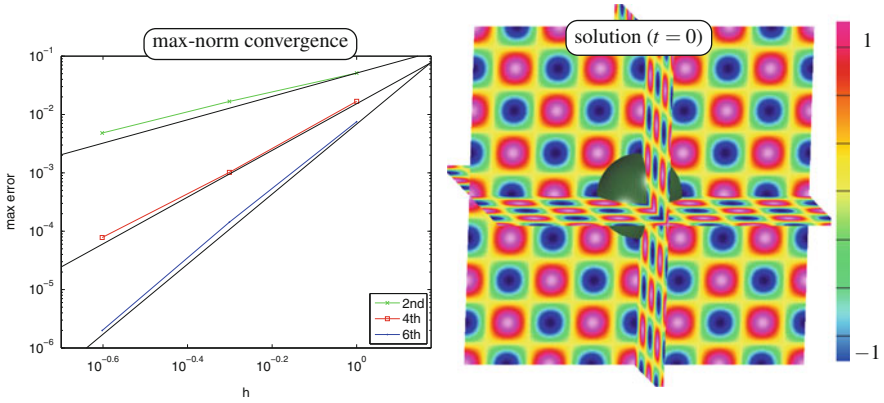


Fig. 5 At *left* are the results of a max-norm convergence test for the various schemes while at *right* is the exact solution for the sphere in a box test

4 Conclusions

In this work we have extended the upwind approach for second-order wave equations developed in [2] to curvilinear and overlapping grids. Upwinding is incorporated through the definition of the numerical flux function by embedding a localized form of d'Alembert's exact solution. A high-order accurate single-step space-time scheme is developed by employing a Cauchy-Kovalevskaya (Lax-Wendroff) procedure. The overall approach is shown to be stable in the presence of overlapping grid interpolation in two and three space dimensions using the method of manufactured solutions. Future work includes incorporation of physical boundary conditions, development of implicit schemes to handle small mesh cells arising from fine geometrical features, and optimization of the schemes.

References

1. D. Appelö, J.W. Banks, W.D. Henshaw, D.W. Schwendeman, Numerical methods for solid mechanics on overlapping grids: linear elasticity. J. Comput. Phys. **231** 6012–6050 (2012)
2. J.W. Banks, W.D. Henshaw, Upwind schemes for the wave equation in second-order form. J. Comput. Phys. **231**(17), 5854–5889 (2012)
3. J.W. Banks, D.W. Schwendeman, A.K. Kapila, W.D. Henshaw, A high-resolution Godunov method for compressible multi-material flow on overlapping grids. J. Comput. Phys. **223**, 262–297 (2007)
4. J.W. Banks, W.D. Henshaw, J.N. Shadid, An evaluation of the FCT method for high-speed flows on structured overlapping grids. J. Comput. Phys. **228**(15), 5349–5369 (2009)
5. J.P. Boris, D.L. Book, Flux-corrected transport. I. SHASTA, a fluid transport algorithm that works. J. Comput. Phys. **11**, 38–69 (1973)

6. G. Chesshire, W. Henshaw, Composite overlapping meshes for the solution of partial differential equations. J. Comput. Phys. **90**, 1–64 (1990)
7. B. Cockburn, C.W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite-element method for conservation-laws 2: general framework. Math. Comput. **52**, 411–435 (1989)
8. P. Colella, P.R. Woodward, The piecewise parabolic method (PPM) for gas-dynamical simulations. J. Comput. Phys. **54**(1), 174–201 (1984)
9. R. Courant, E. Isaacson, M. Rees, On the solution of nonlinear hyperbolic differential equations by finite differences. Commun. Pure. Appl. Math. **5**, 243–255 (1952)
10. B. Fornberg, *A practical Guide to Pseudospectral Methods* (Cambridge University Press, Cambridge, 1996)
11. B. Gustafsson, H.-O. Kreiss, J. Oliger, *Time Dependent Problems and Difference Methods* (Wiley, New York, 1995)
12. A. Harten, B. Engquist, S. Osher, S. Chakravarthy, Uniformly high order accurate essentially non-oscillatory schemes, III. J. Comput. Phys. **71**, 231–303 (1987)
13. W.D. Henshaw, A high-order accurate parallel solver for Maxwell's equations on overlapping grids. SIAM J. Sci. Comput. **28**(5), 1730–1765 (2006)
14. W.D. Henshaw, D.W. Schwendeman, Moving overlapping grids with adaptive mesh refinement for high-speed reactive and non-reactive flow. J. Comput. Phys. **216**(2), 744–779 (2006)
15. G.-S. Jiang, C.-W. Shu, Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**(1), 202–228 (1996)
16. B. van Leer, Towards the ultimate conservative difference scheme, V. A second-order sequel to Godunov's method. J. Comput. Phys. **32**, 101–136 (1979)

Well-Posedness, Stability and Conservation for a Discontinuous Interface Problem: An Initial Investigation

Cristina La Cognata and Jan Nordström

Abstract A robust interface treatment for the discontinuous coefficient advection equation satisfying time-independent jump conditions is presented. The aim of the investigation is to show how the different concepts like well-posedness, conservation and stability are related. The equations are discretized using high order finite difference methods on Summation By Parts (SBP) form. The interface conditions are weakly imposed using the Simultaneous Approximation Term (SAT) procedure. Spectral analysis and numerical simulations corroborate the theoretical findings.

1 Introduction

In this paper we study fundamental properties such as well-posedness, stability and conservation for an advection equation, which changes wave-speed at the interface separating two spatial domains. The solution satisfies a time-independent jump-condition, which makes it discontinuous. The first goal is to show that for any piecewise constant advection velocity and interface jump condition the continuous problem is always well-posed. We provide a straightforward condition for checking conservation despite the presence of discontinuities. Applications where this is of interest include acoustic electromagnetism, seismology and fluid dynamics, [6, 9, 11, 12].

Stability and conservation at interfaces have also been studied in [2–4] for the case of identical velocities in the two domains. We extend this investigation by showing how well-posedness, conservation and stability are related in a more general setting. SBP-SAT schemes, [1, 13, 14], up to fifth order of accuracy are used to exemplify that the interface treatment is stable and accurate for all theoretically meaningful cases.

C. La Cognata (✉) • J. Nordström
Department of Mathematics, Computational Mathematics, Linköping University, SE-581 83 Linköping, Sweden
e-mail: cristina.la.cognata@liu.se; jan.nordstrom@liu.se

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_11

2 The Discontinuous Interface Problem

Consider two advection equations with different real positive advection velocities, a and b

$$\begin{aligned} &u_t + au_x = 0,\ x \le 0, t \ge 0, \\ &v_t + bv_x = 0,\ x \ge 0, t \ge 0, \end{aligned} \quad \text{with} \quad v(0,t) = cu(0,t), \tag{1}$$

where c is a real constant which makes the solution discontinuous at the interface point $x = 0$ when it is different from one.

2.1 Well-Posedness and Conservation

Proposition 1 *The interface problem defined by the coupled equations (1) is well-posed for any positive a, b and any constant $c \in \mathbb{R}$.*

Proof By applying the energy method to (1) using a modified L^2-norm we get

$$\int_{-\infty}^{0} u\,[u_t + au_x]\,dx + \int_{\infty}^{1} \alpha_c v\,[v_t + bv_x]\,dx = 0$$

where α_c is a positive free weight parameter. By ignoring the outer boundary terms, integration by parts leads to an energy estimate if α_c verifies

$$-a + \alpha_c b c^2 \le 0. \tag{2}$$

Uniqueness of the solution can be proved by using the same technique. Existence can be proved by using the Laplace transform technique for the initial boundary value problem, see [5, 7] for details. □

Proposition 2 *The interface problem (1) is conservative if*

$$c = \frac{a}{b}. \tag{3}$$

Proof The weak formulation of (1) is given by

$$\int_{-\infty}^{\infty} [\phi\ w]_0^t\ dx - \int_{-\infty}^{\infty}\int_0^t [\phi_t + \phi_x\ \bar{u}]\,w\ dxdt + \int_0^t \phi w\,[a - bc]_{x=0}\ dt = 0, \tag{4}$$

where $\phi(x,t) \in C^\infty$ with compact support in the spatial interval $[-\infty, \infty]$ while $\bar{u} = a$ for $x \le 0$ and $\bar{u} = b$ for $x > 0$. Thus, all the terms at the interface vanish, resulting in a conservative problem if $c = a/b$. □

3 The Semi-discrete Approximation

The first derivative in space is approximated using summation-by-parts (SBP) finite difference operators $u_x \approx D\mathbf{u} = P^{-1}Q\mathbf{u}$, introduced in [13, 14]. $\mathbf{u}$ is the discrete grid function approximating the solution. From now on we indicate the difference operator with $P_{l,r}^{-1}Q_{l,r}$, which are related to the left and right spatial intervals respectively. By ignoring the boundaries we can write the approximation of (1) together with the SAT procedure [3, 4], for interface conditions as

$$\begin{aligned} \mathbf{u}_t + aP_l^{-1}Q_l\mathbf{u} &= P_l^{-1}\sigma_L(cu_N - v_0)e_N \\ \mathbf{v}_t + bP_r^{-1}Q_r\mathbf{v} &= P_r^{-1}\sigma_R(v_0 - cu_N)e_0, \end{aligned} \tag{5}$$

where $e_N = (0, \ldots, 0, 1)$ and $e_0 = (1, 0 \ldots, 0)$ have length of the left and right domain grid, respectively. Equation (5) can be written in a compact matrix form as follows

$$\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix}_t = P^{-1}\tilde{Q}\begin{pmatrix} \mathbf{u} \\ \mathbf{v} \end{pmatrix} \tag{6}$$

where

$$P = \begin{bmatrix} P_l & 0 \\ 0 & P_r \end{bmatrix}, \tilde{Q} = -Q_\Lambda + \Sigma, \text{ and } Q_\Lambda = \begin{bmatrix} aQ_l & 0 \\ 0 & bQ_r \end{bmatrix}.$$

The penalty matrix Σ which contains the penalties coefficients is zero everywhere except at the boundary and interface points.

3.1 *Stability and Conservation Properties of the Semi-discrete Approximation*

Similar to the continuous case we apply the semi-discrete energy method to (5) to derive stability conditions. To do so we multiply (5) with $u^T P_l, v^T P_r$ from the left and we obtain

$$\frac{d}{dt}\left[\|\mathbf{u}\|_{P_l}^2 + \alpha_d\|\mathbf{v}\|_{P_r}^2\right] = \text{IT} \tag{7}$$

where α_d is a positive weight (not necessarily the same as in the continuous case). IT is a quadratic form given by

$$\text{IT} = \begin{pmatrix} u_N \\ v_0 \end{pmatrix}^T H \begin{pmatrix} u_N \\ v_0 \end{pmatrix}, \quad H = \begin{bmatrix} (-a + 2c\sigma_L) & -(\sigma_L + \alpha_d c\sigma_R) \\ -(\sigma_L + \alpha_d c\sigma_R) & \alpha_d(b + 2\sigma_R) \end{bmatrix}. \tag{8}$$

We have an energy estimate if IT ≤ 0, which require H to be a negative semi-definite matrix. Hence, we need a condition on σ_L and σ_R to ensure this. In the full paper [8] we prove

Proposition 3 *The semi-discrete schemes (5) for the coupled advection equations (1) has a stable interface treatment when*

$$\begin{aligned}(-a+2c\sigma_L)+\alpha_d(b+2\sigma_R)\leq 0,\\ (-a+2c\sigma_L)\alpha_d(b+2\sigma_R)-(\sigma_L+\alpha_d c\sigma_R)^2\geq 0.\end{aligned} \tag{9}$$

One can also prove, see [8], that

Proposition 4 *The conditions in (9) imply that $P^{-1}\tilde{Q}$ has eigenvalues with negative semi-definite real parts.*

As in the continuous case we rewrite (5) in a weak formulation to derive the conservation condition. We obtain the following discrete conservation criteria

Proposition 5 *The semi-discretization (5) with the continuous conservation condition (3) is a conservative approximation if*

$$\sigma_R=\sigma_L-b. \tag{10}$$

The conservative approximation requires a conservative continuous problem.

4 The Relation Between Stability and Conservation

In this section we present explicit stability condition for the penalty coefficients $\sigma_{L,R}$ for different type of continuous problems and approximations. All the conditions are algebraically derived from (9).

Proposition 6 *Consider the most general well-posed interface problem without assuming any conservation conditions. The semi-discrete approximation (5) is stable for all parameters a, b, c when the penalty coefficients σ_L, σ_R satisfy*

$$\sigma_R\leq\frac{-b}{2}. \tag{11}$$

$$\frac{b+\sigma_R-\sqrt{(b+2\sigma_R)(b-\theta\left(\frac{a}{c}\right))}}{\theta}\leq\sigma_L\leq\frac{b+\sigma_R+\sqrt{(b+2\sigma_R)\left(b-\theta\left(\frac{a}{c}\right)\right)}}{\theta}, \tag{12}$$

where $\theta=1/(\alpha_d c)\geq bc/a$ must hold for real penalty coefficients.

Proposition 7 *The continuous conservation condition (3) leads to a stable semi-discrete approximation if the penalty parameters σ_L, σ_R satisfy (11) and*

$$\frac{b+\sigma_R-\sqrt{b(b+2\sigma_R)(1-\theta)}}{\theta} \leq \sigma_L \leq \frac{b+\sigma_R+\sqrt{b(b+2\sigma_R)(1-\theta)}}{\theta} \tag{13}$$

with $\theta = b/(a\alpha_d) \geq 1$.

Note that conservation and stability are two independent properties of the approximation (5). We have a stable and non-conservative semi-discretization if the assumptions of Proposition 6 are satisfied. Note also that for one norm, the stability requirements in Proposition 6 also lead to conservation. That norm is given by $\alpha_d = b/a$.

Proposition 8 *The conditions (3) and (10) lead to a stable and conservative scheme if*

$$\frac{b}{1-\sqrt{\theta}} \leq \sigma_L \leq \frac{b}{1+\sqrt{\theta}} \quad \textit{with} \quad \theta = b/(a\alpha_d) \geq 1. \tag{14}$$

5 Numerical Results

In order to show the effect of the interface treatment we must restrict ourselves to a finite spatial domain, we choose $[-1, 1]$.

In Fig. 1 we show a few of the frames of the time-evolution of a conservative solution of (1), namely $\sin(4\pi(-1+3t))$. The wave propagates with velocity a = 2

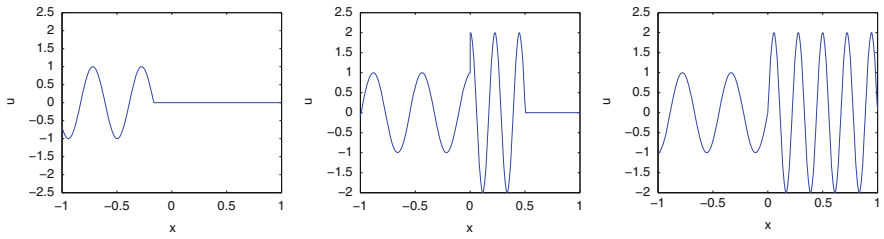


Fig. 1 Time-evolution of the wave function $\sin(4\pi(-1+3t))$ satisfying a conservative jump condition. The solution is computed with a conservation approximation (Proposition 5). Zero initial data in both domains and boundary condition given by $\sin(4\pi(-1+3t))$. The parameters are: a = 2, b = 1 and c = 2

Table 1 Convergence rate as a function of grid N points for the non-conservative interface problem (1) and semi-discretization (5)

L^2	SBP21		SBP42		SBP63		SBP84	
N	u_l	u_r	u_l	u_r	u_l	u_r	u_l	u_r
80	2.0124	2.0267	3.0397	3.0096	3.6801	3.8770	4.5847	5.0897
160	2.0086	2.0102	3.0713	3.0083	3.8480	4.0149	4.7510	5.0624
320	2.0059	2.0044	3.0359	3.0068	3.9590	4.0052	4.9033	5.0176

u_l and u_r are the computed solutions in the left and the right domain respectively. Parameters setting: a = 3, b = 2, c = 3. Interface penalties $\sigma_{L,R}$ satisfying the stability conditions of Proposition 6

in the left domain, b = 1 in right domain and jump condition c = 2. The initial data is zero in both domains. The computations are done using RK4 in time and SBP84 in space, with CFL = 0.1 and 300 grid points in each domain. The penalty $\sigma_{L,R}$ satisfy the conservative assumptions of Proposition 8.

5.1 Order of Accuracy

Next we establish the order of accuracy of our scheme by using the method of manufactured solutions with periodic boundary conditions. In Table 1 we present the accuracy of SBP21, SPB42, SBP63 and SBP84 operators in the L^2 norm for a non-conservative problem and approximation (stability conditions from Proposition 6). Table 1 shows that the solutions computed with the considered SBP operators converge with the correct second, third, fourth and fifth order, respectively. We obtain analogous results for a conservative problem with both conservative and non-conservative approximation.

5.2 The Spectrum

Given that our numerical scheme is accurate, we are now interested in showing that the interface treatment produces a negative semi-definite spectrum, which converges to the continuous spectrum. The semi-discrete spectrum is given by the eigenvalues of $P^{-1}\tilde{Q}$ defined in (6). The continuous spectrum of (1) is derived by using the Laplace Transform technique [5, 7] and is given by the infinite sequence

$$s = \frac{ab}{a+b}\left[log(|cd|) + 2i\pi k\right], \quad k \in \mathbf{Z} \quad \text{with} \quad cd \neq 0. \tag{15}$$

The real constant d defines the boundary closure through $u(-1,t) - dv(1,t) = 0$. In Fig. 2 we plot the semi-discrete spectrum vs the continuous one for a conservative

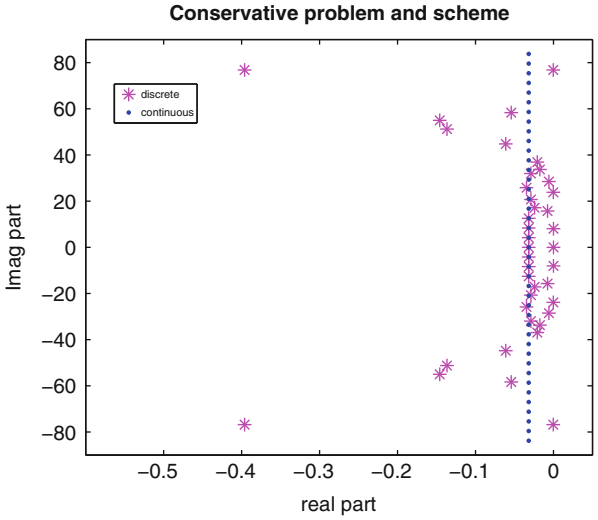


Fig. 2 Continuous and semi-discrete spectrum of fourth order SBP-SAT approximation. Penalty coefficients σ_L, σ_R as in Proposition 7. Parameter setting: $a = 2$, $b = 1$, and $c = 2$

Table 2 Convergence rate of the semi-discrete spectra as a function of N grid points for the non-conservative interface problem (1) and semi-discretization (5)

N	SBP21	SBP42	SBP63	SBP84
40	2.4430	5.2086	6.1259	10.1153
80	2.0485	4.2217	6.9556	8.9885
160	2.0197	4.0813	5.9620	8.8797

Parameters setting: a = 3, b = 2, c = 3. Interface penalties $\sigma_{L,R}$ satisfying the stability conditions of Proposition 5

problem and approximation. We can see that the spectra have eigenvalues with negative real parts, which implies well-posedness and a stable semi-discretization as stated in Proposition 4. We get similar plots for the other problems and schemes.

Table 2 show the order of convergence for the semi-discrete spectra to the continuous spectra for SBP21, SPB42, SBP63 and SBP84 operators. The convergence rate is computed by measuring the distance between the eigenvalues from the semi-discrete spectrum and the eigenvalues from the continuous spectrum. Note that Table 2 show that the convergence is the same as the order of the internal approximation.

Figure 2 also show that a few discrete eigenvalues are located to the right of the continuous spectrum. According to the definition of strict stability, [5, 7], the time growth rate of a strictly stable approximation is bounded by the growth rate of the corresponding continuous problem. Hence, we prefer that the eigenvalues of the semi-discrete spectrum lies on the left side of the continuous spectrum. By adding suitable artificial dissipation to the semi-discretization (5), we can move the discrete spectrum to the left side of the continuous one without reducing accuracy. Figure 3

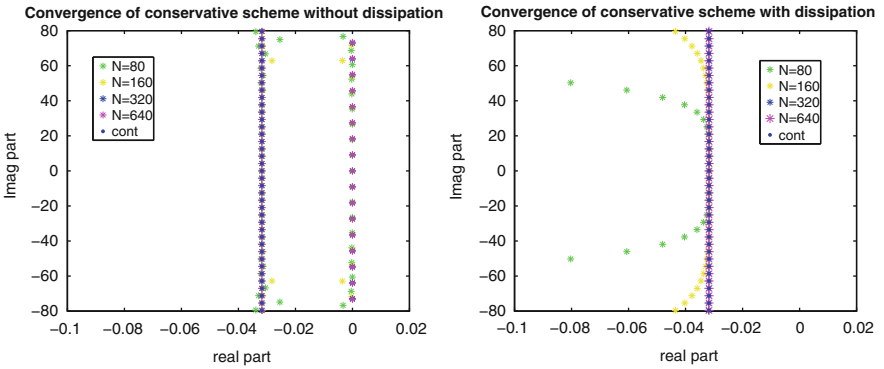


Fig. 3 Close-up of the spectra of the conservative problem plotted in Fig. 2 (*left* figure). The *right* figure shows the same problem approximated by using artificial dissipation

show a close-up of the spectrum of the conservative approximation of the same case with and without artificial dissipation. The semi-discrete eigenvalues on the right in Fig. 3 converge from the left side implying strict stability. For a discussion on how to build artificial dissipation operators for SBP operators without losing accuracy and stability, see [10].

6 Conclusions

We have presented a complete analysis of the discontinuous coefficient interface problem. We have shown that a such problem is always well-posed and we have investigated when it is conservative. We have derived a stable SBP-SAT scheme which can be made conservative or non-conservative depending on our choice. We have also shown that the approximation can be made strictly stable by adding artificial dissipation without reducing the accuracy.

References

1. M.H. Carpenter, D. Gottlieb, S. Abarbanel, Time-stable boundary conditions for finite difference schemes solving hyperbolic systems: methodology and applications to high-order compact schemes. J. Comput. Phys. **129**, 220–236 (1994)
2. M.H. Carpenter, J. Nordström, D. Gottlieb, A stable and conservative interface treatment of arbitrary spatial accuracy. J. Comput. Phys. **148**(2), 341–365 (1999)
3. M.H. Carpenter, J. Nordström, D. Gottlieb, Revisiting and extending interface penalties for multi-domain summation-by-parts operators. J. Sci. Comput. **45**, 118–150 (2010)
4. J. Gong, J. Nordström, Interface procedures for finite difference approximations of the advection-diffusion equation. J. Comput. Appl. Math. **236**(5), 602–620 (2011)

5. B. Gustafsson, H.O. Kreiss, A. Sundström, Stability theory of difference approximations for mixed initial boundary value problems, II. Math. Comput. **26**(119), 649–686 (1972)
6. J.E. Kozdon, E.M. Dunham, J. Nordström, Simulation of dynamic earthquake ruptures in complex geometries using high-order finite difference methods. J. Sci. Comput. **55**(1), 92–124 (2013)
7. H.O. Kreiss, Stability theory of difference approximations for mixed initial boundary value problems, I. Math. Comput. **22**(104), 703–714 (1968)
8. C. La Cognata, J. Nordström, *Well-Posedness, Stability and Conservation for a Discontinuous Interface Problem*. LiTH-MAT-R–2014/16–SE, Department of Mathematics, Linköping University, 2014
9. K. Mattsson, J. Nordström, High order finite difference methods for wave propagation in discontinuous media. J. Comput. Phys. **200**, 249–269 (2006)
10. K. Mattsson, M. Svärd, J. Nordström, Stable and accurate artificial dissipation. J. Sci. Comput. **21**(1), 57–79 (2004)
11. J. Nordström, R. Gustafsson, High order finite difference approximations of electromagnetic wave propagation close to material discontinuities. J. Sci. Comput. **18**(2), 214–234 (2003)
12. J. Nordström, J. Gong, E. Van der Weide, M. Svärd, A stable and conservative high order multi-block method for the compressible Navier-Stokes equations. J. Comput. Phys. **228**(24), 9020–9035 (2009)
13. B. Strand, Summation by parts for finite difference approximation for d/dx. J. Comput. Phys. **110**(1), 47–67 (1994)
14. M. Svärd, J. Nordström, Review of summation-by-parts schemes for initial-boundary-value problems. J. Comput. Phys. **268**, 17–38 (2014)

An Adaptive Fourier Filter for Relaxing Time Stepping Constraints for Explicit Solvers

Dennis Denker, Rick Archibald, and Anne Gelb

Abstract Filtering is necessary to stabilize piecewise smooth solutions. The resulting diffusion stabilizes the method, but may fail to resolve the solution near discontinuities. Moreover, high order filtering still requires cost prohibitive time stepping. This paper introduces an adaptive filter that controls spurious modes of the solution, but is not unnecessarily diffusive. Consequently we are able to stabilize the solution with larger time steps, but also take advantage of the accuracy of a high order filter.

1 Introduction

Filters are often used to stabilize piecewise smooth solutions. In order to maintain spectral accuracy away from discontinuities, such filters must decay with high order smoothness, [2]. Unfortunately, high order filters require small time steps to maintain stability in partially filtered modes; and achieving high order smoothness results in diffusion in some innately stable modes. Apart from using filters to improve accuracy of under-resolved solutions, the resolution of the solution space can be increased; but this comes at the cost of even smaller step sizes and greater computational effort per step.

As a way of better balancing accuracy and computational cost, we introduce an adaptive filter, which maintains stability without diffusion when the numerical solution is well resolved, but acts as a high order filter when spectral support is large. The modification to a standard filter is simple to implement and has negligible computational cost. The numerical tests show this filter can achieve a

D. Denker (✉) • A. Gelb
School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ 85287-1804, USA
e-mail: dennydenker@cox.net

R. Archibald
Computer Science and Mathematics Division, Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA
e-mail: archibaldrk@ornl.gov

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_12

lasting increase in solution accuracy even after the time when solutions become permanently under-resolved and traditional filtering is required.

This paper is organized as follows: In Sect. 2 the necessary background in filter construction and the sources of instability are reviewed. In Sect. 3 a simple chop filter is introduced, which is further refined into an adaptive filter. In Sect. 4 we present the results of a variety of numerical tests using this filter.

2 Background

It is well known that the pseudo-spectrum of the spatial discretization must sit within the stability region of the time integration scheme, [5]. Violating this condition leads to exponential growth of modes with eigenvalues outside the region. Even when solution spectral support is limited to stable modes, numerical noise can perturb modes with growth factors larger than one, which then grow exponentially, [3].

A direct solution to this problem is to reduce time step size, which also increases accuracy, but may be prohibitively expensive computationally. For piecewise smooth solutions, filtering promotes stability and can control modes with large growth factors, [1]. The requirements for high quality filters that promote spectral accuracy has been well studied, [2].

Definition 1 Let the ratio of a given Fourier frequency to the highest allowable frequency in the solution space be given by: $\eta = \frac{|j|}{N}, j = -N \ldots N$. An even function, $\sigma(\eta) \geq 0$ is a filter of order $q \geq 2$ provided that:

- $\sigma(\eta) \in C^{q-1}[-\infty, \infty]$
- $\sigma(0) = 1$ and $\sigma(\eta) = 0, \eta \geq 1$
- $\sigma^{(m)}(0) = \sigma^{(m)}(1) = 0, \forall m \in [1, \ldots, q-1]$

A commonly used filter is the exponential filter, given by:

$$\sigma(\eta) = \begin{cases} 1, & \eta \leq \eta_c \\ e^{-\alpha\left(\frac{\eta-\eta_c}{1-\eta_c}\right)^p}, & \eta > \eta_c \end{cases} \tag{1}$$

Typically α is chosen such that $\sigma(1) = \mathcal{O}(\varepsilon_{machine})$. The filter acts on modes starting at η_c and when $\eta_c = 0$, we have:

$$\sigma(\eta) = e^{-\alpha\eta^p} \tag{2}$$

Reconstruction quality in smooth regions can be improved by increasing the power of the exponential filter, p. Stability is better for smaller p, but filter induced diffusion extends into low frequency modes with smaller values of p and η_c, as illustrated in Fig. 1. Thus we see that balancing spectral accuracy with performance leads to filters that introduce some level of diffusion in otherwise stable modes.

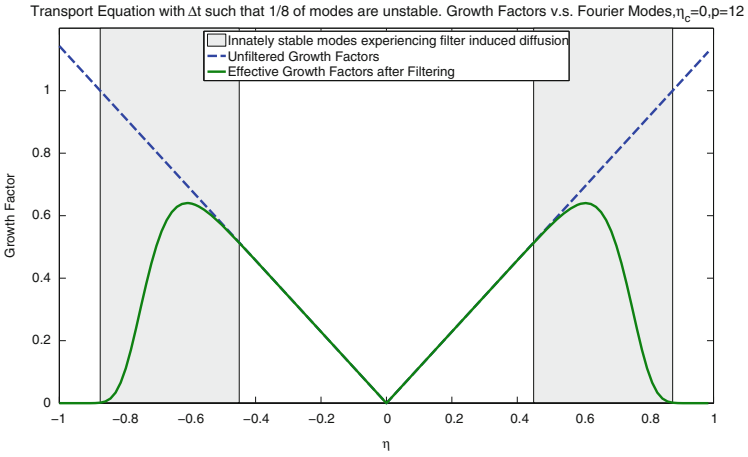


Fig. 1 Illustration of the diffusive effects from a well formed filter that extend into stable modes

3 The Proposed Filter

When a solution has small spectral support it is possible that the exact solution is zero in the unstable modes. Until the solution expands into unstable regions the only trigger for instabilities is numerical noise. In particular, one often uses the FFT in solving nonlinear problems and round-off errors in the FFT are sufficient to perturb unstable modes that then grow exponentially, [7].

Consider the following filter:

$$chop\left(\hat{u}_j\right) = \begin{cases} \hat{u}_j, & \left|\hat{u}_j\right| > \lambda \\ 0, & \left|\hat{u}_j\right| \leq \lambda \end{cases}, \quad \lambda \approx 5N\varepsilon_{Machine} \tag{3}$$

where $\hat{u}_j$ is a Fourier coefficient for an individual mode of the solution.

Equation (3) represents a modification of shrink operator used in l_1 regularization problems and the proposal for solving PDEs by maintaining solution sparsity using this operator, [4, 6]. It was observed that a noise reducing side-effect of the shrink operator could be applied to a different class of PDEs. For well scaled and well posed problems, $\left|\hat{u}_j\right|$ in modes associated with the actual solution will exceed the threshold and will be unaffected leaving only noise driven modes to be corrected to zero. Non-linear PDE terms can introduce data in modes that are indistinguishable from noise, but in practice for well scaled problems these effects are fleeting and exist at a level many orders of magnitude smaller than the accuracy of the numerical scheme.

Algorithm 1: Adaptive Filter

1: **procedure** ADAPTIVE_FILTER $\hat{u}$,λ,τ
2: **for all** $\hat{u}_j$ **do**
3: **if** $\left|\hat{u}_j\right| < \lambda$ **then**
4: $\hat{u}_j \leftarrow 0$
5: $support \leftarrow \max\left(\{|j| : \left|\hat{u}_j\right| > \lambda\}\right)$
6: **if** $support \geq \tau$ **then**
7: **for all** $\hat{u}_j$ **do**
8: $\hat{u}_j \leftarrow \hat{u}_j e^{-\alpha\eta^p}$ ▷ α, η, p as defined in (2)

The chop filter thus allows time steps that may exceed the CFL condition without introducing diffusion; and the solution remains stable as long as its instantaneous support sits in the stability region of the numerical scheme. Nevertheless, problems of interest will have support that spends some time in unstable regions. During these periods (3) is not contractive and is completely ineffective at controlling instabilities. In order to stabilize the solution once the true spectrum expands into the region of instability a standard filter can be used.

A new threshold parameter, τ, can be compared against the size of the spectral support of the solution to determine whether the filter should merely chop noise or chop noise as well as apply an exponential filter. The resulting hybrid filter maintains stability while minimizing diffusive effects when spectral support is small. The optimal choice for τ coincides with the highest stable mode for a given time step. Larger choices lead to instability, while smaller choices weaken the accuracy gained with the adaptive filter. A good choice for τ, with minimal accuracy impact, can be found by a bisection-like process to estimate the last stable mode and setting τ to a point one or two modes fewer. The choice for the power of the exponential filter, p, is the highest possible value that stabilizes the scheme for a given time step.

Algorithm 1 can be optimized substantially. Determination of the size of the spectrum on line 5 can be found as a side effect of the chop filter on lines 2–4. Other than making this determination, the chop filter affects each mode independently and can be made parallel. In addition, the chop operator, (3), requires the determination of the magnitude of a complex number. Numerical tests show that (4) is a less expensive alternative to (3) and has no measurable impact on the calculated solution.

$$\text{chop}^*\left(\hat{u}_j\right) = \begin{cases} \hat{u}_j, & \left|\text{Re}\left(\hat{u}_j\right)\right| + \left|\text{Im}\left(\hat{u}_j\right)\right| > \lambda^* \\ 0, & \textit{otherwise} \end{cases} \tag{4}$$

The primary requirement for the proposed filter to provide some accuracy gain at larger step sizes is that the support for the spectrum of the initial condition solution be smaller than that established by τ. Assuming τ is chosen as described above, the

level of improvement achieved by the proposed filter is determined by the growth rate of the spectral support, whether the spectrum contracts periodically, and the necessary strength of the exponential filter, p.

4 Numerical Results

In the results that follow we will make use of the following definitions:

Definition 2 Time step acceleration factor, ω: The ratio between the smallest stable time step size for an unfiltered solution and the smallest stable time step size for the solution using the adaptive filter.

Definition 3 Enduring relative accuracy, Ψ: The ratio of the accuracy in the solution using only (2) vs. Algorithm 1 at some time after the solution has become under-resolved and the accuracy of each method is well established.

To demonstrate our new algorithm, we consider the following example:

Example 1

$$u_t + c\,(x)\,u_x = 0 \tag{5a}$$

$$c\,(x) = \frac{1}{2}\sin^2\,(\beta x) + \frac{1}{\gamma}; \quad u\,(x,0) = \cos\,(x) \tag{5b}$$

We used a fourth order Runge-Kutta scheme for time stepping and a spatial discretization with N Fourier modes.

The form of (5) was chosen to help illustrate the effects of the adaptive filter, Algorithm 1. With properly chosen constants and initial conditions, the solution starts out stable and well resolved, the spectrum then grows into the region of instability and beyond to the region where aliasing can occur.

Figure 2 shows a comparison of the accuracy using the adaptive filter, Algorithm 1, vs. the exponential filter, (2), alone using the equation and filter parameters in Table 1. In the region $0 \leq t < 2.26$, the solution is stable and the spectrum sits in a region where the value of the exponential filter is essentially one. The two filters perform identically. In the region $2.26 \leq t < 2.98$, the spectrum has grown to the point where the solution is still stable and the adaptive filter is only chopping, but the exponential filter has become diffusive and the accuracy of the solution suffers. In the region $2.98 \leq t < 4.76$, the adaptive threshold is periodically exceeded and the adaptive filter must apply diffusion during some time steps. In this region, the exponential filter is continually diffusive and accuracy decreases more rapidly than with the adaptive filter. Finally in the region $4.76 \leq t$ the spectrum sits in the region of instability and aliasing. The adaptive filter frequently acts like the exponential filter, but the early accuracy gains persist.

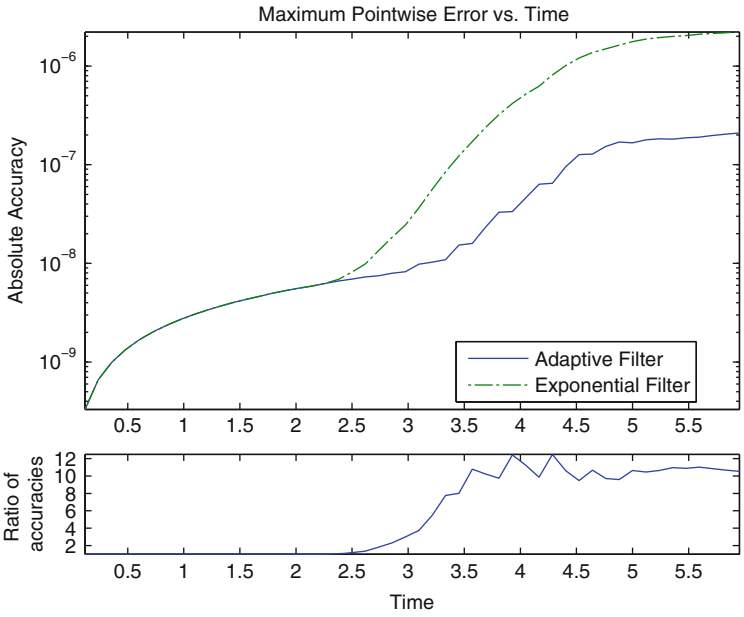


Fig. 2 Illustration of the accuracy gains from the adaptive filter as compared to the exponential filter

Table 1 Equation and numerical parameters used to produce the results in Figure 2

Equation parameters			Filter parameters			Results	
β	γ	N modes	p power	λ, chop threshold	τ, adaptive threshold	ω accel.	Ψ accuracy
2	20	1000	12	5×10^{-13}	0.96	1.06	10.54

Figure 3 shows the source of this enduring accuracy gain for the results Fig. 2. With the strong diffusion of the exponential filter, the spectral support never grows beyond 75 % of the available modes in the numerical solution space. When the adaptive filter operates as a chop filter, the spectral support grows until it reaches the adaptive threshold value, $\tau = 0.96$. At this point the adaptive filter acts as the exponential filter and further support growth is limited. The effective resolution of the method is increased vs. the exponential filter alone.

Using the parameters in Table 2, the problem is solved at a lower resolution and lower filter power. The results are shown in Fig. 4. Even though the time step improvement is better, the lower resolution and smaller adaptive threshold, τ, lead to marginal accuracy gains. The solution spends very little time in the region where the adaptive filter chops and the exponential filter is diffusive.

The parameters and results in Table 3 compare the behavior of the filter at various spatial resolutions. Such a comparison is difficult. We require that the support of the true solution spectrum exceeds the capacity of the numerical solution space even for high resolutions; but at low resolutions the same spectrum causes truncation error

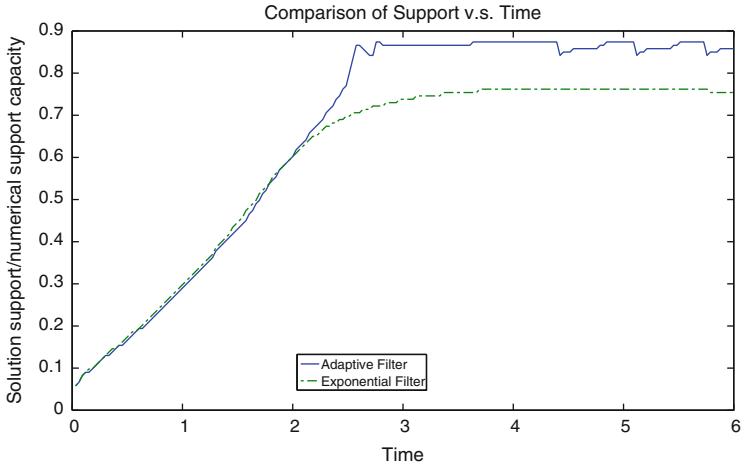


Fig. 3 Illustration of the larger effective numerical solution support possible with the adaptive filter

Table 2 Equation and numerical parameters used to produce the results in Figure 4

Equation parameters			Filter parameters			Results	
β	γ	N modes	p power	λ, chop threshold	τ, adaptive threshold	ω accel.	Ψ accuracy
1	18	400	8	5×10^{-13}	0.85	1.68	1.21

to dominate. To achieve a balance we use the same equation for all resolutions, but compare the results at the moment when the support of the true solution is 1.75 times the maximum support resolved by the numerical solution space. Additionally, time step sizes that are unstable at high resolutions become stable at low resolutions eliminating the need for filtering. So, to produce a reasonable comparison the time step size needs to be dependent on spatial resolution. We use:

$$\Delta t = \alpha \Delta t_{exponential} + (1 - \alpha) \Delta t_{stable} \tag{6}$$

where $\Delta t_{exponential}$ is the smallest stable time step using the exponential filter and Δt_{stable} is the smallest stable time step with no filtering. We set α to 0.9 which achieves nearly the time step gains of the exponential filter (2), but provides room for accuracy gains from the adaptive filter Algorithm 1. Having a different time step associated with each resolution inevitably creates a resolution dependence in the error resulting from the accuracy of the time stepping scheme. For the following tests we use:

$$c(x) = \frac{1}{2}\left(\sin^2(x) + \frac{3}{2}\sin^2(2x)\right) + \frac{1}{20} \tag{7}$$

with the chop threshold, λ: 5×10^{-13} and the adaptive filter threshold, τ: 0.98.

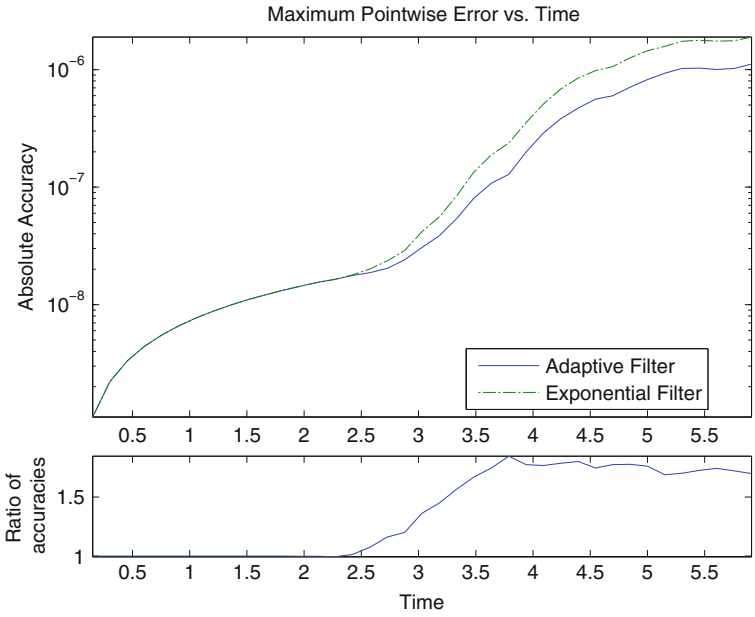


Fig. 4 Illustration of the effect of lower resolution on adaptive filter accuracy gains

Table 3 A comparison of adaptive filter accuracy gains using different spatial resolutions

N modes	Δt time step	p power	ω *acceleration*	Ψ accuracy	$\left\|u - u_{adaptive}\right\|_\infty$ error
256	0.0317	12	1.43	2.52	2.56e−5
512	0.0154	12	1.45	4.53	3.657e−5
768	0.0104	12	1.47	4.70	6.39e−5
1024	0.0077	12	1.47	5.35	9.54e−5

We see a modest improvement in accuracy between the adaptive filter and the exponential filter. The ratio gets better as the resolution grows, because the solution spends more time in the region where the adaptive filter can operate without the exponential filter. Contrary to what is normally expected, the absolute accuracy of the solution does not improve with better resolution. This is not a failure of the technique, but instead a consequence of the tests being designed to have a consistent portion of true solution support in the numerical solution space when measurements are made. The same test is performed in Table 4, but with lower filter power. With the lower power, larger time steps are possible, but there is also more diffusion; and the adaptive filter's accuracy improves compared to the exponential filter.

Similar numerical tests were also performed on Burger's equation and the KDV equation, showing modest accuracy improvements as well.

Table 4 A comparison of adaptive filter accuracy gains using different spatial resolutions, but with a lower exponential filter power

N modes	Δt time step	p power	ω *acceleration*	Ψ accuracy	$\|u - u_{adaptive}\|_\infty$ error
256	0.0359	8	1.65	4.56	0.0003
512	0.0179	8	1.71	7.08	0.0005
768	0.0120	8	1.71	8.36	0.0008
1024	0.009	8	1.73	9.40	0.0011

5 Conclusion

The accuracy gains achieved with the adaptive filter are highly dependent on the PDE being solved and the particular parameters that are chosen as well as the degree to which the CFL condition is exceeded. Certain configurations result in very small accuracy gains. Nonetheless, in all numerical tests that were performed the adaptive filter with stability maintaining parameters outperformed the exponential filter alone. The computational cost and development time of the adaptive filter are negligible, making it a simple addition to standard filtering techniques. With the current algorithm, the choice of the adaptive threshold parameter, τ, is left to the implementer. Maximum accuracy gains occur when this parameter is just below the first unstable mode, future versions of this algorithm could determine the proper value for the parameter by analyzing the growth factors for the equation in question dynamically.

Acknowledgements The works of Dennis Denker and Anne Gelb are supported in part by grants NSF-DMS 1216559 and AFOSR FA9550-12-1-0393.

The submitted manuscript is based upon work of Rick Archibald, authored in part by contractors [UT-Battelle LLC, manager of Oak Ridge National Laboratory (ORNL)], and supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research, Applied Mathematics program under Contract No. DE-AC05-00OR22725.

References

1. A. Gelb, E. Tadmor, Enhanced spectral viscosity approximations for conservation laws. Appl. Numer. Math. **33**, 3–21 (2000)
2. J.S. Hesthaven, S. Gottlieb, D. Gottlieb, *Spectral Methods for Time-Dependent Problems* (Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, Sao Paulo, 2007)
3. R.J. Leveque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, New York, Melbourne, Madrid, Cape Town, Singapore, São Paulo, Delhi, Tokyo, Mexico City, 2002)
4. Y. Li, S. Osher, Coordinate descent optimization for l^1 minimization with application to compressed sensing; a Greedy algorithm. Inverse Prob. Imaging **3**(3), 487–503 (2009)
5. S.C. Reddy, L.N. Trefethen, Stability of the method of lines. Numer. Math. **62**, 235–267 (1992)

6. H. Schaeffer, R. Caflisch, C. Hauck, S. Osher, Sparse dynamics for partial differential equations. Proc. Natl. Acad. Sci. USA **110**(17), 6634–6639 (2013)
7. J.C. Schatzman, Accuracy of the discrete fourier transform and the fast fourier transform. SIAM J. Sci. Comput. **17**(5), 1150–1166 (1996)

High Order Finite Difference Schemes for the Heat Equation Whose Convergence Rates are Higher Than Their Truncation Errors

A. Ditkowski

Abstract Typically when a semi-discrete approximation to a partial differential equation (PDE) is constructed a discretization of the spatial operator with a truncation error τ is derived. This discrete operator should be semi-bounded for the scheme to be stable. Under these conditions the Lax–Richtmyer equivalence theorem assures that the scheme converges and that the error will be, at most, of the order of $\|\tau\|$. In most cases the error is in indeed of the order of $\|\tau\|$.

We demonstrate that for the Heat equation stable schemes can be constructed, whose truncation errors are τ, however, the actual errors are much smaller. This gives more degrees of freedom in the design of schemes which can make them more efficient (more accurate or compact) than standard schemes. In some cases the accuracy of the schemes can be further enhanced using post-processing procedures.

1 Introduction

Consider the differential problem:

$$\begin{aligned} \frac{\partial u}{\partial t} &= P\left(\frac{\partial}{\partial x}\right) u \quad , \qquad x \in \Omega \subset \mathbb{R}^d \, , t \geq 0 \\ u(t=0) &= f(x) \, . \end{aligned} \tag{1}$$

where $u = (u_1, \ldots, u_m)^T$ and $P(\partial / \partial x)$ is a linear differential operator with appropriate boundary conditions. It is assumed that this problem is well posed, i.e. $\exists K(t) < \infty$ such that $||u(t)|| \leq K(t)||f||$, where typically $K(t) = Ke^{\alpha t}$.

A. Ditkowski (✉)
School of Mathematical Sciences, Tel Aviv University, Tel Aviv 69978, Israel
e-mail: adid@post.tau.ac.il

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_13

Let Q be the discretization of $P\,(\partial\,/\partial x)$ where we assume:

Assumption 1: The discrete operator Q is based on grid points $\{x_j\}, j = 1, \ldots, N$.
Assumption 2: Q is semi-bounded in some equivalent scalar product $(\cdot,\cdot)_H = (\cdot, H\cdot)$, i.e

$$(\mathbf{w}, Q\mathbf{w})_H \;\le\; \alpha\,(\mathbf{w},\mathbf{w})_H \;=\; \alpha\,\|\mathbf{w}\|_H^2\;. \tag{2}$$

Assumption 3: The local truncation error vector of Q is $\mathbf{T}_e$ which is defined, at each entry j by

$$(\mathbf{T}_e)_j \;=\; \big(Pw(x_j)\big) \;-\; (Q\mathbf{w})_j\,, \tag{3}$$

where $w(x)$ is a smooth function and $\mathbf{w}$ is the projection of $w(x)$ onto the grid. It is assumed that $\|\mathbf{T}_e\| \xrightarrow{N\to\infty} 0$.

Consider the semi-discrete approximation:

$$\begin{aligned} \frac{\partial\,\mathbf{v}}{\partial t} &= Q\mathbf{v}\;\;, \qquad t \ge 0 \\ \mathbf{v}(t=0) &= \mathbf{f}\,. \end{aligned} \tag{4}$$

Proposition *Under Assumptions 1–3 The semi-discrete approximation converges.*

Proof Let $\mathbf{u}$ is the projection of $u(x,t)$ onto the grid. Then, from assumption 3,

$$\frac{\partial\,\mathbf{u}}{\partial t} \;=\; P\mathbf{u} \;=\; Q\mathbf{u} + \mathbf{T}_e\;. \tag{5}$$

Let $\mathbf{E} \;=\; \mathbf{u} - \mathbf{v}$ then by subtracting (4) from (5) one obtains the equation for $\mathbf{E}$, namely

$$\frac{\partial\,\mathbf{E}}{\partial t} \;=\; Q\mathbf{E} + \mathbf{T}_e\;. \tag{6}$$

Next, by taking the H scalar product with $\mathbf{E}$, using assumption 2 and the Schwartz inequality the following estimate can be derived

$$\begin{aligned} \left(\mathbf{E}, \frac{\partial\,\mathbf{E}}{\partial t}\right)_H \;=\; \frac{1}{2}\frac{\partial}{\partial t}\,(\mathbf{E},\mathbf{E})_H \;=\; \|\mathbf{E}\|_H\,\frac{\partial}{\partial t}\,\|\mathbf{E}\|_H \;=\; (\mathbf{E}, Q\mathbf{E})_H + (\mathbf{E},\mathbf{T}_e)_H \\ \le\; \alpha\,(\mathbf{E},\mathbf{E})_H \;+\; \|\mathbf{E}\|_H\,\|\mathbf{T}_e\|_H\;. \end{aligned}$$

Thus

$$\frac{\partial}{\partial t}\,\|\mathbf{E}\|_H \;\le\; \alpha\,\|\mathbf{E}\|_H + \|\mathbf{T}_e\|_H\;. \tag{7}$$

Therefore:

$$\|\mathbf{E}\|_H(t) \le \|\mathbf{E}\|_H(t=0)\mathrm{e}^{\alpha t} + \frac{\mathrm{e}^{\alpha t}-1}{\alpha}\max_{0\le\tau\le t}\|\mathbf{T}_e\|_H \xrightarrow{N\to\infty} 0\,. \tag{8}$$

Here we assumed that $\|\mathbf{E}\|_H(t=0)$ is either 0, or at least of the order of machine accuracy. Equation (8) establishes the fact that if the scheme is stable and consistent, the numerical solution $\mathbf{v}$ converges to the projection of the exact solution onto the grid, $\mathbf{u}$. Furthermore, it assures that the error will be at most in the truncation error $\|\mathbf{T}_e\|_H$. This is one part of the landmark Lax–Richtmyer equivalence theorem for semi-discrete approximation. See e.g. [6].

Due to this and similar results the common way for constructing finite difference schemes is to derive a semi-bounded Q with proper truncation error. Typically the error $\|\mathbf{E}\|_H$ is indeed of the order of $\|\mathbf{T}_e\|_H$.

It should be noted, however, that (6) is the exact equation for the error dynamics, while (8) is an estimate. In this paper we present finite difference schemes in which the errors are smaller than their truncation errors. It is well known that boundary conditions can be of one order lower accuracy without destroying the convergence rate expected from the approximation at inner points, see e.g. [1, 4, 5, 9]. In [5, 9] it was shown that for parabolic, incompletely parabolic and 2nd-order hyperbolic equations the boundary conditions can be of two order less. Here, however, we consider low order truncation errors in most or all of the grid points.

This paper is constructed as follows; in Sect. 2 we present a preliminary example and illustrate the mechanism which reduces the error. In Sect. 3 we present a two-point block scheme which has a first order truncation error but has a second or third order error. In Sect. 4 two three-point block schemes are presented. Discussion and remarks are presented in Sect. 5.

2 Preliminary Example

Consider the heat equation

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial^2 u}{\partial x^2} + F(x,t)\quad , \qquad x\in[0,2\pi)\,, t\ge 0 \\ u(t=0) &= f(x) \end{aligned} \tag{9}$$

with periodic boundary conditions.

Let the scheme be:

$$\begin{aligned} \frac{\partial v_j}{\partial t} &= D_+D_-v_j + (-1)^j c\, v_j + F_j(t)\ ; \qquad x_j = jh,\ \ h = 2\pi/N\,,\ \ N \text{ is even} \\ v_j(t=0) &= f_j \end{aligned} \tag{10}$$

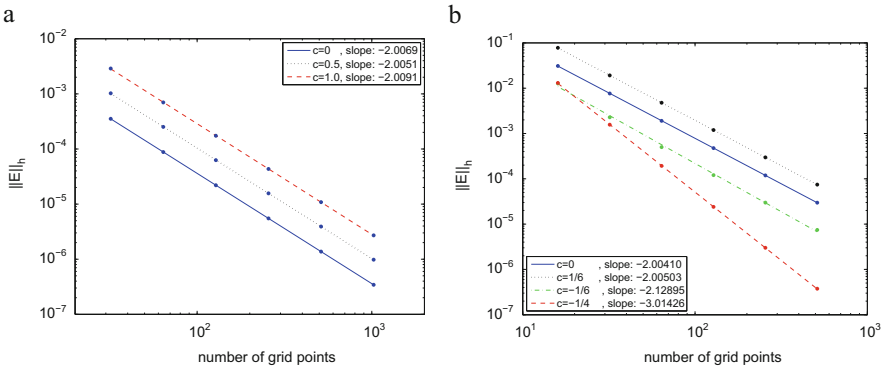


Fig. 1 Convergence plots, $\log_{10}\|\mathbf{E}\|$ vs. $\log_{10} N$, for different values of c. (**a**): Scheme (10). (**b**): Scheme (16)

where $F_j(t)$ is the projection of $F(x,t)$ onto the spatial grid. The truncation error is

$$(T_e)_j = \frac{h^2}{12}\left(u_j\right)_{xxxx} + O(h^2) - (-1)^j c\, v_j = O(1)\ . \tag{11}$$

Formally, this scheme is not consistent. The scheme (10) was run with the initial condition $f(x) = \cos(x)$, $F = 0$ and $N = 32, 64, \ldots, 1024$ with forward-Euler time propagator. The plots of $\log_{10}\|\mathbf{E}\|$ vs. $\log_{10} N$, at $t_{\text{final}} = 2\pi$, for $c = 0, 0.5, 1$ are presented in Fig. 1a. It can be seen that this is a second order scheme.

In order to understand this phenomenon let us consider one high frequency mode of the error.[1] Denote $(T_c)_j = (-1)^j c\, v_j$. This term can also be written as:

$$(T_c)_j = (-1)^j c\, h^\alpha = c h^\alpha\, \mathrm{e}^{iNx_j/2} \tag{12}$$

For the scheme (10) $\alpha = 0$. The equation for the error term caused by T_c is

$$\frac{\partial \mathbf{E}_c}{\partial t} = D_+ D_- \mathbf{E}_c + \mathbf{T}_c \tag{13}$$

Since $(\mathbf{E}_c)_j = \hat{\mathbf{E}}_c\, \mathrm{e}^{iNx_j/2}\sqrt{2\pi}$, $\hat{\mathbf{E}}_c \in \mathbb{C}$, then the equation for $\hat{\mathbf{E}}_c$ is

$$\frac{\partial \hat{\mathbf{E}}_c}{\partial t} = -\left(\frac{N}{2}\right)^2 \hat{\mathbf{E}}_c + c' h^\alpha \tag{14}$$

[1] This scheme was presented for demonstrating the phenomenon that the error, due to high frequency modes, is lower than the truncation error. As this is not a practical scheme, full analysis of the error is not presented, only a demonstration of the dynamics of high frequency error modes is presented. Full analysis is given for the scheme presented in the next section.

Therefore,

$$\|\mathbf{E}_c\|(t) = \left|\hat{\mathbf{E}}_c\right|(t) = \left|\hat{\mathbf{E}}_c\right|(0)\mathrm{e}^{-\left(\frac{N}{2}\right)^2 t} + \left(\frac{2}{N}\right)^2 \left(1 - \mathrm{e}^{-\left(\frac{N}{2}\right)^2 t}\right) c' h^{\alpha} \tag{15}$$
$$\leq \left|\hat{\mathbf{E}}_c\right|(0)\mathrm{e}^{-\left(\frac{N}{2}\right)^2 t} + O\left(h^{\alpha+2}\right)$$

Note that the actual error, $\|\mathbf{E}_c\|(t)$, is two orders lower that the truncation error, $\|T_c\|$. In the next sections we present practical schemes which utilize this idea.

3 Two-Point Block, 3rd Order Scheme

Let the grid be: $x_j = jh$, $h = 2\pi/(N+1)$ and $x_{j+1/2} = x_j + h/2, j = 0, \ldots N$. Altogether there are $2(N+1)$ points with spacing of $h/2$. For simplicity, we assume that N is even.

Consider the approximation:

$$\frac{d^2}{dx^2}u_j \approx \frac{1}{(h/2)^2}\left[\left(u_{j-1/2} - 2u_j + u_{j+1/2}\right) + c\left(-u_{j-1/2} + 3u_j - 3u_{j+1/2} + u_{j+1}\right)\right] \tag{16}$$
$$\frac{d^2}{dx^2}u_{j+1/2} \approx \frac{1}{(h/2)^2}\left[\left(u_j - 2u_{j+1/2} + u_{j+1}\right) + c\left(u_{j-1/2} - 3u_j + 3u_{j+1/2} - u_{j+1}\right)\right]$$

The truncation errors are:

$$(T_e)_j = \frac{1}{12}\left(\frac{h}{2}\right)^2 (u_j)_{xxxx} + c\left[\left(\frac{h}{2}\right)(u_j)_{xxx} + \frac{1}{2}\left(\frac{h}{2}\right)^2 (u_j)_{xxxx}\right] + O(h^3)$$
$$= O(h) \tag{17}$$
$$(T_e)_{j+\frac{1}{2}} = \frac{1}{12}\left(\frac{h}{2}\right)^2 \left(u_{j+\frac{1}{2}}\right)_{xxxx} + c\left[-\left(\frac{h}{2}\right)\left(u_{j+\frac{1}{2}}\right)_{xxx} + \frac{1}{2}\left(\frac{h}{2}\right)^2 \left(u_{j+\frac{1}{2}}\right)_{xxxx}\right]$$
$$+O(h^3) = O(h)$$

The motivation leading to this scheme is that the highly oscillating $O(h)$ error terms will be dissipated, as in the previous example, while the $O(h^2)$ terms will be canceled, for the proper value of c.

3.1 Analysis

Let $\omega \in \{-N/2, \ldots, N/2\}$ and

$$\nu = \begin{cases} \omega - (N+1) & \omega > 0 \\ \\ \omega + (N+1) & \omega \le 0 \end{cases} \tag{18}$$

Then

$$e^{i\omega x_j} = e^{i\nu x_j} \text{ and } e^{i\omega x_{j+1/2}} = -e^{i\nu x_{j+1/2}} . \tag{19}$$

We look for eigenvectors in the form of:

$$\psi_k(\omega) = \frac{\alpha_k}{\sqrt{2\pi}} \begin{pmatrix} \vdots \\ e^{i\omega x_j} \\ e^{i\omega x_{j+1/2}} \\ \vdots \end{pmatrix} + \frac{\beta_k}{\sqrt{2\pi}} \begin{pmatrix} \vdots \\ e^{i\nu x_j} \\ e^{i\nu x_{j+1/2}} \\ \vdots \end{pmatrix} \tag{20}$$

where, for normalization, it is require that $|\alpha_k|^2 + |\beta_k|^2 = 1$, $k = 1, 2$. The expressions for α_k, β_k and the eigenvalues (symbols) $\hat{Q}_k$ are:

$$\alpha_1 = \left[\sqrt{1 + \frac{\begin{array}{r} c^2 \cos(4(h/2)\omega) + 4(2c-1)\Delta \cos((h/2)\omega) + 4(c(7c-8)+2) \\ \times \cos(2(h/2)\omega) + (35c-32)c + 8 \end{array}}{2c^2(2\sin((h/2)\omega) + \sin(2(h/2)\omega))^2}} \right]^{-1} \tag{21}$$

$$\beta_1 = -\frac{i((8c-4)\cos((h/2)\omega) + \Delta)}{2c(2\sin((h/2)\omega) + \sin(2(h/2)\omega))\alpha_1^{-1}} \tag{22}$$

$$\beta_2 = \left[\sqrt{1 + \frac{2c^2(2\sin((h/2)\omega) + \sin(2(h/2)\omega))^2}{\begin{array}{r} c^2 \cos(4(h/2)\omega) + 4(1-2c)\Delta \cos((h/2)\omega) + 4(c(7c-8)+2) \\ \times \cos(2(h/2)\omega) + (35c-32)c + 8 \end{array}}} \right]^{-1} \tag{23}$$

$$\alpha_2 = -\frac{2ic(2\sin((h/2)\omega) + \sin(2(h/2)\omega))}{((4-8c)\cos((h/2)\omega) + \Delta)\beta_2^{-1}} \tag{24}$$

where

$$\Delta = \sqrt{2c^2\cos(4(h/2)\omega) + 38c^2 + 8(c-1)(3c-1)\cos(2(h/2)\omega) - 32c + 8} \tag{25}$$

and the

$$\hat{Q}_{1,2}(\omega) = \frac{-4 + 2c(\cos(2(h/2)\omega) + 3) \pm \Delta}{2(h/2)^2} \,. \tag{26}$$

It can be shown that the eigenvalues are real and non positive for $|c| < 1/2$. Therefore the scheme is stable.

For $\omega h \ll 1$ the eigenvalues and eigenvectors are:

$$\hat{Q}_1(\omega) = -\omega^2 + \frac{(1+4c)\omega^4}{12-24c}\left(\frac{h}{2}\right)^2 + O(h^4) \tag{27}$$

$$\alpha_1 = 1 - \frac{c^2}{32(1-2c)^2}\left(\frac{\omega h}{2}\right)^6 + O\left(\mathrm{h}^7\right) \,, \quad \beta_1 = -\frac{ic}{4-8c}\left(\frac{\omega h}{2}\right)^3 + O(h^5) \tag{28}$$

and

$$\hat{Q}_2(\omega) = -\frac{4-8c}{(\mathrm{h}/2)^2} + (1-4c)\omega^2 + O(h^2) \tag{29}$$

$$\alpha_2 = \frac{ic}{2c-1}\left(\frac{\omega h}{2}\right) + O(h^3) \,, \quad \beta_2 = 1 + O(h^2) \tag{30}$$

If the initial condition is

$$\mathbf{v}_j(0) = \mathrm{e}^{i\omega x_j} \,, \mathbf{v}_{j+\frac{1}{2}}(0) = \mathrm{e}^{i\omega x_{j+\frac{1}{2}}} \quad ; \quad \omega^2 h \ll 1 \tag{31}$$

then

$$\begin{aligned}(\mathbf{v})_j(t) = \mathrm{e}^{-\omega^2 t}&\left(1 - \frac{(1+4c)\omega^2 t}{12-24c}\left(\frac{\omega h}{2}\right)^2 + O(h^4)\right)\mathrm{e}^{i\omega x_j} \\ &+\left(-\frac{ic}{4-8c}\left(\frac{\omega h}{2}\right)^3 + O(h^5)\right)\mathrm{e}^{iv x_j}\end{aligned} \tag{32}$$

The same expression hold for $x_{j+\frac{1}{2}}$. Therefore the scheme is, in general, 2nd order and it is 3rd order if $c = -1/4$. Note that by naive analysis of the truncation error terms, (17), one would expect to get 3rd order with $c = -1/6$.

We used the approximation (16) for solving the heat equation (9), where $F(x,t)$ and the initial condition were chosen such that the exact solution is $u(x,t) =$

$\exp(\cos(x-t))$. The scheme was run with $N = 32, 64, \ldots, 1024$. 4th order Runge–Kutta scheme was used for time integration. The plots of $\log_{10}\|\mathbf{E}\|$ vs. $\log_{10} N$ for $c = 0, 1/6, -1/6, -1/4$ are presented in Fig. 1b. As can be seen, the results are as predicted by the analysis.

4 Three-Point Block

In this section we briefly present two schemes which are based on three-point block. Here we use the grid, $x_j = jh$, $h = 2\pi/(N+1)$ with the internal block nodes $x_{j+1/3} = x_j + h/3$ and $x_{j+2/3} = x_j + 2h/3$, $j = 0, \ldots N$. Altogether there are $3(N+1)$ points with spacing of $h/3$.

three-point block, 3rd order scheme

Consider the approximation:

$$\begin{aligned}
\frac{d^2}{dx^2}u_j &= \frac{1}{4(h/3)^2}\left[\left(4u_{j-1/3} - 8u_j + 4u_{j+1/3}\right)\right. \\
&\quad \left. +c\left(-u_{j-1/3} + 3u_j - 3u_{j+1/3} + u_{j+2/3}\right)\right] + O(h) \\
\frac{d^2}{dx^2}u_{j+1/3} &= \frac{1}{4(h/3)^2}\left[\left(4u_j - 8u_{j+1/3} + 4u_{j+2/3}\right)\right] + O(h^2) \\
\frac{d^2}{dx^2}u_{j+2/3} &= \frac{1}{4(h/3)^2}\left[\left(4u_{j+1/3} - 8u_{j+2/3} + 4u_{j+1}\right)\right. \\
&\quad \left. +c\left(u_j - 3u_{j+1/3} + 3u_{j+2/3} - u_{j+1}\right)\right] + O(h)
\end{aligned} \tag{33}$$

This scheme was run under the same conditions as the example in Sect. 3. As in the previous example the truncation error is of $O(h)$, however this is a 2nd order scheme and 3rd order for $c = 1.340$. The convergence results are presented in Fig. 2a.

three-point block, 5th order scheme

$$\begin{aligned}
\frac{d^2}{dx^2}u_j &= \frac{1}{12(h/3)^2}\left[\left(-u_{j-2/3} + 16u_{j-1/3} - 30u_j + 16u_{j+1/3} - u_{j+2/3}\right) + \right. \\
&\quad \left. c\left(u_{j-2/3} - 5u_{j-1/3} + 10u_j - 10u_{j+1/3} + 5u_{j+2/3} - u_j\right)\right] + O(h^3) \\
\frac{d^2}{dx^2}u_{j+1/3} &= \frac{1}{12(h/3)^2}\left[\left(-u_{j-1/3} + 16u_j - 30u_{j+1/3} + 16u_{j+2/3} - u_j\right)\right] + O(h^4) \\
\frac{d^2}{dx^2}u_{j+2/3} &= \frac{1}{12(h/3)^2}\left[\left(-u_j + 16u_{j+1/3} - 30u_{j+2/3} + 16u_{j+1} - u_{j+4/3}\right) + \right. \\
&\quad \left. c\left(-u_{j-1/3} + 5u_j - 10u_{j+1/3} + 10u_{j+2/3} - 5u_{j+1} + u_{j+4/3}\right)\right] + O(h^3)
\end{aligned} \tag{34}$$

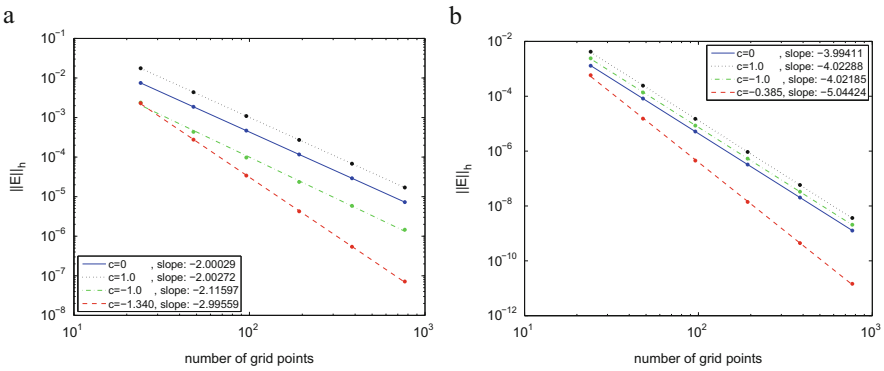


Fig. 2 Convergence plots, $\log_{10} \|\mathbf{E}\|$ vs. $\log_{10} N$, for different values of c. (**a**):Scheme (33). (**b**): Scheme (34)

This scheme was run under the same conditions as the previous examples, with the exception that now a 6th order Runge–Kutta scheme was used for time integration. Here the truncation error is of $O(h^3)$, however this is a 4th order scheme and 5th order for $c = -0.385$. The convergence results are presented in Fig. 2b.

It should be noted that by taking $c = 1$ the coefficients of $u_{j-2/3}$ and $u_{j+4/3}$ are 0. Therefore the scheme is more 'compact' than standard explicit 4th order scheme in the sense that the scheme depends only on one term, on each side, out of the three-point block $u_j, u_{j+1/3}, u_{j+2/3}$. Potentially, this thinner stencil helps in the derivation of boundary schemes for initial-boundary value problems.

5 Summary

In this paper we presented a few block-finite-difference schemes in which the actual errors are much smaller than their truncation errors. This reduction of error was achieved by constructing the truncation errors to be oscillatory and using the dissipative property of the scheme.

A comparison between standard and block finite difference schemes in terms of the number of points out of the cell and operation count is presented in Table 1. As can be seen, in terms of accuracy and computational cost the 3rd and 5th order schemes are between the standards 2nd and 4th order and 4th and 6th order schemes respectively.

If $c = -1/4$ in the two-point block, 3rd order scheme (16), the leading term in the error is highly oscillatory, see (32). It was suggested by Jennifer K. Ryan (J. Ryan, Private communication) that this term can be filtered by post-processing. In this technique the high frequency error terms are filtered using convolution with a proper kernel. This method was successfully applied to the discontinuous Galerkin method. As this filtering is done only once, after the final time step, the cost is minimal, see

Table 1 Comparison between standard finite difference schemes and the ones presented on Sects. 3, 4

Scheme	Points out the block (at each side)	Number of operations		Scheme	Points out the block (at each side)	Number of operations	
		+	×			+	×
Standard 2nd order	1	2	3	2-point block 3rd order	1	3	4
Standard 4th order	2	4	5	3-point block 3rd order	1	$2\frac{2}{3}$	$3\frac{2}{3}$
Standard 6th order	3	6	7	3-point block 5th order	2	$4\frac{2}{3}$	$5\frac{2}{3}$
				3-point block 4th order compact	1	4	5

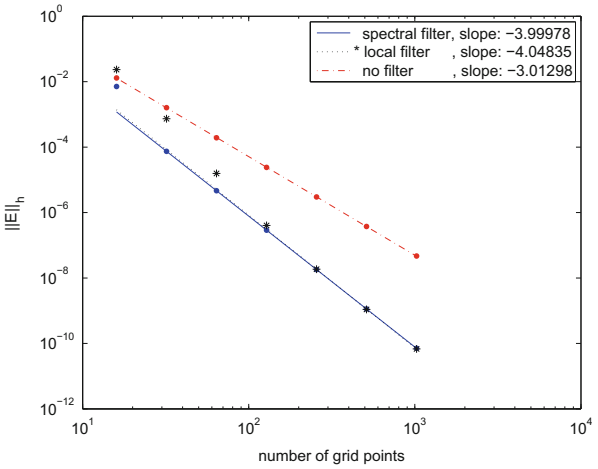


Fig. 3 Convergence plots, $\log_{10}\|\mathbf{E}\|$ vs. $\log_{10} N$, for scheme (16), $c = 1/4$. with no post-processing, with spectral filter and with the filter suggested in [3]

e.g. [2, 3, 8]. Here we used a global spectral filter and the local filter suggested in [3]. As can be seen in Fig. 3, the filtered scheme is 4th order accurate. The difference between both kernels is that the global filter is more computationally expensive, $O(N \log N)$ for proper values of N, but is accurate for small values of N while the local filter requires only $O(N)$ operation but is accurate only for large values of N. The investigation of which is the optimal filter for these schemes is a topic for future research.

It should be noted that finite difference representations of discontinuous Galerkin schemes for the heat equation have a similar form to the schemes presented above. They may also present similar enhancing of accuracy. See [10]. This manuscript was the inspiration to the current work. As was also pointed out in [10] the increasing of accuracy may be related to a phenomenon called Supra-Convergence [7].

Further research of the properties and implementations of these schemes as well as the existence of these kinds of schemes for hyperbolic problems are topics for future research.

Acknowledgements The author would like to thank Jennifer K. Ryan, Chi-Wang Shu and Sigal Gottlieb for the fruitful discussions and their help. The author would also like to thank the anonymous reviewers for their useful remarks.

References

1. S. Abarbanel, A. Ditkowski, B. Gustafsson, On error bounds of finite difference approximations to partial differential equations—temporal behavior and rate of convergence. J. Sci. Comput. **15**(1), 79–116 (2000)

2. B. Cockburn, M. Luskin, C.-W. Shu, E. Suli, *Post-Processing of Galerkin Methods for Hyperbolic Problems* (Springer, New York 2000)
3. B. Cockburn, M. Luskin, C.-W. Shu, E. Suli, Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. Math. Comput. **72**(242), 577–606 (2003)
4. B. Gustafsson, The convergence rate for difference approximations to mixed initial boundary value problems. Math. Comput. **29**(130), 396–406 (1975)
5. B. Gustafsson, The convergence rate for difference approximations to general mixed initial-boundary value problems. SIAM J. Numer. Anal. **18**(2), 179–190 (1981)
6. B. Gustafsson, H.-O. Kreiss, J. Oliger, *Time-Dependent Problems and Difference Methods* (Wiley, New York, 1995)
7. H.-O. Kreiss, T.A. Manteuffel, B. Swartz, B. Wendroff, A.B. White, Supra-convergent schemes on irregular grids. Math. Comput. **47**(176), 537–554 (1986)
8. J. Ryan, C.-W. Shu, H. Atkins, Extension of a post processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem. SIAM J. Sci. Comput. **26**(3), 821–843 (2005)
9. M. Svärd, J. Nordström, On the order of accuracy for difference approximations of initial-boundary value problems. J. Comput. Phys. **218**(1), 333–352 (2006)
10. M. Zhang, C.-W. Shu, An analysis of three different formulations of the discontinuous Galerkin method for diffusion equations. Math. Models Methods Appl. Sci. **13**(03), 395–413 (2003)

Hybrid Compact-WENO Finite Difference Scheme For Detonation Waves Simulations

Yanpo Niu, Zhen Gao, Wai Sun Don, Shusen Xie, and Peng Li

Abstract The performance of a hybrid compact (Compact) finite difference scheme and characteristic-wise weighted essentially non-oscillatory (WENO) finite difference scheme (Hybrid) for the detonation waves simulations is investigated. The Hybrid scheme employs the nonlinear 5*th*-order WENO-Z scheme to capture high gradients and discontinuities in an essentially non-oscillatory manner and the linear 6*th*-order Compact scheme to resolve the fine scale structures in the smooth regions of the solution in an efficient and accurate manner. Numerical oscillations generated by the Compact scheme is mitigated by the high order filtering. The high order multi-resolution algorithm is employed to detect the smoothness of the solution. The Hybrid scheme allows a potential speedup up to a factor of three or more for certain classes of shocked problems. The simulations of one-dimensional shock-entropy wave interaction and classical stable detonation waves, and the two-dimensional detonation diffraction problem around a 90° corner show that the Hybrid scheme is more efficient, less dispersive and less dissipative than the WENO-Z scheme.

Y. Niu • W.S. Don (✉) • S. Xie
School of Mathematical Sciences, Ocean University of China, Qingdao, China
e-mail: yanponiu@163.com; waisundon@gmail.com; shusenxie@ouc.edu.cn

Z. Gao
Key Laboratory of Marine Environment & Ecology, Ministry of Education, Qingdao, China

School of Mathematical Sciences, Ocean University of China, Qingdao, China
e-mail: zhengao@ouc.edu.cn

P. Li
State Key Laboratory of Explosion Science and Technology, Beijing Institute of Technology, Beijing, China
e-mail: weilailp@gmail.com

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_14

1 Introduction

Detonation is a complex phenomenon that involves a shock front followed by a reaction zone. Accurate and efficient numerical simulations of a mathematical model of detonation waves provide a way to obtain insights in the physical problems and guide researchers to have a deeper understanding of the physics and to design better experiments.

Characteristic-wise WENO conservative finite difference schemes on an equidistant stencil as a class of high order/resolution nonlinear scheme for solutions of hyperbolic conservation laws in the presence of shocks and small scale structures was initially developed in [11] (for details and history of WENO scheme, see [14] and references contained therein). It has been shown that the WENO-Z scheme [1, 3] is less dissipative and has higher resolution power than the classical WENO-JS scheme [11] for a larger class of problems. High order compact finite difference (Compact) schemes are sufficiently accurate to resolve both small and large scale structures presented at direct numerical simulation of highly complex flows. However, when applied to simulate the propagation of detonation waves near the detonation front exhibiting high gradients and discontinuities, known as the Gibbs phenomenon, that causes loss of accuracy and numerical instability.

In this work, we aim at the conjugation of high order Compact scheme and the WENO-Z scheme (Hybrid) for numerical simulations of detonation waves. The $5th$-order characteristic-wise WENO-Z finite difference scheme and $6th$-order Compact finite difference scheme are employed to resolve solutions in the non-smooth parts and the smooth parts of the solution respectively. A high order multi-resolution analysis [9] is performed at every Runge-Kutta step to measure the degree of smoothness at a given grid point to maintain the high order/resolution nature of the Hybrid scheme.

The paper is organized as follows. In Sect. 2, a very brief introduction to the WENO-Z scheme, the Compact scheme and the Hybrid scheme for solving hyperbolic conservation laws on uniform cells are given. In Sect. 3, the one-dimensional shock-entropy wave interaction and classical stable detonation waves, and the two-dimensional detonation diffraction problem around a 90° corner are simulated by the Hybrid scheme and their results are discussed. Conclusions are given in Sect. 4.

2 Hybrid Compact-WENO Finite Difference Scheme

The nonlinear system of hyperbolic conservation laws can be written compactly as

$$\frac{\partial \mathbf{Q}}{\partial t} + \nabla \cdot \mathbf{F}(\mathbf{Q}) = \mathbf{S}, \tag{1}$$

where $\mathbf{Q}, \mathbf{F}$ and $\mathbf{S}$ are vectors of the conservative variables, flux and source term respectively.

Consider a uniformly spaced grid defined by the points $x_i = i\Delta x, i = 0, \ldots, N$, which are called cell centers, with cell boundaries given by $x_{i+\frac{1}{2}} = x_i + \frac{\Delta x}{2}$, where Δx is the uniform cell size. The semi-discretized form of (1) is transformed into the system of ordinary differential equations and solved by the method of lines

$$\frac{dQ_i(t)}{dt} = -\left.\frac{\partial f}{\partial x}\right|_{x=x_i}, \quad i = 0, \ldots, N, \tag{2}$$

where $Q_i(t)$ is a numerical approximation to the cell-averaged value $Q(x_i, t)$.

2.1 Weighted Essentially Non-Oscillatory Schemes

The 5*th*-order WENO-Z scheme [1, 3] defines the nonlinear weights ω_k^z as

$$\alpha_k^z = \frac{d_k}{\beta_k^z} = d_k\left(1 + \left(\frac{\tau_5}{\beta_k + \epsilon}\right)^p\right), \quad \omega_k^z = \alpha_k^z / \sum_{l=0}^{2} \alpha_l^z, \quad k = 0, 1, 2, \tag{3}$$

where $\tau_5 = |\beta_0 - \beta_2|$, which has a leading truncation error of order $O(\Delta x^5)$. In contrary, the leading truncation error of β_k are of order $O(\Delta x^2)$ in an absence of critical points [5]. The sensitivity and power parameters are $\epsilon = 10^{-12}$ and $p = 2$, respectively. $\left\{d_0 = \frac{3}{10},\ d_1 = \frac{3}{5},\ d_2 = \frac{1}{10}\right\}$ are the ideal weights that, when the solution is sufficiently smooth, one has $\omega_k \approx d_k$ and the WENO-Z scheme becomes the optimal 5*th*-order central upwind scheme.

2.2 Compact Finite Difference Schemes

A 6*th*-order ($c_r = 6$) compact finite difference scheme [12] approximates the derivative of a function on a uniformly spaced grids can be written compactly as

$$\mathbf{A}\mathbf{g}' = \mathbf{B}\mathbf{g} + \mathbf{b}, \tag{4}$$

where **A** and **B** are the banded coefficient matrices,

$$\mathbf{A} = \begin{pmatrix} 1 & 1/3 & & & & \\ 1/3 & 1 & 1/3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & 1/3 & 1 & 1/3 & \\ & & & 1/3 & 1 & \end{pmatrix}, \quad \mathbf{B} = \frac{1}{36\Delta x} \begin{pmatrix} 0 & 28 & 1 & & & \\ -28 & 0 & 28 & 1 & & \\ -1 & -28 & 0 & 28 & 1 & \\ & \ddots & \ddots & \ddots & \ddots & \ddots \\ & & -1 & -28 & 0 & 28 \\ & & & -1 & -28 & 0 \end{pmatrix}.$$

The vector **b** is

$$\mathbf{b} = \frac{1}{36\Delta x}\left(-g_{-1} - 28g_0, -g_0, 0, \cdots, 0, g_N, 28g_N + g_{N+1}\right)^{\mathrm{T}} - \frac{1}{3}\left(g_0', 0, 0, \cdots, 0, 0, g_N'\right)^{\mathrm{T}},$$

where $g_{-1} = g(x_0 - \Delta x)$ and $g_{N+1} = g(x_N + \Delta x)$ are the ghost points. The derivatives at the two boundary points g_0' and g_N' are computed by the WENO-Z scheme.

2.3 Hybrid Scheme

By performing a high order MR analysis [9, 10] of a given variable (usually density), a MR flag Flag is specified by

$$\text{Flag}_i = \begin{cases} 1, & |d_i| > \epsilon_{MR} \text{ Non-Smooth}, \\ 0, & \text{otherwise Smooth}, \end{cases} \tag{5}$$

where d_i is the MR coefficient at x_i which measures the error in approximating the function by a high degree polynomial ($8th$) and ϵ_{MR} is the MR tolerance parameter ($\epsilon_{MR} = 10^{-3}$ in this study.) Readers are referred to [4, 9] for details on the MR analysis. In the Hybrid scheme, a buffer zone is created around each cell that is flagged as non-smooth with $\text{Flag}_i = 1$ to allow a smooth transition between non-smooth and smooth stencils. Then, the PDEs (1) is solved by the WENO-Z and Compact schemes at non-smooth and smooth stencils respectively. In smooth stencils, the numerical fluxes on the non-smooth zones obtained by the WENO scheme are saved and automatically used as the internal boundary fluxes needed by the Compact scheme, and an $8th$-order finite difference filtering is applied to stabilize the Hybrid scheme ([15]).

The resulting system of ordinary differentiation equations (ODEs) resulted from the spatial discretization is advanced in time via the third order TVD Runge-Kutta scheme [1]. The CFL condition is set to be CFL $= 0.4$ in this study. Readers are referred to [4, 6] for details. We would like to remark that, in implementation, the Hybrid Compact-WENO scheme is different from the Hybrid Central-WENO

scheme [6] where the central scheme can first be applied at all grid points, and the solution in the non-smooth stencils are then updated by the WENO scheme.

3 Governing Equations and Numerical Results

For the one-dimensional unsteady reactive Euler equations with a perfect ideal gas coupled with one step irreversible chemical reaction, one has, from (1),

$$\mathbf{Q} = (\rho,\ \rho u,\ E,\ \rho f_1), \quad \mathbf{F} = (\rho u,\ (\rho u^2 + P),\ (E + P)u,\ \rho f_1 u), \quad \mathbf{S} = (0,\ 0,\ 0,\ \dot{\omega}), \tag{6}$$

where ρ is density, P is pressure, u is velocity, and $0 \leq f_1 \leq 1$ is the reactant mass fraction. The total specific energy, with an addition of energy $\rho f_1 q_0$ generated through the chemical reaction, is given by $E = \frac{P}{\gamma - 1} + \frac{1}{2}\rho u^2 + \rho f_1 q_0$. The source term consists of the energy production term in the form of $\dot{\omega}(T, f_1) = -K\rho f_1 e^{-E_a/T}$ where γ is the ratio of specific-heat ($\gamma = 1.2$ is used in this study), q_0 is the heat-release parameter, E_a is the activation-energy parameter, and K is a pre-exponential factor that sets the spatial and temporal scales. The temperature $T = P/\rho R$, R is the specific gas constant ($R = 1$ in this study). Readers are referred to [7, 8] for details on the initial conditions and the boundary conditions.

3.1 Shock Interaction with Small Entropy Wave

To demonstrate the performance of the Hybrid scheme in terms of accuracy and efficiency, we solve the source-less Euler equations (6) in simulating a right moving Mach 3 shock interacting with a small amplitude sinusoidal perturbation of the entropy in the pre-shock region. The initial condition is

$$(\rho, u, P) = \begin{cases} \left(\frac{27}{7},\ \frac{4\sqrt{35}}{9},\ \frac{31}{3}\right), & x \leq x_0, \\ \left(\exp(-\varepsilon \sin(k(x + x_0))),\ 0,\ 1\right), & x > x_0, \end{cases}$$

where $x \in [-10, 10], \varepsilon = 0.01, x_0 = -9.5$ and $k = 13$. The final time is $t_f = 5$. Since there is no exact solution for this problem, the numerical solution computed by the WENO-JS9 scheme with $N = 10{,}240$ uniform cells is used as the reference solution.

The left figure of Fig. 1 shows that the MR analysis captures the location of the shock very well. In the middle and right figures of Fig. 1, it is clear from the evolution of the small amplitude high frequency entropy waves behind the main shock that the wave form computed by the Hybrid scheme has no discernible dissipation and dispersion errors over time at both lower and higher resolutions.

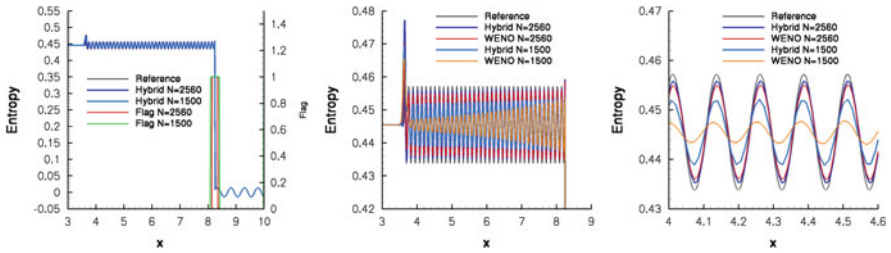


Fig. 1 (*Left*) The entropy and WENO flag (*red* and *green solid lines*) of the Hybrid scheme, (*Middle*) and (*Right*) close-up view of entropy as computed by the WENO-Z and Hybrid schemes with $N = 1500$ and $N = 2560$ at the final time $t_f = 5$

Table 1 Comparative CPU timing and speedup for the shock-entropy wave interaction

$2r-1$	c_r	N	WENO-Z	Hybrid	*Speedup*
5	6	1500	7.3	2.5	2.9
		2560	20.5	5.9	3.5

In contrary, those computed by the WENO-Z scheme are severely dampened at a lower resolution and increasingly less so at the higher resolution. Table 1 gives the comparative CPU timing and speedup of both schemes. We observe that the Hybrid scheme is at least *three* times faster than the WENO-Z scheme.

3.2 One-Dimensional Detonation Waves

Here we evaluate the performance of the Hybrid scheme by simulating the one-dimensional stable detonation waves with the parameters $f = 1.8$, $q_0 = 50$, $E_a = 50$, $K = 145.69$ and the final time $t_f = 100$. The physical domain is set to be $x = [120, 180]$ with PML layer $x = [120, 130]$ and the location of the initial detonation front at $x_d = 160$. Readers are referred to [6] for details. The numerical solution computed by the WENO-Z scheme with $N = 4800$ uniform cells serves as the reference solution.

The left figure of Fig. 2 gives the density spatial profiles showing that the MR analysis captures the location of the detonation front very accurately. The peak pressure temporal histories $P_m(t)$ computed by the Hybrid and WENO-Z schemes with several resolutions at the time $t_f = 100$ are shown in the right figure of Fig. 2, which agree well with those given in [2, 13]. At the lower resolution $N = 1800$, the temporal history of the peak pressure of both schemes is oscillatory and does not seem to reach a steady state. As one increases the resolution to $N = 3000$, both schemes reach the constant steady state solution with a value slightly lower than the reference solution. Table 2 gives the comparative CPU timing and speedup, which shows that the Hybrid scheme is at least *three* times faster than the WENO-Z scheme.

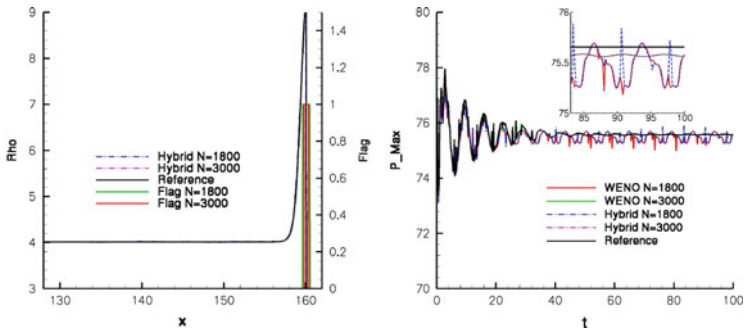


Fig. 2 (*Left*) The density and WENO flag (*green* and *red solid lines*) of the Hybrid scheme and (*Right*) the peak pressure temporal histories $P_m(t)$ of detonation waves with the overdrive factor $f = 1.8$ at the final time $t_f = 100$

Table 2 Comparative CPU timing and speedup for the one-dimensional detonation waves

$2r-1$	c_r	N	WENO-Z	Hybrid	*Speedup*
5	6	1800	295	77	3.8
		3000	815	205	4.0

3.3 Two-Dimensional Detonation Diffraction problem

In this section, we consider the detonation diffraction problem. It is numerically challenging especially for the high order schemes because the pressure and density may drop very close to zero when the shock wave is diffracted around an obstacle making an 90^o angle turn (see Fig. 4). The initial condition is

$$(\rho, u, v, E, f_1) = \begin{cases} (\ 11,\ 6.18,\ 0,\ 970,\ 1), & x < 0.5, \\ (\ \ 1,\ \ \ 0,\ 0,\ \ 55,\ 1), & \text{otherwise}, \end{cases}.$$

The physical domain is set to be $(x, y) = [0, 5] \times [0, 5]$. The boundary conditions are reflective except that at $x = 0$, $(\rho, u, v, E, f_1) = (11, 6.18, 0, 970, 1)$. The uniform cells used are $N_x \times N_y = 400 \times 400$ and $N_x \times N_y = 1000 \times 1000$. The final time is $t_f = 0.6$.

As shown in Fig. 3, the MR analysis captures the detonation front very accurately. In Fig. 4, the density and pressure computed by the Hybrid scheme at $t_f = 0.6$ are in a very good agreement with those in [16]. The solution computed by the WENO-Z scheme is omitted as they are very similar to the one computed by the Hybrid scheme. One can see that the density becomes very small when the flow expands around the corner and is well handled by the Hybrid scheme. In Table 3, the comparative CPU timing and speedup show that the Hybrid scheme is at least *two and half* times faster than the WENO-Z scheme.

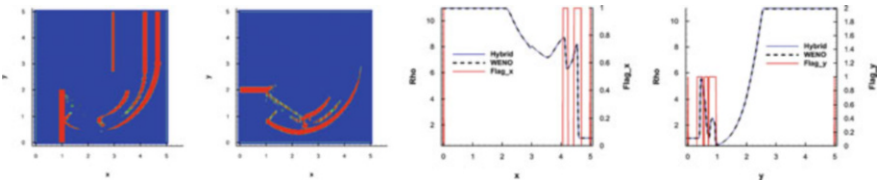


Fig. 3 The multi-resolution flags in the x- and y-directions of detonation diffraction around a 90° corner computed by the Hybrid scheme with the uniform cells $N_x \times N_y = 400 \times 400$

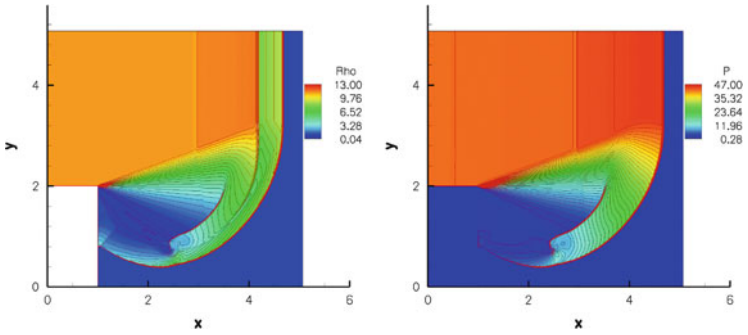


Fig. 4 The density and pressure of detonation diffraction around a 90° corner as computed by the Hybrid scheme with the uniform cells $N_x \times N_y = 400 \times 400$

Table 3 Comparative CPU timing and speedup for the two-dimensional detonation diffraction problem

$2r-1$	c_r	$N_x \times N_y$	WENO-Z	Hybrid	*Speedup*
5	6	400× 400	1697	691	2.5
		1000× 1000	27,980	8430	3.3

4 Conclusion

We studied the performance of the hybrid Compact-WENO finite difference scheme (Hybrid) in the simulations of detonation waves. The Hybrid scheme is used to keep the solutions parts displaying high gradients and discontinuities always captured by the WENO-Z scheme in an essentially non-oscillatory manner while the smooth parts are highly resolved by a more efficient and high resolution compact finite difference scheme and to speedup the computation of the overall scheme. Here, the 5*th*-order WENO-Z schemes and the 6*th*-order Compact scheme are conjugated in the discontinuous and smooth parts respectively. To detect the smooth and discontinuous parts of the solutions, a high order multi-resolution algorithm was used. The 8*th*-order finite difference filter was used to mitigate the numerical oscillations of the Compact scheme. We conducted several numerical comparisons between the WENO-Z and Hybrid schemes in the simulations of the one-dimensional shock-entropy wave interaction, stable detonation waves and two-dimensional detonation diffraction problem. The results showed that the Hybrid

scheme can be *three* times faster than and as accurate as the WENO-Z scheme. The FORTRAN 95 program is written based on subroutines contained in the high performance software library HOPEpack.

Acknowledgements The authors would like to acknowledge the funding support of this research by National Natural Science Foundation of China (11201441), China Postdoctoral Science Foundation (2012M521374, 2013T60684) and Fundamental Research Funds for the Central Universities (201362033). The author (Don) also likes to thank the Ocean University of China for providing the startup fund (201412003) that is used to support this work. Part of the work was performed during the Second Summer Workshop of Advanced Research in Applied Mathematics and Scientific Computing 2014 and the authors are grateful for the support provided by the School of Mathematical Sciences at Ocean University of China.

References

1. R. Borges, M. Carmona, B. Costa, W.S. Don, An improved weighted essentially non-oscillatory scheme for hyperbolic conservation laws. J. Comput. Phys. **227**, 3191–3211 (2008)
2. A. Bourlioux, A.J. Majda, V. Roytburd, Theoretical and numerical structure for unstable one-dimensional detonations. SIAM J. Appl. Math. **51**, 303–343 (1991)
3. M. Castro, B. Costa, W.S. Don, High order weighted essentially non-oscillatory WENO-Z schemes for hyperbolic conservation laws. J. Comput. Phys. **230**, 1766–1792 (2011)
4. B. Costa, W.S. Don, High order hybrid central-WENO finite difference scheme for conservation laws. J. Comput. Appl. Math. **204**(2), 209–218 (2007)
5. W. S. Don, R. Borges, Accuracy of the weighted essentially non-oscillatory conservative finite difference schemes. J. Comput. Phys. **205**, 347–372 (2013)
6. Z. Gao, W.S. Don, Mapped hybrid central-WENO finite difference scheme for detonation waves simulations. J. Sci. Comput. **55**, 351–371 (2012)
7. Z. Gao, W.S. Don Z. Li, High order weighted essentially non-oscillation schemes for one-dimensional detonation wave simulations. J. Comput. Math. **29**, 623–638 (2011)
8. Z. Gao, W.S. Don, Z. Li, High order weighted essentially non-oscillation schemes for two-dimensional detonation wave simulations. J. Sci. Comput. **53**, 80–101 (2012)
9. A. Harten, High resolution schemes for hyperbolic conservation laws. J. Comput. Phys. **49**, 357–393 (1983)
10. A. Harten, Adaptive multiresolution schemes for shock computations. J. Comput. Phys. **115**, 319–338 (1994)
11. G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**, 202–228 (1996)
12. S.A. Lele, Compact finite difference schemes with spectral-like resolution. J. Comput. Phys. **103**(1), 16–42 (1992)
13. M.V. Papalexandris, A. Leonard, P.E. Dimotakis, Unsplit schemes for hyperbolic conservation laws with source terms in one space detonation. J. Comput. Phys. **134**, 31–61 (1997)
14. C.W. Shu, High order weighted essentially nonoscillatory schemes for convection dominated problems. SIAM Rev. **51**(1), 82–126 (2009)
15. O. Vasilyev, T. Lund, P. Moin, A general class of commutative filters for LES in complex geometries. J. Comput. Phys. **146**(1), 82–104 (1998)
16. C. Wang, X. Zhang, C.-W. Shu, J. Ning, Robust high order discontinuous Galerkin schemes for two-dimensional gaseous detonations. J. Comput. Phys. **231**, 653–665 (2012)

Higher Order Accurate Solutions for Flow in a Cavity: Experiences and Lessons Learned

Peter Eliasson, Marco Kupiainen, and Jan Nordström

Abstract Experiences from using a higher order accurate finite difference multi-block solver to compute the time dependent flow over a cavity is summarized. The work has been carried out as part of a work in a European project called IDIHOM in a collaboration between the Swedish Defense Research Agency (FOI) and University of Linköping (LiU). The higher order code is based on Summation By Parts operators combined with the Simultaneous Approximation Term approach for boundary and interface conditions. The spatial accuracy of the code is verified by calculations over a cyclinder by monitoring the decay of the errors of known wall quantities as the grid is refined. The focus is on the validation for a test case of transonic flow over a rectangular cavity with hybrid RANS/LES calculations. The results are compared to reference numerical results from a second order finite volume code as well as with experimental results with a good overall agreement between the results.

1 Introduction

The project IDIHOM (Industrialization of High-Order Methods—a Top-Down Approach) is a collaboration between 21 European partners from universities, research establishments and industries. The objective of the project is to industrialize the higher order methods and CFD codes to handle industrial type of applications. Within the scope of IDIHOM, FOI and LiU have jointly been further developing a higher order finite difference CFD code for multiblock structured grids. The current paper focuses on the validation of a calculation using a formally third

P. Eliasson (✉)
Department of Aeronautics and Autonomous Systems, FOI, Swedish Defense Research Agency, SE-164 90 Stockholm, Sweden
e-mail: peter.eliasson@foi.se

M. Kupiainen • J. Nordström
Department of Mathematics, Computational Mathematics, University of Linköping, SE-581 83 Linköping, Sweden

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_15

order accurate approach in space for calculations of the unsteady, turbulent transonic flow over a rectangular cavity with experimental comparisons.

The next section summarizes the computational approach including a short description of a tool that has been used to make reference calculations. Section 3 describes the verification for flow over a cylinder by a study of the decay of errors of known wall quantities as the grid is refined. The cavity flow calculations are then presented and in Sect. 4 we summarize and make concluding remarks.

2 Computational Tools

The higher order finite difference solver, called Essense, is based on Summation By Parts (SBP) operators combined with the Simultaneous Approximation Term (SAT) approach with penalty terms that guarantee accuracy and stability [1–5]. The code is able to handle arbitrary order of spatial accuracy, but is currently limited to fifth order. The code uses central difference operators for the approximation of the first derivative, $DU = P^{-1}QU$, where P is a block diagonal positive matrix containing the step size and where Q is an almost block skew symmetric difference matrix. The Navier-Stokes equations are transformed to a curvilinear coordinate system where the differentiation is carried out separately in each direction. The transformation contains metric first derivatives which are differentiated with the same technique. Second differences, as in the viscous terms, are computed by applying the first difference operator twice. Since the difference operator is central, artificial dissipation is added to stabilize the computations [6]. No shock capturing has been applied for the computations described here.

Weak boundary conditions are applied on all boundaries including wall boundary conditions, far-field boundary conditions and interface conditions between blocks. A common feature for all boundary and interface conditions is that they are enforced through penalty terms multiplying the difference of the unknown quantity and the corresponding prescribed data. The data, the size of the penalties and the number of boundary conditions depend of the specific boundary condition. Explicit time stepping with a fourth order accurate additive Runge-Kutta scheme is used to integrate the governing equations in time.

The parallel implementation utilizes domain decomposition for each block of the multiblock grid. Point-to-point communication is done by using so called halo-zones of half the width of the central SBP-operator. The communication across the different grids yields a possibly one-to-many and many-to-one communication pattern, which may have an adverse effect on load-balancing and scalability [7].

The higher order results for the cavity are compared to reference results from a formally second order accurate CFD solver, the Edge code, being an edge- and node-based Navier-Stokes flow solver applicable for both structured and unstructured grids [8–10]. Edge is based on a finite volume formulation where a median dual grid forms the control volumes with the unknowns allocated in the centers. The

governing equations are integrated with a multistage Runge-Kutta scheme to steady state and with acceleration by FAS agglomeration multigrid [11].

3 Computed Results

3.1 Verification for Flow Over a Cylinder

The computed results from the higher order code Essense have been verified for a number of test cases. Here we describe the verification for flow over a cylinder where the decay of errors of known wall quantities are studied as the grids are successively refined. The flow conditions are $M_\infty = 0.3$ and $Re = 50$ where the Reynolds number is based on the cylinder diameter. Five successively refined grids are used, the grids are of O-type structured grids where the coarsest grid contains 51×51 nodes. Two spatial operators denoted 42- and 84-operators are used where the first figure in the notation denotes the interior accuracy and the second the accuracy at the boundary. The two operators are formally third and fifth order accurate, respectively.

Figure 1 shows the decay of the errors of the wall velocity for a calculation with isothermal wall boundary conditions (left) and the errors of the normal wall temperature gradient using adiabatic boundary conditions (right). The errors follow more or less the expected design order; the decay of the errors is actually slightly higher than the design order. The results indicate the correct behavior and implementation of the numerical schemes.

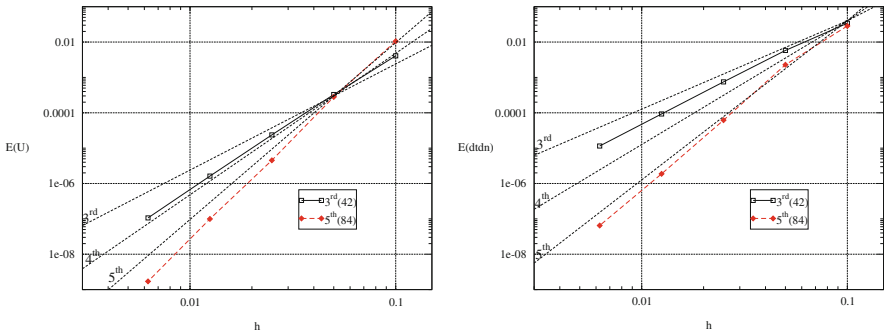
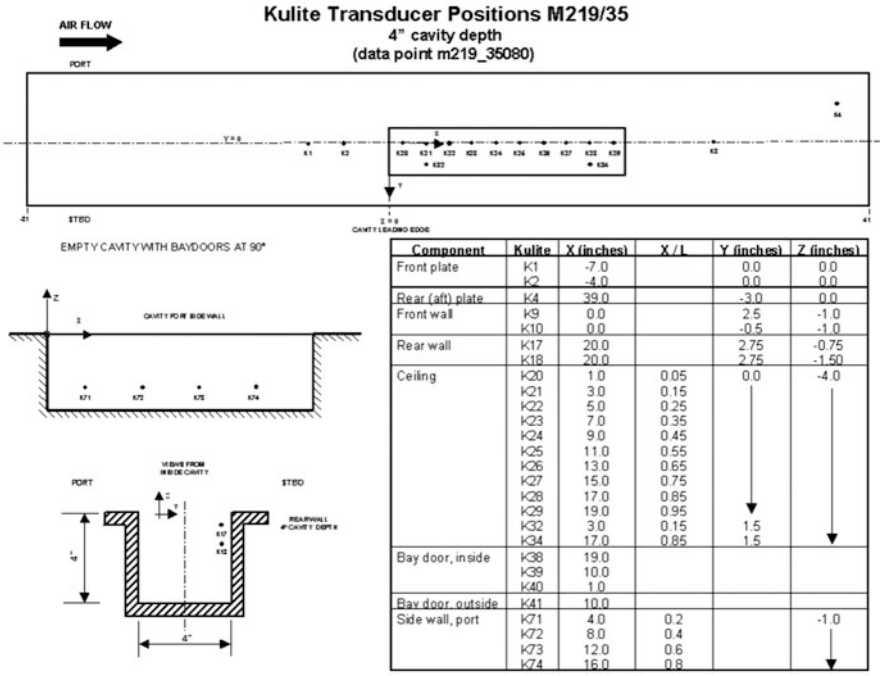


Fig. 1 Decay of errors of the velocity using isothermal wall boundary conditions (*left*); decay of temperature gradient using adiabatic wall boundary conditions (*right*)

3.2 Validation for Flow Over a Cavity

All partners within IDIHOM validated higher order results on different industrially relevant test cases. FOI and LiU chose a test case with transonic flow over a rectangular cavity, the test case is also denoted M219 in the literature [12]. The test case is suitable for Large Eddy Simulations (LES) or for hybrid RANS/LES calculations due to the turbulent fluctuations over the cavity. Experimental, time dependent data exist on the cavity walls and floor [12], computational results are available in many past references, e.g. [13, 14]. Several different cavity geometries exist; the one used here is the cavity with 5:1:1 length-to-depth-to-width relation. The geometry as well as the locations of the pressure probes are reproduced in Fig. 2.

The free stream values are $M_\infty = 0.85$ and $Re = 6.8 \times 10^6$ where the Reynolds number is based on the cavity length (20 in.). The cavity is experimentally measured on a device inside a wind tunnel. For the higher order calculations with Essense, calculations were carried out on a flat plate with the cavity embedded. A two block structured grid was created for these calculations where one block is located inside the cavity. To have a single boundary condition per block side, the block on top of the cavity block was split up resulting in ten blocks all together.



Component	Kulite	X (inches)	X / L	Y (inches)	Z (inches)
Front plate	K1	-7.0		0.0	0.0
	K2	-4.0		0.0	0.0
Rear (aft) plate	K4	39.0		-3.0	0.0
Front wall	K9	0.0		2.5	-1.0
	K10	0.0		-0.5	-1.0
Rear wall	K17	20.0		2.75	-0.75
	K18	20.0		2.75	-1.50
Ceiling	K20	1.0	0.05	0.0	-4.0
	K21	3.0	0.15		
	K22	5.0	0.25		
	K23	7.0	0.35		
	K24	9.0	0.45		
	K25	11.0	0.55		
	K26	13.0	0.65		
	K27	15.0	0.75		
	K28	17.0	0.85		
	K29	19.0	0.95		
	K32	3.0	0.15	1.5	
	K34	17.0	0.85	1.5	
Bay door, inside	K38	19.0			
	K39	10.0			
	K40	1.0			
Bay door, outside	K41	10.0			
Side wall, port	K71	4.0	0.2		-1.0
	K72	8.0	0.4		
	K73	12.0	0.6		
	K74	16.0	0.8		

Fig. 2 Cavity geometry, experimental setup and location of the pressure probes recording unsteady pressure fluctuations [12]

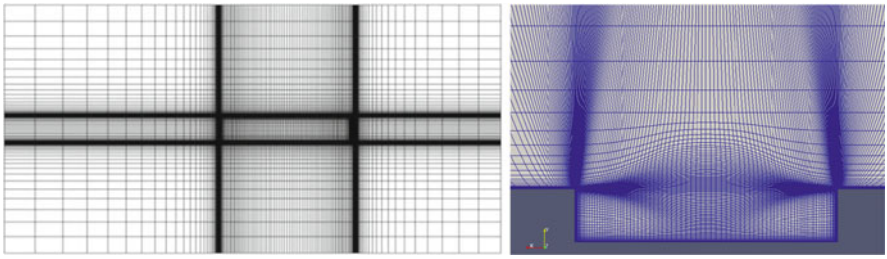


Fig. 3 Mesh pictures of the cavity test case used for Essense

Table 1 Details of the computational grids for the cavity

Grid	Solver	# vol. nodes	# boundary nodes in cavity	# volume nodes in cavity	Near wall distance (m)
Structured	Essense	2.6×10^6	41×10^3	0.73×10^6	1.2×10^{-5}
Unstructured	Edge	6.2×10^6	77×10^3	2.0×10^6	4.0×10^{-6}

The computational grid is depicted in Fig. 3. The stretching of the grid near the boundaries is relaxed in the interior to have a more uniform resolution which causes the grid to become curvilinear. For the reference calculations with Edge [15], the calculations were carried out on a hybrid unstructured grid that has been generated by EADS (European Aeronautic Defence and Space Company) and contains a grid over the cavity, the device on which the cavity is integrated and the entire test section of the wind tunnel. The main data of the two grids are displayed in Table 1. The grid for the higher order calculations has fewer grid points than that for the reference calculations with Edge.

The computational grids are designed to carry out hybrid RANS/LES calculations with RANS in the near wall region and LES off wall in the cavity. Both calculations were initiated from poorly converged steady state calculations with local time steps. Only one higher order calculation was performed using a third order accurate calculation (42-operator), no model was used for modeling the turbulence. Adiabatic weak wall boundary conditions were used inside the cavity and on the plate. Far-field boundary conditions were used elsewhere. The higher order calculation use explicit time stepping and progress for about 60 through flows (L/U_∞ where L is the cavity length and U_∞ the free stream velocity), the solutions from last 40 through flows are used for the statistics.

The reference calculations use the second order implicit backward difference method and dual time stepping in each time step. An algebraic RANS/LES model is used to model the turbulence [15]. The calculations progress for about 120 through flows with statistics from the last 80 through flows. Some computational parameters are given in Table 2.

In Fig. 4 the sound pressure level (SPL) at two locations on the cavity floor are presented. The overall sound pressure level (OASPL) is displayed and compared to experimental values in Fig. 5, the OASPL is obtained by integrating SPL for all

Table 2 Sizes of time steps and number of inner iterations for the reference cavity calculations

Grid	Solver	Δt	$\Delta t/T$	N inner iterations
Structured	Essense	1.0×10^{-8}	182×10^{3}	1
Unstructured	Edge	2.0×10^{-5}	91	32

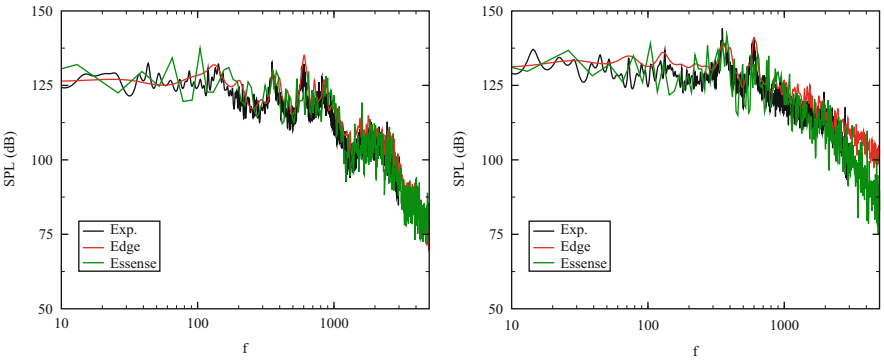


Fig. 4 Sound pressure level at cavity floor at kulits k21, $x/L = 15\,\%$ (*left*); k25, $x/L = 55\,\%$ (*right*)

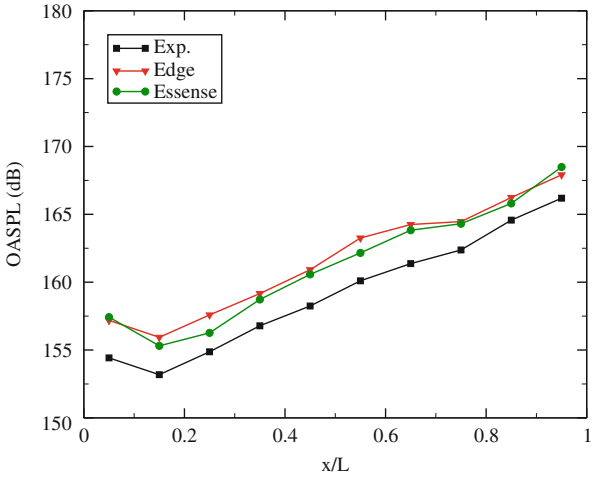


Fig. 5 Overall sound pressure level at cavity floor

frequencies. The higher order results compare reasonably well to the experimental values of SPL. The main tonal peaks are captured. The higher order results have a tendency to have somewhat larger amplitudes of the oscillations compared to the reference results. This may be due to the lack of RANS/LES model for the higher order calculations or possibly a too short and coarse sampling interval. The computed OASPL from the higher order scheme and from the reference

calculations with the unstructured grid agree well with the experimental values, an over prediction of OASPL is common.

In an attempt to quantify the deviation from the experimental OASPL and to make a cross comparison between different results we defined the normalized deviation D as

$$D = \frac{\| O_{CFD} - O_{exp} - \overline{\delta O} \|_2}{\overline{O}_{exp}} = \frac{\sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_{i,CFD} - O_{i,exp} - \overline{\delta O})^2}}{\overline{O}_{exp}} \quad (1)$$

where $\overline{O}_{exp} = 159.2\,\text{dB}$, $N = 10$. The intention with the derived formula for the deviation is to define a measure that gives a zero value if the shape of OASPL is identical to the shape of the experimental OASPL. It allows for a shift in absolute level though. As it turns out the deviation obtained with the higher order scheme is the same as that obtained with Edge indicating that the two solutions follow experimental OASPL equally well.

4 Summary and Conclusions

A higher order, provable stable, finite difference solver has been verified for a 2D cylinder flow case for which design order accuracy was obtained. The higher order solver has been applied to an industrially relevant case, the transonic flow over a 3D rectangular cavity at a high Reynolds number for which third order accurate time dependent solution were obtained. The quality of the solutions was good in terms of SPL and OASPL indicating that the higher order solution can be of use for industrial applications. The next phase in the development of Essense includes steady state convergence acceleration and implicit time integration.

Acknowledgements This work has been carried out within the EU project IDIHOM under contract No. FP7-AAT-2010-RTD-1-2657808.

References

1. J. Nordström, S. Eriksson, P. Eliasson, Weak and strong wall boundary procedures and convergence to steady-state of the Navier–Stokes equations. J. Comput. Phys. **231**, 4867–4884 (2012)
2. J. Berg, J. Nordström, Superconvergent functional output for time-dependent problems using finite differences on summation-by-parts form. J. Comput. Phys. **231**, 6846–6860 (2012)
3. J. Berg, J. Nordström, On the impact of boundary conditions on dual consistent finite difference discretizations. J. Comput. Phys. **236**, 41–55 (2013)
4. M.H. Carpenter, J. Nordström, D. Gottlieb, A stable and conservative interface treatment of arbitrary spatial accuracy. J. Comput. Phys. **148**(2), 341–365 (1999)

5. J. Berg, J. Nordström, Stable Robin solid wall boundary conditions for the Navier-Stokes equations. J. Comput. Phys. **230**, 7519–7532 (2011)
6. K. Mattsson, S. Svärd, J. Nordström, Stable and accurate artificial dissipation. J. Sci. Comput. **21**(1), 57–79 (2004)
7. J. Rantakokko, Partitioning strategies for structured multiblock grids. Parallel Comput. **26**, 1661–1680 (2000)
8. P. Eliasson, P. Weinerfelt, Recent applications of the flow solver edge, in *Proceedings to 7th Asian CFD Conference*, Bangalore (2007)
9. P. Eliasson, J. Nordström, P. Weinerfelt, Application of a line-implicit scheme on stretched unstructured grids. AIAA Paper 2009–163 (2009)
10. P. Eliasson, S. Eriksson, J. Nordström, The influence of weak and strong solid wall boundary conditions on the convergence to steady-state of the Navier-Stokes equations. AIAA Paper 2009–3551 (2009)
11. A. Brandt, O.E. Livne, Multigrid techniques, 1984 guide with applications to fluid dynamics, in *Applied Mathematics*. SIAM, vol. 67 (1984). ISBN-13: 978-1611970746
12. X. Chen, N.D. Sandham, X. Zhang, Cavity Flow Noise Predictions. Report No. AFM-07/05, School of Engineering Sciences, University of Southampton (2007)
13. R.M. Ashworth, Prediction of acoustic resonance phenomena for weapon bays using detached eddy simulation. Aeronaut. J. **109**(1102), 631–638 (2005)
14. S.-H. Peng, Simulation of flow past a rectangular open cavity using DES and unsteady RANS. AIAA Paper 2006–2827 (2006)
15. S.-H. Peng, Hybrid RANS-LES modelling based on zero- and one-equation models for turbulent flow simulation, in *Proceedings of 4th Internal Symposium on Turbulent and Shear Flow Phenomena*, vol. 3, pp. 1159–1164, 2005

On the Solution of the Elliptic Interface Problems by Difference Potentials Method

Yekaterina Epshteyn and Michael Medvinsky

Abstract Designing numerical methods with high-order accuracy for problems in irregular domains and/or with interfaces is crucial for the accurate solution of many problems with physical and biological applications. The major challenge here is to design an efficient and accurate numerical method that can capture certain properties of analytical solutions in different domains/subdomains while handling arbitrary geometries and complex structures of the domains. Moreover, in general, any standard method (finite-difference, finite-element, etc.) will fail to produce accurate solutions to interface problems due to discontinuities in the model's parameters/solutions. In this work, we consider Difference Potentials Method (DPM) as an efficient and accurate solver for the variable coefficient elliptic interface problems.

1 Introduction

In this paper, we consider Difference Potentials Method (DPM) as an efficient and accurate solver for variable coefficient elliptic interface problems. DPM can be understood as the discrete version of the method of generalized Calderon's potentials and Calderon's boundary equations with projections in the theory of partial differential equations (PDEs). DPM introduces a computationally simple auxiliary domain. The original domain of the problem is embedded into an auxiliary domain, and the auxiliary domain is discretized using simple structured grids, e.g. Cartesian grids. After that, the main idea of DPM is to define a Difference Potentials operator, and to reformulate the original discretized PDEs (without imposed boundary/interface conditions yet) as equivalent discrete generalized Calderon's boundary equations with projections (BEP). These BEP are supplemented by the given boundary/interface conditions (the resulting BEP are always well-posed, as long as the original problem is well-posed), and solved to obtain the values of the solution at the points near the continuous boundary of the original domain

Y. Epshteyn (✉) • M. Medvinsky
Department of Mathematics, The University Of Utah, Salt Lake City, UT, USA
e-mail: epshteyn@math.utah.edu; mmedvin@math.utah.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_16

(at the points of the discrete grid boundary which approximates the continuous boundary from the inside and outside of the domain). Using the obtained values of the solution at the discrete grid boundary, the approximation to the solution in the original domain is constructed through the discrete generalized Green's formula. *DPM offers geometric flexibility (without the use of unstructured meshes or "body-fitted" meshes), but does not require explicit knowledge of the fundamental solution, is not limited to constant coefficient problems or linear problems, does not involve singular integrals, and can handle general boundary and/or interface conditions.* The reader can consult [14–16] for a detailed theoretical study of the methods based on Difference Potentials, and ([1, 4, 5, 7, 8, 11–13, 16–21], etc.) for the recent developments and applications of DPM.

In this paper, we extend the work on DPM for the elliptic interface problems started in [7, 19, 20] to variable coefficient elliptic interface models in 2D. A more detailed presentation of DPM for elliptic (and parabolic interface problems) in 2D with different high-order accurate discretizations, as well as the analysis of DPM for the interface problems will be part of the future publications [2, 6].

The paper is organized as follows. In Sect. 2, we introduce the formulation of the problem. Next, in Sect. 2.1 we briefly describe the main building blocks of the DPM. Finally, we illustrate the performance of the proposed DPM, as well as compare DPM with the Mayo's method [3, 10] and the Immersed Interface Method (IIM) [3, 9] in several challenging numerical experiments (performed by M. Medvinsky) in Sect. 2.2.

2 Elliptic Interface Problem

In this work we consider interface/composite domain problem defined in some bounded domain $D^0 \subset \mathbb{R}^2$:

$$L_D u = \begin{cases} L_1 u_{D_1} = f_1(x,y) & (x,y) \in D_1 \\ L_2 u_{D_2} = f_2(x,y) & (x,y) \in D_2 \end{cases} \tag{1}$$

subject to the appropriate interface conditions:

$$u_{\overline{D}_1}\big|_\Gamma - u_{\overline{D}_2}\big|_\Gamma = \phi_1(x,y), \quad \left.\frac{\partial u_{\overline{D}_1}}{\partial n}\right|_\Gamma - \left.\frac{\partial u_{\overline{D}_2}}{\partial n}\right|_\Gamma = \phi_2(x,y) \tag{2}$$

and boundary conditions

$$u|_{\partial D} = \psi(x,y) \tag{3}$$

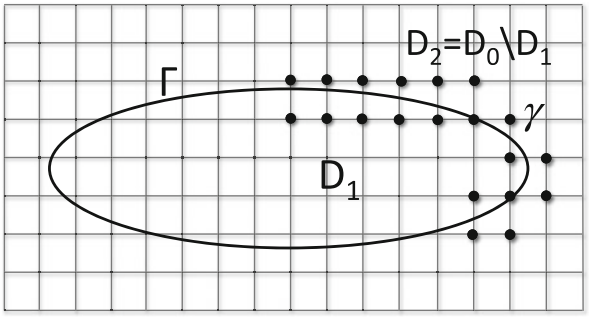


Fig. 1 Example of an auxiliary domain D^0, original domains D_1 and D_2 separated by the interface Γ, and the example of the points in the discrete grid boundary set γ for the 5-point stencil of the second-order method. Auxiliary domain D^0 coincides with D here

where $D_1 \cup D_2 = D$ and $D \subset D^0$, see Fig. 1. Here, we assume L_s , $s \in \{1, 2\}$ are the second-order linear elliptic differential operators of the form

$$L_s u_{D_s} \equiv \frac{\partial}{\partial x}\Big(a_s(x,y)\frac{\partial u_{D_s}}{\partial x}\Big) + \frac{\partial}{\partial y}\Big(b_s(x,y)\frac{\partial u_{D_s}}{\partial y}\Big), \quad s \in \{1,2\}.$$

The functions $a_s(x, y) \geq 1$ and $b_s(x, y) \geq 1$ are sufficiently smooth and defined in a larger auxiliary subdomains $D_s \subset D_s^0$. The functions $f_s(x, y)$ are sufficiently smooth functions defined in each subdomain D_s^0. We assume that the continuous problem (1)–(3) is well-posed. *Moreover, we assume that the operators L_s are well-defined on some larger auxiliary domain D_s^0. More precisely, we assume that for any sufficiently smooth functions $f_s(x, y)$ the equations $L_s u_{D_s^0} = f_s(x, y)$ have a unique solution $u_{D_s^0}$ on D_s^0 that satisfy the given boundary conditions on ∂D_s^0.*
Note, here and below, the upper/or lower index $s \in \{1, 2\}$ is introduced to distinguish between the subdomains.

2.1 Difference Potentials Method for Interface/Composite Domain Problems

Here we discuss the development of high-order methods based on Difference Potentials approach for the elliptic interface/composite domain problem (1)–(3). Below, we only briefly discuss main ideas of DPM for interface problems. The reader can consult [1, 7, 16, 19, 20] and future publications [2, 6] for more details. Also, the reader can consult [16] for the detailed discussion on the general theory and numerical analysis of DPM. Let us briefly describe the main steps of the algorithm.

Introduction of the Auxiliary Domain Place the original domains D_s, $s \in \{1, 2\}$ in the auxiliary computationally simple domains $D_s^0 \subset \mathbb{R}^2$ that we will choose to be squares. Next, introduce a Cartesian mesh for each D_s^0, with points $x_j^s = j\Delta x^s, y_k^s = k\Delta y^s, (k, j = 0, \pm 1, \ldots)$. Let us assume for simplicity that $\Delta x^s = \Delta y^s := h^s$.

Select discretization of the continuous model (1), for example here we will consider a finite-difference approximation. Next, define a finite-difference stencil $N^s_{j,k}$ with its center placed at (x^s_j, y^s_k) (like a 5 node "dimension by dimension stencil" for the second-order scheme, or a 9 node "dimension by dimension stencil" for the classical fourth-order scheme, etc.). Additionally, introduce the point sets M^0_s (the set of all the mesh nodes (x^s_j, y^s_k) that belong to the interior of the auxiliary domain D^0_s), $M^+_s := M^0_s \cap D_s$ (the set of all the mesh nodes (x^s_j, y^s_k) that belong to the interior of the original domain D_s), and by $M^-_s := M^0_s \backslash M^+_s$ (the set of all the mesh nodes (x^s_j, y^s_k) that are inside of the auxiliary domain D^0_s but don't belong to the interior of the original domain D_s). Define $N^+_s := \{\bigcup_{j,k} N^s_{j,k} | (x^s_j, y^s_k) \in M^+_s\}$ (the set of all points covered by the stencil $N^s_{j,k}$ when center point (x^s_j, y^s_k) of the stencil goes through all the points of the set $M^+_s \subset D_s$). Similarly, define $N^-_s := \{\bigcup_{j,k} N^s_{j,k} | (x_j, y_k) \in M^-_s\}$ (the set of all points covered by the stencil $N^s_{j,k}$ when center point (x^s_j, y^s_k) of the stencil goes through all the points of the set M^-_s).

Introduce $\gamma_s := N^+_s \cap N^-_s$. The set γ_s is called the *discrete grid boundary*. The mesh nodes from set γ_s straddle the boundary ∂D_s. $N^0_s := \{\bigcup_{j,k} N^s_{j,k} | (x^s_j, y^s_k) \in M^0_s\} \subset \overline{D^0_s}$. The sets N^0_s, M^0_s, N^+_s, N^-_s, M^+_s, M^-_s, γ_s will be used to develop the method based on the Difference Potentials approach, Fig. 1.

Difference Equations The discrete reformulation of the model problem (1) in each auxiliary domain D^0_s is: solve for $u^s_{j,k} \in N^+_s$

$$L^s_h[u^s_{j,k}] = F^s_{j,k}, \quad (x^s_j, y^s_k) \in M^+_s \tag{4}$$

where $L^s_h[u^s_{j,k}]$ is the discrete linear elliptic operator obtained using finite-difference approximation of order r (for example, the second-order $r = 2$ or the fourth-order $r = 4$, etc.). $F^s_{j,k}$ denotes the discrete right-hand side. The unknowns are $u^s_{j,k} :\approx u_{D_s}(x^s_j, y^s_k)$, where (x^s_j, y^s_k) is a mesh point of the Cartesian grid.

We need to complete the linear system of difference equations (4) with the appropriate choice of the numerical boundary and interface conditions to construct a unique accurate approximation of the continuous problem (1)–(3) in domain D. Thus, to design an efficient algorithm for any type of boundary and interface conditions, we will consider a numerical method based on the idea of the Difference Potentials.

Step 1: Construction of a Particular Solution: Denote by $u^s_{j,k} := G^h_s F^s_{j,k}$, $u^s_{j,k} \in N^+_s$ the particular solution of the discrete problem (4), which we will construct as the solution (restricted to set N^+_s) of the simple auxiliary problem (AP) of the following form:

$$L^s_h[u^s_{j,k}] = \begin{cases} F^s_{j,k}, & (x^s_j, y^s_k) \in M^+_s, \\ 0, & (x^s_j, y^s_k) \in M^-_s, \end{cases} \tag{5}$$

$$u^s_{j,k} = 0, \quad (x^s_j, y^s_k) \in N^0_s \backslash M^0_s \tag{6}$$

Step 2: Difference Potentials and Construction of the BEP: We now introduce a linear space $\mathbf{V}_{\gamma_s}$ of all the grid functions denoted by v_{γ_s} defined on γ_s [7, 16, 19, 20], etc. We will extend the value v_{γ_s} by zero to other points of the grid N_s^0.

Definition 1 The Difference Potential with any given density $v_{\gamma_s} \in \mathbf{V}_{\gamma_s}$ is the grid function $u_{j,k}^s := \mathbf{P}_{N^+\gamma_s} v_{\gamma_s}$, defined on N_s^+, and coincides on N_s^+ with the solution $u_{j,k}^s$ of the simple auxiliary problem (AP) of the following form:

$$L_h^s[u_{j,k}^s] = \begin{cases} 0, & (x_j^s, y_k^s) \in M_s^+, \\ L_h^s[v_{\gamma_s}], & (x_j^s, y_k^s) \in M_s^-, \end{cases} \tag{7}$$

$$u_{j,k}^s = 0, \quad (x_j^s, y_k^s) \in N_s^0 \backslash M_s^0 \tag{8}$$

Here, $\mathbf{P}_{N^+\gamma_s}$ denotes the operator which constructs the Difference Potential $u_{j,k}^s = \mathbf{P}_{N^+\gamma_s} v_{\gamma_s}$ from the given density $v_{\gamma_s} \in V_{\gamma_s}$. The operator $\mathbf{P}_{N^+\gamma_s}$ is the linear operator of the density v_{γ_s}. Hence, it can be easily constructed [7, 19, 20]. We will now state the most important theorem of the method:

Theorem 1 *Density u_{γ_s} is the trace of some solution $u_{j,k}^s \in N_s^+$ to the Difference Equations (4): $u_{\gamma_s} \equiv Tr_{\gamma_s} u_{j,k}^s$, if and only if, u_{γ_s} satisfies Generalized Calderon's Boundary Equations with Projections (BEP)*

$$u_{\gamma_s} - \mathbf{P}_{\gamma_s} u_{\gamma_s} = G_s^h F_{\gamma_s}, \tag{9}$$

where $G_s^h F_{\gamma_s} := Tr_{\gamma_s}(G_s^h F_{j,k}^s)$ is the trace (or restriction) of the particular solution $G_s^h F_{j,k}^s \in N_s^+$ constructed in (5)–(6) on the grid boundary γ_s, and $\mathbf{P}_{\gamma_s} u_{\gamma_s} := Tr_{\gamma_s}(\mathbf{P}_{N^+\gamma_s} u_{\gamma_s})$ is the trace of the Difference Potential $\mathbf{P}_{N^+\gamma_s} u_{\gamma_s} \in N_s^+$ in (7)–(8) on the grid boundary γ_s.

Remark The BEP (9) are constructed for each subdomain and solved efficiently together with the boundary and interface conditions for the unknown densities u_{γ_s} using the idea of the extension operator for u_{γ_s}, and the spectral approach for the approximation of the Cauchy data $(u_{D_s}, \frac{\partial u_{D_s}}{\partial n})|_{\partial D_s}$ ([12, 19, 20], etc.).

Step 3: Construction of the Approximate Solution to the Model Problem (1)–(3) from the density u_{γ_s} obtained in Step 2:

Statement 1 (Generalized Green's Formula) *The discrete solution $u_{j,k}^s := \mathbf{P}_{N^+\gamma_s} u_{\gamma_s} + G_s^h F_{j,k}^s$ is the approximation to the solution $u_{j,k}^s \approx u_{D_s}(x_j^s, y_k^s)$, $(x_j^s, y_k^s) \in N_s^+ \cap D_s$ of the continuous problem (1)–(3) (see [14–16] for a general theory of DPM and [1, 6, 7, 19, 20]).*

The expected accuracy of the proposed method for domains with the smooth boundaries and under sufficient regularity of the exact solutions will be $O(h^{r-\varepsilon})$ in the discrete Hölder norm of order $2+\varepsilon$ (if the continuous second-order linear elliptic operator L is approximated with rth order of accuracy by the discrete operator L_h,

and the extension operator for u_{γ_s} is constructed with sufficient accuracy), see [14–16], [1, 6, 7, 19, 20] and Sect. 2.2. Here, ε is an arbitrary number with $0 < \varepsilon < 1$.

2.2 Numerical Examples

In the numerical examples below, we consider a second-order centered finite-difference approximation (with 5-node stencil) as the underlying discretization for DPM. The numerical experiments for the fourth-order approximation will be presented in future publication [6]. The first test problem that we present here is the problem from the paper [3]:

$$\Delta u_{D_s} = f_s(x, y), \quad (x, y) \in D_s, \quad s \in \{1, 2\} \tag{10}$$

where the interface between two subdomains D_1 and D_2 (see Fig. 1) is given by an ellipse with semi-axes $(a, b) = (0.9, 0.1)$, and the curvature is $\kappa = -90$ at $(\pm a, 0)$ which leads to a quite challenging tests [3]. The exact solution here is

$$u_1 = \sin x \cos y, \quad u_2 = 0, \tag{11}$$

which is discontinuous at the interface. The results for the test problem (10)–(11) are presented in Table 1, which shows the relative error in the maximum norm of the solution and its derivatives. To match the settings of the numerical experiments in paper [3], we consider auxiliary domains (here and below) $D_1^0 = D_2^0 \equiv D = [-1.1, 1.1] \times [-1.1, 1.1]$ for the subdomains D_1 and D_2 respectively, Fig. 1. Note, that in these settings, $h^1 = h^2 = h$ (however, DPM handles as easily different auxiliary problems/non-matching meshes [1, 5, 7, 19, 20]). As observed from the Table 1 here, and from the Table 1 (bottom), on page 111 in paper [3], the accuracy in the solution for the test problem (10)–(11) obtained by DPM is very close to the accuracy obtained by Mayo's Method and by IIM. But, the accuracy in the derivatives of the solution obtained by DPM is superior to the accuracy obtained by Mayo's Method or IIM.

Table 1 Test problem (10)–(11) with $a = 0.9$, $b = 0.1$ from paper [3]

N	L_∞-error in u	Rate	L_∞-error in u_x	Rate	L_∞-error in u_y	Rate
40	$1.7474e-06$		$1.0559e-06$		$1.0041e-06$	
80	$5.2910e-07$	1.72	$1.7733e-07$	2.57	$1.6081e-07$	2.64
160	$1.2986e-07$	2.03	$2.5886e-08$	2.78	$2.1461e-08$	2.91
320	$3.1742e-08$	2.03	$1.7307e-09$	3.90	$1.3500e-09$	3.99
640	$7.8701e-09$	2.01	$2.0067e-10$	3.11	$1.3030e-10$	3.37

Here N corresponds to half of the number of subintervals (the same number of subintervals in x and y-direction), similarly to the results in Table 1 (bottom), page 111 in [3]. Relative L_∞ error in the solution and in its derivatives

Table 2 Test problem (10), (12) with $a = 0.9$, $b = 0.1$ from paper [3]

N	L_∞-error in u	Rate	L_∞-error in u_x	Rate	L_∞-error in u_y	Rate
40	$1.0000e+00$		$8.3442e-01$		$1.0000e+00$	
80	$2.6622e-01$	1.91	$2.2263e-01$	1.91	$3.3108e-01$	1.59
160	$3.8645e-02$	2.78	$2.2076e-02$	3.33	$5.0801e-02$	2.70
320	$9.0971e-03$	2.09	$2.7015e-03$	3.03	$7.7708e-03$	2.71
640	$2.3838e-03$	1.93	$3.3376e-04$	3.02	$1.0421e-03$	2.90

Here N corresponds to half of the number of subintervals (the same number of subintervals in x and y-direction), similarly to the results in Table 3, page 113 in [3]. Relative L_∞ error in the solution and its derivatives

The second test problem is again from [3] and has the same settings as the first test problem (10)–(11), but now the exact solution is defined as:

$$u_1 = x^9 y^8, \quad u_2 = 0. \tag{12}$$

The results for this test problem are presented in Table 2. DPM errors for this test problem (10), (12) are again close to the errors for Mayo's method and IIM, reported in Table 3, page 113 in [3]. As the last and more challenging test problem, we consider the interface problem with variable coefficients as described below:

$$\frac{\partial}{\partial x}\left(a_s(x,y)\frac{\partial u_{D_s}}{\partial x}\right) + \frac{\partial}{\partial y}\left(b_s(x,y)\frac{\partial u_{D_s}}{\partial y}\right) = f_s(x,y), \quad (x,y) \in D_s, \quad s \in \{1,2\} \tag{13}$$

where $a_1 = (3 + 0.5\sin(2x + y))$ $b_1 = (2 + 0.5\cos(4x + 3y))$ and $a_2 = b_2 = 10^6$. The interface curve for this problem is again given by the ellipse with semi-axes $(a, b) = (0.9, 0.1)$. The exact solution for this test problem (13) is set to

$$u_1 = \sin(y^2 x)\sin(x^3 y), \quad u_2 = \sin(2x)\sin(3y). \tag{14}$$

The interface problem (13)–(14) is much more challenging than the previous test problems since it has discontinuous solution at the interface, as well as a large jump ratio between diffusion coefficients in subdomains D_1 and D_2, Fig. 2. The results for this test problem are presented in Table 3, which shows the relative error of the solution and its derivatives in the maximum norm. As in the previous numerical examples, DPM preserves overall second-order (and even slightly better in the derivative) accuracy in the solution and its derivatives. The observed numerically in Tables 1, 2, and 3 slightly higher order of accuracy in the derivatives could be due to the specifics of the considered test problems and the properties of the extension operator for u_{γ_s}.

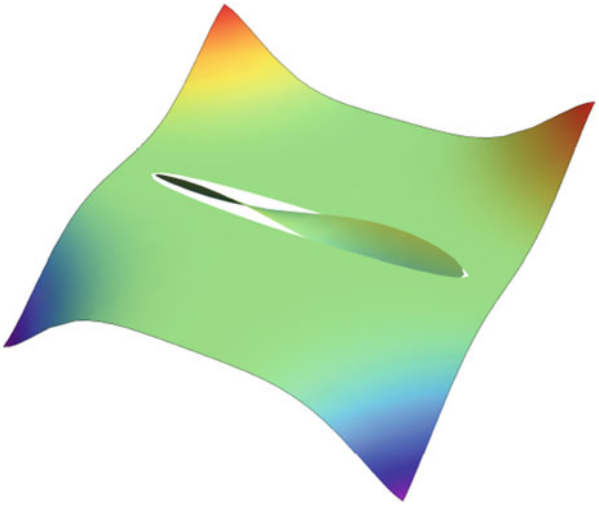


Fig. 2 Exact solution to the test problem (13)–(14)

Table 3 Test problem (13)–(14) with $a = 0.9$, $b = 0.1$

N	L_∞-error in u	Rate	L_∞-error in u_x	Rate	L_∞-error in u_y	Rate
40	$4.5671e-04$		$1.3639e-04$		$1.3981e-03$	
80	$1.1520e-04$	1.99	$2.2087e-05$	2.63	$3.1356e-04$	2.16
160	$2.8329e-05$	2.02	$2.3138e-06$	3.25	$3.5176e-05$	3.16
320	$7.0319e-06$	2.01	$3.1931e-07$	2.86	$4.6670e-06$	2.91
640	$1.7578e-06$	2.00	$4.9421e-08$	2.69	$7.2111e-07$	2.69

Here N corresponds to half of the number of subintervals (the same number of subintervals in x and y-direction), similarly to previous examples. Relative L_∞ error in the solution and its derivatives

Acknowledgements We are grateful to Jason Albright and Kyle R. Steffen for the comments that helped to improve the manuscript. The research of Yekaterina Epshteyn and Michael Medvinsky is supported in part by the National Science Foundation Grant # DMS-1112984.

References

1. J. Albright, Y. Epshteyn, K.R. Steffen, High-order accurate difference potentials methods for parabolic problems. Appl. Numer. Math. **93**, 87–106 (2015). http://dx.doi.org/10.1016/j.apnum.2014.08.002
2. J. Albright, Y. Epshteyn, Q. Xia, High-order difference potentials methods for 2D parabolic interface problems (September 2015, in preparation)
3. J.T. Beale, A.T. Layton, On the accuracy of finite difference methods for elliptic problems with interfaces. Commun. Appl. Math. Comput. Sci. **1**, 91–119 (electronic) (2006)
4. Y. Epshteyn, Upwind-difference potentials method for Patlak-Keller-Segel chemotaxis model. J. Sci. Comput. **53**(3), 689–713 (2012)

5. Y. Epshteyn, Algorithms composition approach based on difference potentials method for parabolic problems. Commun. Math. Sci. **12**(4), 723–755 (2014)
6. J. Albright, Y. Epshteyn, M. Medvinsky, Q. Xia, High-order numerical schemes based on difference potentials for 2D elliptic problems with material interfaces (2015, submitted for publication)
7. Y. Epshteyn, S. Phippen, High-order difference potentials methods for 1D elliptic type models. Appl. Numer. Math. **93**, 69–86 (2015). http://dx.doi.org/10.1016/j.apnum.2014.02.005
8. E. Kansa, U. Shumlak, S. Tsynkov, Discrete Calderon's projections on parallelepipeds and their application to computing exterior magnetic fields for FRC plasmas. J. Comput. Phys. **234**, 172–198 (2013). http://dx.doi.org/10.1016/j.jcp.2012.09.033
9. R.J. LeVeque, Z.L. Li, The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. SIAM J. Numer. Anal. **31**(4), 1019–1044 (1994)
10. A. Mayo, The fast solution of Poisson's and the biharmonic equations on irregular regions. SIAM J. Numer. Anal. **21**(2), 285–299 (1984)
11. M. Medvinsky, High order numerical simulation of waves using regular grids and non-conforming interfaces. Ph.D. Dissertation, Tel Aviv University (2013)
12. M. Medvinsky, S. Tsynkov, E. Turkel, The method of difference potentials for the Helmholtz equation using compact high order schemes. J. Sci. Comput. **53**(1), 150–193 (2012)
13. M. Medvinsky, S. Tsynkov, E. Turkel, High order numerical simulation of the transmission and scattering of waves using the method of difference potentials. J. Comput. Phys. **243**, 305–322 (2013)
14. A.A. Reznik, Approximation of surface potentials of elliptic operators by difference potentials. Dokl. Akad. Nauk SSSR **263**(6), 1318–1321 (1982)
15. A.A. Reznik, Approximation of surface potentials of elliptic operators by difference potentials and solution of boundary value problems. Ph.D, Moscow, MPTI (1983)
16. V.S. Ryaben'kii, *Method of Difference Potentials and Its Applications*. Springer Series in Computational Mathematics, vol. 30 (Springer, Berlin, 2002)
17. V.S. Ryaben'kiĭ, Difference potentials analogous to Cauchy integrals. Usp. Mat. Nauk **67**(3(405)), 147–172 (2012)
18. V.S. Ryaben'kii, S. Utyuzhnikov, An algorithm of the method of difference potentials for domains with cuts. Appl. Numer. Math. **93**, 254–261 (2015)
19. V.S. Ryaben'kii, V.I. Turchaninov, Ye.Yu. Epshteyn, The numerical example of algorithms composition for solution of the boundary-value problems on compound domain based on difference potential method. Keldysh Institute for Applied Mathematics, Russia Academy of Sciences, Moscow, No. 3 (2003). http://library.keldysh.ru/preprint.asp?lg=&id=2003-3
20. V.S. Ryaben′kiĭ, V.I. Turchaninov, E.Yu. Èpshteĭn. An algorithm composition scheme for problems in composite domains based on the method of difference potentials. Zh. Vychisl. Mat. Mat. Fiz. **46**(10), 1853–1870 (2006)
21. S.V. Utyuzhnikov, Nonlinear problem of active sound control. J. Comput. Appl. Math. **234**(1), 215–223 (2010)

Generalized Summation by Parts Operators: Second Derivative and Time-Marching Methods

David C. Del Rey Fernández, Pieter D. Boom, and David W. Zingg

Abstract This paper describes extensions of the generalized summation-by-parts (GSBP) framework to the approximation of the second derivative with a variable coefficient and to time integration. GSBP operators for the second derivative lead to more efficient discretizations, relative to the classical finite-difference SBP approach, as they can require fewer nodes for a given order of accuracy. Similarly, for time integration, time-marching methods based on GSBP operators can be more efficient than those based on classical SBP operators, as they minimize the number of solution points which must be solved simultaneously. Furthermore, we demonstrate the link between GSBP operators and Runge-Kutta time-marching methods.

1 Introduction

In this paper, we present an overview of generalized summation-by-parts (GSBP) operators [7] for the approximation of the second derivative with a variable coefficient and as time integration methods. Further details can be found in [2, 6]. The benefit of the GSBP approach is that it broadens the applicability of the SBP approach to a wider class of operators and provides a straightforward methodology to construct novel operators with the summation-by-parts (SBP) property. This enables the use of simultaneous approximation terms (SATs) for the weak imposition of initial and boundary conditions and inter-element/block coupling, leading to schemes that are provably consistent, conservative, and stable.

The GSBP framework extends the definition of SBP operators given by Kreiss and Scherer [13] to those that have a combination of (1) no repeating interior point operator, (2) nonuniform nodal distributions, and (3) operators that do not include one or both boundary nodes. The GSBP framework leads to operators

D.C. Del Rey Fernández (✉) • P.D. Boom • D.W. Zingg
University of Toronto Institute for Aerospace Studies, 4925 Dufferin St., Toronto, ON, Canada M3H 5T6
e-mail: dcdelrey@gmail.com; pieter.boom@mail.utoronto.ca; dwz@oddjob.utias.utoronto.ca

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_17

that approximate the first derivative and that mimic the integration-by-parts (IBP) property of the first derivative in a similar way as [13] for such operators.

The vast majority of work on SBP-SAT schemes has been in the context of classical finite-difference SBP operators, typified by uniform nodal spacing, in computational space, and a repeating interior point operator (see the two review papers [8, 19]). There have been a number of extensions to more general operators; for example, Carpenter and Gottlieb [4] realized that the SBP property defined by Kreiss and Scherer [13] applies to a broad class of operators. They proved that using the Lagrangian interpolant, operators with the SBP property can be constructed on nearly arbitrary nodal distributions. The GSBP framework [7] unifies many of these extensions. In contrast, the extensions to the classical definition found in Carpenter et al. [5] and Abarbanel and Chertock [1] are not unified in the GSBP framework.

Using GSBP operators, derivatives can be approximated using a traditional finite-difference approach where h-refinement is performed by increasing the number of nodes in the mesh. Alternatively, the discretization can be implemented using an element approach where the domain is subdivided into a number of elements, each of which contains a fixed number of nodes, and h-refinement is carried out by increasing the number of elements. GSBP operators that have a repeating interior point operator can be applied in the traditional approach, while those that have a fixed nodal distribution can only be applied using an element approach.

Nordström and Lundquist [15, 18] have applied classical SBP operators as time integrators. They constructed fully discrete approximations that are provably consistent, conservative, and stable. The ideas of [15, 18] equally apply to GSBP operators, enabling the use of smaller operators for the same order of accuracy and hence more efficient time-marching methods.

The objectives of this paper are to present the extensions of the GSBP approach to the second derivative with a variable coefficient and to time integration.

2 Generalized Summation-by-Parts Operators for the Second Derivative

In this section, we review the construction of GSBP operators for the second derivative with a variable coefficient [6] that lead to stable and conservative schemes for partial differential equations (PDEs) that contain first, second, and mixed-derivative terms. GSBP operators for the first derivative are defined as follows [7]:

Definition 1 (First-Derivative GSBP Operator) A matrix operator, $\mathsf{D}_1 \in \mathbb{R}^{N\times N}$, on a nodal distribution $\mathbf{x}$, approximating the first derivative $\frac{\partial \mathscr{U}}{\partial x}$, is a GSBP operator

of order p if it is exact for the restriction of monomials up to degree p and

- $\mathsf{D}_1 = \mathsf{H}^{-1}\mathsf{Q}$, where H is a symmetric positive-definite matrix;
- $\mathsf{Q} + \mathsf{Q}^{\mathrm{T}} = \mathsf{E}$; and
- $\mathsf{E} = \mathbf{s}_\beta \mathbf{s}_\beta{}^{\mathrm{T}} - \mathbf{s}_\alpha \mathbf{s}_\alpha{}^{\mathrm{T}}$;

where the projection vectors $\mathbf{s}_\beta$ and $\mathbf{s}_\alpha$ are constructed such that $\mathbf{s}_\beta{}^{\mathrm{T}}\mathbf{u}$ and $\mathbf{s}_\alpha{}^{\mathrm{T}}\mathbf{u}$ are at least $p+1$ order approximations to $\mathscr{U}(\beta)$ and $\mathscr{U}(\alpha)$, respectively.

We note that for diagonal-norm GSBP operators, $\mathbf{v}^{\mathrm{T}}\mathsf{H}\mathbf{u}$ is an order $2p$ approximation to the L_2 inner product $\int_\alpha^\beta \mathscr{V}\mathscr{U}\,\mathrm{d}x$ [7].

The application of first-derivative GSBP operators twice leads to stable and conservative schemes. However, for operators with a repeating interior point operator, they lead to an unnecessarily wide interior point operator, and in general, lead to approximations of the second-derivative that are one order less accurate than the first-derivative operator. Alternatively, we can construct distinct GSBP operators approximating the second derivative that are one order more accurate than the application of the first-derivative operator twice while retaining the ability to prove stability using the energy method—we denote such operators as order-matched. To maintain stability of the semi-discrete or fully-discrete forms of the class of PDEs of interest, certain relations need to exist between the operators used to discretize the first-derivative, second-derivative, and mixed-derivative terms. One approach is to use operators that are compatible with the first-derivative operator used to discretize mixed-derivative terms [17]. In this paper, we concentrate on diagonal-norm compatible and order-matched operators, since for the variable-coefficient case, it is unclear how to construct stable schemes using dense-norm compatible and order-matched operators (see Mattsson and Almquist [16] for a discussion and potential solution). To motivate the form of compatible and order-matched GSBP operators for the second derivative with a variable coefficient, consider the following decomposition of the application of first-derivative GSBP operators twice:

$$\mathsf{D}_1\mathsf{B}\mathsf{D}_1 = \mathsf{H}^{-1}\left[-\mathsf{D}_1{}^{\mathrm{T}}\mathsf{H}\mathsf{B}\mathsf{D}_1 + \mathsf{E}\mathsf{B}\mathsf{D}_1\right]. \tag{1}$$

We construct compatible and order-matched GSBP operators as the application of the first-derivative operator twice plus corrective terms in order to increase the order of the resultant operator. These ideas lead to the following definition [6]:

Definition 2 (Compatible and Order-Matched Second-Derivative GSBP Operator) A diagonal-norm order-matched GSBP operator, $\mathsf{D}_2(\mathsf{B}) \in \mathbb{R}^{N\times N}$, approximating the second derivative $\frac{\partial}{\partial x}\left(\mathscr{B}\frac{\partial \mathscr{U}}{\partial x}\right)$, is compatible with the first-derivative GSBP operator, D_1 of order p, on a nodal distribution $\mathbf{x}$, if it is exact for the restriction of monomials up to degree $p+1$ and is of the form

$$\mathsf{D}_2(\mathsf{B}) = \mathsf{H}^{-1}\left[-\mathsf{D}_1{}^{\mathrm{T}}\mathsf{H}\mathsf{B}\mathsf{D}_1 + \sum_{i=1}^{N}\mathsf{B}(i,i)\mathsf{R}_i + \mathsf{E}\mathsf{B}\tilde{\mathsf{D}}_1\right]. \tag{2}$$

The matrices R_i are symmetric negative semidefinite. Furthermore, the matrix $\tilde{\mathsf{D}}_1$ is an approximation to the first derivative of at least order $p+1$ and the matrix B is constructed from the restriction of the variable coefficient $\mathscr{B}$ onto its diagonal. The remainder of the matrices are given in Definition 1.

Stability can be proven if all derivative operators share the same norm H, and the compatibility that is enforced on the second-derivative operator is with respect to the first-derivative operator used to approximate mixed-derivative terms.

Definition 2 leads to stable schemes if the operator satisfies an SBP property. In fact, GSBP operators are constructed such that they mimic the IBP property of the continuous PDE. For example, consider the linear convection-diffusion equation with a variable coefficient on the domain $x \in [\alpha, \beta]$:

$$\frac{\partial \mathscr{U}}{\partial t} = -\frac{\partial \mathscr{U}}{\partial x} + \frac{\partial}{\partial x}\left(\mathscr{B}\frac{\partial \mathscr{U}}{\partial x}\right). \tag{3}$$

Applying the energy method to (3), i.e., multiplying by the solution, integrating in space, and using integration by parts, leads to

$$\frac{\mathrm{d}\|\mathscr{U}\|^2}{\mathrm{d}t} = -\mathscr{U}^2\Big|_\alpha^\beta + 2\mathscr{U}\mathscr{B}\frac{\partial \mathscr{U}}{\partial x}\Big|_\alpha^\beta - 2\int_\alpha^\beta \frac{\partial \mathscr{U}}{\partial x}\mathscr{B}\frac{\partial \mathscr{U}}{\partial x}\mathrm{d}x. \tag{4}$$

With appropriate boundary conditions, (4) can be used to show that the solution is bounded in terms of the data of the problem (for more information see [9, 10, 12]). The semi-discrete version of (3), using a diagonal-norm first-derivative GSBP operator and a compatible and order-matched GSBP operator, is given as

$$\frac{\mathrm{d}\mathbf{u}}{\mathrm{d}t} = -\mathsf{D}_1\mathbf{u} + \mathsf{H}^{-1}\left[-\mathsf{D}_1{}^{\mathrm{T}}\mathsf{H}\mathsf{B}\mathsf{D}_1 + \sum_{i=1}^{N}\mathsf{B}(i,i)\mathsf{R}_i + \mathsf{E}\mathsf{B}\tilde{\mathsf{D}}_1\right]\mathbf{u}. \tag{5}$$

Applying the discrete energy method to (5), i.e., multiplying by $\mathbf{u}^{\mathrm{T}}\mathsf{H}$ and adding the transpose of the product to itself, leads to

$$\frac{\mathrm{d}\|\mathbf{u}\|_{\mathsf{H}}^2}{\mathrm{d}t} = \overbrace{-\mathbf{u}^{\mathrm{T}}\mathsf{E}\mathbf{u} + 2\mathbf{u}^{\mathrm{T}}\mathsf{E}\mathsf{B}\tilde{\mathsf{D}}_1\mathbf{u} - 2\left(\mathsf{D}_1\mathbf{u}\right)^{\mathrm{T}}\mathsf{H}\mathsf{B}\mathsf{D}_1\mathbf{u}}^{\approx -\mathscr{U}^2\big|_\alpha^\beta + 2\mathscr{U}\mathscr{B}\frac{\partial \mathscr{U}}{\partial x}\big|_\alpha^\beta - 2\int_\alpha^\beta \frac{\partial \mathscr{U}}{\partial x}\mathscr{B}\frac{\partial \mathscr{U}}{\partial x}\mathrm{d}x} + 2\sum_{i=1}^{N}\mathbf{u}\mathsf{R}_i\mathbf{u}, \tag{6}$$

where $\|\mathbf{u}\|_{\mathsf{H}}^2 = \mathbf{u}^{\mathrm{T}}\mathsf{H}\mathbf{u}$. We see that (6) is mimetic of the continuous case (4) with the addition of a negative semidefinite term of the order of the discretization. Using appropriate SATs for the imposition of boundary conditions and inter-element coupling, (6) can be shown to be stable.

The main difficulty in deriving compatible and order-matched operators is ensuring that the R_i are symmetric negative semidefinite, since it is necessary to ensure that the eigenvalues of N matrices are non-positive. This means that it is necessary to solve the eigenvalue problem of N matrices of size $N \times N$ to determine additional constraints. For compatible and order-matched operators, an alternative method is to construct the variable-coefficient operator from the constant-coefficient operator. This means that it is only necessary to solve the eigenvalue problem for one matrix. The resultant operator has the following form [6]:

$$\mathsf{D}_2(\mathsf{B}) = \mathsf{H}^{-1}\left[-\mathsf{D}_1{}^{\mathrm{T}}\mathsf{H}\mathsf{B}\mathsf{D}_1 + \frac{1}{N}\sum_{i=1}^{N}\mathsf{B}(i,i)\mathsf{R}_{\mathrm{c}} + \mathsf{E}\mathsf{B}\tilde{\mathsf{D}}_1\right], \tag{7}$$

where R_{c} and $\tilde{\mathsf{D}}_1$ are from the constant-coefficient operator. As an example, consider the approximation of the first and second derivative of order 3 on $x \in [-1, 1]$ using 5 Chebyshev-Gauss quadrature nodes. This nodal distribution does not have nodes on the boundary of the domains and is given as, to 5 digits of precision, $\mathbf{x} = \left[-0.95106, -0.58779, 0.0, 0.58779, 0.95106\right]$. The first-derivative GSBP operator has a norm matrix that is an order 6 approximation to the L_2 inner product; to 5 digits of precision, these operators are given by

$$\mathsf{D}_1 = \begin{bmatrix} -4.7488 & 6.6022 & -2.6552 & 1.0967 & -0.29478 \\ -1.1708 & -0.13670 & 1.7169 & -0.53833 & 0.12892 \\ 0.32492 & -1.3764 & 0.0 & 1.3764 & -0.32492 \\ -0.12892 & 0.53833 & -1.7169 & 0.13670 & 1.1708 \\ 0.29478 & -1.0967 & 2.6552 & -6.6022 & 4.7488 \end{bmatrix}, \quad \mathsf{H} = \begin{bmatrix} 0.16778 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.52555 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.61333 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.52555 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.16778 \end{bmatrix}.$$

The projection vectors used in the decomposition of E and to construct SATs (see Sect. 3 where this is shown for time integration) are given as
$\mathbf{s}_\beta{}^{\mathrm{T}} = \left[0.031677, -0.10191, 0.20000, -0.39252, 1.2628\right]$,
$\mathbf{s}_\alpha{}^{\mathrm{T}} = \left[1.2628, -0.39252, 0.20000, -0.10191, 0.031677\right]$. For the second derivative, the remaining matrices are given as

$$\mathsf{R}_{\mathrm{c}} = \begin{bmatrix} 0.13011 & -0.34065 & 0.42106 & -0.34065 & 0.13011 \\ -0.34065 & 0.89182 & -1.1024 & 0.89182 & -0.34065 \\ 0.42106 & -1.1024 & 1.3626 & -1.1024 & 0.42106 \\ -0.34065 & 0.89182 & -1.1024 & 0.89182 & -0.34065 \\ 0.13011 & -0.34065 & 0.42106 & -0.34065 & 0.13011 \end{bmatrix}, \quad \tilde{\mathsf{D}}_1 = \begin{bmatrix} -4.9798 & 7.2068 & -3.4026 & 1.7013 & -0.52573 \\ -1.0515 & -0.44903 & 2.1029 & -0.85065 & 0.24822 \\ 0.32492 & -1.3764 & 0.0 & 1.3764 & -0.32492 \\ -0.24822 & 0.85065 & -2.1029 & 0.44903 & 1.0515 \\ 0.52573 & -1.7013 & 3.4026 & -7.2068 & 4.9798 \end{bmatrix}.$$

3 Time-Marching Methods Based on Generalized Summation-by-Parts Operators

This section describes the application of GSBP operators to the solution of initial value problems

$$\frac{\mathrm{d}y}{\mathrm{d}t} = f(y,t), \quad \text{with} \quad y(\alpha) = y_\alpha \quad \text{and} \quad \alpha \le t \le \beta. \tag{8}$$

This is an extension of the work presented in [15, 18] for time-marching methods based on classical SBP operators. It also draws on the concepts of dual-consistency and superconvergence presented in [11] for classical SBP operators.

The primary advantage of the GSBP approach in time is the significantly smaller number of solution points required per block for a given order of accuracy. With careful selection of SAT coefficients in a multiblock implementation, the pointwise solution within each block is decoupled from the solution in subsequent blocks. As a result, each block can be solved sequentially in time, though the pointwise solution within each block remains in general fully coupled. Thus, the reduced size of the operators possible with the GSBP approach can significantly improve the efficiency of the time integration.

Consider the application of a single-block classical SBP or GSBP time-marching method to the initial value problem (8):

$$\mathsf{D}_1 \mathbf{y}_\mathrm{d} = \mathsf{H}^{-1}\mathsf{Q}\mathbf{y}_\mathrm{d} = \mathbf{f}(\mathbf{y}_\mathrm{d}, \mathbf{t}) + \sigma \mathsf{H}^{-1}\mathbf{s}_\alpha(\mathbf{s}_\alpha{}^\mathrm{T}\mathbf{y}_\mathrm{d} - y_\alpha), \tag{9}$$

where the second term on the right-hand side, the SAT penalty term, weakly enforces the initial condition. The most practical choice of SAT coefficient σ is -1, which renders the temporal discretization dual consistent and L-stable [2]. In addition, if the norm associated with the GSBP operator is diagonal, then the discretization becomes algebraically stable [2]. This choice of SAT coefficient implies the superconvergence of the pointwise solution projected to the boundary at β, $\mathbf{s}_\beta{}^\mathrm{T}\mathbf{y}_\mathrm{d}$, as well as linear functionals of the solution, $\int_\alpha^\beta g(t)y(t)dt$, integrated with the norm of the discretization [2]. These properties all extend to the multiblock case with appropriate choice of interface SAT coefficients.

Time-marching methods based on classical SBP and GSBP operators are a subclass of Runge-Kutta (RK) methods, which are written as

$$\tilde{y}^{[i]} = \tilde{y}^{[i-1]} + h\sum_{j=1}^{n} \mathbf{b}_j \mathbf{f}(\mathbf{y}_j, t^{[i-1]} + \mathbf{c}_j h), \tag{10}$$

with internal stage approximations:

$$\mathbf{y}_k = \tilde{y}^{[i-1]} + h\sum_{j=1}^{n} \mathsf{A}_{kj} \mathbf{f}(\mathbf{y}_j, t^{[i-1]} + \mathbf{c}_j h) \quad \text{for } k = 1, \ldots, n, \tag{11}$$

where A and $\mathbf{b}$ are the coefficient matrices of the method, $\mathbf{c}$ is the abscissa, and h is the step size. With a dual consistent choice of SAT coefficients, classical SBP or GSBP temporal discretizations can be rearranged and written in this form. The pointwise solution mimics the RK stage approximations, and the projection of the pointwise solution to the boundary at β becomes the solution update [2]. The

coefficient matrices of the equivalent RK scheme, written in terms of the SBP-SAT discretization (9), are [2]:

$$\mathsf{A} = \tfrac{1}{h}\left(\mathsf{Q} + \mathbf{s}_\alpha \mathbf{s}_\alpha{}^{\mathrm{T}}\right)^{-1}\mathsf{H}, \quad \mathbf{b}^{\mathrm{T}} = \mathbf{s}_\beta{}^{\mathrm{T}}\mathsf{A} = \tfrac{1}{h}\mathbf{s}_\beta{}^{\mathrm{T}}\left(\mathsf{Q} + \mathbf{s}_\alpha \mathbf{s}_\alpha{}^{\mathrm{T}}\right)^{-1}\mathsf{H} = \tfrac{1}{h}\mathbb{1}^{\mathrm{T}}\mathsf{H}, \tag{12}$$

where $\mathbf{c} = \frac{\mathbf{t}-\mathbb{1}\alpha}{h}$, $h = \beta - \alpha$, and $\mathbb{1} = [1, \ldots, 1]^{\mathrm{T}}$.

This characterization of classical SBP and GSBP time-marching methods enables a direct comparison with traditional time-marching methods. It also enables common time-marching ideas to be transferred back into the GSBP realm, for example diagonally-implicit methods, where the pointwise solution within each block can be solved sequentially in time (See [2] for examples). It also highlights the fact that dual-consistent SBP and GSBP time-marching methods do not define a new class of methods. Nevertheless, the GSBP framework provides a relatively simple way of constructing high-order implicit RK schemes with high-stage order and L-stability. Furthermore, if the norm is diagonal then the resulting scheme will be algebraically stable [2].

As an example, consider dual-consistent time-marching methods based on classical SBP and Legendre-Gauss GSBP operators which are exact for third-order polynomials. The minimum number of solution points per block required for the diagonal and dense-norm classical SBP operators is 12 and 8, respectively. The rate of superconvergence obtained for these classical SBP time-marching methods is 6 and 4, respectively. In contrast, only 4 solution points per block are required for a Legendre-Gauss based GSBP operator to be exact for third-order polynomials. Furthermore, the rate of superconvergence obtained is 7 [2], higher than both of the classical SBP time-marching methods. This translates to significantly more efficient time integration.

The first derivative GSBP operator and diagonal norm of the 4-point Legendre-Gauss GSBP time-marching method discussed above are:

$$\mathsf{D}_1^{(3)} = \begin{bmatrix} -3.3320 & 4.8602 & -2.1088 & 0.58063 \\ -0.75756 & -0.38441 & 1.4707 & -0.32870 \\ 0.32870 & -1.4707 & 0.38441 & 0.75756 \\ -0.58063 & 2.1088 & -4.8602 & 3.3320 \end{bmatrix}, \quad \mathsf{H} = \begin{bmatrix} 0.34785 & 0 & 0 & 0 \\ 0 & 0.65215 & 0 & 0 \\ 0 & 0 & 0.65215 & 0 \\ 0 & 0 & 0 & 0.34785 \end{bmatrix}.$$

This is derived for the quadrature points $\mathbf{t} = [-0.86114,\ -0.33998,\ 0.33998,\ 0.86114]$, defined for the interval $[-1, 1]$. The equivalent RK scheme has the coefficient matrices:

$$\mathsf{A} = \begin{bmatrix} 0.095040 & -0.047061 & 0.033084 & -0.011632 \\ 0.17721 & 0.19067 & -0.055518 & 0.017647 \\ 0.17810 & 0.32632 & 0.19067 & -0.025102 \\ 0.16941 & 0.33390 & 0.33222 & 0.095040 \end{bmatrix}, \quad \mathbf{b}^{\mathrm{T}} = \begin{bmatrix} 0.086964,\ 0.16304,\ 0.16304,\ 0.086964 \end{bmatrix},$$

with abscissa: $\mathbf{c} = [0.069432,\ 0.33001,\ 0.66999,\ 0.93057]$. This RK scheme differs from the well-known Kuntzmann-Butcher Gauss RK methods [3, 14] which are one order higher, but forfeit L-stability.

4 Conclusions

The developments reviewed in this paper extend the GSBP approach to the second derivative with a variable coefficient as well as to time marching. The benefit of the GSBP approach, relative to the classical SBP approach, is that for a given order of accuracy, operators that require fewer nodes can be constructed. This leads to more efficient methods.

References

1. S.S. Abarbanel, A.E. Chertock, A. Yefet, Strict stability of high-order compact implicit finite-difference schemes: the role of boundary conditions for hyperbolic PDES, I. J. Comput. Phys. **160**, 42–66 (2000)
2. P.D. Boom, D.W. Zingg, High-order implicit time-marching methods based on generalized summation-by-parts operators. SIAM J. Sci. Comput. (accepted)
3. J.C. Butcher, Implicit Runge-Kutta processes. Math. Comput. **18**(85), 50–64 (1964)
4. M.H. Carpenter, D. Gottlieb, Spectral methods on arbitrary grids. J. Comput. Phys. **129**(1), 74–86 (1996)
5. M.H. Carpenter, D. Gottlieb, S. Abarbanel, Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes. J. Comput. Phys. **111**(2), 220–236 (1994)
6. D.C. Del Rey Fernández, D.W. Zingg, Generalized summation-by-parts operators for the second derivative with a variable coefficient. SIAM J. Sci. Comput. (accepted)
7. D.C. Del Rey Fernández, P.D. Boom, D.W. Zingg, A generalized framework for nodal first derivative summation-by-parts operators. J. Comput. Phys. **266**(1), 214–239 (2014)
8. D.C. Del Rey Fernández, J.E. Hicken, D.W. Zingg, Review of summation-by-parts operators with simultaneous approximation terms for the numerical solution of partial differential equations. Comput. Fluids **95**(22), 171–196 (2014)
9. B. Gustafsson, *High Order Difference Methods for Time Dependent PDE* (Springer, Berlin, 2008)
10. B. Gustafsson, H.O. Kreiss, J. Oliger, *Time-Dependent Problems and Difference Methods*, 2nd edn. Pure and Applied Mathematics (Wiley, New York, 2013)
11. J.E. Hicken, D.W. Zingg, Superconvergent functional estimates from summation-by-parts finite-difference discretizations. SIAM J. Sci. Comput. **33**(2), 893–922 (2011)
12. H.O. Kreiss, J. Lorenz, *Initial-Boundary Value Problems and the Navier-Stokes Equations*. Classics in Applied Mathematics, vol. 47 (SIAM, Philadelphia, 2004)
13. H.O. Kreiss, G. Scherer, Finite element and finite difference methods for hyperbolic partial differential equations, in *Mathematical Aspects of Finite Elements in Partial Differential Equations* (Academic, New York/London, 1974), pp. 195–212
14. J. Kuntzmann, Neuere Entwicklungen der Methode von Runge und Kutta. ZAMM J. Appl. Math. Mech. / Z. Angew. Math. Mech. **41**(S1), T28–T31 (1961)
15. T. Lundquist, J. Nordström, The SBP-SAT technique for initial value problems. J. Comput. Phys. **270**(1), 86–104 (2014)
16. K. Mattsson, M. Almquist, A solution to the stability issues with block norm summation by parts operators. J. Comput. Phys. **15**, 418–442 (2013)
17. K. Mattsson, M. Svärd, M. Shoeybi, Stable and accurate schemes for the compressible Navier-Stokes equations. J. Comput. Phys. **227**(4), 2293–2316 (2008)

18. J. Nordström, T. Lundquist, Summation-by-parts in time. J. Comput. Phys. **251**, 487–499 (2013)
19. M. Svärd, J. Nordström, Review of summation-by-parts schemes for initial-boundary-value-problems. J. Comput. Phys. **268**(1), 17–38 (2014)

3D Viscoelastic Anisotropic Seismic Modeling with High-Order Mimetic Finite Differences

Miguel Ferrer, Josep de la Puente, Albert Farrés, and José E. Castillo

Abstract We present a scheme to solve three-dimensional viscoelastic anisotropic wave propagation on structured staggered grids. The scheme uses a fully-staggered grid (FSG) or Lebedev grid (Lebedev, J Sov Comput Math Math Phys 4:449–465, 1964; Rubio et al. Comput Geosci 70:181–189, 2014), which allows for arbitrary anisotropy as well as grid deformation. This is useful when attempting to incorporate a bathymetry or topography in the model. The correct representation of surface waves is achieved by means of using high-order mimetic operators (Castillo and Grone, SIAM J Matrix Anal Appl 25:128–142, 2003; Castillo and Miranda, Mimetic discretization methods. CRC Press, Boca Raton, 2013), which allow for an accurate, compact and spatially high-order solution at the physical boundary condition. Furthermore, viscoelastic attenuation is represented with a generalized Maxwell body approximation, which requires of auxiliary variables to model the convolutional behavior of the stresses in lossy media. We present the scheme's accuracy with a series of tests against analytical and numerical solutions. Similarly we show the scheme's performance in high-performance computing platforms. Due to its accuracy and simple pre- and post-processing, the scheme is attractive for carrying out thousands of simulations in quick succession, as is necessary in many geophysical forward and inverse problems both for the industry and academia.

1 Introduction

Seismic waves occur when the subsurface is excited, by an internal event (e.g. an earthquake, an underground explosion) or an external event (e.g. the impact of a meteorite, a landslide). The behaviour of such waves can be described by means of a

M. Ferrer (✉) • J. de la Puente • A. Farrés
Computer Applications in Science and Engineering, Barcelona Supercomputing Center, Jordi Girona 29, 08034 Barcelona, Spain
e-mail: miguel.ferrer@bsc.es; josep.delapuente@bsc.es; albert.farres@bsc.es

J.E. Castillo
Computational Science Research Center, San Diego State University, 5500 Campanile Drive, San Diego, CA 92182-7720, USA
e-mail: jcastillo@mail.sdsu.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_18

hierarchy of physical laws that represent ever more accurately observed phenomena. For certain applications, waves can be represented as rays, although some wave phenomena require of mechanical laws that properly describe their properties. At medium to long scales, seismic waves can be fully described with an anisotropic viscoelastic theory. Anisotropy describes the properties of some solids to support waves moving with different speeds when they travel in different directions. In rocks, anisotropy can be due to intrinsic crystalline properties or a macroscopic representation of fine sediment layering. Viscoelasticity is a macroscopic property which accounts for energy losses observed in the subsurface. When having good models of the subsurface properties, anisotropic viscoelastic modelling allows us to obtain synthetic seismic waves which behave very similarly to observed waves. A very popular approach for modelling seismic waves is the staggered-grid time-domain finite-difference method [14, 16, 20, 21]. This method is very efficient for large simulations. However, it presents limitations when modelling strong anisotropy [9] or topography [8]. An improvement to the method is the fully-staggered grid (FSG) method [13, 15] which naturally supports arbitrary anisotropy. More recently [4] showed that the method can be further modified by using mimetic operators and deformed grids to model topography with high precision. In this paper we show how the mimetic FSG finite-difference method can be improved by adding support for viscoelastic materials with Generalized Maxwell Body (GMB, see [5]) rheology.

2 Viscoelastic Wave Propagation

Viscoelastic waves, in time-domain velocity-stress formulation, are governed by the PDE

$$\begin{aligned} \frac{\partial \mathbf{S}}{\partial t} &= \mathbf{C} * \mathbf{E}\,, \\ \frac{\partial \mathbf{v}}{\partial t} &= \frac{1}{\rho}\mathbf{T}\,, \end{aligned} \tag{1}$$

where the stress tensor in vector form is $\mathbf{S} = (\sigma_{xx}, \sigma_{yy}, \sigma_{zz}, \sigma_{yz}, \sigma_{xz}, \sigma_{xy})^T$, the strain-rate tensor in vector form is $\mathbf{E} = (\dot{\varepsilon}_{xx}, \dot{\varepsilon}_{yy}, \dot{\varepsilon}_{zz}, \dot{\varepsilon}_{yz}, \dot{\varepsilon}_{xz}, \dot{\varepsilon}_{xy})^T$, $\mathbf{C}$ is the stiffness matrix, $\mathbf{v}$ is the particle velocity vector, ρ the density and $\mathbf{T} \equiv \partial\sigma_{ij}/\partial x_j$ which is related to the gradients of the tractions in planes perpendicular to all three Cartesian directions x, y and z. Equation (1) is sufficient to describe waves propagating through a solid lossy material. In general, both compressional P and shear S wave modes are supported in viscoelastic media. The convolution in the equation becomes a normal product in case the medium is lossless (e.g. elastic). Viscoelastic effects are generally accounted for with quality factors Q_P and Q_S which are lower the more attenuated the wave mode is and, additionally, are reported to be almost frequency independent. Many mechanical models exist to represent accurately viscoelastic

effects in geophysics, although one of the most accurate is the Generalized Maxwell Body (GMB) rheology (see [17] for a complete overview). When using GMB, we can rewrite (1) as

$$\begin{aligned} \frac{\partial \mathbf{S}}{\partial t} &= \tilde{\mathbf{C}}\mathbf{E} - \sum_{l=1}^{n} \mathbf{Y}^l \mathbf{A}^l \,, \\ \frac{\partial \mathbf{v}}{\partial t} &= \frac{1}{\rho}\mathbf{T} \,, \\ \frac{\partial \mathbf{A}^l}{\partial t} &= \omega_l \left(\mathbf{A}^l - \mathbf{E}\right) \,, \end{aligned} \tag{2}$$

where n denotes the number of Maxwell mechanisms used, $\mathbf{Y}^l$ are viscoelastic coefficient matrices, $\mathbf{A}^l = (a_{xx}, a_{yy}, a_{zz}, a_{yz}, a_{xz}, a_{xy})^T$ are the anelastic variables related to the strain rates and ω_l is the characteristic frequency of each mechanism. Notice that the stiffness matrix in (2) refers to the unrelaxed stiffness $\tilde{\mathbf{C}}$, which refers to the value of $\mathbf{C}$ at very high frequencies. Equation (2) allows us to model viscoelastic waves without convolution operators, which have been substituted by extra (anelastic) variables in our system. This makes the simulation of viscoelastic waves affordable in the time domain. Explicitly, we have

$$\tilde{\mathbf{C}} = \begin{pmatrix} \tilde{c}_{11} & \tilde{c}_{12} & \tilde{c}_{13} & c_{14} & c_{15} & c_{16} \\ \tilde{c}_{12} & \tilde{c}_{22} & \tilde{c}_{23} & c_{24} & c_{25} & c_{26} \\ \tilde{c}_{13} & \tilde{c}_{23} & \tilde{c}_{33} & c_{34} & c_{35} & c_{36} \\ c_{14} & c_{24} & c_{34} & \tilde{c}_{44} & c_{45} & c_{46} \\ c_{15} & c_{25} & c_{35} & c_{45} & \tilde{c}_{55} & c_{56} \\ c_{16} & c_{26} & c_{36} & c_{46} & c_{56} & \tilde{c}_{66} \end{pmatrix}, \mathbf{Y}^l = \begin{pmatrix} Y_l^P & Y_l^\lambda & Y_l^\lambda & 0 & 0 & 0 \\ Y_l^\lambda & Y_l^P & Y_l^\lambda & 0 & 0 & 0 \\ Y_l^\lambda & Y_l^\lambda & Y_l^P & 0 & 0 & 0 \\ 0 & 0 & 0 & Y_l^S & 0 & 0 \\ 0 & 0 & 0 & 0 & Y_l^S & 0 \\ 0 & 0 & 0 & 0 & 0 & Y_l^S \end{pmatrix}, \tag{3}$$

so that the material can be anisotropic but we only accept isotropy in the attenuative properties of the material. The ω_l values are chosen to cover our desired bandwidth evenly in the logarithmic scale. Then, the coefficients in $\mathbf{Y}^l$ can be found, if we use auxiliary halfway ω_k points with $k = 1, \cdots, 2n-1$, by using

$$\begin{aligned} Q_v^{-1}(\omega_k) &= \sum_{l=1}^{n} \frac{\omega_k \omega_l + \omega_l^2 Q_v^{-1}(\omega_k)}{\omega_l^2 + \omega_k^2} Y_l^v \text{ with } v = P, S \,. \\ Y_l^\lambda &= \frac{P}{L} Y_l^P - \frac{2S}{L} Y_l^S \end{aligned} \tag{4}$$

where Q_P and Q_S are locally constant values and $P = c_{ii}/3$ with $i = 1, 2, 3$, $S = c_{ii}/3$ with $i = 4, 5, 6$ and $L = P - 2S$. Finally, the unrelaxed stiffness components must be found so that the input velocities match as well as possible

the phase velocity at our peak frequency ω_0. This can be achieved using

$$\begin{aligned} \Theta_1^v &= 1 - \sum_{l=1}^{n} Y_l^v \frac{1}{1+(\omega_0/\omega_l)^2}\,, \quad \Theta_2^v = \sum_{l=1}^{n} Y_l^v \frac{\omega_0/\omega_l}{1+(\omega_0/\omega_l)^2}\,, \\ U^v &= \frac{\sqrt{\left(\Theta_1^v\right)^2 + \left(\Theta_2^v\right)^2} + \Theta_1^v}{2\left(\Theta_1^v\right)^2 + 2\left(\Theta_2^v\right)^2}\,, \qquad \text{with } v = P, S. \end{aligned} \tag{5}$$

The values U^P and U^S can be used to obtain the unrelaxed stiffness values following

$$\begin{aligned} \tilde{c}_{ii} &= c_{ii} U^P\,, && \text{for } i = 1,2,3\,, \\ \tilde{c}_{ii} &= c_{ii} U^S\,, && \text{for } i = 4,5,6\,, \\ \tilde{c}_{ij} &= c_{ij} \frac{PU^P - 2U^S}{L}\,, && \text{for } i,j = 1,2,3 \text{ and } i \neq j\,. \end{aligned} \tag{6}$$

After initializing the parameters with Eqs. (3)–(6), we solve system (2) using an FSG finite-difference method, where all spatial derivatives are substituted by mimetic operators [2–4, 18]. In addition, we use a leap-frog explicit scheme for the time integration which benefits from the corrections in [11] for reducing storage in this configuration. Notice that in time-domain explicit seismic modelling applications, time integration beyond second order is rare, as errors in the form of dispersion are dominated by the spatial discretization at the relevant frequency and propagation distances [6].

3 2D Homogeneous Test

First of all we wish to verify the accuracy of our scheme when handling elastic and viscoelastic wave propagation. To that goal we set up a simple 2D test in homogenous material for which an analytical solution exists [1]. We use a material with $v_P = 6000\,\text{m/s}$, $v_S = 3464\,\text{m/s}$ and $\rho = 2700\,\text{kg/m}^3$. We then set $Q_P = 60$ and $Q_S = 30$ for the viscoelastic case and $Q_P = Q_S = \infty$ for the elastic case. The source is a force acting with a Ricker wavelet having its energy peak at 10 Hz. A receiver is placed 1500 m away from the source along the source direction. Viscoelasticity is modelled with $n = 3$ using a bandwidth of 100 Hz centered on the source's peak frequency. The model was discretized with a 201×201 grid of spacing 30 m. In Fig. 1 left, we plot the wave velocities depending on the frequency as a consequence of the GMB mechanisms. Similarly we can observe how the desired Q value is fitted along our bandwidth. In Fig. 1 right, we show the fit between analytical and numerical solutions for both the elastic and viscoelastic cases. We observe no quality degradation due to using viscoelasticity in our algorithm. Furthermore, we can observe how the P-wave (earlier) and S-wave (later) arrivals are both damped, being the energy loss stronger for the S-wave due to the lower (i.e. more attenuating) Q

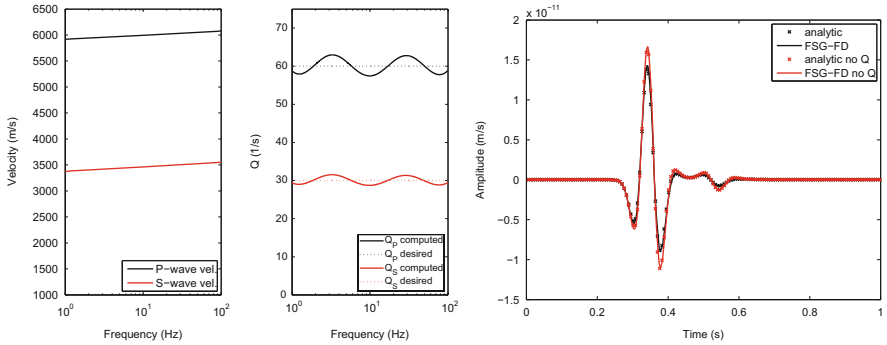


Fig. 1 *Left*: velocity dispersion and attenuation fit for both P- and S-wave. *Right*: simulation and analytical solutions for elastic and viscoelastic case

value for this wave mode. Similarly, we can observe that our coefficient computation algorithm enabled a nearly zero phase difference between the elastic and viscoelastic runs, as is expected from our waves concentrating energy around the peak of the wavelet.

4 3D Heterogeneous Test

We have built a large 3D elastic and its equivalent viscoelastic model. The model is cubic and composed of 27 small subcubes, each of them with different physical properties. The model is challenging because it displays very large contrasts in the material properties. P-wave velocities range from 1000 to 5000 m/s and Poisson ratios from values of 0.2–0.45, which results in S-wave velocities ranging from 408 to 3535 m/s. Densities range from 1200 to 2700 kg/m^3. In the viscoelastic case, Q_P takes values from 50 to 250 and Q_S from 20 to 176. We have an explosive source located at the middle of the domain with a Ricker wavelet having peak frequency at 20 Hz. A total of six receivers are located at the center of each of the domain's quadrilateral faces. The time sampling is $\Delta t = 0.00016$ s and the spatial sampling is equal in all directions to 2.5 m. The volume is composed of $501 \times 501 \times 501$ cells and the simulation lasts for 10,000 iterations. CPML [10] boundary conditions are set everywhere. The results of the simulation with and without viscous mechanisms can be seen in Fig. 2. We observe that the strong heterogeneity generates many wave arrivals. The viscoelastic simulation is mostly in-phase with the elastic one, displaying different degrees of energy loss depending on the actual arrival and receiver. We conclude that the method is robust in strongly heterogeneous cases, including attenuation in 3D, when using a wide range of realistic values for the material properties. For large scenarios like this, we employ a hybrid parallel approach OpenMP/MPI for distributed memory computer clusters.

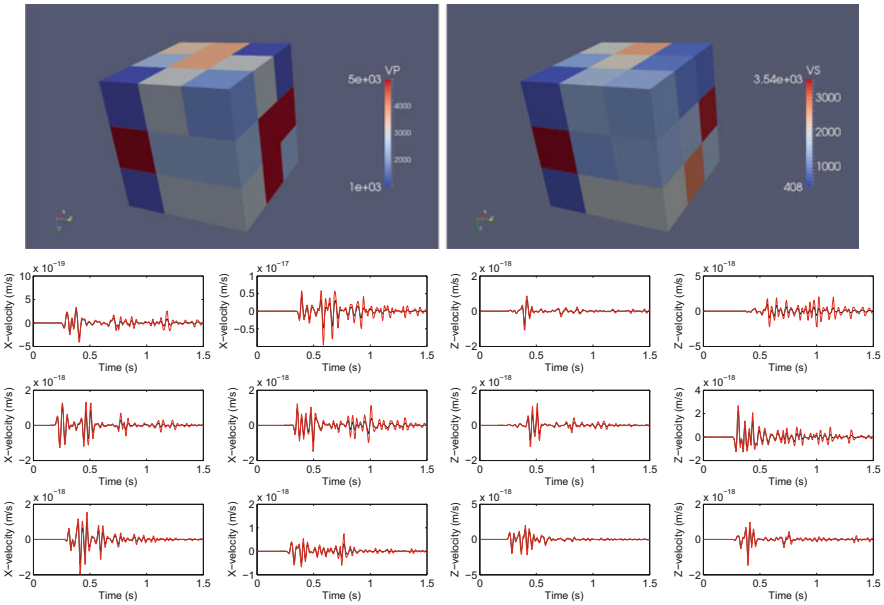


Fig. 2 Description of model for P-wave velocity and S-wave velocity (*top*) and seismograms recorded at all six locations, for the x and z components of the particle velocity vector (*bottom*)

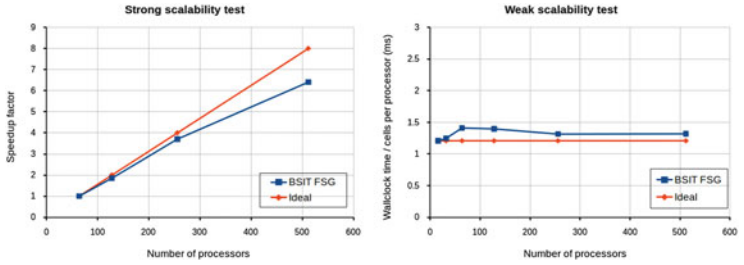


Fig. 3 Strong and weak scalability tests performed on the MareNostrum supercomputer at the Barcelona Supercomputing Center

The simulation code has been developed with BSIT [7, 19] achieving 69 GFLOPS per Intel E5-2670 16-core node. A scalability test is provided in Fig. 3.

5 Discussion

As a final check for the correctness of our results we can quantify the dispersion and amplitude differences for our two examples. We perform a time-frequency analysis of the elastic and viscoelastic solutions using the definitions of Kristekova

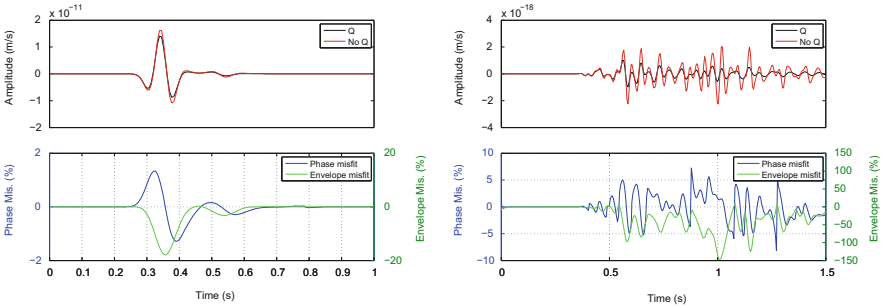


Fig. 4 Phase and envelope misfits for the homogeneous case and for a randomly chosen receiver in the 3D heterogeneous case

et al. [12]. This allows us to completely separate phase misfits from envelope misfits. By looking at the misfits for the homogeneous case, in Fig. 4 left, we can see that the envelope misfits are negative everywhere, which corresponds to the observed (and expected) amplitude loss in the viscoelastic case. More interesting is the phase misfit. We observe that, for both the P and S arrivals, we have first a slight phase misfit increase which is followed by a slight misfit decrease after the wave peak. This indicates that each wave arrival is being dispersively separated into faster wave components and slower wave components. This is what we expect, and corresponds to the dispersion curves in Fig. 1 where higher frequency modes travel faster than lower frequency modes. We remark again that this dispersive behaviour is expected in physically sound viscoelastic rheologies [1]. Furthermore, the phase misfit tends to average out along during each arrival, indicating that the central frequencies travel at the correct velocity. In Fig. 4 right we have a more complex scenario, but nevertheless displaying the same behaviour: phase misfits increase and then decrease for an overall in-phase propagation although with signs of dispersion. The envelope misfit, however is always negative and quite large. Notice that as waves have travelled more cycles, the effects of dispersion and attenuation are also stronger.

6 Conclusions

We have described the upgrade of the FSG time-domain mimetic finite-difference method to support viscoelastic attenuation accurately. Our approach is based on a Generalized Maxwell Body mechanism which allows us to correctly model the dispersive behaviour of viscoelastic waves. We make an effort in finding ways to obtain attenuating parameters that respect our wave velocity at the center of our frequency bandwidth and are quasi-flat throughout it. The resulting algorithm has been tested against an analytical solution obtaining an excellent agreement with it.

Furthermore, we have shown that the method is robust enough to tackle cases of extreme heterogeneity in large 3D scenarios.

Acknowledgements The authors want to thank Repsol for the permission to publish the present research, carried out at the Repsol-BSC Research Center as a part of the Kaleidoscope Project. This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement No 644602.

References

1. J. Carcione, *Wave Fields in Real Media: Wave Propagation in Anisotropic, Anelastic and Porous Media*. Handbook of Geophysical Exploration, Seismic Exploration (Pergamon Press, Oxford, 2002)
2. J. Castillo, R. Grone, A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law. SIAM J. Matrix Anal. Appl. **25**, 128–142 (2003)
3. J.E. Castillo, G.F. Miranda, *Mimetic Discretization Methods* (CRC Press, Boca Raton, 2013)
4. J. de la Puente, M. Ferrer, M. Hanzich, J.E. Castillo, J.M. Cela, Effects of free-surface topography on moving-seismic-source modeling. Geophysics **79**(3), T125–T141 (2014)
5. H. Emmerich, M. Korn, Incorporation of attenuation into time-domain computations of seismic wave fields. Geophysics **52**, 1252–1264 (1987)
6. A. Fichtner, *Full Seismic Waveform Modelling and Inversion* (Springer, Heidelberg, 2010)
7. M. Hanzich, J. Rodriguez, N. Gutierrez, J. de la Puente, J. Cela, Using HPC software frameworks for developing BSIT: a geophysical imaging tool. In: *Proceedings of WCCM XI, ECCM V, ECFD VI*, vol. III (2014), pp. 2019–2030
8. S. Hestholm, Three-dimensional finite difference viscoelastic wave modelling including surface topography. Geophys. J. Int. **139**(3), 852–878 (1999)
9. H. Igel, P. Mora, B. Riollet, Anisotropic wave propagation through finite-difference grids. Geophysics **60**, 1203–1216 (1995)
10. D. Komatitsch, R. Martin, An unsplit convolutional perfectly matched layer improved at grazing incidence for the seismic wave equation. Geophysics **72**(5), SM155–SM167 (2007)
11. J. Kristek, P. Moczo, Seismic-wave propagation in viscoelastic media with material discontinuities: a 3D fourth-order staggered-grid finite-difference modeling. Bull. Seismol. Soc. Am. **93**, 2273–2280 (2003)
12. M. Kristeková, J. Kristek, P. Moczo, S. Day, Misfit criteria for quantitative comparison of seismograms. Bull. Seismol. Soc. Am. **96**(5), 1836–1850 (2006)
13. V. Lebedev, Difference analogies of orthogonal decompositions of basic differential operators and some boundary value problems. J. Sov. Comput. Math. Math. Phys. **4**, 449–465 (1964)
14. A.R. Levander, Fourth-order finite difference P-SV seismograms. Geophysics **53**, 1425–1436 (1988)
15. V. Lisitsa, D. Vishnevskiy, Lebedev scheme for the numerical simulation of wave propagation in 3D anisotropic elasticity. Geophys. Prospect. **58**(4), 619–635 (2010)
16. R. Madariaga, Dynamics of an expanding circular fault. Bull. Seismol. Soc. Am. **65**, 163–182 (1976)
17. P. Moczo, J. Kristek, M. Galis, P. Pazak, M. Balazovjech, The finite-difference and finite-element modeling of seismic wave propagation and earthquake motion. Acta Phys. Slovaca **57**(2), 177–406 (2007)
18. O. Rojas, S. Day, J. Castillo, L.A. Dalguer, Modelling of rupture propagation using high-order mimetic finite differences. Geophys. J. Int. **172**(2), 631–650 (2008)

19. F. Rubio, M. Hanzich, A. Farrés, J. de la Puente, J. María Cela, Finite-difference staggered grids in GPUs for anisotropic elastic wave propagation simulation. Comput. Geosci. **70**, 181–189 (2014)
20. J. Virieux, SH-wave propagation in heterogeneous media: velocity-stress finite-difference method. Geophysics **49**, 1933–1942 (1984)
21. J. Virieux, P-SV wave propagation in heterogeneous media: velocity-stress finite-difference method. Geophysics **51**, 889–901 (1986)

A Locally Conservative High-Order Least-Squares Formulation in Curvilinear Coordinates

Marc Gerritsma and Pavel Bochev

Abstract We present a locally conservative spectral least-squares formulation for the scalar diffusion-reaction equation in curvilinear coordinates. Careful selection of a least squares functional and compatible finite dimensional subspaces for the solution space yields the conservation properties. Numerical examples confirm the theoretical properties of the method.

1 Introduction

Least-squares finite element methods for partial differential equations reformulate PDEs into unconstrained minimization problems. The sum of weighted equation residuals measured in suitable Sobolev norms defines the least-squares functional. Norm-equivalent least-squares functionals give rise to symmetric and strongly coercive variational problems. These properties are inherited on conforming finite dimensional subspaces of the solution space. Therefore, conforming finite element discretizations circumvent inf-sup conditions and are always symmetric, positive definite, which make these discrete systems amenable to well-established iterative solvers.

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

M. Gerritsma (✉)
TU Delft, Kluyverweg 1, 2629 HS Delft, The Netherlands
e-mail: M.I.Gerritsma@tudelft.nl

P. Bochev
Computational Mathematics, Sandia National Laboratories, Mail Stop 1320, Albuquerque, NM 87185, USA
e-mail: pbboche@sandia.gov

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_19

Exceptional stability of least-squares formulations has led to the widespread use of standard C^0 elements in their discretization. Unfortunately, resulting finite element methods are only approximately conservative, which generally leads to violation of fundamental physical properties, such as loss of mass or artificial vorticity generation in potential flows. In many cases this drawback can outweigh potential advantages of least squares methods; see [10, 19]. As a result, improving conservation properties of least-squares methods has attracted significant attention [1–4, 7, 8, 10, 12–14].

2 Conservative Least-Squares Functional

We explain our approach using the following diffusion-reaction problem [5]

$$-\nabla \cdot \mathbb{A}\nabla\phi + \gamma\phi = f \text{ in } \Omega\ , \quad \phi = g \text{ on } \Gamma_D\ , \quad \boldsymbol{n} \cdot \mathbb{A}\nabla\phi = h \text{ on } \Gamma_N\ , \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, has a Lipschitz-continuous boundary $\partial\Omega = \Gamma_D \cup \Gamma_N$ and $\boldsymbol{n}$ is the outward unit normal to $\partial\Omega$. We assume that $\mathbb{A}$ is a symmetric positive definite tensor and γ is a real-valued, strictly positive function, i.e., there exist constants $a_{min}, a_{max}, \gamma_{min}, \gamma_{max} > 0$ such that $a_{min}\boldsymbol{\xi}^T\boldsymbol{\xi} \leq \boldsymbol{\xi}^T\mathbb{A}(\mathbf{x})\boldsymbol{\xi} \leq a_{max}\boldsymbol{\xi}^T\boldsymbol{\xi}$ and $\gamma_{min} \leq \gamma(\mathbf{x}) \leq \gamma_{max}$ for all $\boldsymbol{x} \in \Omega$ and vectors $\boldsymbol{\xi}$. The tensor $\mathbb{A}$ and the function γ describe material properties. For instance, in heat transfer applications $\mathbb{A}$ is the thermal conductivity of the material and γ can be related to the specific heat capacity.

This scalar problem can be recast as an equivalent four-field problem, given by

$$\begin{aligned} \nabla \cdot \mathbf{u} + \psi &= 0 \text{ in } \Omega\ , \quad \mathbf{v} = \mathbb{A}^{-1}\mathbf{u} \quad \text{in } \Omega\ , \\ \mathbf{v} + \nabla\phi &= 0 \text{ in } \Omega\ , \quad \psi = \gamma\phi - f \text{ in } \Omega\ , \end{aligned} \quad \text{and} \quad \begin{aligned} \phi &= g \text{ on } \Gamma_D\ , \\ -\boldsymbol{n} \cdot \mathbf{u} &= h \text{ on } \Gamma_N\ . \end{aligned} \tag{2}$$

We will refer to the equations $\nabla \cdot \mathbf{u} + \psi = 0$ and $\mathbf{v} + \nabla\phi = 0$ as the *conservation laws*. The first one expresses the fact that the net amount of outflow, $\mathbf{u}$, over the surface of any body $\omega \subset \Omega$ balances the volumetric production terms ψ. The second equation states that circulation of $\mathbf{v}$ over any closed loop is zero. We call such equations *topological* because they are independent of material parameters and only involve geometric concepts like surface, body and closed loop. With proper selection of discrete variables these equations can be satisfied exactly.

On the other hand, the equations $\mathbf{v} = \mathbb{A}^{-1}\mathbf{u}$ and $\psi = \gamma\phi - f$ depend explicitly on the material parameters $\mathbb{A}$ and γ and the right hand side term f. We refer to these equations as the *constitutive relations*. Their association with geometry is less obvious; for instance $\mathbf{v} = \mathbb{A}^{-1}\mathbf{u}$ equates circulation of $\mathbf{v}$ along a curve to the flux of $\mathbf{u}$ across a surface. This geometrical incompatibility between the variables is an important source of errors in many numerical methods.

The two sets of equations play very different mathematical and physical roles. The constitutive relations prescribe functional relationships between the variables, which represent simplified summaries of more complex physical phenomena, i.e., these equations are *based on modeling assumptions*. The material-dependent data is generally obtained through experiment and is not known exactly. On the other hand, the conservation laws express fundamental balance relationships between global quantities that hold universally, i.e., these equations *do not involve modeling assumptions*. Let

$$H_D^1(\Omega) = \left\{\phi \in H^1(\Omega) \,|\, \phi = 0 \text{ on } \Gamma_D\right\} \ ,$$

$$H_N(div, \Omega) = \left\{\mathbf{u} \in H(div, \Omega) \,|\, \mathbf{u} = 0 \text{ on } \Gamma_N\right\} \ .$$

In this paper we consider a least-squares functional originally proposed in [5]:

$$\begin{aligned} \mathscr{J}((\phi, \mathbf{v}), (\psi, \mathbf{u}); f) = \frac{1}{2}\Big(&\left\| \mathbb{A}^{-1/2}\left(\mathbf{u} + \mathbb{A}\nabla\phi\right)\right\|_0^2 + \\ &\left\|\gamma^{-1/2}\left(\gamma\phi + \nabla\cdot\mathbf{u} - f\right)\right\|_0^2 + \left\|\mathbf{v} + \nabla\phi\right\|_0^2 + \left\|\nabla\cdot\mathbf{u} + \psi\right\|_0^2\Big) , \end{aligned} \tag{3}$$

and its associated least-squares principle

$$\min_{(\phi,\mathbf{v})\in U, (\psi,\mathbf{u})\in V} \mathscr{J}((\phi, \mathbf{v}), (\psi, \mathbf{u}); f) \tag{4}$$

where $U = H_D^1(\Omega) \times (L^2(\Omega))^n$ and $V = L^2(\Omega) \times H_N(div, \Omega)$.

Proposition 1 *The least-squares functional (3) is norm-equivalent with respect to the solution space $U = H_D^1(\Omega) \times (L^2(\Omega))^n$ and $V = L^2(\Omega) \times H_N(div, \Omega)$.*

Proof *See [5].*

Corollary 1 *Let $U^h \subset U$, $V^h \subset V$ and $(\phi^h, \mathbf{v}^h) \in U^h$, $(\psi^h, \mathbf{u}^h) \in V^h$ satisfy*

$$\{(\phi^h, \mathbf{v}^h), (\psi^h, \mathbf{u}^h)\} = \arg\min \mathscr{J}((\hat{\phi}^h, \hat{\mathbf{v}}^h), (\hat{\psi}^h, \hat{\mathbf{u}}^h); f)$$

Then, there exists a positive constant C such that

$$\begin{aligned} &\|\phi^h - \phi\|_1 + \|\mathbf{v}^h - \mathbf{v}\|_0 + \|\psi^h - \psi\|_0 + \|\mathbf{u}^h - \mathbf{u}\|_{div} \\ &\leq C \inf_{(\hat{\phi}^h,\hat{\mathbf{v}}^h)\in U^h, (\hat{\psi}^h,\hat{\mathbf{u}}^h)\in V^h} \left(\|\hat{\phi}^h - \phi\|_1 + \|\hat{\mathbf{v}}^h - \mathbf{v}\|_0 + \|\hat{\psi}^h - \psi\|_0 + \|\hat{\mathbf{u}}^h - \mathbf{u}\|_{div}\right) \end{aligned}$$

Proof Norm-equivalence of (3) implies that the associated Euler-Lagrange equation has coercive and bounded bilinear form. Then, by Céa's Theorem, the error in the least-squares solution is bounded by a constant times the best approximation of the exact solution out of the conforming space $U^h \times V^h$.

Proposition 2 *The solution of (4) satisfies the conservation laws in the L^2 sense.*

Proof *The proof follows by taking variations of (3) with respect to* $\mathbf{v}$ *and* ψ.

3 A Mimetic Least-Squares Method

Because strong coercivity is inherited on subspaces, conforming finite element spaces of $H^1_D(\Omega)$ and $H_N(div,\Omega)$ such as standard C^0 elements will give a well-posed least-squares finite element method. Since the inception of least-squares methods this has often been quoted as one of its principal advantages. However, if we want Proposition 2 to hold at the discrete level, we need to ensure that the discrete conservation laws, $\nabla\cdot\mathbf{u}+\psi=0$ and $\mathbf{v}+\nabla\phi=0$, can be represented on these subspaces, i.e. if $(\phi^h,\mathbf{v}^h)\in G^h\times C^h$ with $G^h\subset H^1_D(\Omega)$ and $C^h\subset\left(L^2(\Omega)\right)^n$, we need to have that $\nabla\phi^h\in C^h$ for all $\phi^h\in G^h$. Similarly, for $(\psi^h,\mathbf{u}^h)\in S^h\times D^h$ with $S^h\subset L^2(\Omega)$ and $D^h\subset H_N(div,\Omega)$ we require that $\nabla\cdot\mathbf{u}^h\in S^h$ for all $\mathbf{u}^h\in D^h$. Thus, the finite dimensional spaces forming $U^h=G^h\times C^h$ and $V^h=S^h\times D^h$ need to belong to a discrete DeRham complex, [4, 17]. With the spectral element basis functions from [11] this is indeed the case. With these spectral basis functions, the conservation laws can be exactly satisfied and reduce to simple relations between the expansion coefficients. In addition, the discrete conservation laws do not depend on the size or shape of the grid and will be independent of the order of the spectral element approximation. The discrete conservation laws only depend on the topology of the grid, see for instance [5, 16, 18] for a more extensive explanation.

Let $\Omega_0=[-1,1]^2$ be the reference spectral element with coordinates (ξ,η) and $\Phi:\Omega_0\to\Omega$, $(x,y)=\Phi(\xi,\eta)$. We expand the pullback of the potential, $\Phi^\star\phi^h$, in terms of a tensor product of Lagrange polynomials, h_i, associated with the GLL points of polynomial degree N in both ξ- and η-direction, see also [9] and [5] for the transformations

$$\Phi^\star\phi^h(\xi,\eta)=\sum_{i=0}^{N}\sum_{j=0}^{N}\phi_{i,j}h_i(\xi)h_j(\eta)\ , \tag{5}$$

and $\Phi^\star\mathbf{v}$ as

$$\Phi^\star\mathbf{v}^h(\xi,\eta)=\sum_{i=1}^{N}\sum_{j=0}^{N}u_{i,j}e_i(\xi)h_j(\eta)+\sum_{i=0}^{N}\sum_{j=1}^{N}v_{i,j}h_i(\xi)e_j(\eta)\ , \tag{6}$$

where the edge $e_i(\xi)$ are given by, [11], $e_i(\xi) = -\sum_{k=0}^{i-1} \mathrm{d}h_k(\xi)$. In terms of these expansions the conservation law $\mathbf{v} + \nabla\phi = 0$ assumes the form

$$\begin{aligned} \Phi^\star \mathbf{v}^h + \nabla\Phi^\star \phi^h = &\sum_{i=1}^{N}\sum_{j=0}^{N}(u_{i,j} + \phi_{i,j} - \phi_{i-1,j})e_i(\xi)h_j(\eta) + \\ &\sum_{i=0}^{N}\sum_{j=1}^{N}(v_{i,j} + \phi_{i,j} - \phi_{i,j-1})h_i(\xi)e_j(\eta) = 0 \,. \end{aligned} \tag{7}$$

Since basis functions are linear independent, (7) holds if and only if

$$u_{i,j} + \phi_{i,j} - \phi_{i-1,j} = 0 \quad \text{and} \quad v_{i,j} + \phi_{i,j} - \phi_{i,j-1} \,. \tag{8}$$

The pullback of the fluxes, $\Phi^\star \mathbf{u}^h$, is expanded in terms of tensor products of edge functions and Lagrange polynomials as, see [5, 11, 15] for details

$$\Phi^\star \mathbf{u}^h(\xi,\eta) = \sum_{i=0}^{N}\sum_{j=1}^{N} p_{i,j}h_i(\xi)e_j(\eta) - \sum_{i=1}^{N}\sum_{j=0}^{N} q_{i,j}e_i(\xi)h_j(\eta) \,. \tag{9}$$

Finally, the pullback of ψ^h, $\Phi^\star\psi^h$ is expanded as

$$\Phi^\star \psi^h(\xi,\eta) = \sum_{i=1}^{N}\sum_{j=1}^{N} \psi_{i,j}e_i(\xi)e_j(\eta) \,. \tag{10}$$

With these particular expansions the conservation law $\nabla\cdot\mathbf{u} + \psi = 0$ can be expressed as a relation between the expansion coefficients

$$p_{i,j} - p_{i-1,j} + q_{i,j} - q_{i,j-1} + \psi_{i,j} = 0 \,. \tag{11}$$

4 Numerical Example

In [5] we demonstrated the conservation properties of (3) on affine elements. In this paper we extend these results to non-affine, curvilinear grids.

In order to show that even in curvilinear coordinates the conservation laws are satisfied up to machine precision we solve the scalar diffusion-reaction problem on the spectral element grid shown in Fig. 1. The spectral element mesh consists of $K \times K$ elements

$$\begin{aligned} x(\xi,\eta) &= \xi + c\sin(\pi\xi)\sin(\pi\eta) \,, \\ y(\xi,\eta) &= \eta + c\sin(\pi\xi)\sin(\pi\eta) \,, \end{aligned} \qquad (\xi,\eta) \in [-1,1]^2 \,. \tag{12}$$

This curvilinear mesh was also used in [6, 9, 17].

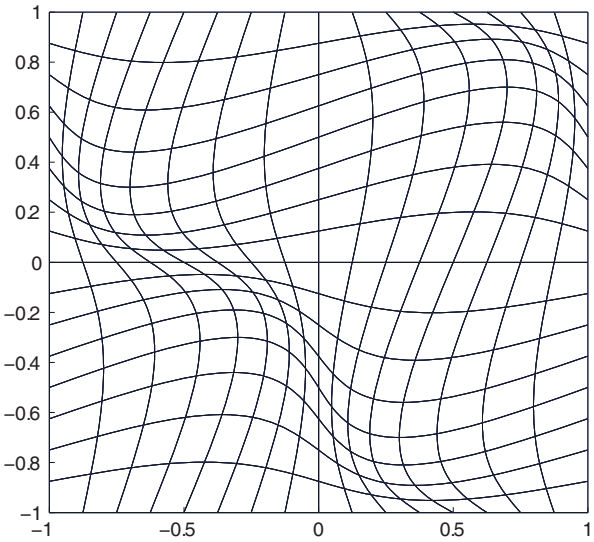


Fig. 1 Curvilinear coordinate system generated by the mapping (12) for $K = 16$

For this test problem we use $\mathbb{A} = \mathbb{I}$ and $\gamma = 1$ and as exact reference solution $\phi_{ex}(x, y) = \sin(\pi x)\sin(\pi y)$. Although the material parameters are trivial in the (x, y)-coordinates, this is no longer the case when the equations are transformed to (ξ, η)-coordinates, see [5]. In Fig. 2 h-convergence of the unknowns ϕ, $\mathbf{v}$, $\mathbf{u}$ and ψ in the L^2-norm is depicted for $K = 1, \ldots, 16$ and $N = 1, \ldots, 6$. The convergence rates are optimal in all unknowns, although the errors are higher than for an orthogonal grid. In Fig. 3 the residuals of $\nabla \cdot \mathbf{u} + \psi$ and $\nabla \times \mathbf{v}$ are plotted in the L^∞-norm as a function of $h = 2/K$ and N. The conservation relations are satisfied up to machine precision, independent of the mesh size, the particular mesh shape (i.e. curved grid) and polynomial degree. The slight increase in error with h-refinement and p-enrichment is a result of the increase in condition number, since in this study the full system resulting from (3) was solved. In practice this is not necessary, because if we know a priori that we can satisfy the conservation laws exactly, we might as well use the reduced functional

$$\mathscr{J}^R((\phi, \mathbf{u}); f) = \frac{1}{2}\left(\left\|\mathbb{A}^{-1/2}\left(\mathbf{u} + \mathbb{A}\nabla\phi\right)\right\|_0^2 + \left\|\gamma^{-1/2}\left(\gamma\phi + \nabla \cdot \mathbf{u} - f\right)\right\|_0^2\right), \tag{13}$$

and determine $\mathbf{v}$ from ϕ and ψ from $\mathbf{u}$ afterwards using (8) and (11). In summary,

when the reduced least-squares functional (13) is used to calculate ϕ^h and $\mathbf{u}^h$ and $\mathbf{v}^h$ and ψ^h are derived in a post-processing step using (8) and (11) and the associated expansions (7) and (10) for $\mathbf{v}^h$ and ψ^h, then for all meshes and all

(continued)

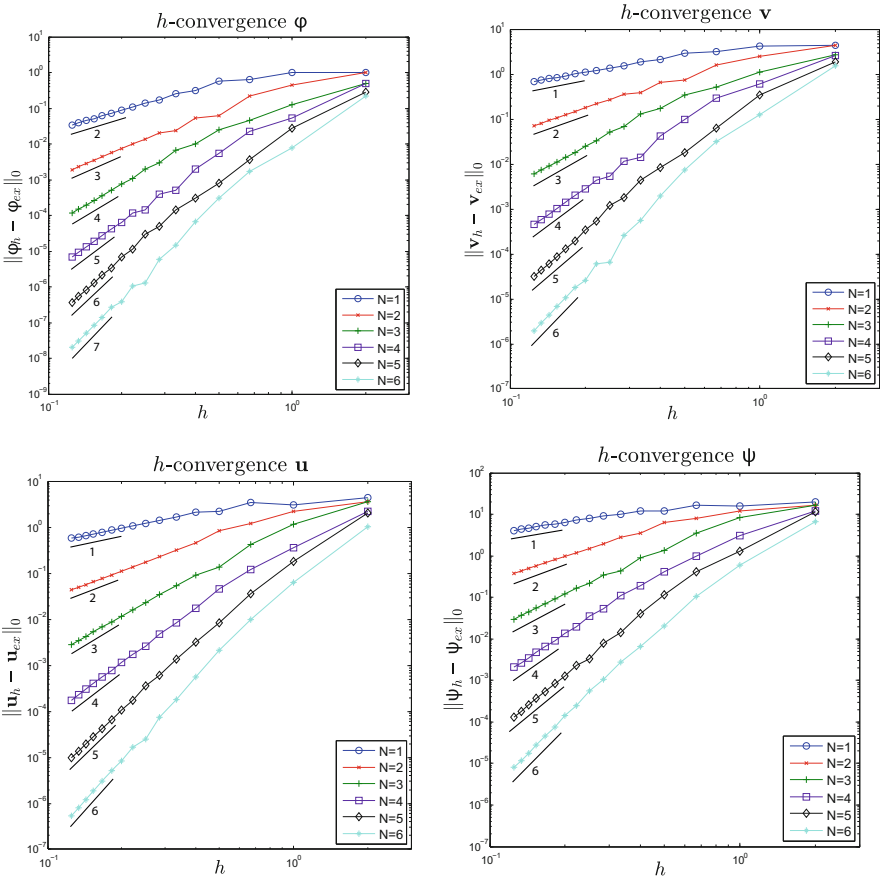


Fig. 2 Convergence plots of ϕ, $\mathbf{v}$, $\mathbf{u}$ and ψ with h-refinement for various polynomial approximations

polynomial degrees

$$\|\nabla \times \mathbf{v}^h\|_{L^\infty} = 0 \qquad \|\nabla \cdot \mathbf{u}^h + \psi^h\|_{L^\infty} = 0\,,$$

that is, the least-squares formulation is exactly locally conservative. Exact conservation is observed in the computations.

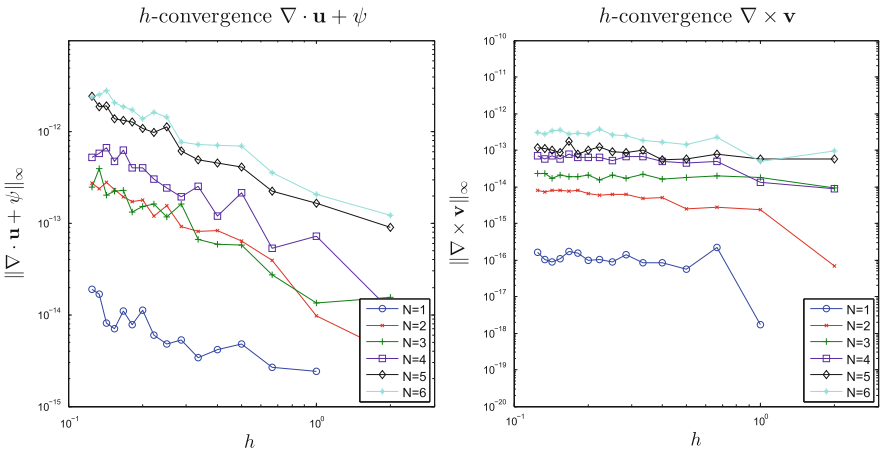


Fig. 3 Convergence plots of $\nabla \cdot \mathbf{u} + \psi$, and $\nabla \times \mathbf{v}$ with h-refinement for various polynomial approximations

5 Conclusions

Despite all its advantages, lack of conservation is one of the major drawbacks of least-squares finite element methods implemented using standard C^0 elements. In this paper we have shown that by combining an appropriate choice of a least-squares functional with compatible finite element spaces, one can define a least-squares method that is conservative up to a machine accuracy.

In practice, one can use the reduced functional (13) in which case the conservation laws are identically satisfied regardless of the coarseness and shape of the grid as well the approximation order. The price we pay is that we can no longer use our favorite C^0-elements.

Acknowledgements This material is based upon work supported by the U.S. Department of Energy, Office of Science, Office of Advanced Scientific Computing Research (ASCR).

References

1. J.H. Adler, P.S. Vassilevski, Error analysis for constrained first-order system least-squares finite element methods. SIAM J. Sci. Comput. **38**(3), A 1071–A 1088 (2014)
2. P.B. Bochev, M.D. Gunzburger, A locally conservative least-squares method for Darcy flows. Commun. Numer. Methods Eng. **24**, 97–110 (2008)
3. P.B. Bochev, M.D. Gunzburger, *Least-Squares Finite Element Methods* (Springer, New York, 2009)
4. P.B. Bochev, M.D. Gunzburger, A locally conservative mimetic least-squares finite element method for the Stokes equations, in *Proceedings of LSSC 2009*, ed. by I. Lirkov, S. Margenov,

J. Wasniewski. Springer Lecture Notes in Computer Science, vol. 5910 (Springer, Berlin/Heidelberg, 2009), pp. 637–644
5. P.B. Bochev, M.I. Gerritsma, A spectral mimetic least-squares method. Comput. Math. Appl. **68**, 1480–1502 (2014). http://dx.doi.org/10.1016/j.camwa.2014.09.014
6. P.B. Bochev, D. Ridzal, Rehabilitation of the lowest-order Raviart-Thomas element on quadrilateral grids. SIAM J. Numer. Anal. **47**(1), 487–507 (2008)
7. P.B. Bochev, J. Lai, L. Olson, A non-conforming least-squares finite element method for incompressible fluid flow problems. Int. J. Numer. Methods Fluids **72**, 375–402 (2013)
8. P. Bolton, R.W. Thatcher, On mass conservation in least-squares methods. J. Comput. Phys. **203**(1), 287–304 (2005)
9. M. Bouman, A. Palha, J.J. Kreeft, M.I. Gerritsma, A conservative spectral element method on curvilinear domains, in *Spectral and Higher Order Methods for Partial Differential Equations*, ed. by J. Hesthaven, R. Rønquist. Springer Lecture Notes in Computational Science and Engineering, vol. 76 (Springer, Berlin/Heidelberg, 2011), pp. 111–119
10. C.L. Chang, J.J. Nelson, Least-squares finite element method for the Stokes problem with zero residual of mass conservation. SIAM J. Numer. Anal. **34**(2), 480–489 (1997)
11. M.I. Gerritsma, Edge functions for spectral element methods, in *Spectral and Higher Order Methods for Partial Differential Equations*, ed. by J. Hesthaven, R. Rønquist. Springer Lecture Notes in Computational Science and Engineering, vol. 76 (Springer, Berlin/Heidelberg, 2011), pp. 199–208
12. J.J. Heys, E. Lee, T.A. Manteuffel, S.F. McCormick, An alternative least-squares formulation for the Navier-Stokes equations with improved mass conservation. J. Comput. Phys. **226**(1), 994–1006 (2007)
13. J.J. Heys, E. Lee, T.A. Manteuffel, S.F. McCormick, J.W. Ruge, Enhanced mass conservation in least-squares methods for Navier-Stokes equations. SIAM J. Sci. Comput. **31**(3), 2303–2321 (2009)
14. T. Kattelans, W. Heinrichs, Conservation of mass and momentum of the least-squares spectral element collocation scheme for the Stokes problem. J. Comput. Phys. **228**(13), 4649–4664 (2009)
15. J.J. Kreeft, M.I. Gerritsma, Mixed mimetic spectral element method for Stokes flow: a pointwise divergence-free solution. J. Comput. Phys. **240**, 284–309 (2013)
16. J.J. Kreeft, A. Palha, M.I. Gerritsma, Mimetic framework on curvilinear quadrilaterals of arbitrary order (2011). `arXiv:1111.4304`
17. A. Palha, M.I. Gerritsma, Spectral element approximations of the Hodge-$\star$ operator in curved elements, in *Spectral and Higher Order Methods for Partial Differential Equations*, ed. by J. Hesthaven, R. Rønquist. Springer Lecture Notes in Computational Science and Engineering, vol. 76 (Springer, Berlin/Heidelberg, 2011), pp. 283–291
18. A. Palha, P. Rebelo, R. Hiemstra, J.J. Kreeft, M.I. Gerritsma, Physics-compatible discretization techniques on single and dual grids, with application to the Poisson equation for volume forms. J. Comput. Phys. **257**, 1394–1422 (2014)
19. M.M.J. Proot, M.I. Gerritsma, Mass- and momentum conservation of the least-squares spectral element method for the Stokes problem. J. Sci. Comput. **27**, 389–401 (2006)

Nonlinear Compact Finite-Difference Schemes with Semi-Implicit Time Stepping

Debojyoti Ghosh and Emil M. Constantinescu

Abstract Atmospheric flows are characterized by a large range of length scales as well as strong gradients. The accurate simulation of such flows requires numerical algorithms with high spectral resolution, as well as the ability to provide nonoscillatory solutions across regions of high gradients. These flows exhibit a large range of time scales as well—the slowest waves propagate at the flow velocity and the fastest waves propagate at the speed of sound. Time integration with explicit methods are thus inefficient, although algorithms with semi-implicit time integration have been used successfully in past studies. We propose a finite-difference method for atmospheric flows that uses a weighted compact scheme for spatial discretization and implicit-explicit additive Runge-Kutta methods for time integration. We present results for a benchmark atmospheric flow problem and compare our results with existing ones in the literature.

1 Introduction

The simulation of atmospheric flows requires accurate numerical solutions of the compressible Navier-Stokes equations or the inviscid Euler equations if the physical viscosity and heat conduction are neglected. Such flows are characterized by localized flow structures and strong gradients, and numerical algorithms need a high spectral resolution and must be nonoscillatory across regions of strong gradients. Algorithms used for numerical weather prediction include finite-difference methods [13], finite-volume methods [1], and discontinuous Galerkin and spectral element methods [9, 10]. Although standard finite-difference methods suffer from poor spectral resolution, compact finite-difference methods [15] have significantly higher spectral resolution and have been applied to applications such as large eddy simulations and direct numerical simulations of turbulent flows [14, 18].

D. Ghosh (✉) • E.M. Constantinescu
Mathematics & Computer Science Division, Argonne National Laboratory, Argonne, IL, USA
e-mail: ghosh@mcs.anl.gov; emconsta@mcs.anl.gov

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_20

In this study, we propose a high-order finite-difference method for atmospheric flows based on compact-reconstruction weighted essentially nonoscillatory (CRWENO) schemes [5, 6, 8]. The CRWENO schemes combine the high spectral resolution of linear compact schemes with the solution-dependent stencil adaptation method of the WENO schemes [11, 17] to produce nonoscillatory solutions. Although discontinuities such as shock waves are not encountered in atmospheric flows, strong gradients often form that are resolved by very few grid points. The CRWENO schemes are thus well suited for simulating such flows. We explore implicit-explicit time-integration schemes based on a separation of stiff and nonstiff components of the governing equations [9]. We present results for a benchmark atmospheric flow problem.

2 Governing Equations

We consider the conservative form of the Euler equations based on the mass, momentum, and potential temperature for mesoscale flows (neglecting the Coriolis forces) [9]. These are given by

$$\frac{\partial}{\partial t}\begin{bmatrix}\rho'\\ \rho\mathbf{u}\\ \rho\theta\end{bmatrix} + \nabla\cdot\begin{bmatrix}\rho\mathbf{u}\\ \rho\mathbf{u}\otimes\mathbf{u}+p'\mathscr{I}\\ \rho\theta\mathbf{u}\end{bmatrix} = \begin{bmatrix}0\\ -\rho' g\hat{\mathbf{k}}\\ 0\end{bmatrix} \tag{1}$$

where ρ is the density, $\mathbf{u}$ is the velocity vector, p is the pressure, $\mathscr{I}$ is the identity matrix, and g is the acceleration due to gravity acting along the z-axis of the coordinate system with unit vector $\hat{\mathbf{k}}$. The potential temperature θ is given by

$$\theta = \frac{T}{\pi};\ \pi = \left(\frac{p}{p_0}\right)^{\frac{R}{C_P}}, \tag{2}$$

where T is the temperature, π is the Exner pressure, p_0 is the pressure at the surface (or reference altitude), R is the universal gas constant, and C_P is the constant pressure specific heat. The system of equations is completed by the equation of state, $p = p_0\left(\frac{\rho R\theta}{p_0}\right)^{\frac{C_P}{C_V}}$, where C_V is the constant volume specific heat. Equation (1) is expressed in terms of the density, pressure, and potential temperature perturbations (ρ', p', θ') that can be expressed as $(\cdot)' = (\cdot)(x,y,z,t) - \bar{(\cdot)}(z)$, where $\bar{(\cdot)}$ is the mean density, pressure, or potential temperature in hydrostatic balance $C_P\bar{\theta}\frac{d\bar{\pi}}{dz} = -g$. The governing equations form a system of hyperbolic partial differential equations (PDEs) and are solved by a conservative finite-difference algorithm.

3 Numerical Methodology

Equation (1) can be expressed as a system of hyperbolic conservation laws with a source term

$$\frac{\partial \mathbf{U}}{\partial t} + \frac{\partial \mathbf{f}_i(\mathbf{U})}{\partial x_i} = \mathbf{s}(\mathbf{U}), \ i = 1, \cdots, D, \tag{3}$$

where $\mathbf{U}$ is the solution, $\mathbf{f}_i$ is the flux along the ith dimension, $\mathbf{s}$ is the source term, and D is the number of dimensions. We describe the discretization of (3) in one dimension ($D = 1$); it can be trivially extended to multiple dimensions. A conservative, finite-difference spatial discretization of (3) on this grid results in a semi-discrete ordinary differential equation (ODE) in time,

$$\frac{d\mathbf{U}_j}{dt} + \frac{1}{\Delta x}\left[\hat{\mathbf{f}}_{j+1/2} - \hat{\mathbf{f}}_{j-1/2}\right] = \mathbf{s}_j, \ j = 1, \cdots, N, \tag{4}$$

where j denotes the grid index, $\mathbf{U}_j = \mathbf{U}(x_j)$ is the cell-centered solution, $\hat{\mathbf{f}}_{j+1/2}$ is the numerical flux at the cell interface $x_{j+1/2}$, and $\mathbf{s}_j$ is the source term evaluated at the cell center.

3.1 Reconstruction

We use the CRWENO scheme [5, 6, 8] to reconstruct the interface fluxes $\hat{\mathbf{f}}_{j+1/2}$ from the cell-centered flux $\mathbf{f}_j$. We briefly summarize the scheme in this section; a more complete description is available in [5]. The fifth-order CRWENO scheme (CRWENO5) is constructed by considering three third-order-accurate compact interpolation schemes for the flux function at the $(j + 1/2)$th interface:

$$\frac{2}{3}\hat{f}_{j-1/2} + \frac{1}{3}\hat{f}_{j+1/2} = \frac{1}{6}\left(f_{j-1} + 5f_j\right); \quad c_1 = \frac{2}{10}, \tag{5}$$

$$\frac{1}{3}\hat{f}_{j-1/2} + \frac{2}{3}\hat{f}_{j+1/2} = \frac{1}{6}\left(5f_j + f_{j+1}\right); \quad c_2 = \frac{5}{10}, \tag{6}$$

$$\frac{2}{3}\hat{f}_{j+1/2} + \frac{1}{3}\hat{f}_{j+3/2} = \frac{1}{6}\left(f_j + 5f_{j+1}\right); \quad c_3 = \frac{3}{10}. \tag{7}$$

Multiplying (5)–(7) with their optimal coefficients ($c_k, \ k = 1, 2, 3$) and adding, we obtain the fifth-order-accurate compact interpolation scheme,

$$\frac{3}{10}\hat{f}_{j-1/2} + \frac{6}{10}\hat{f}_{j+1/2} + \frac{1}{10}\hat{f}_{j+3/2} = \frac{1}{30}f_{j-1} + \frac{19}{30}f_j + \frac{1}{3}f_{j+1}. \tag{8}$$

We now compute weights ω_k based on the local smoothness of the solution [11] such that they converge to the corresponding optimal coefficient c_k when the solution is locally smooth, and approach zero at or near a discontinuity. They can be expressed as

$$\omega_k = \frac{\alpha_k}{\sum_k \alpha_k};\ \alpha_k = \frac{c_k}{(\epsilon + \beta_k)^p};\ k = 1, 2, 3, \tag{9}$$

where $\epsilon = 10^{-6}$ is a small number to prevent division by zero. The smoothness indicators (β_k) measure the local smoothness of the solution and are given by

$$\beta_1 = \frac{13}{12}(f_{j-2} - 2f_{j-1} + f_j)^2 + \frac{1}{4}(f_{j-2} - 4f_{j-1} + 3f_j)^2, \tag{10}$$

$$\beta_2 = \frac{13}{12}(f_{j-1} - 2f_j + f_{j+1})^2 + \frac{1}{4}(f_{j-1} - f_{j+1})^2, \tag{11}$$

$$\text{and } \beta_3 = \frac{13}{12}(f_j - 2f_{j+1} + f_{j+2})^2 + \frac{1}{4}(3f_j - 4f_{j+1} + f_{j+2})^2. \tag{12}$$

Multiplying (5)–(7) with ω_k instead of c_k, and adding, we obtain the CRWENO5 scheme:

$$\begin{aligned}\left(\frac{2}{3}\omega_1 + \frac{1}{3}\omega_2\right)\hat{f}_{j-1/2} + \left[\frac{1}{3}\omega_1 + \frac{2}{3}(\omega_2 + \omega_3)\right]\hat{f}_{j+1/2} + \frac{1}{3}\omega_3\hat{f}_{j+3/2} \\ = \frac{\omega_1}{6}f_{j-1} + \frac{5(\omega_1 + \omega_2) + \omega_3}{6}f_j + \frac{\omega_2 + 5\omega_3}{6}f_{j+1}.\end{aligned} \tag{13}$$

This scheme is fifth-order accurate when the solution ($\omega_k \to c_k$) is smooth, and it yields a nonoscillatory solution across discontinuities by biasing the interpolation stencil away from it. The standard fifth-order WENO scheme [11] is used to compute the flux at the physical boundaries [5]. Equation (13) requires the solution to a tridiagonal system of equations at each time-integration step or stage; however, past studies [5] demonstrated the higher computational efficiency of the CRWENO scheme compared with a standard finite-difference scheme. A scalable and efficient parallel implementation of the CRWENO5 scheme is discussed in [7]. This discussion describes the left-biased computation of the interface flux; the corresponding expressions for the right-biased interface flux can be similarly obtained. The final flux at a given interface is computed from the left- and right-biased approximations by using the Rusanov upwinding scheme [16].

3.2 Time Integration

Equation (4) is integrated in time by using explicit Runge-Kutta (ERK) and implicit-explicit additive Runge-Kutta (ARKIMEX) methods. Efficient implementations of these methods are available in the `TS` (time-stepping) module of `PETSc` [3, 4]. ERK methods are often inefficient, however, because the time-step size is restricted by the acoustic (fastest) wave. Implicit-explicit time-integration methods have been previously applied to atmospheric flows [9, 10]. We briefly summarize the separation of stiff and nonstiff components of the governing equations and its implicit-explicit discretization in time.

Equation (1) can be rearranged such that the right-hand side comprises a nonstiff term and a linear stiff term [9],

$$\frac{\partial \mathbf{U}}{\partial t} = \mathbf{S}(\mathbf{U}) + \mathbf{L}(\mathbf{U}), \tag{14}$$

$$\mathbf{U} = \begin{bmatrix} \rho' \\ \rho\mathbf{u} \\ \rho\theta' \end{bmatrix}, \ \mathbf{S}(\mathbf{u}) = -\nabla\cdot\begin{bmatrix} 0 \\ \rho\mathbf{u}\otimes\mathbf{u} \\ \rho\theta\mathbf{u} - \rho\bar{\theta}\mathbf{u} \end{bmatrix}, \ \mathbf{L}(\mathbf{u}) = -\begin{bmatrix} \nabla\cdot\rho\mathbf{u} \\ \nabla p' + g\rho'\hat{\mathbf{k}} \\ \nabla\cdot\rho\bar{\theta}\mathbf{u} \end{bmatrix},$$

where the pressure perturbation is linearized as $p' = \frac{\gamma\bar{p}}{\bar{\rho}\bar{\theta}}\left(\rho\theta - \bar{\rho}\bar{\theta}\right)$, with $\gamma = C_P/C_V$ as the specific heat ratio. The nonstiff component, $\mathbf{S}(\mathbf{U})$, of the right-hand side of (14) consists of terms that are second and higher order perturbations around the hydrostatic balance; and the linear stiff component, $\mathbf{L}(\mathbf{U})$, consists of terms that are first order perturbations. Equation (14) is spatially discretized and integrated in time by using the ARKIMEX methods [2, 12, 19], where an ERK method is applied to the nonstiff term and an ARK method is applied to the stiff term. This multistage procedure can be expressed as

$$\mathbf{U}^{(k)} = \mathbf{U}_n + \Delta t\sum_{i=1}^{k-1} a_{ki}\hat{\mathbf{S}}\left(\mathbf{U}^{(i)}\right) + \Delta t\sum_{i=1}^{k}\tilde{a}_{ki}\hat{\mathbf{L}}\left(\mathbf{U}^{(i)}\right), \ k = 1,\cdots,s, \tag{15}$$

$$\mathbf{U}_{n+1} = \mathbf{U}_n + \Delta t\sum_{i=1}^{s} b_i\hat{\mathbf{S}}\left(\mathbf{U}^{(i)}\right) + \Delta t\sum_{i=1}^{s}\tilde{b}_i\hat{\mathbf{L}}\left(\mathbf{U}^{(i)}\right), \tag{16}$$

where s is the number of stages, the superscripts of $\mathbf{U}$ indicate the stage index, and the subscripts of $\mathbf{U}$ indicate the time step. The coefficients a_{ki} and b_i specify the ERK method, and the coefficients $\tilde{a}_{ki}$ and $\tilde{b}_i$ specify the ARK method. $\hat{\mathbf{S}}$ and $\hat{\mathbf{L}}$ are the spatially discretized forms of $\mathbf{S}(\mathbf{U})$ and $\mathbf{L}(\mathbf{U})$, respectively.

Past applications of implicit-explicit time-integration to atmospheric flows [9, 10] used discontinuous Galerkin or spectral element methods for the discretization of spatial derivatives; these approaches resulted in (15) being a linear system. We, however, use a nonlinear finite-difference operator to discretize the spatial

derivative, as given by (4) and (13). Thus, $\hat{\mathbf{L}}$ is nonlinear even though $\mathbf{L}$ is linear, and (15) is a nonlinear system of equations. We make two comments on our algorithm in this context.

- We ensure that the discretized right-hand side $(\hat{\mathbf{S}}+\hat{\mathbf{L}})$ is consistent with the right-hand side of (14) by using the *same* finite-difference operator to discretize both $\mathbf{S}$ and $\mathbf{L}$. The nonlinear weights in (13) are computed based on the smoothness of $\mathbf{S}+\mathbf{L}$, and the resulting CRWENO5 scheme is applied to both terms.
- We linearize the finite-difference operator at each stage such that (15) is a linear system of equations. We compute the nonlinear weights in (13) at the beginning of stage k based on the smoothness of $(\mathbf{S}+\mathbf{L})\left(\mathbf{U}^{(k-1)}\right)$ (or $(\mathbf{S}+\mathbf{L})(\mathbf{U}_n)$ for $k=1$); and we solve (15) as a linear system (since, once the nonlinear weights are fixed, (13) is a linear operator).

The linear system is solved using the generalized residual method (GMRES) method [20] implemented in the `KSP` (linear equations solvers) module of `PETSc`. The current implementation does not apply any preconditioning; the derivation of effective preconditioners for this application is a subject of active research.

4 Results

We verify our algorithm by solving the two-dimensional inertia-gravity wave problem [13]. The domain is a periodic channel with dimensions 300,000 × 10,000 m. Zero-flux boundary conditions are specified at the top and bottom boundaries. The initial atmosphere has a mean flow of 20 m/s and is uniformly stratified with a Brunt-Vaisala frequency of $\mathscr{N} = 0.01$/s [9, 13]. A perturbation in the potential temperature is introduced as

$$\theta' = \theta_c \left[\sin\left\{(\pi_c z)\,(h_c)^{-1}\right\}\right]\left[1+\left\{(x-x_c)\,a_c^{-1}\right\}\right]^{-2}, \tag{17}$$

where $\theta_c = 0.01$ K, $h_c = 10{,}000$ m, $a_c = 5000$ m, $x_c = 100{,}000$ m, and π_c is the trigonometric constant. Solutions are obtained at a final time of 3000 s. Figure 1a shows the potential temperature perturbation (θ') contours for a solution obtained with the CRWENO5 scheme on a grid with 1200 × 50 points. The solution is integrated in time with the second-order-accurate, two-stage ARKIMEX 2C method at a CFL of 8. We observe good agreement with results in the literature [1, 9, 13]. The cross-sectional variation of the potential temperature perturbation through $z = 5000$ m is shown in Fig. 1b for the solutions obtained with the CRWENO5 as well as the fifth-order WENO (WENO5) [11] schemes. The explicit four-stage, fourth-order Runge-Kutta (RK4) and the three-stage, third-order ARKIMEX (ARKIMEX3) methods are used to integrate the solution in time. The absolute and relative tolerances for the linear solver are specified as 10^{-6}. Excellent agreement is observed for all the methods with the reference solution, obtained by using the

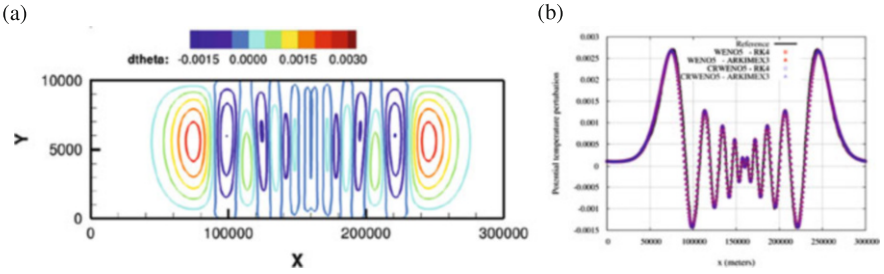


Fig. 1 Solutions of the inertia-gravity wave problem obtained on a grid with 1200 × 50 points. (**a**) Potential temperature perturbation contours. (**b**) Cross-sectional variation of potential temperature perturbation

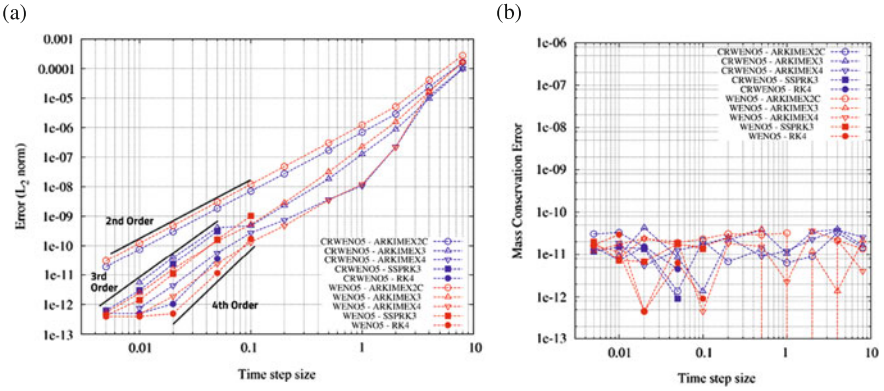


Fig. 2 Error analysis on a grid with 8192 × 256 points. (**a**) L_2 norm of the error as a function of time step size. (**b**) Mass conservation error as a function of time step size

spectral element method with 10th-order polynomials and 250-m grid resolution [9]. Figure 2a shows the L_2 norm of the error as a function of the time-step sizes for solutions obtained on a grid with 8192 × 256 points. The reference solution is computed with the strong-stability-preserving three-stage, third-order Runge-Kutta (SSPRK3) scheme and a small time-step size of 0.0005. We consider two ERK schemes, SSPRK3 and RK4, and three ARKIMEX schemes, ARKIMEX2C, ARKIMEX3, and ARKIMEX4 (four-stage, fourth-order). The semi-implicit solutions are obtained by specifying the absolute and relative tolerances for the linear solver as 10^{-12} and 10^{-10}, respectively. The methods converge at their theoretical convergence rates. Figure 2b shows the error in mass conservation for the various methods and time-step sizes. Mass is conserved to round-off error for all the methods considered.

5 Conclusions

A high-order-accurate finite-difference method for the simulation of atmospheric flows is proposed in this paper. The algorithm uses the CRWENO scheme for spatial discretization and the ARKIMEX schemes for time integration. The semi-implicit ARKIMEX schemes result in a time-step size that is not restricted by the acoustic waves. The algorithm is applied to a benchmark atmospheric flow problem, and solutions show excellent agreement with existing results in the literature. The semi-implicit time-integrators exhibit optimal convergence and conservative behavior.

Acknowledgements This material is based upon work supported by the U.S. Department of Energy, Office of Science, Advanced Scientific Computing Research, under contract DE-AC02-06CH11357.

References

1. N. Ahmad, J. Lindeman, Euler solutions using flux-based wave decomposition. Int. J. Numer. Methods Fluids **54**(1), 47–72 (2007). doi:10.1002/fld.1392
2. U.M. Ascher, S.J. Ruuth, R.J. Spiteri, Implicit-explicit Runge-Kutta methods for time-dependent partial differential equations. Appl. Numer. Math. **25**(2–3), 151–167 (1997). doi:10.1016/S0168-9274(97)00056-1
3. S. Balay, J. Brown, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Users Manual. Tech. Rep. ANL-95/11 - Revision 3.4, Argonne National Laboratory (2013)
4. S. Balay, J. Brown, K. Buschelman, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, H. Zhang, PETSc Web page (2013). http://www.mcs.anl.gov/petsc
5. D. Ghosh, J.D. Baeder, Compact reconstruction schemes with weighted ENO limiting for hyperbolic conservation laws. SIAM J. Sci. Comput. **34**(3), A1678–A1706 (2012). doi:10.1137/110857659
6. D. Ghosh, J.D. Baeder, Weighted non-linear compact schemes for the direct numerical simulation of compressible, turbulent flows. J. Sci. Comput. **61**(1), 61–89 (2014). doi:10.1007/s10915-014-9818-0
7. D. Ghosh, E.M. Constantinescu, J. Brown, Efficient implementation of nonlinear compact schemes on massively parallel platforms. SIAM J. Sci. Comput. **37**(3), C354–C383 (2015). doi:10.1137/140989261
8. D. Ghosh, S. Medida, J.D. Baeder, Application of compact-reconstruction weighted essentially nonoscillatory schemes to compressible aerodynamic flows. AIAA J. **52**(9), 1858–1870 (2014). doi:10.2514/1.J052654
9. F. Giraldo, M. Restelli, M. Läuter, Semi-implicit formulations of the Navier-Stokes equations: application to nonhydrostatic atmospheric modeling. SIAM J. Sci. Comput. **32**(6), 3394–3425 (2010). doi:10.1137/090775889
10. F. Giraldo, J. Kelly, E. Constantinescu, Implicit-explicit formulations of a three-dimensional nonhydrostatic unified model of the atmosphere (NUMA). SIAM J. Sci. Comput. **35**(5) (2013). doi:10.1137/120876034
11. G.S. Jiang, C.W. Shu, Efficient implementation of weighted ENO schemes. J. Comput. Phys. **126**(1), 202–228 (1996). doi:10.1006/jcph.1996.0130

12. C.A. Kennedy, M.H. Carpenter, Additive Runge-Kutta schemes for convection-diffusion-reaction equations. Appl. Numer. Math. **44**(1–2), 139–181 (2003). doi:10.1016/S0168-9274(02)00138-1
13. J.B. Klemp, W.C. Skamarock, J. Dudhia, Conservative split-explicit time integration methods for the compressible nonhydrostatic equations. Mon. Weather Rev. **135**, 2897–2913 (2007). doi:10.1175/MWR3440.1
14. C. Lee, Y. Seo, A new compact spectral scheme for turbulence simulations. J. Comput. Phys. **183**(2), 438–469 (2002). doi:10.1006/jcph.2002.7201
15. S.K. Lele, Compact finite difference schemes with spectral-like resolution. J. Comput. Phys. **103**(1), 16–42 (1992). doi:10.1016/0021-9991(92)90324-R
16. R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems*. Cambridge Texts in Applied Mathematics (Cambridge University Press, Cambridge, 2002)
17. X.D. Liu, S. Osher, T. Chan, Weighted essentially non-oscillatory schemes. J. Comput. Phys. **115**(1), 200–212 (1994). doi:10.1006/jcph.1994.1187
18. S. Nagarajan, S.K. Lele, J.H. Ferziger, A robust high-order compact method for large eddy simulation. J. Comput. Phys. **191**(2), 392–419 (2003). doi:10.1016/S0021-9991(03)00322-X
19. L. Pareschi, G. Russo, Implicit-explicit Runge-Kutta schemes and applications to hyperbolic systems with relaxation. J. Sci. Comput. **25**(1–2), 129–155 (2005). doi:10.1007/BF02728986
20. Y. Saad, M.H. Schultz, GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. SIAM J. Sci. Stat. Comput. **7**(3), 856–869 (1986). doi:10.1137/0907058

Unsteady Simulations of Rotor Stator Interactions Using SBP-SAT Schemes: Status and Challenges

G. Giangaspero, M. Almquist, K. Mattsson, and E. van der Weide

Abstract Recent developments in the SBP-SAT method have made available high-order interpolation operators (Mattsson and Carpenter, SIAM J Sci Comput 32(4):2298–2320, 2010). Such operators allow the coupling of different SBP methods across nonconforming interfaces of multiblock grids while retaining the three fundamental properties of the SBP-SAT method: strict stability, accuracy, and conservation. As these interpolation operators allow a more flexible computational mesh, they are appealing for complex geometries. Moreover, they are well suited for problems involving sliding meshes, like rotor/stator interactions, wind turbines, helicopters, and turbomachinery simulations in general, since sliding interfaces are (almost) always nonconforming. With such applications in mind, this paper presents an accuracy analysis of these interpolation operators when applied to fluid dynamics problems on moving grids. The classical problem of an inviscid vortex transported by a uniform flow is analyzed: the flow is governed by the unsteady Euler equations and the vortex crosses a sliding interface. Furthermore, preliminary studies on a rotor/stator interaction are also presented.

1 Introduction

The SBP-SAT framework has been developed considerably during the last two decades. While it has already been successfully applied to many different problems [6], it is not yet suitable for turbomachinery cases. One of the main reasons for this is the lack of a consistent (stable and accurate) treatment of sliding interfaces, and of nonconforming interfaces in general. In the industrial environment, turbomachinery problems are typically solved with low-order discretization techniques for which

G. Giangaspero (✉) • E. van der Weide
University of Twente, Enschede, The Netherlands
e-mail: g.giangaspero@utwente.nl; e.t.a.vanderweide@utwente.nl

M. Almquist • K. Mattsson
Uppsala University, Uppsala, Sweden
e-mail: martin.almquist@it.uu.se; ken.mattsson@it.uu.se

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_21

ways of handling such interfaces are readily available. This is not the case for high-order (not only SBP-SAT) schemes: nonconforming interfaces are still a major problem. However, high-order schemes are receiving a constantly growing attention thanks to their better computational efficiency [7], that is less computational work for a given accuracy. Computational efficiency is highly appreciated by turbomachinery designers, who have to perform computations with large number of unknowns in tight turn-around times. Moreover, general high-order interpolation operators for nonconforming interfaces, besides being necessary for sliding interfaces, would have the added benefit of allowing a more flexible computational mesh for complex geometries.

Within the SBP-SAT framework, consistent interpolation operators have been recently proposed for static interfaces [2]. In this work, we adapt the interpolation operators in [2] to sliding interfaces (see Sect. 2). In Sect. 3, we verify their design accuracy with the classic test case of the Euler vortex. Then, in Sect. 4 we apply them to an academic rotor-stator interaction test case. Finally, Sect. 5 provides our conclusions and future work.

2 Interpolation Operators

In conforming meshes, the SAT term of a vertex on one side of the interface between computational blocks is proportional to the difference between the solution in that vertex and the penalty state, i.e. the target solution, which is the solution in the overlapping vertex on the other side of the interface [4]. By definition, in a conforming mesh there is a 1to1 matching between the vertices at the interface, so the penalty state is clearly defined. In a sliding interface, however, such 1to1 relations cease to exist and interpolation is needed. Therefore, our aim is to find the narrowest possible interpolation stencil that both preserves high-order accuracy and leads to a stable discretization. In this work, we make the following simplifying assumptions:

1. the problem is 2D, thus the interpolation is 1D
2. the mesh spacing is constant on both sides of the interface
3. same number of points on both sides of the interface (1:1 compression ratio)
4. the problem is periodic in the direction parallel to the interface.

Assumptions 2 and 3 imply that the sliding interface can reside on one face only of the computational block. Assumption 4 implies that there is no boundary closure and therefore the same stencil can be used throughout the interface. For example, referring to Fig. 2, periodic boundary conditions are applied at the top and bottom computational boundaries and hence, no special boundary treatment is needed. The procedure to construct the operators for a more general case, where assumptions 3 and 4 are removed, can be found in [2]. Here we present the second and fourth order accurate interpolation operators resulting from the assumptions outlined above.

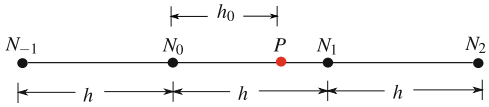


Fig. 1 Stencil for fourth order interpolation operator

Figure 1 shows the stencil for the fourth order operator. There we introduce a general notation for all stencils: P is the vertex on one side of the interface where we wish to calculate the penalty state; the penalty state in P is a weighted sum of the solution in the neighboring points N_i, which reside on the other side of the interface. The second order stencil uses two neighbors and their interpolation weights are given in Eq. (1).

$$Sol(P) = w_0 Sol(N_0) + w_1 Sol(N_1) \tag{1a}$$

$$w_0 = (h - h_0)/h, \qquad w_1 = h_0/h \tag{1b}$$

The fourth order stencil uses four neighbors, and their interpolation weights are given in Eq. (2). The sixth and eighth order stencils are presented in the Appendix.

$$Sol(P) = \sum_{i=-1}^{2} w_i Sol(N_i), \qquad \alpha = (h - h_0)/h \tag{2a}$$

$$w_{-1} = -1/6(\alpha - 2)(\alpha - 1)(\alpha), \qquad w_0 = +1/2(\alpha - 2)(\alpha - 1)(1 + \alpha), \tag{2b}$$

$$w_{+1} = -1/2(\alpha - 2)(\alpha)(1 + \alpha), \qquad w_{+2} = +1/6(\alpha - 1)(\alpha)(1 + \alpha). \tag{2c}$$

In order to have a consistent discretization, we use the interpolation operator that corresponds to the accuracy of the interior stencil of the scheme. We employ SBP-SAT schemes with diagonal norms (see, for example, [5] to explain this terminology), for which the design accuracy is s in a few points near the boundary and p in the interior, where $s = p/2$. This leads to a scheme with global accuracy of order $s + 1$ measured in the L^2-norm. Therefore, the fourth order interpolation operator is used with the third order scheme (which is second order accurate at the boundary and fourth in the interior); the sixth order interpolation operator is used with the fourth order scheme, and the eighth order interpolation operator is used with the fifth order scheme.

Referring again to [2], for stability to be proven, the following condition must be met:

$$H_R^y I_{L2R} = I_{R2L}^T H_L^y \tag{3}$$

where H_L^y and H_R^y are the norms in the direction parallel to the interface in the left and right domain, respectively; I_{L2R} and I_{R2L} are the interpolation operators. Because of assumption 4, H_L^y and H_R^y reduce to the identity matrix. It remains to show that $I_{L2R} = I_{R2L}^T$, which is easily verified by computing the weights w_i on both sides of the interface.

3 Accuracy Study: The Euler Vortex Test Case

This classic test case is used here to verify the design accuracy of the interpolation operators. The unsteady 2D Euler equations govern the simulation, which consists of a 2D vortex transported by a uniform flow across two rectangular computational domains of dimensions $(x, y) = (-L_x, 0) \times (-0.5L_y, 0.5L_y)$ and $(x, y) = (0, L_x) \times (-0.5L_y, 0.5L_y)$, see Fig. 2. The initial configuration of the vortex, centered in (x_c, y_c) and superimposed onto the uniform (infinity) flow, is given by the following equations:

$$\rho_0 = \rho_\infty \left(T_0/T_\infty\right)^{1/(\gamma-1)}, \qquad u_0 = U_\infty + \delta u, \qquad v_0 = \delta v, \qquad T_0 = T_\infty - \delta T \tag{4}$$

where

$$\delta u = -(\beta U_\infty)\frac{y - y_c}{R} e^{-r^2/2}, \quad \delta v = (\beta U_\infty)\frac{x - x_c}{R} e^{-r^2/2}, \quad \delta T = \frac{0.5}{C_p}(\beta U_\infty)^2 e^{-r^2}$$

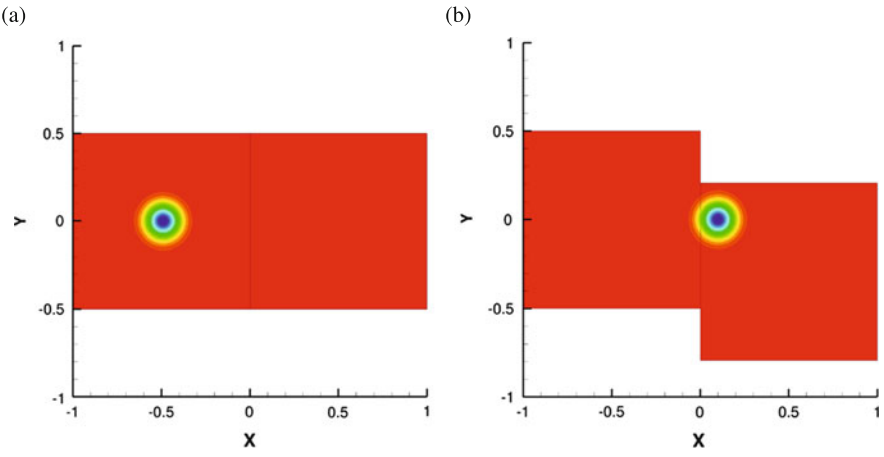


Fig. 2 Euler vortex problem. $N_x = N_y = 65$, fifth order solution (density). (**a**) Initial solution. (**b**) Solution at $t = 0.6t_{final} = 0.6L_x/U_\infty$

and $C_p = \gamma R_{gas}/(\gamma - 1)$, $r = \sqrt{(x - x_c)^2 + (y - y_c)^2}/R$. The variable R represents the vortex characteristic radius while β defines its strength; $\gamma = 1.4$ is the constant specific heat ratio, $R_{gas} = 287.87\,\mathrm{J/(kg\,K)}$ is the gas constant and $U_\infty = M_\infty \sqrt{\gamma R_{gas} T_\infty}$ and $\rho_\infty = p_\infty/(T_\infty R_{gas})$ are the velocity and density of the unperturbed flow, respectively. We set $M_\infty = 0.5$, $\beta = 1/2\pi$, $R = 0.1$, $L_x = L_y = 1$, $x_c = -0.5$, $y_c = 0.0$.

The (analytic) solution is steady in the frame of reference moving with the free-stream. The flow is periodic in the y direction, therefore no boundary closure is needed at the interface and there we can use the interpolation operators. The analytic solution is used as boundary data at the left and right boundaries. While the left block is fixed, the right block is oscillating with a frequency of $f = U_\infty/L_x$ and an amplitude of $0.5L_y$.

For the SBP-SAT discretization of the Euler equations, we refer to [2], where static interfaces were considered. To cope with the sliding interface, the interpolation operators I_{C2F} and I_{F2C} presented therein must be replaced by the novel operators I_{L2R} and I_{R2L}, derived in the present study.

The solution is advanced in time with an explicit third order TVD Runge-Kutta scheme for 1 characteristic time ($t_{final} = L_x/U_\infty$), such that the vortex has to travel across the interface once. The time step is chosen such that the error due to the temporal discretization is negligible with respect to the error due to the spatial discretization. In this configuration, 3000 time steps were used. As the problem is almost linear, no artificial dissipation was necessary for this case.

Since the vortex should be transported without distortion, the L^2 norm of the error can be defined as

$$L^2_{\text{error}} = \left[\left(\sum_{i=1}^{N_{tot}} \text{error}_i^2 \right) / N_{tot} \right]^{1/2}, \quad \text{error}_i = \phi_i^{\text{final}} - \phi_i^{\text{analytic}}, \quad i = 1, \ldots, N_{tot}$$

where ϕ is one of the conserved variables; ϕ^{final} is the numerical solution at $t = t_{final}$; the analytic solution is computed from Eq. (4) with ($x_c = 0.5, y_c = 0.0$), and N_{tot} is the total number of grid points.

The convergence rates of the density error are reported in Table 1; similar values were obtained for the other conserved variables. The global accuracy of the schemes is verified, hence the interpolation operators show the design accuracy.

Table 1 Euler vortex: L^2_{error} norms and convergence rates for the density error obtained with the different schemes

	Second order		Third order		Fourth order		Fifth order	
N	L^2_{error}	Conv. Rate	L^2_{error}	Conv. Rate	L^2_{error}	Conv. Rate	L^2_{error}	Conv. Rate
33	1.176e−4	–	3.365e−5	–	6.372e−5	–	5.623e−5	–
65	3.216e−5	1.87	4.128e−6	3.03	5.369e−6	3.57	2.032e−6	4.79
129	8.042e−6	2.00	4.971e−7	3.05	3.618e−7	3.89	5.802e−8	5.13

N ($= N_x = N_y$) is the number of grid points in the two directions for each block

4 A Rotor-Stator Interaction Problem

To further illustrate the applicability of the sliding interface treatment, we consider a linear cascade problem. Originally designed at the Duke University [1], the cascade consists of a $1\&1/2$ stages (stator-rotor-stator) compressor. In order to reduce the grid complexity and computational costs, the geometry has been scaled down to a 3-4-5 configuration from the original 16-20-25 configuration. The coarse grid, used to obtain a cheap initial solution, is shown in Fig. 3. The fine mesh consists of 54 computational blocks, $\approx$ 650,000 vertices. The mesh topology was chosen in order to comply with the simplifying assumptions of Sect. 2; that is, only one block on each side of the interface and $1:1$ matching. However, this forced us to introduce sub-faces in the computational blocks involved in the sliding interfaces, see the red block in Fig. 3b. Sub-faces, also known as T-junctions, are not accounted for in the stability of the SBP-SAT schemes of third order and higher, and are a relatively recent topic of research. A possible solution has been proposed in [3], however it has not been implemented yet in our code and no ad-hoc treatment has been employed.

Characteristic far-field boundary conditions are used at inlet and outlet, and the flow is assumed to be periodic in the y direction. The rotor travels at a speed of 1.25 m/s in vertical direction and the total compression ratio is 1.6. The flow is inviscid and subsonic, and the solution is advanced in time until the initial disturbances are smoothened out and a periodic solution is obtained.

We computed the second and third order solution. The fourth and fifth order schemes were not stable. The cause of this behavior is under investigation but we believe the most likely culprit is the non-stable handling of sub-faces. Figure 4a shows the instantaneous pressure field at a particular time during the periodic

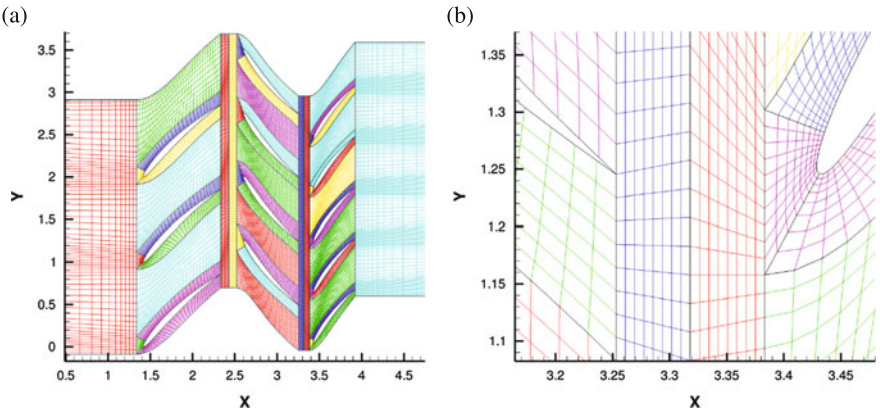


Fig. 3 Coarse mesh for the linear cascade problem. (**a**) Coarse mesh. (**b**) 1:1 matching at the interface and sub-faces (right boundary of the red block)

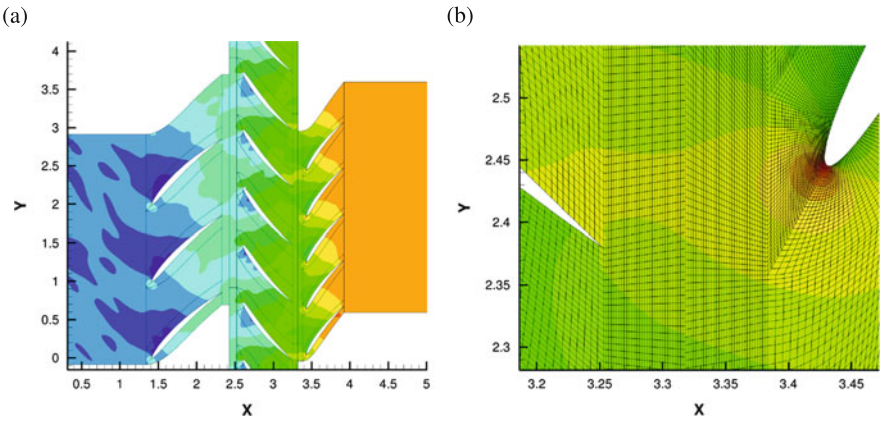


Fig. 4 Pressure contours of the third order instantaneous solution of the stator-rotor-stator calculation. (**a**) Global view. (**b**) Zoom at the sliding interface: the solution is smooth, the mesh is not

regime. The third order accurate method is used. Figure 4b shows the pressure contours and the mesh close to the rotor-stator interface (the same considerations hold for the stator-rotor interface). The mesh is clearly not continuous, but the solution is.

5 Conclusions and Future Work

We have constructed stable and accurate SBP-SAT interpolation operators for sliding interfaces under some simplifying assumptions. In order to verify their design accuracy, we have applied them to the classic test case of the Euler vortex (Sect. 3). A more involved test case, a linear cascade, has been computed as well (Sect. 4). Promising results were obtained: design accuracy is verified, the solution is smooth over the sliding interfaces and there are no reflections. However, the limitations of the operators presented in this work and in [2] are simply too restricting for real-life (industrial) applications. For those kind of problems we need generalized interpolation operators, which should be able to handle 2D interpolation, any compression/expansion ratio, non-constant spacings, and arbitrary number of (sub-)faces. Generalized operators are not only necessary for the calculations of sliding interfaces, but would also be very beneficial for nonconforming fixed interfaces in general. For example, the mesh generation of some complicated geometrical features, like blade cooling holes, would be greatly simplified. While this work is a step forward towards a fully consistent SBP-SAT discretization technique for turbomachinery problems, there are clearly many more to take. However, once generalized operators become available, real-life turbomachinery problems will be at reach.

Acknowledgements The authors would like to thank Mark Carpenter for his contribution to this work. His name is not amongst the authors' due to administrative reasons. Part of the research leading to these results has received funding through the project COPA-GT (European Union's Seventh Framework Programme FP7/2007–2013, REA grant agreement No. PITN-GA-2011-290042).

Appendix

The interpolation weights for the sixth order operator are (see Fig. 1):

$$Sol(P) = \sum_{i=-2}^{3} w_i Sol(N_i), \qquad \alpha = (h - h_0)/h$$

$$\begin{aligned}
w_{-2} &= -1/120 \quad (\alpha-3)(\alpha-2) \quad (\alpha-1)(\alpha) \quad (1+\alpha)\\
w_{-1} &= +1/24 \quad (\alpha-3)(\alpha-2) \quad (\alpha-1)(\alpha) \quad (2+\alpha)\\
w_{0} &= -1/12 \quad (\alpha-3)(\alpha-2) \quad (\alpha-1)(1+\alpha) \quad (2+\alpha)\\
w_{+1} &= +1/12 \quad (\alpha-3)(\alpha-2) \quad (\alpha)(1+\alpha) \quad (2+\alpha)\\
w_{+2} &= -1/24 \quad (\alpha-3)(\alpha-1) \quad (\alpha)(1+\alpha) \quad (2+\alpha)\\
w_{+3} &= +1/120 \quad (\alpha-2)(\alpha-1) \quad (\alpha)(1+\alpha) \quad (2+\alpha)
\end{aligned}$$

The interpolation weights for the eighth order operator are (see Fig. 1):

$$Sol(P) = \sum_{i=-3}^{4} w_i Sol(N_i), \qquad \alpha = (h - h_0)/h$$

$$\begin{aligned}
w_{-3} &= -1/5040 \quad (\alpha-4)(\alpha-3) \quad (\alpha-2)(\alpha-1) \quad (\alpha)(\alpha+1) \quad (\alpha+2)\\
w_{-2} &= +1/720 \quad (\alpha-4)(\alpha-3) \quad (\alpha-2)(\alpha-1) \quad (\alpha)(\alpha+1) \quad (\alpha+3)\\
w_{-1} &= -1/240 \quad (\alpha-4)(\alpha-3) \quad (\alpha-2)(\alpha-1) \quad (\alpha)(\alpha+2) \quad (\alpha+3)\\
w_{0} &= +1/144 \quad (\alpha-4)(\alpha-3) \quad (\alpha-2)(\alpha-1) \quad (\alpha+1)(\alpha+2) \quad (\alpha+3)\\
w_{+1} &= -1/144 \quad (\alpha-4)(\alpha-3) \quad (\alpha-2)(\alpha) \quad (\alpha+1)(\alpha+2) \quad (\alpha+3)\\
w_{+2} &= +1/240 \quad (\alpha-4)(\alpha-3) \quad (\alpha-1)(\alpha) \quad (\alpha+1)(\alpha+2) \quad (\alpha+3)\\
w_{+3} &= -1/720 \quad (\alpha-4)(\alpha-2) \quad (\alpha-1)(\alpha) \quad (\alpha+1)(\alpha+2) \quad (\alpha+3)\\
w_{+4} &= +1/5040 \quad (\alpha-3)(\alpha-2) \quad (\alpha-1)(\alpha) \quad (\alpha+1)(\alpha+2) \quad (\alpha+3)
\end{aligned}$$

References

1. K. Ekici, K.C. Hall, Nonlinear analysis of unsteady flows in multistage turbomachines using harmonic balance. AIAA J. **45**(5), 1047–1057 (2007)
2. K. Mattsson, M. Carpenter, Stable and accurate interpolation operators for high-order multiblock finite difference methods. SIAM J. Sci. Comput. **32**(4), 2298–2320 (2010)
3. A. Nissen, K. Kormann, M. Grandin, K. Virta, Stable difference methods for block-oriented adaptive grids. J. Sci. Comput. 1–26 (2014). doi:10.1007/s10915-014-9969-z
4. J. Nordström, J. Gong, E. van der Weide, M. Svärd, A stable and conservative high order multi-block method for the compressible Navier-Stokes equations. J. Comput. Phys. **228**(24), 9020–9035 (2009)
5. M. Svärd, On coordinate transformations for Summation-by-Parts operators. J. Sci. Comput. **20**, 29–42 (2004)
6. M. Svärd, J. Nordström, Review of Summation-By-Parts schemes for initial-boundary-value problems. J. Comput. Phys. **268**, 17–38 (2014)
7. Z.J. Wang, et al., High-order CFD methods: current status and perspective. Int. J. Numer. Methods Fluids **72**, 811–845 (2013)

Degree and Wavenumber [In]dependence of Schwarz Preconditioner for the DPG Method

Jay Gopalakrishnan and Joachim Schöberl

Abstract This note describes an implementation of a discontinuous Petrov Galerkin (DPG) method for acoustic waves within the framework of high order finite elements provided by the software package NGSolve. A technique to impose the impedance boundary condition weakly is indicated. Numerical results from this implementation show that a multiplicative Schwarz algorithm, with no coarse solve, provides a p-preconditioner for solving the DPG system. The numerical observations suggest that the condition number of the preconditioned system is independent of the frequency k and the polynomial degree p.

1 A Petrov Galerkin Formulation

We consider the Helmholtz equation modeling time harmonic acoustic waves in a homogenous medium,

$$-\Delta u - k^2 u = f \qquad \text{on } \Omega \tag{1a}$$

$$u = 0 \qquad \text{on } \partial\Omega. \tag{1b}$$

Here we have set the simplest Dirichlet boundary condition (postponing the case of impedance boundary condition to later), and Ω is a polygonal (2D) or polyhedral (3D) domain, partitioned into a simplicial finite element mesh Ω_h. When k^2 is not an eigenvalue of $-\Delta$, this problem has a unique solution. We want to study its approximation by the so-called primal discontinuous Petrov Galerkin (DPG) method [4] (cf. [1, 2]). This approximation is based on a Petrov Galerkin weak formulation.

J. Gopalakrishnan (✉)
Portland State University, PO Box 751, Portland, OR 97207, USA
e-mail: gjay@pdx.edu

J. Schöberl
Wiedner Hauptstraße 8-10, TU Wien, 1040 Wein, Austria
e-mail: joachim.schoeberl@tuwien.ac.at

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_22

The derivation of the formulation begins, as in other standard finite element formulations, by multiplying the equation by a smooth enough complex-valued test function v and integrating by parts. The difference in the DPG case is that v is allowed to be discontinuous across element interfaces. Hence the appearance of interelement fluxes is inevitable, i.e.,

$$\sum_{K\in\Omega_h}\left(\int_K \operatorname{grad} u\cdot\overline{\operatorname{grad} v}-\int_K k^2u\overline{v}-\int_{\partial K}(n\cdot\operatorname{grad} u)\overline{v}\right)=\sum_{K\in\Omega_h}\int_K f\,\overline{v}.$$

Here, n generically denotes the unit outward normal of any domain under consideration, f is assumed to be square integrable (although this can be relaxed), and as usual, the integral over ∂K must be interpreted as a duality pairing if u is not sufficiently regular. Letting $n\cdot\operatorname{grad} u$ be an independent unknown, denoted by $n\cdot q$, this leads to the following weak formulation: *Find $u\in U$ and $q\in Q$ such that*

$$(\operatorname{grad} u,\operatorname{grad} v)_{\Omega_h}-k^2(u,v)_{\Omega_h}-\langle n\cdot q,v\rangle_{\partial\Omega_h}=(f,v)_\Omega,\quad\forall v\in Y,\tag{2}$$

where $(r,s)_{\Omega_h}=\sum_{K\in\Omega_h}(r,s)_K$ and $(\cdot,\cdot)_D$, for any domain D, denotes the complex $L^2(D)$-inner product, $\langle\ell,w\rangle_{\partial\Omega_h}=\sum_{K\in\Omega_h}\langle\ell,w\rangle_{1/2,\partial K}$ where $\langle\ell,\cdot\rangle_{1/2,\partial K}$ denotes the action of a functional ℓ in $H^{-1/2}(\partial K)$,

$$U=H_0^1(\Omega),\qquad Y=\prod_{K\in\Omega_h}H^1(K),\qquad Q=H(\operatorname{div},\Omega)\Big/\prod_{K\in\Omega_h}H_0(\operatorname{div},K).$$

Here $H_0(\operatorname{div},K)=\{q\in H(\operatorname{div},K):q\cdot n|_{\partial K}=0\}$. Formulation (2) is clearly of the Petrov-Galerkin kind as the trial space $X=U\times Q$ is different from the test space Y. Adapting the techniques in [3, 4], it is possible to prove that this weak formulation has a unique solution whenever k^2 is not a cavity resonance. However, the focus of this note is on practical implementation.

The method we shall implement is not based on the above Petrov-Galerkin form, but rather on an equivalent mixed Bubnov-Galerkin form. To describe it, first let us set the sesquilinear form $b(\cdot,\cdot)$ by

$$b((u,q),v)=(\operatorname{grad} u,\operatorname{grad} v)_{\Omega_h}-k^2(u,v)_{\Omega_h}-\langle n\cdot q,v\rangle_{\partial\Omega_h}$$

and the Y-inner product by

$$(y,v)_Y=(\operatorname{grad} y,\operatorname{grad} v)_{\Omega_h}+k^2(y,v)_{\Omega_h}.$$

The equivalent mixed formulation is to find $(\varepsilon,u,q)\in Y\times X$ such that

$$(\varepsilon,y)_Y+b((u,q),y)=(f,y)_{\Omega_h}\tag{3a}$$

$$b((w,r),\varepsilon)=0,\tag{3b}$$

for all $(y, w, r) \in Y \times X$. One can show (see e.g., [5]) that the solution (u, q) of (2) together with $\varepsilon = 0$ is the unique solution of (3).

2 A DPG Method for the Helmholtz Equation

The DPG method we want to study is a Galerkin method obtained directly from (3), i.e., the DPG approximation $(\varepsilon_h, u_h, q_h)$ is in a discrete subspace $Y_h \times U_h \times Q_h$ of $Y \times U \times Q$ and satisfies

$$(\varepsilon_h, y)_Y + b((u_h, q_h), y) = (f, y)_{\Omega_h} \tag{4a}$$

$$b((w, r), \varepsilon_h) = 0 \tag{4b}$$

for all $(y, w, r) \in Y_h \times U_h \times Q_h$. (A different DPG method for the Helmholtz equation based on an ultra-weak formulation can be found in [3].)

The discrete spaces are set, as recommended in [2, 4], for any degree $p \geq 0$, by

$$\begin{aligned} Y_h &= \{v \in Y : \ v|_K \in P_{p+2}(K), \forall K \in \Omega_h\}, \\ U_h &= \{w \in U : \ w|_K \in P_{p+1}(K), \forall K \in \Omega_h\}, \\ Q_h &= \{r \in Q : \ q|_K \in R_p^\partial(K), \forall K \in \Omega_h\}, \end{aligned}$$

where $P_p(K)$ denotes the space of polynomials of degree at most p on K and $R_p^\partial(K)$ is defined as follows. Recall that the Raviart-Thomas space in N space dimensions $R_p(K) = P_p(K)^N + xP_p(K)$ (where $x \in \mathbb{R}^N$ is the coordinate vector), can be split into a subspace $R_p^0(K) = R_p(K) \cap H_0(\mathrm{div}, K)$ and a linearly independent remainder $R_p^\partial(K)$. The decomposition $R_p(K) = R_P^0(K) \oplus R_p^\partial(K)$ depends on the choice of the basis for $R_p(K)$, but since the sesquilinear form $b(\cdot, \cdot)$ uses only the trace $n \cdot q$ of function q in Q, its value is independent of the choice of the basis representation. The trace space of $R_p(K)$ and $R_p^\partial(K)$ coincide. Indeed, we may even use a space other than the Raviart-Thomas space, as long as its traces coincide with that of the Raviart-Thomas space of index p (i.e., polynomials of degree at most p on each $(N-1)$-subsimplex of K).

3 The Matrix Form of the Method

Let $\{v_j\}, \{w_l\}, \{r_m\}$ denote some bases for Y_h, U_h, and Q_h, respectively. Then, defining the matrices A, B, C by

$$A_{ij} = (v_j, v_i)_Y = \sum_K \left(\int_K \mathrm{grad}\, v_j \cdot \mathrm{grad}\, \overline{v}_i + k^2 \int_K v_j \overline{v}_i \right)$$

$$B_{lj} = \overline{b((w_l,0),v_j)} = \sum_K \left(\int_K \operatorname{grad} v_j \cdot \operatorname{grad} \overline{w}_l - k^2 \int_K v_j \overline{w}_l \right)$$

$$C_{mj} = \overline{b((0,r_m),v_j)} = -\sum_K \left(\int_{\partial K} v_j \, n \cdot \overline{r}_m \right),$$

we can write the matrix form of the DPG method as

$$\begin{bmatrix} A & B^* & C^* \\ B & 0 & 0 \\ C & 0 & 0 \end{bmatrix} \begin{bmatrix} x_\varepsilon \\ x_u \\ x_q \end{bmatrix} = \begin{bmatrix} F \\ 0 \\ 0 \end{bmatrix}, \tag{5}$$

where * denote conjugate transpose. Clearly, the system is Hermitian. It is possible to prove that this discrete system inherits invertibility from the well-posedness of the exact problem whenever Y_h is of sufficiently high degree, but in practice we choose Y_h to be of degree $p+2$ as already stated.

Since functions in Y_h have no continuity constraints across element interfaces, the matrix A is block diagonal (in addition to being Hermitian and positive definite) with one block per element, and is thus easy to invert. Therefore, the preferred matrix system for inversion is not (5), but rather its positive definite Schur complement computed as follows. With $L^* = [B^* \;\; C^*]$ and $x_{uq^*} = [x_u^* \;\; x_q^*]$, rewriting (5) as

$$\begin{bmatrix} A & L^* \\ L & 0 \end{bmatrix} \begin{bmatrix} x_\varepsilon \\ x_{uq} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}, \tag{6}$$

and eliminating x_ε, we obtain

$$(LA^{-1}L^*)x_{uq} = LA^{-1}F. \tag{7}$$

This is a Hermitian and positive definite system whenever (5) is invertible. Hence we are able to use the *preconditioned conjugate gradient method* as an iterative solver even though the original Helmholtz problem is indefinite. The remaining component x_ε can be recovered by $x_\varepsilon = A^{-1}(F - L^* x_{uq})$.

4 Implementation in NGSolve

We use several facilities provided by the package NGSolve [7, 8] to implement the above DPG method. First, the spaces Y_h and U_h are standard finite element spaces provided by the classes `L2HighOrderFESpace` and `H1HighOrderFESpace`, respectively. The space Q_h can be implemented by removing all interior degrees of

freedom from the NGSolve class `HDivHighOrderFESpace`. A built-in facility for this removal is provided via the option `-orderinner` which allows one to restrict the degree of interior shape functions (those with zero normal traces on the element boundary). One then makes a compound space using these components. All of this can be done in the standard `pde`-input file format of NGSolve, as shown:

```
# Finite element spaces        (p = 2 case)
fespace fs1 -type=l2ho   -order=4 -complex                # Yh
fespace fs2 -type=h1ho   -order=3 -complex                # Uh
fespace fs3 -type=hdivho -order=2 -complex -orderinner=1  # Qh
fespace fs  -type=compound -spaces=[fs1,fs2,fs3] -complex  # Yh x Uh x Qh
```

Next, we must define all the sesquilinear forms in (4). The first form $(\cdot,\cdot)_Y$ in (4a) can be input in the `pde`-file using the built-in "integrator" classes `laplace` and `mass` provided in NGSolve. The $b(\cdot,\cdot)$ form however is nonstandard and is not available in NGSolve. We therefore exploit NGSolve's extensibility via shared library additions by writing new integrator classes. They use the dynamic polymorphism in NGSolve, inheriting properties from the abstract NGSolve class `BilinearFormIntegrator`. The new integrator classes are used to build a shared library of forms often needed in DPG methods. With the integrators for the $b(\cdot,\cdot)$ form made (subsumed under `[custom_integrators]` below) we can now define the sesquilinear form:

```
bilinearform dpg -fespace=fs -linearform=lf -nonsym -eliminate_internal
[custom_integrators]                  # b( (u,q), v)
laplace             (1.0)  --comp=1   # (grad e, grad v)
mass                (k*k)  --comp=1   # k*k* (e,v)
```

Of particular interest to us is the option `-eliminate_internal` above. Each degree of freedom in an NGSolve finite element space is marked if it is "inner" or not. An inner degree of freedom on one element does not interact with another inner degree of freedom on another element. By virtue of this stored information, the code can automatically perform static condensation of all inner degrees of freedom. In particular, all degrees of freedom of `L2HighOrderFESpace` within an element are marked to be inner. This means that the elimination of x_ε that allowed us to go from (5) to (7) is automatically performed by the code once the flag `-eliminate_internal` is given. To be precise, in addition to condensing (5) to (7), the code does a further condensation that eliminates all inner degrees of freedom of U_h.

Thus the *condensed system* consists only of degrees of freedom of Q_h (which by definition are associated only to element interfaces) and those degrees of freedom of U_h at the element interfaces (see Fig. 1). This final system, being another Schur

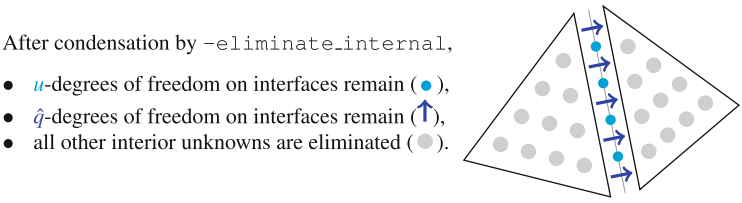


Fig. 1 Schematic of degrees of freedom left after condensation

complement of the Hermitian positive definite Schur complement (7), is Hermitian and positive definite. We solve it by conjugate gradients, preconditioned by the Schwarz procedure discussed later.

This and other input files in their entirety as well as the code for the DPG shared library is publicly available at [https://github.com/jayggg/DPG]

5 The Impedance Boundary Condition

Previously, we built the Dirichlet boundary condition (1b) into the weak formulation by essentially imposing it in U. Now suppose we are given, instead of (1b), the impedance condition

$$\frac{\partial u}{\partial n} - \hat{\imath} k u = 0, \qquad \text{on } \partial\Omega,$$

where $\hat{\imath}$ denotes the imaginary unit. Then instead of setting U to $H_0^1(\Omega)$, we now set $U = H^1(\Omega)$. Using the flux approximation given explicitly in the DPG formulations, the impedance boundary condition can be rewritten as

$$n \cdot q - \hat{\imath} k u = 0, \qquad \text{on } \partial\Omega. \tag{8}$$

Being a constraint tying two of the component spaces, a natural implementation would be by a Lagrange multiplier technique. However, this can result in loss of positive definiteness.

We pursue a different approach that imposes condition (8) weakly. The idea is to use the test function components w and r, i.e., we would like to impose the additional conditions $\int_{\partial\Omega} (n \cdot q_h - \hat{\imath} k u_h)\, \overline{n \cdot r} = 0$ and $\int_{\partial\Omega} (n \cdot q_h - \hat{\imath} k u_h)\, \overline{w} = 0$ without over-constraining the system. Since ε_h is an approximation to zero, we are motivated to build an approximate version of these conditions into the system by adding the term

$$\pm \int_{\partial\Omega} (n \cdot q_h - \hat{\imath} k u_h)\, \overline{(n \cdot r - \hat{\imath} k w)} \tag{9}$$

to the left hand side of (4b). This then perturbs the original system (6) to

$$\begin{bmatrix} A & L^* \\ L & D \end{bmatrix} \begin{bmatrix} x_\varepsilon \\ x_{uq} \end{bmatrix} = \begin{bmatrix} F \\ 0 \end{bmatrix}. \tag{10}$$

This system can also be condensed to get an analogue of (7):

$$(LA^{-1}L^* - D)x_{uq} = LA^{-1}F. \tag{11}$$

Now the choice of the sign in (9) becomes important: If we want (11) to be positive definite, we must choose the negative sign in (9) so that D is negative semidefinite.

6 The Condensed Schwarz Preconditioner

We now study a preconditioner for (11) constructed using a block Gauss-Seidel operator with overlapping blocks. The block Gauss-Seidel algorithm is standard, so we omit all details, except the specification of the blocks for our application. The blocks consists of all degrees of freedom after condensation, associated to a vertex patch. In 2D, one such block consists of all degrees of freedom of U_h and Q_h associated to the edges which meet at a single vertex (see Fig. 2). The block corresponding to a vertex in the 3D case consists of all degrees of freedom on all the mesh edges and the mesh faces containing that vertex. There are as many blocks as there are mesh vertices. The block Gauss-Seidel iteration multiplicatively updates an iterate by block inverses of certain residuals. These block inverses exist because they are principal submatrices of the positive definite matrix in (11). The action of our preconditioner consists simply of a block Gauss-Seidel relaxation algorithm followed by its adjoint given by the same relaxation done in the reverse block ordering.

- Set up diagonal blocks from the *condensed* DPG matrix, with vertex patches blocks, as indicated.
- One preconditioner action:
  - Perform one block Gauss-Seidel relaxation sweep with these blocks.
  - Perform a reverse block Gauss-Seidel sweep, using the same blocks, but in the reverse order, for symmetrization.
  - The result is the action of the linear operator used as a preconditioner in conjugate gradients.

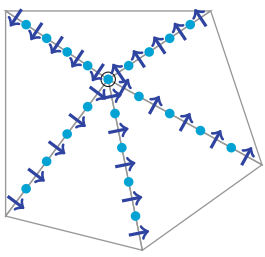


Fig. 2 Gauss-Seidel blocks

7 Numerical Results

We now report a result that is typical of our numerical experience with this method. We simulated a plane wave propagating in the x-direction on a uniform 4×4 triangular mesh of the unit square by providing the needed non-homogenous data to the impedance boundary condition. After assembling the condensed system (11), we used conjugate gradients, preconditioned by the above-mentioned block Gauss-Seidel algorithm, as an iterative solver. We stopped the iterations when successive iterates differed by less than 10^{-10}. The number of iterations are reported in Table 1. Each column of the table reports iteration counts obtained using a fixed wavenumber $k = 2\pi \times n_\lambda$ where n_λ (indicated atop the table) is the number of wavelengths that fit into the unit square.

The grayed out entries give iteration counts as well as indicate that computed solution did not resolve the wave. As is typical of all finite element type methods for wave problems, when meshes are too coarse, waves are not resolved. However, unlike many other methods, the DPG system remains solvable, no matter how coarse the mesh is. Moreover, the preconditioned conjugate gradient algorithm seems to converge at a degree-independent rate even on such coarse meshes. The bold entries also give the iteration numbers, but additionally indicate that in these cases the converged solution clearly showed the wave features. For example, in the $k = 2\pi \times 4$ case, it appears that we need at least $p = 8$ to resolve the wave. Note that we are able to go to polynomial degrees as high as 32 due to the good conditioning properties of the integrated Legendre shape functions implemented in NGSolve.

For comparison, we provide results from a simple diagonal preconditioning in a separate table. Clearly, the results from the block preconditioner are better. Entries marked "***" indicate that stopping criterion was not met even at 1000 iterations.

Our main conclusion from these observations is that the preconditioner seems to be uniform in p and k. (Similar observations were reported in [6] using an analogous preconditioner within GMRES for a different method. That method yields an indefinite system, while the current DPG method yields positive definite systems, so we may reliably use conjugate gradients on the latter.) Other (unreported) experiments in other wave directions in 2D, as well as in 3D tetrahedral meshes,

Table 1 Preconditioned conjugate gradient iteration counts

Schwarz preconditioner				
	Number of waves			
Degree p	2	4	8	16
1	16	14	12	11
2	22	13	12	10
4	**28**	27	12	12
8	**28**	**30**	32	11
16	**29**	**30**	**30**	32
32	**29**	**30**	**30**	**30**

Diagonal preconditioner				
	Number of waves			
Degree p	2	4	8	16
1	63	59	54	51
2	180	178	166	121
4	**261**	468	416	398
8	**328**	**612**	***	***
16	**662**	**894**	***	***
32	***	***	***	***

all appear to confirm the uniformity of the preconditioner on k and p. Finally, we note that the preconditioner is not uniform in mesh size h. One usually needs to use a "coarse" solution to get h-uniformity. But for wave propagation, a good coarse problem is still a subject of debate.

Acknowledgements The authors wish to thank graduate student Lukas Kogler for his assistance in developing an initial version of the DPG code. This work was partially supported by the NSF under grant DMS-1318916 and by the AFOSR under grant FA9550-12-1-0484.

References

1. D. Broersen, R. Stevenson, A Petrov-Galerkin discretization with optimal test space of a mild-weak formulation of convection-diffusion equations in mixed form. IMA J. Numer. Anal. doi:10.1093/imanum/dru003. (2014, to appear in print)
2. L. Demkowicz, J. Gopalakrishnan, A class of discontinuous Petrov-Galerkin methods. Part II: optimal test functions. Numer. Methods Partial Differ. Equ. **27**, 70–105 (2011)
3. L. Demkowicz, J. Gopalakrishnan, I. Muga, J. Zitelli, Wavenumber explicit analysis for a DPG method for the multidimensional Helmholtz equation. Comput. Methods Appl. Mech. Eng. **213/216**, 126–138 (2012)
4. L. Demkowicz, J. Gopalakrishnan, A primal DPG method without a first-order reformulation. Comput. Math. Appl. **66**, 1058–1064 (2013)
5. J. Gopalakrishnan, Five lectures on DPG methods (2013). arXiV: 1306.0557
6. P. Monk, J. Schöberl, A. Sinwel, Hybridizing Raviart-Thomas elements for the Helmholtz equation. Electromagnetics **30**, 149–176 (2010)
7. J. Schöberl, NETGEN – an advancing front 2D/3D-mesh generator based on abstract rules. Comput. Visual. Sci. **1**, 41–52 (1997)
8. J. Schöberl, NGSolve [Computer Software]. Retrieved from http://sourceforge.net/projects/ngsolve/ (2014)

An HDG Method for Unsteady Compressible Flows

Alexander Jaust, Jochen Schütz, and Michael Woopen

Abstract Recent gain of interest in discontinuous Galerkin (DG) methods shows their success in computational fluid dynamics. One potential drawback is the high number of globally coupled unknowns. By means of hybridization, this number can be significantly reduced. The hybridized DG (HDG) method has proven to be beneficial especially for steady flows. In this work we apply it to a time-dependent flow problem with shocks. Due to its inherently implicit structure, time integration methods such as diagonally implicit Runge-Kutta (DIRK) methods present themselves as natural candidates. Furthermore, as the application of flux limiting to HDG is not straightforward, an artificial viscosity model is applied to stabilize the method.

1 Introduction

A prominent class of high-order methods for computational fluid dynamics are so-called discontinuous Galerkin methods [3, 4, 7, 9–12]. Based on a partitioning of the domain into a set of N elements, the solution is approximated by piecewise polynomials on each of the elements. This allows local (non-conforming) refinement by varying the number of elements N or the degree p of the polynomials used on each of the elements. However, these methods suffer from a large number of globally coupled unknowns when being used with implicit time-stepping methods or in the context of stationary problems. An approach to reduce this number is to use hybridization, presented for DG by Cockburn et al. [6]. This leads to

A. Jaust (✉)
MathCCES, RWTH Aachen University, Schinkelstraße 2, 52062 Aachen, Germany
e-mail: jaust@mathcces.rwth-aachen.de

J. Schütz
IGPM, RWTH Aachen University, Templergraben 55, 52062 Aachen, Germany
e-mail: schuetz@igpm.rwth-aachen.de

M. Woopen
AICES, RWTH Aachen University, Schinkelstraße 2, 52062 Aachen, Germany
e-mail: woopen@aices.rwth-aachen.de

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_23

the hybridized discontinuous Galerkin (HDG) method, see, e.g., [15–19, 21]. By introducing an additional unknown λ_h having support on the element interfaces, the system of equations can be formulated such that it is only globally coupled in this hybrid variable. Therefore, the number of globally coupled unknowns changes asymptotically from $\mathcal{O}\left(p^d N\right)$ to $\mathcal{O}\left(p^{d-1}\hat{N}\right)$ where $\hat{N}$ is the number of element interfaces in the mesh and d the spatial dimension.

In this paper we focus on supersonic inviscid flows. These flows can be described by the compressible Euler equations, and it is known that they tend to develop discontinuities. These discontinuities are a severe issue for high-order methods, as they usually show oscillatory behavior that leads to stability issues. In order to capture discontinuities and stabilize computations we use an shock-capturing method that has been introduced by Persson and Peraire [20].

2 Numerical Method

In this section, we give a brief introduction to the hybridized discontinuous Galerkin method. For more details, we refer to, e.g., [13, 15]. We shortly describe the applied time integration and shock-capturing schemes. Please note that we focus on the two dimensional case in this work, i.e., $d = 2$. Therefore, we will refer to element interfaces as edges from here on.

2.1 Governing Equations

We consider supersonic inviscid flows that can be described using the compressible Euler equations, given by

$$\frac{\partial w}{\partial t} + \nabla \cdot f(w) = 0 \quad \forall\, (x,t) \in \Omega \times [0,\infty) \tag{1}$$

on a domain Ω, equipped with appropriate initial and boundary conditions. The vector of conserved variables is $w = (\rho, \rho u_1, \rho u_2, E)^T$ and involves the density ρ, velocities u_1 and u_2 and total energy E. Convective fluxes $f = (f_1, f_2)$ are given by

$$\begin{aligned} f_1 &= (\rho u_1, P + \rho u_1^2, \rho u_1 u_2, u_1(E+P))^T, \\ f_2 &= (\rho u_2, \rho u_1 u_2, P + \rho u_2^2, u_2(E+P))^T. \end{aligned} \tag{2}$$

Pressure P is determined using the ideal gas law with the adiabatic constant $\gamma = 1.4$ for air.

2.2 The Hybridized Discontinuous Galerkin Method

In order to discretize Eq. (1), we assume a partitioning of Ω as $\Omega = \bigcup_{k=1}^{N} \Omega_k$. Additionally, we define the following ansatz spaces

$$V_h := \{f \in L^2(\Omega) \mid f_{|\Omega_k} \in \Pi^p(\Omega_k)\ \forall k = 1, \ldots, N\}^4 \tag{3}$$

$$M_h := \{f \in L^2(\Gamma) \mid f_{|e_k} \in \Pi^p(e_k)\ \forall k = 1, \ldots, \hat{N},\ e_k \in \Gamma\}^4, \tag{4}$$

where Γ is the set of all edges. Given these definitions, the semi-discrete formulation of the HDG method can be written as

$$\sum_{k=1}^{N} \left(((w_h)_t, \varphi_h)_{\Omega_k} - (f(w_h), \nabla\varphi_h)_{\Omega_k} + \langle \hat{f} \cdot n, \varphi_h \rangle_{\partial\Omega_k} \right) = 0 \quad \forall \varphi_h \in V_h \tag{5}$$

$$\sum_{k=1}^{\hat{N}} \langle [\![\hat{f} \cdot n]\!], \mu_h \rangle_{\Gamma} = 0 \quad \forall \mu_h \in M_h. \tag{6}$$

$(\cdot, \cdot)$ and $\langle \cdot, \cdot \rangle$ denote the element and edge scalar product, respectively. The fluxes over edges have been substituted by the numerical fluxes

$$\hat{f} := f(\lambda_h) - \alpha(\lambda_h - w_h^-)n, \tag{7}$$

with positive real parameter α and w_h^- denoting w_h evaluated on the element's interior. $w_h \in V_h$ and $\lambda_h \in M_h$ are the unknowns.

Equation (5) is very similar to the weak formulation one obtains for standard DG methods. However, the coupling between cells is established via the hybrid variable λ_h. The second Eq. (6) is derived from the original problem and is necessary to both determine λ_h and ensure that the total flux has a weak divergence. Together with the Rusanov-/Lax-Friedrichs inspired flux the locality of the scheme is retained. In this setting, *locality* means that the system of equations on each element only depends on the information given on the element and on its edges. This allows the application of static condensation [6]. Therefore, the linearized system of equations can be written such that it is globally coupled only in λ_h. This may lead to an extensive reduction of globally coupled unknowns.

2.3 Time Integration

In Eqs. (5)–(6), only the temporal derivative of w_h occurs. Thus, the discretization leads to a system of differential-algebraic equations (DAEs) of index 1. This poses a severe restriction to the time integration methods that can be applied [8]. Therefore, only implicit methods with adequate stability properties can be used with

HDG [16–18]. In our case, we found backward differentiation (BDF) methods and diagonally implicit Runge-Kutta (DIRK) methods [1, 2, 5, 8] to be reliable time integrators for the HDG method [13, 14, 22].

2.4 Shock-Capturing

High-order methods such as the DG or the HDG method tend to show oscillatory behavior near discontinuities or steep gradients if they are not suitably stabilized. For DG, flux limiters and artificial viscosity models are a generic choice. However, the HDG method relies on the locality of the method, which makes it hard or even impossible to employ flux limiters. The use of artificial viscosity models, however, is possible. In this work, we use a shock-capturing method introduced by Persson and Peraire [20], which relies on adding an additional diffusive term to the equations. In our case, we add an (inconsistent) discretization of the Laplacian, $(\varepsilon_k \nabla w_h, \nabla \phi_h)$, to Eq. (5).

The cell-wise constant viscosity ε_k is determined using a shock sensor

$$s_k := \log_{10}\left(\frac{(\bar{w} - w, \bar{w} - w)_{L_2(\Omega_k)}}{(w, w)_{L_2(\Omega_k)}}\right) \tag{8}$$

where $\bar{w}$ represents the L^2-projection of w from $\Pi^p(\Omega_k)$ to $\Pi^{p-1}(\Omega_k)$. Please note, that this projection step is very cheap as we employ orthogonal basis functions. Hence, s_k measures the discrete smoothness of a solution by comparing its highest order terms to the complete solution.

Then, the actual amount of viscosity ε_k for each element is computed from

$$\varepsilon_k := \begin{cases} 0 & , \quad s_k < s_0 - \kappa_{\text{av}} \\ \frac{\varepsilon_0}{2}\left(1 + \sin\left(\frac{\pi(s_k - s_0)}{2\kappa_{\text{av}}}\right)\right) & , \quad s_0 - \kappa_{\text{av}} \leq s_k \leq s_0 + \kappa_{\text{av}} \\ \varepsilon_0 & , \quad s_k > s_0 + \kappa_{\text{av}} \end{cases} \tag{9}$$

where $\varepsilon_0 \sim \frac{h}{p}$, $s_0 \sim \log(p)$ and κ_{av} are problem-dependent parameters. They have to be chosen such that the method is sufficiently stabilized while keeping discontinuities sharp.

3 Numerical Results

In this section we present results obtained from the described method. Based on the work in [13, 14, 20] we present results for a double Mach reflection at a wedge. At each time step, we employ a Newton-Krylov solver based on a restarted GMRES

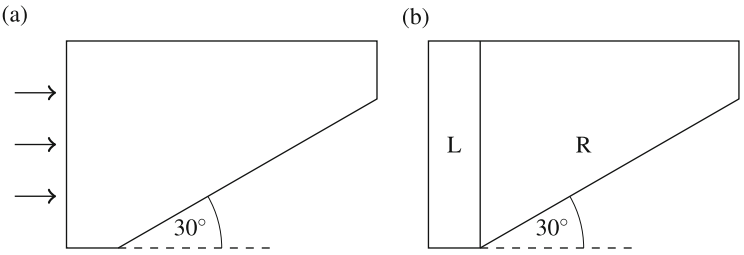


Fig. 1 Sketches of computational domain and how initial data is distributed. (**a**) Sketch of the physical domain. (**b**) Partitioning of the domain for setting the initial conditions

for the resulting linear system. As a preconditioner we use an incomplete LU factorization without additional levels of fill.

This famous test case is taken from the paper of Woodward and Colella [23]. Supersonic flow enters the domain from the left and hits a wedge of 30 degree angle (see Fig. 1a). Instead of a rectangular domain as in the original paper, we choose the domain such that the flow enters the domain on the left and is parallel to the x_1-direction. Therefore, we do not have to prescribe the shock on the upper boundary. We have inflow boundary conditions on the left, slip-wall boundary conditions at the wedge and symmetry boundary conditions everywhere else.

The domain is initialized with pre-shock values, $(\rho_L = 8.0, u_{1,L} = 8.25, u_{2,R} = 0, P_L = 116.5)^T$, in front of the wedge, denoted by L, and post-shock values, $(\rho_R = 1.4, u_{1,R} = 0, u_{2,R} = 0, P_R = 1.0)^T$, everywhere else (see Fig. 1b). We run the simulation up to $t = 0.2$ and use a DIRK method of second order with two stages described by Alexander [2]. Due to limited deflection angles for the given flow conditions, the shock is reflected such that a complex structure occurs. A convex shock beginning at the tip of the wedge is created and a region with interacting shocks develops close to where the shock hits the wedge.

We run two simulations with $N_1 = 3167$ and $N_2 = 8395$ elements. In both cases polynomials of degree $p = 3$ are employed. For the coarse mesh, see Fig. 2, and the fine mesh, see Fig. 3, we show the mesh, the artificial viscosity, the density and isolines of the density. For both meshes, the general structure of the solution is captured while on the finer mesh the shocks are sharper. The shocks are detected by the smoothness sensor such that artificial viscosity is only applied in these regions. In the region where the shock interacts with an occurring jet much less viscosity is added than at the outer shocks. However, on neither of the meshes Kelvin-Helmholtz instabilities in the jet can be seen. There is possibly too much dissipation due to the applied viscosity, the mesh resolution or the applied time integration method.

Note that there are small oscillations in the area in front of the shock that cannot be seen in this figures. These may be reduced by using other parameters for the shock-capturing. However, these are actually that small, such that the shock sensor hardly recognizes them.

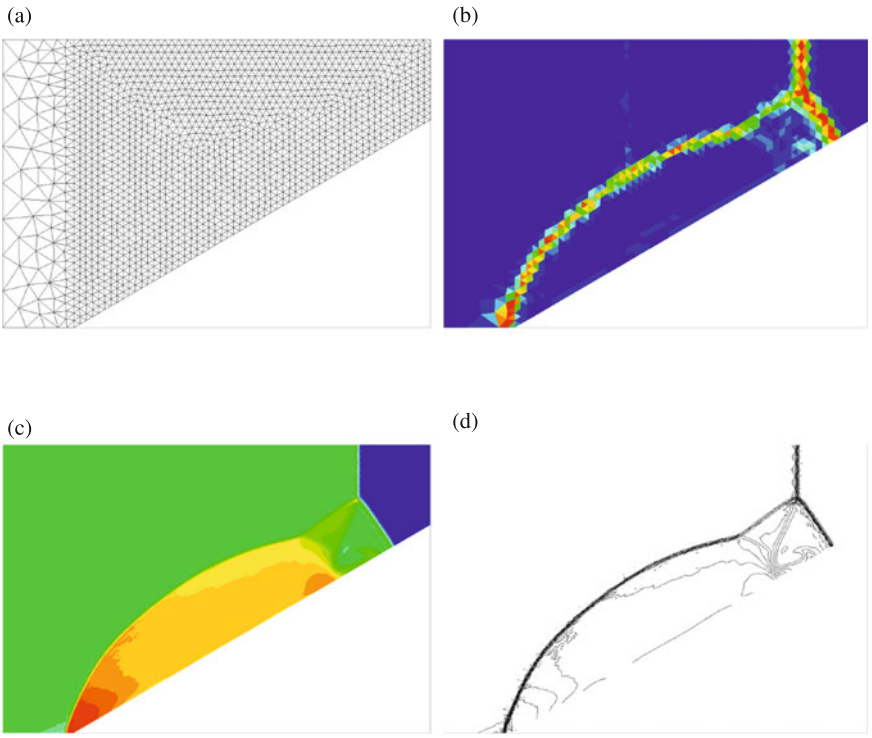


Fig. 2 Mesh and solution at $t = 0.2$. (**a**) Mesh with 3167 elements. (**b**) Cellwise artificial viscosity using 14 levels from $\varepsilon_{k,\min} = 0.0$ and $\varepsilon_{k,\max} = 0.014$ at $t = 0.2$. (**c**) Density distribution for 20 levels with $\rho_{\min} = 1$ and $\rho_{\max} = 19$ at $t = 0.2$. (**d**) Isolines of the density distribution for 20 levels with $\rho_{\min} = 1$ and $\rho_{\max} = 19$ at $t = 0.2$

4 Conclusion and Outlook

We have presented a hybridized DG method for an unsteady compressible flow problem. By applying an artificial viscosity model we can stabilize the method successfully to approximate flows with shocks.

Future work will include more detailed studies regarding the optimal parameters for the shock-capturing scheme as well as the behavior on refined grids. The meshes used here rely on uniformly large elements which is not suitable for flows with sharp flow features such as shocks. Therefore, local adaptation of the polynomial degree p and the mesh is work in progress. The latter is a challenging task for unsteady flows since the flow features are likely to move. This introduces the need for both mesh refinement and coarsening to keep the number of elements low.

In addition to adaptation, we also plan to try further shock-capturing strategies to see whether there are methods in particular well-suited for the HDG method. This also includes different ways of applying the artificial viscosity. As suggested in the

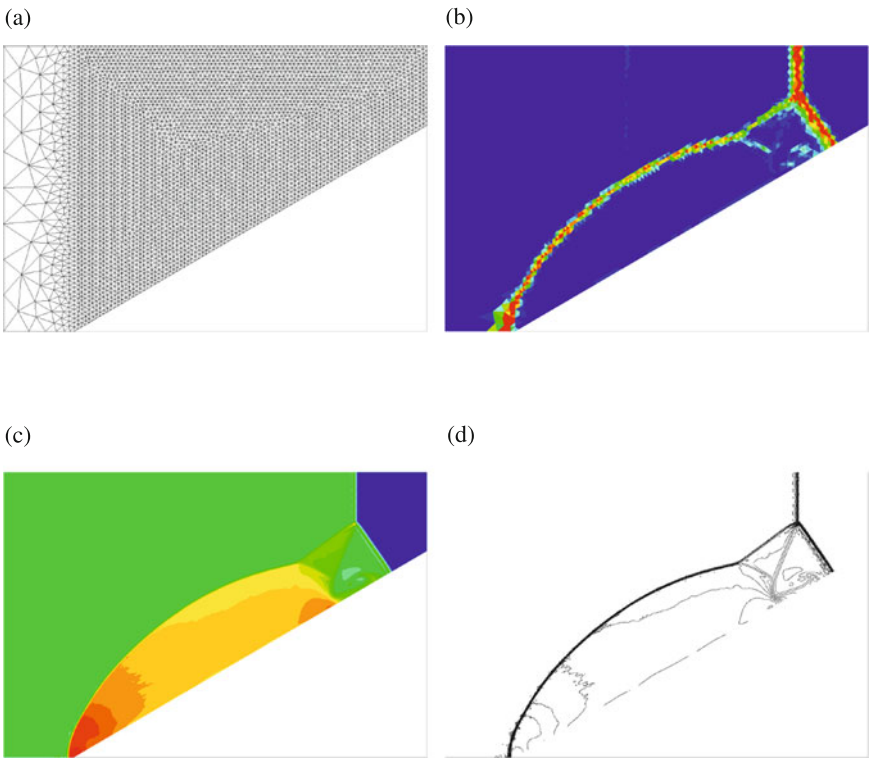


Fig. 3 Mesh and solution at $t = 0.2$. (**a**) Mesh with 8395 elements. (**b**) Cellwise artificial viscosity using 15 levels from $\varepsilon_{k,\min} = 0.0$ and $\varepsilon_{k,\max} = 0.007$ at $t = 0.2$. (**c**) Density distribution for 20 levels with $\rho_{min} = 1$ and $\rho_{max} = 19$ at $t = 0.2$. (**d**) Isolines of the density distribution for 20 levels with $\rho_{\min} = 1$ and $\rho_{\max} = 19$ at $t = 0.2$

work by Persson and Peraire [20] it may be beneficial to use the physical diffusive terms present in the Navier-Stokes equations instead of a Laplacian.

Another point to address are time integration schemes. So far, classical one step and multistep methods have been applied, but it may be worthwhile to also consider other concepts such as general linear methods.

References

1. A.H. Al-Rabeh, Embedded DIRK methods for the numerical integration of stiff systems of ODEs. Int. J. Comput. Math. **21**(1), 65–84 (1987)
2. R. Alexander, Diagonally implicit Runge-Kutta methods for stiff O.D.E.'s. SIAM J. Numer. Anal. **14**(14), 1006–1021 (1977)
3. D.N. Arnold, F. Brezzi, B. Cockburn, L.D. Marini, Unified analysis of discontinuous Galerkin methods for elliptic problems. SIAM J. Numer. Anal. **39**(5), 1749–1779 (2002)

4. F. Bassi, S. Rebay, A high-order accurate discontinuous finite-element method for the numerical solution of the compressible Navier-Stokes equations. J. Comput. Phys. **131**, 267–279 (1997)
5. J.R. Cash, Diagonally implicit Runge-Kutta formulae with error estimates. J. Inst. Math. Appl. **24**, 293–301 (1979)
6. B. Cockburn, J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problems. SIAM J. Numer. Anal. **47**(2), 1319–1365 (2009)
7. K. Fidkowski, T. Oliver, J. Lu, D. Darmofal, *p*-Multigrid solution of high-order discontinuous Galerkin discretizations of the compressible Navier-Stokes equations. J. Comput. Phys. **207**, 92–113 (2005)
8. E. Hairer, G. Wanner, *Solving Ordinary Differential Equations II, Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics (Springer, Berlin, 1991)
9. R. Hartmann, P. Houston, Symmetric interior penalty DG methods for the compressible Navier-Stokes equations I: method formulation. Int. J. Numer. Anal. Model. **3**(1), 1–20 (2006)
10. R. Hartmann, P. Houston, Symmetric interior penalty DG methods for the compressible Navier-Stokes equations II: goal-oriented a posteriori error estimation. Int. J. Numer. Anal. Model. **3**(2), 141–162 (2006)
11. J. Hesthaven, T. Warburton, *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Texts in Applied Mathematics, vol. 54 (Springer, Berlin, 2008)
12. P. Houston, E. Süli, hp-adaptive discontinuous Galerkin finite element methods for first order hyperbolic problems. SIAM J. Sci. Comput. **23**(4), 1226–1252 (2001)
13. A. Jaust, J. Schütz, A temporally adaptive hybridized discontinuous Galerkin method for instationary compressible flows. Comput. Fluids **98**, 177–185 (2014)
14. A. Jaust, J. Schütz, M. Woopen, A hybridized discontinuous Galerkin method for unsteady flows with shock-capturing. AIAA Paper 2014-2781, in *44th AIAA Fluid Dynamics Conference*, 2014
15. N.C. Nguyen, J. Peraire, Hybridizable discontinuous Galerkin methods for partial differential equations in continuum mechanics. J. Comput. Phys. **231**, 5955–5988 (2012)
16. N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. J. Comput. Phys. **228**(9), 3232–3254 (2009)
17. N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for nonlinear convection-diffusion equations. J. Comput. Phys. **228**(23), 8841–8855 (2009)
18. N.C. Nguyen, J. Peraire, B. Cockburn, High-order implicit hybridizable discontinuous Galerkin methods for acoustics and elastodynamics. J. Comput. Phys. **230**, 3695–3718 (2011)
19. J. Peraire, N.C. Nguyen, B. Cockburn, A hybridizable discontinuous Galerkin method for the compressible Euler and Navier-Stokes equations. AIAA Paper 2010-362, 48th AIAA Aerospace Sciences Meeting and Exhibit, 2010
20. P.-O. Persson, J. Peraire, Sub-cell shock capturing for discontinuous Galerkin methods. AIAA Paper 2006-0112, American Institute of Aeronautics and Astronautics, 2006
21. J. Schütz, G. May, A hybrid mixed method for the compressible Navier-Stokes equations. J. Comput. Phys. **240**, 58–75 (2013)
22. J. Schütz, M. Woopen, G. May. A combined hybridized discontinuous Galerkin / hybrid mixed method for viscous conservation laws. in *Hyperbolic Problems: Theory, Numerics, Applications*, ed. by F. Ancona, A. Bressan, P. Marcati, A. Marson, pp. 915–922 (American Institute of Mathematical Sciences, Springfield 2012)
23. P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks. J. Comput. Phys. **54**, 115–173 (1984)

Thermal Boundary Condition of First Type in Fourier Pseudospectral Method

D. Kinoshita, A. da Silveira Neto, F.P. Mariano, and R.A.P. Silva

Abstract The purpose of this paper is to extend a novel numerical methodology, combining thermal immersed boundary and Fourier pseudospectral methods called IMERSPEC. This methodology has been developed for incompressible fluid flow problems modeled using Navier-Stokes, mass and energy equations. The numerical algorithm consists of Fourier pseudospectral method (FPSM), where Dirichlet boundary condition is modeled using an immersed boundary method (multi-direct forcing method). The new method combines the advantages of high accuracy and low computational cost provided by FPSM to the possibility of managing complex and non periodical geometries given by immersed boundary method. In the present work this new methodology is applied to the problem of heat transfer for natural convection in the annulus between horizontal concentric cylinders and conducted to validate the capability and efficiency of present method. Results for this application are presented and good agreement with available data in the literature have been achieved.

D. Kinoshita (✉) • A. da Silveira Neto
Department of Mechanical Engineering, Laboratory of Mechanical of Fluids, Federal University of Uberlandia, Campus Sta. Monica, Av:Joao Naves de Avila, 2121, Bl 5p. CEP:38400-902, Uberlandia, Brazil
e-mail: denikino@doutorado.ufu.br; aristeus@mecanica.ufu.br

F.P. Mariano
School of Electric, Mechanical and Computational Engineering, Federal University of Goias, Av.: Universitaria, 1488, Bl: A, Piso: 3, Goiania - GO. CEP: 74.605-010, Brazil
e-mail: fpmariano@ufg.com.br

R.A.P. Silva
Faculty of Computing, Federal University of Uberlandia, Campus Sta. Monica, Av:Joao Naves de Avila, 2121, Room 1A236. CEP:38400-902, Uberlandia, Brazil
e-mail: rpimentel@ufu.br

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_24

1 Introduction

The search for accurate methods to solving the Navier-Stokes, mass and energy equations is of great interest to computational fluid mechanics for the solution of physical phenomena for which only high accuracy methodologies allows one to obtain a representative solution.

In terms of high accuracy, the classical Fourier pseudospectral collocation method is probably impressive, due to its extremely high accuracy and its low computational cost. These classical methods, however, are barely applicable over complex geometries, since a periodic domain is required, [1, 2].

Seeking to contributions to the solution of such problem have been developed, alternatively, the methodologies based on the concept of immersed boundary. It can handle complex and moving geometries, using Cartesian mesh.

In the present work, the main goal is to verify and validate the proposed methodology and the numerical implementation. We employ the IMERSPEC method presented by Mariano et al. [6], which combines a classical Fourier pseudospectral method with an immersed boundary method and extend for flows with internal energy transfer.

In order to verify the IMERSPEC methodology, a synthesized or manufactured solution for Taylor-Green problem was used, which also considers thermal effects. For that, an analytical solution of the velocity field, pressure and temperature field was given. Besides to validate the numerical code, developed in the present work, the problem of natural convection in a horizontal concentric cylinder is simulated and compared with the literature.

2 Mathematical Modeling

The mathematical model for incompressible flows of Newtonian fluids with thermal energy transfer is composed by the mass conservation, Eq. (1), the Navier-Stokes equations, Eq. (2) and the energy conservation, Eq. (3). Such equations present source terms that model the boundary conditions for momentum and thermal energy transfer, as well as, the synthesized solution, originated from the method of manufactured solutions.

$$\frac{\partial u_j}{\partial x_j} = 0, \tag{1}$$

$$\frac{\partial u_i}{\partial t} = \underbrace{-\frac{\partial (u_i u_j)}{\partial x_j} - \frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} + g_i \beta (T - T_0) + \frac{1}{\rho} f_{iss}}_{RHS} + \frac{1}{\rho} f_i, \tag{2}$$

$$\frac{\partial T}{\partial t} = \underbrace{\alpha \frac{\partial^2 T}{\partial x_j \partial x_j} - \frac{\partial (u_j T)}{\partial x_j} + \frac{1}{\rho Cp} f_{Tss}}_{RHS_T} + \frac{1}{\rho Cp} f_T. \tag{3}$$

2.1 Dirichlet Boundary Condition for Energy Equation

For the first type of boundary condition, the temperature $T_\Gamma(\mathbf{X}, t)$, is provided over the immersed boundary, where Γ indicates that the points are in the boundary, which gives the reference temperature to assess the forcing term, defined as:

$$T_{REF}(\mathbf{X}, t) \equiv T_\Gamma(\mathbf{X}, t), \tag{4}$$

where $T_{REF}(\mathbf{X}, t)$ is used in order to calculate the forcing term $f_T(\mathbf{x}, t)$. The temperature $T_\Gamma(\mathbf{X}, t)$ is given in terms of the physical condition for each problem. In the present paper, this forcing term is calculated using the thermal direct forcing, with a procedure similar to that used by Wang et al. [8]. So, if we discretize Eq. (3) using the Euler time discretization method, as demonstration, we obtain:

$$\frac{T^{t+\Delta t}(\mathbf{x}) - T^t(\mathbf{x})}{\Delta t} = RHS_T^t(\mathbf{x}) + \frac{1}{\rho Cp} f_T(\mathbf{x}). \tag{5}$$

By adding and subtracting a temporal parameter for the temperature, $T^*(\mathbf{x})$ (estimation of variable T), on the left hand side of Eq. (5), it gives:

$$\frac{T^{t+\Delta t}(\mathbf{x}) - T^t(\mathbf{x})}{\Delta t} + \frac{T^*(\mathbf{x}) - T^*(\mathbf{x})}{\Delta t} = RHS_T^t(\mathbf{x}) + \frac{1}{\rho Cp} f_T(\mathbf{x}). \tag{6}$$

This equation can be decomposed in to Eqs. (7) and (8):

$$\frac{T^*(\mathbf{x}) - T^t(\mathbf{x})}{\Delta t} = RHS_T^t(\mathbf{x}), \tag{7}$$

$$f_T(\mathbf{x}) = \rho Cp \frac{T^{t+\Delta t}(\mathbf{x}) - T^*(\mathbf{x})}{\Delta t}. \tag{8}$$

Equation (8), which is valid for any material particle, can be rewriten for a material particle placed over the interface, in the other words, over the immersed boundary:

$$F_T(\mathbf{X}) = \rho Cp \frac{T^{t+\Delta t}(\mathbf{X}) - T^*(\mathbf{X})}{\Delta t}, \tag{9}$$

where $T^{t+\Delta t}(\mathbf{X}) \equiv T^{t+\Delta t}_{REF}(\mathbf{X})$ is given by the physical characteristic of each problem. On the other hand, $T^*(\mathbf{X})$ is obtained by the interpolation of $T^*(\mathbf{x})$, which is obtained by the solution of Eq. (7). This interpolation can be defined mathematically by the following equation:

$$T^*(\mathbf{X}) = \sum_{\Omega} T^*(\mathbf{x}) D_h(\mathbf{x} - \mathbf{X}) h^2, \tag{10}$$

where $\mathbf{x} \in \Omega$, which represent the cartesian and periodical domain (Eulerian points) and $\mathbf{X} \in \Gamma$, which represent the non cartesian and non periodical domain (Lagrangian points). The distribution function, $D_h(\mathbf{x} - \mathbf{X})$, is given by the cubic function, proposed by Tornberg and Engquist [7].

Once $F_T(\mathbf{X})$ is obtained, determined by Eq. (9), it is distributed over Ω. With the force term distributed to the Eulerian points, $f_T(\mathbf{x})$, we can finally update the temperature, using Eq. (8), rewriten as:

$$T^{t+\Delta t, it+1}(\mathbf{x}) = T^{*,it}(\mathbf{x}) + \frac{\Delta t}{\rho C p} f_T^{it+1}(\mathbf{x}), \tag{11}$$

where $T^*(\mathbf{x})$ is obtained by Eq. (7) and *it* is the multi-direct forcing process which is given by the minimum value of the error between the calculated temperature at the immersed boundary and the desired temperature and the times of exerting direct heat source. Note that this error measures how good is the model for the boundary condition. Likewise, Dirichlet boundary condition is given for Navier-Stokes equations, this boundary condition is characterized by the 'non-slip' physical condition.

2.2 Mathematical Model in the Fourier Spectral Space

Given the mathematical model in the physical space, the next step is to transform it to the Fourier spectral space. For instance, the Fourier transform of the mass conservation equation Eq. (1) is given by:

$$\iota k_i \hat{u}_i(\mathbf{k}, t) = 0. \tag{12}$$

where $\hat{u}_i(\mathbf{k}, t)$ stands for the Fourier transform of the velocity field $u_i(\mathbf{x}, t)$ [1]. Equation (12) shows that, for incompressible flows, the transformed velocity field

is orthogonal to the wave number vector. The transformed Eqs. (2) and (3) are given by:

$$\frac{\partial \hat{u}_i(\mathbf{k},t)}{\partial t} = -\nu k^2 \hat{u}_i(\mathbf{k},t) + \wp_{im}\left\{\frac{1}{\rho}\left[\hat{f}_{ssm}(\mathbf{k},t) + \hat{f}_m(\mathbf{k},t)\right]\right\} - \tag{13}$$

$$- \wp_{im}\left\{\iota k_j \int_{\mathbf{k}=\mathbf{r}+\mathbf{s}} \hat{u}_m(\mathbf{r},t)\widehat{u_j}(\mathbf{k}-\mathbf{r},t)d\mathbf{r} + \beta g_m(\hat{T} - \widehat{T_r})\right\}$$

and

$$\frac{\partial \hat{T}(\mathbf{k},t)}{\partial t} = -\alpha k^2 \hat{T}(\mathbf{k},t) + \tag{14}$$

$$+ \frac{1}{\rho Cp}\left[\hat{f}_{ssT}(\mathbf{k},t) + \hat{f}_T(\mathbf{k},t)\right] - \iota k_j \int_{\mathbf{k}=\mathbf{r}+\mathbf{s}} \hat{T}(\mathbf{r},t)\widehat{u_j}(\mathbf{k}-\mathbf{r},t)d\mathbf{r},$$

where $\wp_{im}$ is the projection tensor, presented by Canuto et al. [2].

The non-linear terms which appear at the right hand side of Eqs. (13) and (14) are given by the convolution integrals, which are expensive to be solved. Otherwise, they can be solved using the pseudospectral method, presented by Canuto et al. [2]. It consists of evaluating the product in the physical space and then, transforming it to the Fourier spectral space. It is worth to be highlighted that the pressure is eliminated from Navier-Stokes equations as shown by Eq. (13). Nevertheless, the pressure can be recuperated at the post-processing procedure. Details can be seen in [6].

3 Results and Discussion

3.1 Taylor-Green Problem with Thermal Effects

In order to verify the methodology proposed in the present paper, a synthesized or manufactured solution is proposed. It consists of determining a source term from proposed analytical solutions for velocity, temperature and pressure fields. The following equations, proposed by Henshaw [3]—including a similar analytical solution for the energy equation—are used in the present work:

$$u^{an} = U_\infty sen\left(\frac{x}{L}\right) cos\left(\frac{y}{L}\right) cos\left(\frac{2\pi\nu t}{L^2}\right), \tag{15}$$

$$v^{an} = -U_\infty cos\left(\frac{x}{L}\right) sen\left(\frac{y}{L}\right) cos\left(\frac{2\pi\nu t}{L^2}\right), \tag{16}$$

$$p^{an} = \rho U_{\infty}^{2} sen\left(\frac{x}{L}\right) sen\left(\frac{y}{L}\right) cos\left(\frac{2\pi \nu t}{L^2}\right), \tag{17}$$

$$T^{an} = T_r cos\left(\frac{x}{L}\right) cos\left(\frac{y}{L}\right) cos\left(\frac{2\pi \alpha t}{L^2}\right), \tag{18}$$

where, u^{an}, v^{an}, p^{an} and T^{an} are, respectively, the analytical solutions for the velocity components, the pressure and the temperature. The parameters x and y are the components of the coordinate system, t is the time, ρ, ν and α are the fluid density, the kinematic viscosity and thermal diffusivity, respectively. U_{∞} is the reference velocity, T_r is the reference temperature and L is the reference length.

From the analytical solutions imposed to Eqs. (2) and (3), source terms are obtained simultaneously. This case is characterized by the presence of an immersed boundary Γ inside of the domain Ω.

Figure 1 shows both the geometrical domain (a) and the rate of convergence (b) for coinciding collocation nodes. The rate of convergence is approximately eighth order for $\Delta t = 10^{-2}[s]$ with mesh size $h = L/8$, $L/16$, $L/32$ and $L/64$, where $L = L_x = L_y = 2\pi$ and the corresponding number of Lagrangian points are 14, 30, 62 and 126. For non-coinciding collocation nodes fourth order is obtained. The error is greater for non-coinciding collocation nodes than, in the case, of coinciding nodes. This happens due to the interpolation and distribution routines required by the immersed boundary methodology for non-coinciding nodes.

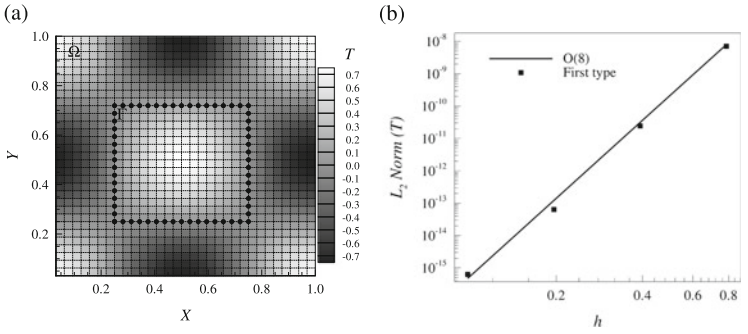


Fig. 1 Coinciding collocation nodes to $\Delta t = 10^{-2}[s]$: (**a**) geometrical domain and (**b**) rate of convergence

3.2 *Natural Convection in a Concentric Horizontal Cylindrical Annulus*

In order to validate the methodology for a physical problem, using non-coinciding collocation nodes, natural convection between concentric cylinders is simulated. The scheme of the computational domain used is seen in Fig. 2.

For Dirichlet boundary condition, it is necessary to impose the temperature in both cylinders: $T_i = T_r + \Delta T$ and $T_0 = T_r - \Delta T$, for the inner and the outer cylinder, respectively, where $T_r = 300$ K. The temperature difference is obtained from both Rayleigh (Ra) and Prandtl (Pr) numbers, given by $Ra = \frac{g\beta\Delta T l^3}{\alpha\nu}$ and $Pr = 0.71$, where $l = R_o - R_i$ $[m]$. The dimensionless parameters to temperature field and the radius are given by $\bar{T} = \frac{T-T_c}{T_h-T_c}$ and $R = \frac{r-R_i}{R_0-R_i}$.

The local Nusselt number is evaluated at both inner and outer cylinders, as proposed by Joo-Sik [4]. Figure 3 shows comparisons with experimental [5] data. One can see the dimensionless temperature profiles along the annulus radius for $\theta = 90°$, in Fig.3a. The local Nusselt number for the inner and the outer cylinders is depicted in Fig.3b. For both plots, a good agreement is observed between the obtained numerical results and the experimental data.

Moreover, the spatial accuracy and the computational cost of the MPEF is investigated for the annulus study as shown in Fig. 4a, b, respectively. Figure 4a shows the related L_2 error of the numerical solution, as function of the mesh spacing. For uniform meshes displaying four refinement levels: $h = L/32, L/64, L/128$

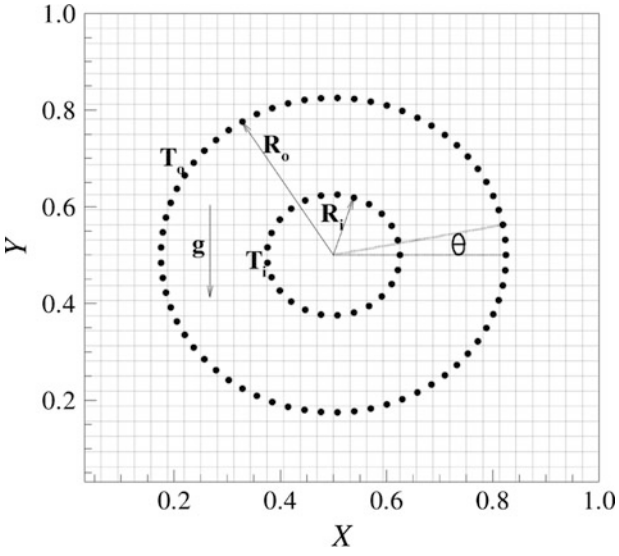


Fig. 2 Scheme for the complete domain for natural convection between horizontal concentric cylinders

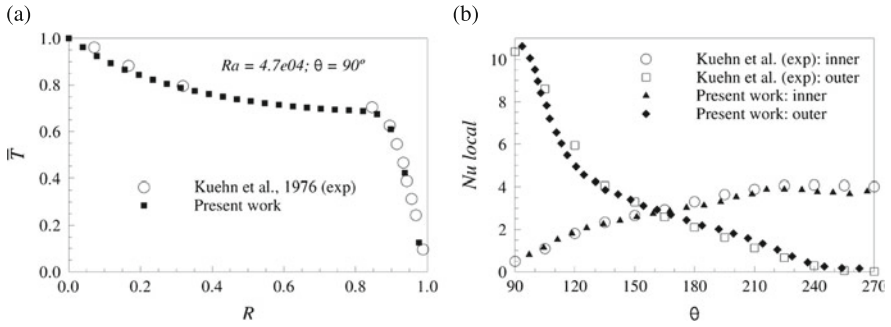


Fig. 3 Comparison with experimental data at $Ra = 4.7 \times 10^4$, (**a**) temperature distribution at 90° and (**b**) distribution of the local Nusselt number

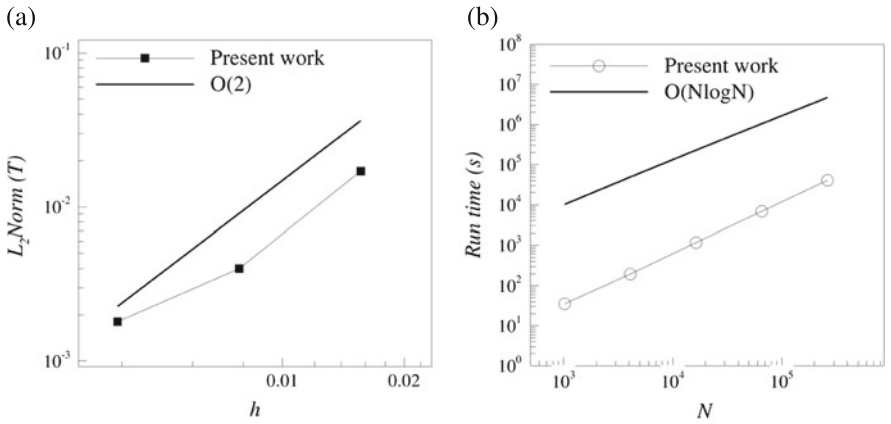


Fig. 4 Analysis of mesh refinement as function of (**a**) the L_2-norm for temperature and (**b**) run time

and $L/256$. Second order is achieved for non-coinciding Lagrangian and Eulerian meshes applied in a physical problem. Furthermore, it was observed in Fig. 4b that the computational run time with refinement of the mesh keep the order $O(NlogN)$, where N is the number of the point of the mesh discretization.

4 Conclusions

A new methodology, combining Fourier pseudospectral method with immersed boundary method, is extended from [6] in order to consider non periodical flows with internal energy transfer. A mathematical model for the first type thermal boundary condition is proposed and introduced to the Fourier pseudospectral numerical code. The code is verified using a synthesized analytical solution for

Navier-Stokes and energy equations. Non periodical problems with internal energy transfer are simulated using a Fourier pseudospectral methodology coupled with the immersed boundary methodology. The comparison to experimental results presents good agreement. Furthermore, it can be observed that this methodology keeps the computational cost at order $O(NlogN)$, as expected, reaching second order when applied to a physical problem. The methodology presented in the paper is currently being extended in order to contemplate second and third type boundary conditions.

5 Responsibility Notice

The authors are the only responsible for the printed material included in this paper.

Acknowledgements The authors would like to thank to PETROBRAS, CAPES, FAPEMIG, FAPEG, CAPES/PROEX, CNPq, UFU and UFG for the support.

References

1. C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods: Fundamentals in Single Domains* (Springer, Berlin, 2006)
2. C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang, *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics* (Springer, Berlin, 2007)
3. W.D. Henshaw, A fourth-order accurate method for the incompressible Navier-Stokes equations on overlapping grids. J. Comput. Phys. **113**, 13–25 (1994)
4. Y. Joo-Sik, Dual steady solutions in natural convection between horizontal concentric cylinders. Int. J. Heat Fluid Flow **17**, 587–593 (1996)
5. T.H. Kuehn, R.J. Goldstein, An experimental and theoretical study of natural convection in the annulus between horizontal concentric cylinders. J. Fluid Mech. **74**, 695–719 (1976)
6. F.P. Mariano, L.Q. Moreira, A. Silveira-Neto, J.C.F. Pereira, A new incompressible Navier-Stokes solver combining Fourier pseudo-spectral and immersed boundary method. Comput. Model. Eng. Sci. **59**, 181–216 (2010)
7. A.K. Tornberg, B. Engquist, Numerical approximations of singular source terms in differential equations. J. Comput. Phys. **200**, 462–488 (2004)
8. Z. Wang, J. Fan, K. Luo, K. Cen, Immersed boundary method for the simulation of flows with heat transfer. Int. J. Heat Mass Transf. **52**, 4510–4518 (2009)

Numerical Dissipation Control in High Order Shock-Capturing Schemes for LES of Low Speed Flows

D.V. Kotov, H.C. Yee, A.A. Wray, and B. Sjögreen

Abstract In Kotov et al. (Proceedings of ICCFD8, 2014) the LES of a turbulent flow with a strong shock by Yee and Sjögreen (Proceedings of ICOSAHOM 09, Trondheim, Norway, 2013) scheme indicated a good agreement with the filtered DNS data. There are vastly different requirements in the minimization of numerical dissipation for accurate turbulence simulations of different compressible flow types and flow speeds. The present study examines the versatility of the Yee and Sjögreen scheme for LES of low speed flows. Special attention is focused on the accuracy performance of this scheme using the Smagorinsky and the Germano-Lilly SGS models.

1 Introduction

For the last decade, high order shock-capturing methods with numerical dissipation controls have been the state-of-the-art numerical approach for direct numerical simulation (DNS) and large eddy simulation (LES) of turbulent flows with shocks. See for example [1–10]. The majority of these methods involve flow sensors with parameter tuning applied depending on the flow type. Some of the flow sensors were designed for certain flow types and might not preserve their high accuracy when used to simulate a different flow type. In a study presented in Johnsen et al. [3], all of the shock-capturing schemes involve tuning of the parameters. It appears that the Yee and Sjogreen filter scheme is not as accurate as the hybrid scheme presented in [3] as the key parameter κ responsible for minimizing the numerical

D.V. Kotov (✉)
Bay Area Environmental Research Institute, Petaluma, CA 94952, USA
e-mail: dmitry.v.kotov@nasa.gov; dmitry.kotov84@gmail.com

H.C. Yee • A. Wray
NASA Ames Research Center, Moffett Field, CA 94035, USA
e-mail: helen.m.yee@nasa.gov; Alan.A.Wray@nasa.gov

B. Sjögreen
Lawrence Livermore National Laboratory, Livermore, CA 94551, USA
e-mail: sjogreen2@llnl.gov

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_25

dissipation in the 2007 Yee and Sjogreen scheme [4] was mandated to set to a constant for all of test cases shown for results presented in [3]. See [2, 5] for a description of better control of numerical dissipation using a local κ. The hybrid scheme presented in [3] which employed the Ducros et al. flow sensor [6] also consists of a key tuning parameter δ. From our study presented below of the same Taylor-Green vortex problem considered in [3], the cut-off parameter δ to be 1 to achieve the best accurate result. On the other hand, for the isotropic turbulence with shocklets test case, the Ducros et al. flow sensor δ parameter has to be reduced, mostly by trial and error. Yet in another study [1] for turbulence interacting with a high speed stationary shock, depending on the Mach number and turbulent Mach number, different δ are required for each case.

In recognizing the different requirements on numerical dissipation control for DNS and LES of a variety of compressible flow types, Yee and Sjogreen, [2], presented a general framework for a local κ and the accompanying variety of flow sensors were introduced into their high order nonlinear filter scheme. Aside from suggesting different local κ formulation, Yee and Sjogreen also proposed the use of a combination of different flow sensors. Their proposed scheme with numerical dissipation control has not been studied extensively. A subset to the sequel to [2] was presented in [5]. This is yet another sequel to Yee and Sjogreen. The goal of this work is to examine the different combinations of flow sensors for DNS and LES of low speed turbulent flows.

2 High Order Nonlinear Filter Schemes

This section gives a brief overview of the high-order nonlinear filter scheme of Yee et al. [2, 4, 5, 7] for accurate computations of DNS and LES of compressible turbulence for a wide range of flow types by introducing as little shock-capturing numerical dissipation as possible.

Preprocessing Step Before the application of a high-order non-dissipative spatial base scheme, a preprocessing step is employed to improve numerical stability. The inviscid flux derivatives of the governing equations are split into the following three ways, depending on the flow types and the desire for rigorous mathematical analysis or physical argument.

- Entropy splitting of [8]. This is non-conservative and the derivation is based on the physical entropy variable and energy norm stability for the compressible Euler equations with boundary closure for the initial boundary value problem.
- The Ducros et al. splitting [9] for systems. This is a conservative splitting and the derivation is based on physical arguments.
- Tadmor entropy conservation formulation for systems [10]. The derivation is based on mathematical analysis. Preliminary study in [10] indicated the Tadmor entropy conservation formulation is more diffusive than the other two splittings.

Base Scheme Step A full time step is advanced using a high-order non-dissipative spatially central scheme on the split form of the governing equations. A summation-by-parts (SBP) boundary operator [11] and matching order conservative high-order free stream metric evaluation for curvilinear grids [12] are used. Note that the base scheme can be a high order compact scheme [13], the standard high order central schemes or spectral methods. However the same entropy stable SBP boundary closure for high order central schemes is not valid for the latter base schemes.

Post-Processing (Nonlinear Filter Step) To further improve the accuracy of the computed solution from the base scheme step, after a full time step of a base scheme step the post-processing step is used to nonlinearly filter the solution by a dissipative portion of a high-order shock-capturing scheme with a local flow sensor. The flow sensor provides locations and amounts of built-in shock-capturing dissipation that can be further reduced or eliminated. At each grid point a local flow sensor is employed to analyze the regularity of the computed flow data. Only the strong discontinuity locations would receive the full amount of shock-capturing dissipation. In smooth regions no shock-capturing dissipation would be added, unless high frequency oscillations are developed, owning to the possibility of numerical instability in long time integrations of nonlinear governing PDEs. In regions with strong turbulence, if needed, a small fraction of the shock-capturing dissipation would be added to improve stability. Note that the filter numerical fluxes only involve the inviscid flux derivatives regardless if the flow is viscous or inviscid. If viscous terms are present, a matching high order central difference operator (as the inviscid difference operator) is included on the base scheme step.

Let U^* be the solution after the completion of the full time step of the base scheme step. The final update of the solution after the filter step is (with the numerical fluxes in the y- and z-directions suppressed as well as their corresponding y- and z-direction indices on the x inviscid flux suppressed)

$$U_{j,k,l}^{n+1} = U_{j,k,l}^* - \frac{\Delta t}{\Delta x}[H_{j+1/2}^* - H_{j-1/2}^*], \quad H_{j+1/2}^* = R_{j+1/2}\overline{H}_{j+1/2}, \tag{1}$$

where $R_{j+1/2}$ is the matrix of right eigenvectors of the Jacobian of the inviscid flux vector in terms of Roe's average states based on U^*. $H_{j+1/2}^*$ and $H_{j-1/2}^*$ are "filter" numerical fluxes in terms of Roe's average states based on U^*. Denote the elements of $\overline{H}_{j+1/2}$ by $\overline{h}_{j+1/2}^l, l = 1, 2, \ldots, 5$, where

$$\overline{h}_{j+1/2}^l = \frac{\kappa_{j+1/2}^l}{2} w_{j+1/2}^l \phi_{j+1/2}^l. \tag{2}$$

Here $w_{j+1/2}^l$ is a flow sensor to activate the nonlinear numerical dissipation portion of a high order shock-capturing scheme $\frac{1}{2}\phi_{j+1/2}^l$, and $\kappa_{j+1/2}^l$ is a flow dependent positive parameter to control the amount of shock-capturing dissipation to be used. The nonlinear dissipative portion of a high-resolution shock-capturing scheme "$\frac{1}{2}\phi_{j+1/2}^l$" can be any shock-capturing scheme. The choice of the parameter $\kappa_{j+1/2}^l$

can be different for different flow types and is automatically chosen by using the local $\kappa^l_{j+1/2}$ described in [2]. The flow sensor $w^l_{j+1/2}$ can be a variety of formulae introduced in the literature or can be switched from one flow sensor to another, depending on the computed flow data at that particular location. For a variety of local flow sensors with automatic selection of the proper parameter, depending on different flow type, see [2]. The form of Tauber-Sandham [14] for the filter numerical flux uses the Ducros et al. flow sensor as $\kappa^l_{j+1/2}$ and the Harten artificial compression method formula (ACM) as the flow sensor indicated in [7] and similarly in [15] are part of the Yee and Sjögreen adaptive numerical dissipation control generalization filter formulae. The form of Ducros et al. flow sensor is $w = (\nabla \mathbf{u})^2/\big((\nabla \mathbf{u})^2 + \omega^2 + \varepsilon\big)$. Here $\mathbf{u}$ is the velocity vector, ω is the vorticity magnitude and ε is a small number to avoid division by zero (e.g., 10^{-6}). The Ducros et al. flow sensor consists of a cut off parameter δ that can be used to switch on or off the dissipative portion of the high order shock-capturing scheme. If δ is set to be one, the dissipation only switches on when it encounters a shock wave. For lower value of the cut off δ parameter, vorticity can be detected.

The current numerical experimental study is confined to the following four forms for the filter numerical flux. It is well known that for certain low speed turbulence flows, the schemes of choice are spectral and high order compact, or central schemes with SBP boundary closures. The nonlinear filter step is not needed and this option using the high order central scheme base scheme only is included as the fifth scheme for comparison (the last bullet below).

- The first form of the filter numerical flux indicated in [2] is where $\kappa^l_{j+1/2}$ is the Mach curve for low speed flow described in [2]. $w_{j+1/2}$ is the wavelet flow sensor. If the tenth-order central base scheme, entropy splitting and the dissipative portion of the ninth-order WENO scheme (WENO9) are employed, it is denoted by WENO9fi-Esplit-Wav $\kappa(i)$. If the Ducros et al. splitting is used, it is denoted by WENO9fi-Dsplit-Wav.
- The second form of the numerical flux is the same as the first form except $\kappa^l_{j+1/2}$ is a constant based on the initial Mach number of the flow. The corresponding schemes are denoted by Esplit-Wav $\kappa = const$ and WENO9fi-Dsplit-Wav $\kappa = const$
- The third form of the numerical flux is where $\kappa^l_{j+1/2}$ is a positive non-zero constant, and $w_{j+1/2}$ is the Ducros et al. flow sensor in conjunction with the δ cut off parameter. The corresponding schemes are denoted by WENO9fi-Esplit-Ducr & WENO9fi-Dsplit-Ducr.
- The fourth form of the numerical flux is where the Ducros et al. flow sensor is used as $\kappa^l_{j+1/2}$, and $w_{j+1/2}$ is the wavelet flow sensor or the ACM flow sensor. For the same base scheme and the dissipative portion of WENO9, it is denoted by WENO9fi-Esplit-WavD & WENO9fi-Dsplit-WavD (WENO9fi-Esplit-AcmD & WENO9fi-Dsplit-AcmD).
- The last form is when no nonlinear filter step is used, i.e., only the base scheme step is employed. It is denoted by C10-Esplit in the case of employing the tenth-order central base scheme with entropy splitting. If the Ducros et al. splitting is used, it is denoted by C10-Dsplit.

The subgrid-scale (SGS) Smagorinsky model denoted by LES1 using $C_s = 0.0085$ [16] and the dynamic Germano model [17, 18] denoted by LES2 are considered. All of the results shown use the third-order Runge-Kutta temporal discretization.

3 Test Cases

This section illustrates the performance of our high-order filter scheme for DNS and LES of two 3D low speed turbulence flows considered in [3]. The first test case is the nearly incompressible (inviscid) Taylor-Green vortex problem and its viscous counterpart. The second test case is the decay of an isotropic turbulence with shocklets for an initial turbulent Mach number $M_{t,0} = 0.6$. For both test cases grid convergence studies are performed using uniform 256^3, 128^3 and 64^3 grids for the DNS simulations. Grid convergence studies also are performed using uniform 128^3, 64^3 and 32^3 grids for LES computations. Studies found that for an accurate numerical dissipation control scheme, a coarse grid DNS using a uniform 64^3 grid compared well with the filtered DNS using a fine grid of 256^3 grid points (spectrally filtered to a 64^3 grid). For the LES computations the 32^3 grid is too coarse for obtaining an accurate solution, whereas, the 128^3 grid solutions are almost on top of the filtered DNS computation on the 256^3 grid. Here, only the results using the 64^3 are briefly discussed. Due to a page limitation, see [19] for extended comparisons with more relevant illustrations that are not able to include here.

Taylor-Green Vortex: The 3D compressible inviscid test case solve the Euler equations with gas constant $\gamma = 5/3$. The computational domain is a 2π square cube using a uniform 64^3 grid. Boundary conditions are periodic in all directions. The initial conditions are:

$$\begin{aligned} &\rho = 1, \quad p = 100 + ([\cos(2z) + 2][\cos(2x) + \cos(2y)] - 2)/16, \\ &u_x = \sin x \cos y \cos z, \quad u_y = -\cos x \sin y \cos z, \quad u_z = 0. \end{aligned} \tag{3}$$

The initial turbulent Mach number is $M_{t,0} = 0.042$ and the final time is $t = 10$. We also consider the viscous counterpart of the Taylor-Green vortex problem. In the viscous case the physical viscosity is assumed to follow a power-law: $\mu/\mu_{ref} = \left(T/T_{ref}\right)^{3/4}$. Here we use $\mu_{ref} = 0.005$ and $T_{ref} = 1$ in non-dimensional units. The initial Reynolds number is $Re_0 = 2040$. For this low-Mach number flow without high shear regime the simulation actually does not require any numerical dissipation. However, we use the same shock-capturing scheme with adaptive numerical dissipation control to demonstrate its accurate performance for such low-Mach number cases. The key study involves the assessment of accuracy of the computed solution using different forms of $\kappa^l_{j+1/2}$ and different values of δ mentioned above.

Inviscid Taylor-Green Vortex—DNS Scheme Comparison: In the inviscid case the kinetic energy should be constant. It can be used as a criterion to judge the accuracy of the four considered filter numerical fluxes. The coarse grid DNS (64^3 grid—no SGS model) comparison among different methods by examining the temporal evolution of the mean kinetic energy and enstrophy comparing with the 256^3 grid filtered DNS reference solution (figure not shown). The preservation of kinetic energy is achieved with C10-split, WENO9fi-Dsplit-WavD and WENO9fi-Dsplit-Wav $\kappa = 10^{-5}$, while WENO9fi-Dsplit-Wav $\kappa(i)$ obtains a small loss in energy after $t \approx 6$. All four methods presented on the enstrophy plot demonstrate good agreement with the semi-analytical solution [20], which is defined on the interval $0 \leq t \leq 3.5$. The enstrophy values obtained using WENO9fi-Dsplit-Wav $\kappa(i)$ are slightly smaller than those obtained using the other three methods.

Viscous Taylor-Green Vortex—DNS and LES Scheme Comparison: The temporal evolution of the mean-square velocity and enstrophy of the coarse grid DNS (no SGS model) results on a 64^3 grid by different methods are shown in [19] The reference solution is the DNS simulation using a 256^3 grid and spectral filtering to the 64^3 grid. For this viscous case the most accurate cut off parameter δ in WENO9fi-Esplit-WavD and WENO9fi-Dsplit-Ducr is when $\delta = 1$. The kinetic energy computed solutions by all considered methods matches the reference solution. The difference between methods is only visible on the enstrophy comparison, though all the results are very close to the reference solution. The methods using Ducros et al. split C10-Dsplit and WENO9fi-Dsplit-Wav $\kappa = 10^{-5}$ as well as WENO9fi-Esplit-Wav $\kappa(i)$ obtain slightly more accurate results than C10-Esplit and WENO9fi-Esplit-WavD.

The results obtained using the LES1 model is shown in Fig. 1. Results obtained in LES1 are closer to the reference solution than the results obtained using the dynamic model LES2 (figure not shown; see [19]). All LES methods underestimate both the

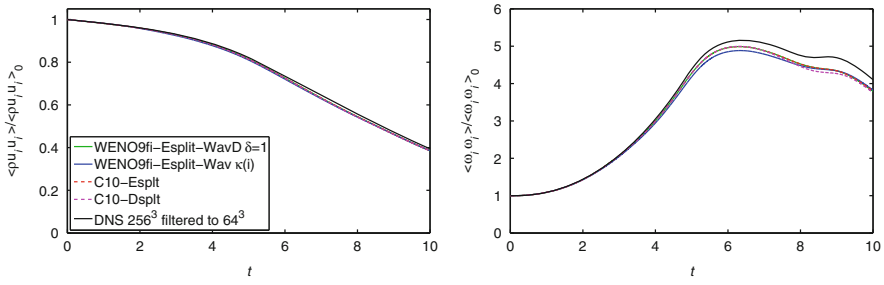


Fig. 1 LES1 comparison for the viscous Taylor-Green vortex problem using a 64^3 grid: Temporal evolution of the kinetic energy (*left*) and enstrophy (*right*). The reference solution is the DNS computation on a 256^3 grid and spectrally filtered to a 64^3 grid

kinetic energy and the enstrophy. WENO9fi-Esplit-Wav $\kappa(i)$ is slightly less accurate than C10-Dsplit and WENO9fi-Esplit-WavD. The accuracy by C10-Esplit and C10-Dsplit are almost the same.

Decaying Isotropic Turbulence with Shocklets: The second test case is the decaying compressible isotropic turbulence with eddy shocklets considered in [3]. For high enough turbulent Mach number, M_t weak shock waves (shocklets) develop spontaneously from the turbulent motions. For the current numerical experiment we set the initial $M_{t,0} = 0.6$. The filtered governing equations are solved using gas constant $\gamma = 1.4$. The computational domain is on the $2\pi^3$ cube with periodic boundary conditions in all directions. The physical viscosity is assumed to follow a power-law.

The initial condition consists of a random solenoidal velocity field $u_{i,0}$ that satisfies $E(k) \sim k^4 \exp(-2(k/k_0)^2)$, $\frac{3}{2}u_{rms,0}^2 = \frac{\langle u_{i,0}u_{i,0}\rangle}{2} = \int_0^\infty E(k)dk$. The brackets here denote averaging over the entire computational domain. For this study we put $u_{rms,0} = 1$ and $k_0 = 4$. The density and pressure fields are initially constant with initial turbulent Mach number $M_{t,0} = 0.6$ and Taylor-scale Reynolds $Re_{\lambda,0} = 100$. These parameters are defined as follows: $M_t = \frac{\sqrt{\langle u_i u_i\rangle}}{\langle c\rangle}$, $Re_\lambda = \frac{\langle\rho\rangle u_{rms}\lambda}{\langle\mu\rangle}$, $u_{rms} = \sqrt{\frac{\langle u_i u_i\rangle}{3}}$, $\lambda = \sqrt{\frac{\langle u_x^2\rangle}{\langle(\partial_x u_x)^2\rangle}}$. The time scale is $\tau = \lambda_0/u_{rms,0}$ and the final time is $t/\tau = 4$. The final turbulent Mach number is $M_t = 0.29$.

Unlike the Taylor-Green vortex case, the most accurate solutions are obtained using a smaller κ and for vales of δ between 0.7 and 1. Comparisons of the temporal evolutions of the mean-square velocity, enstrophy, temperature variance and dilatation using by the various filter numerical fluxes on a 64^3 coarse grid DNS (no SGS model) are shown in [19]. The reference solution was obtained from the DNS simulation using a 256^3 grid and spectral filtering to a 64^3 grid (digitized from [3]). The best results are obtained with C10-AV12, WENO9fi-Dsplit-Wav $\kappa(i)$ and WENO9fi-Esplit-Ducr. The cut-off parameter of the Ducros et al. sensor in WENO9fi-Esplit-WavD is $\delta = 0.7$. However, the results remain almost the same when δ increases slightly beyond 0.7. For the dilatation, the best match with the reference solution is obtained by method C10-AV12 . However, this scheme underestimates the enstrophy, while the rest of the methods either match or slightly overestimate the enstrophy. The results obtained using the LES1 model is shown in Fig. 2. The LES1 computations are closer to the reference solution than the dynamic model LES2 (figure not shown). The best results is obtained with C10-Esplit, WENO9fi-Esplit-Ducr and WENO9fi-Esplit-WavD. The spectra of this isotropic decaying turbulence test case were examined, the computed spectra by these schemes are as expected and results are not shown due to a space limitation. See [19] for the comparison.

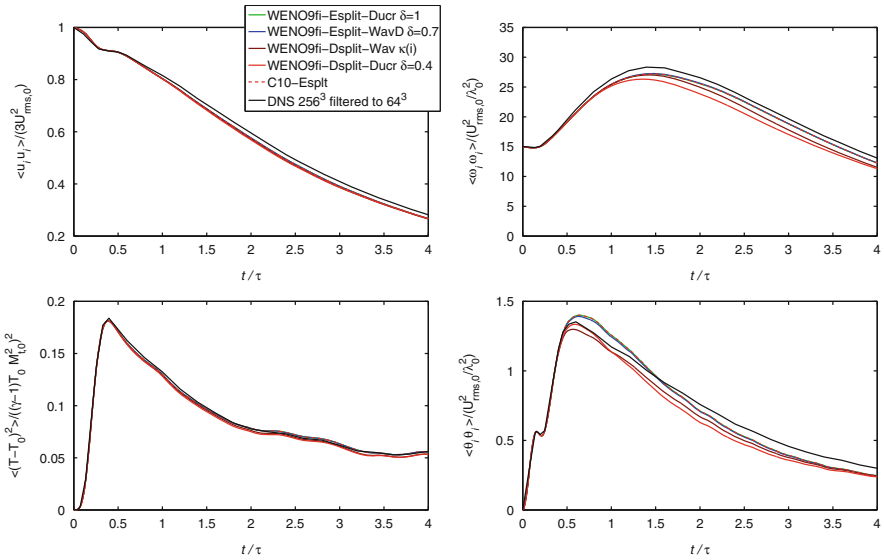


Fig. 2 LES1 comparison for the isotropic turbulence problem using a 64^3 grid: Temporal evolution of kinetic energy (*top left*), enstrophy (*top right*), temperature variance (*bottom left*) and dilatation, $\theta_i = \partial_i u_i$ (*bottom right*). The reference solution is the digitized solution from [3] on a 256^3 grid spectrally filtered to a 64^3 grid

4 Conclusions

The performance of the filter scheme with different flow sensors was demonstrated in LES and DNS of low-Mach number flows. Forms (1)–(4) for the filter numerical flux were chosen to demonstrate that for low speed turbulence flows without strong shear waves, the constant κ vs. the local $\kappa^l_{j+1/2}$ behave similarly. The main difference when using the constant κ parameter is that one has to know the flow structure of the entire evolution a priori in order to select the proper constant κ parameter. Contrary to the considered low speed flow test cases, our previous investigations [2, 5, 7, 21–24] for various complex high speed shock-turbulence interaction flows, employing the local $\kappa^l_{j+1/2}$ would provide an automatic selection of the amount of numerical dissipation needed at each flow location, thus, leading to a more accurate DNS and LES simulation with less tuning of parameters.

References

1. D. Kotov, H.C. Yee, A. Hadjadj, A. Wray, B. Sjögreen, in *Proceedings of ICCFD8* (Chengdu, Sichuan, China; also expanded version submitted to CiCP, 2014, 2014)
2. H.C. Yee, B. Sjögreen, in *Proceedings of ICOSAHOM 09* (Trondheim, Norway, 2013)

3. E. Johnsen, J. Larsson, A. Bhagatwala, W. Cabot, P. Moin, B. Olson, P. Rawat, S. Shankar, B. Sjögreen, H. Yee, X. Zhong, S. Lele, J. Comput. Phys. **229**, 1213 (2010)
4. H.C. Yee, B. Sjögreen, J. Comput. Phys. **225**, 910 (2007)
5. D. Kotov, H.C. Yee, B. Sjögreen, in *Proceedings of the ASTRONUM-2013* (Biarritz, France, 2013)
6. F. Ducros, V. Ferrand, F. Nicoud, C. Weber, D. Darracq, C. Gacherieu, T. Poinsot, J. Comput. Phys. **152**, 517 (1999)
7. H.C. Yee, N. Sandham, M. Djomehri, J. Comput. Phys. **150**, 199 (1999)
8. H.C. Yee, M. Vinokur, M. Djomehri, J. Comput. Phys. **162**, 33 (2000)
9. F. Ducros, F. Laporte, T. Soulères, V. Guinot, P. Moinat, B. Caruelle, J. Comput. Phys. **161**, 114 (2000)
10. B. Sjögreen, H.C. Yee, in *Proceedings of the 8th European Conference on Numerical Mathematics & Advanced Applications (ENUMATH 2009)* (Uppsala University, Uppsala, Sweden, 2009)
11. B. Sjögreen, H.C. Yee, in *Proceedings of the Turbulence and Shear Flow Phenomena 5 (TSFP-5)* (Munich, Germany, 2007)
12. B. Sjögreen, H.C. Yee, M. Vinokur, J. Comput. Phys. **265**, 211 (2014)
13. M. Ciment, Leventhal, Math. Comput. **29**, 985 (1975)
14. E. Touber, N. Sandham, Shock Waves **19**(6), 469 (2011)
15. S.C. Lo, G. Blaisdell, A. Lyrintzis, J. Numer. Methods Fluids **62**(5), 473 (2010)
16. G. Erlebacher, M.Y. Hussaini, C.G. Speziale, T.A. Zang, J. Fluid Mech. **238**, 155 (1992)
17. M. Germano, U. Piomelli, P. Moin, W. Cabot, Phys. Fluids **3**(7), 1760 (1991)
18. D.K. Lilly, Phys. Fluids **4**(3), 633 (1992)
19. D.V. Kotov, H.C. Yee, A. Wray, B. Sjögreen, Annual Research Briefs, Center for Turbulence Research, Stanford pp. 99–108 (2014; also submitted to J. Comput. Phys.)
20. M. Brachet, D. Meiron, S. Orszag, B. Nickel, R. Morf, U. Frisch, J. Fluid Mech. **130**, 411 (1983)
21. N.D. Sandham, Q. Li, H.C. Yee, J. Comput. Phys. **178**, 307 (2002)
22. B. Sjögreen, H.C. Yee, J. Sci. Comput. **20**, 211 (2004)
23. H.C. Yee, B. Sjögreen, Shock Waves **17**, 185 (2007)
24. H.C. Yee, B. Sjögreen, A. Hadjadj, Commun. Comput. Phys. **12**, 1603 (2012)

A Sub-cell Discretization Method for the Convective Terms in the Incompressible Navier-Stokes Equations

N. Kumar, J.H.M. ten Thije Boonkkamp, and B. Koren

Abstract In this contribution we present a sub-cell discretization method for the computation of the interface velocities involved in the convective terms of the incompressible Navier-Stokes equations. We compute an interface velocity by solving a local two-point boundary value problem (BVP) iteratively. To account for the two-dimensionality of the interface velocity we introduce a constant cross-flux term in our computation. The discretization scheme is used to simulate the flow in a lid-driven cavity.

1 Introduction

When solving the incompressible Navier-Stokes equations using a finite-volume method on a staggered grid, it is required to compute the interface velocities involved in the convective terms. Standard methods for computing the interface velocities use linear interpolations (taking the average values of the two neighbouring velocities), or use the upwind value. For incompressible flows the interface velocities attain the average value in case of diminishing flow ($\mathrm{Re} \downarrow 0$) or the upwind value in the limit $\mathrm{Re} \to \infty$. Using standard methods for computing the interface velocities we tend to ignore the nature of the flow in most cases. In the sub-cell computation, we solve a reduced momentum equation locally over an interval to compute the interface velocities. The interface velocities thus computed are consistent with the equations governing fluid flow.

In [1], we presented the idea of including a piecewise linear pressure gradient in the local BVP for the computation of interface velocities. In the present paper we further extend the method, by including the cross-flux term to the right-hand side (RHS) of the local BVP. The inclusion of the cross-flux term provides a two-dimensional character to the computed interface velocities. The sub-cell method

N. Kumar (✉) • J.H.M. ten Thije Boonkkamp • B. Koren
Department of Mathematics and Computer Science, Center for Analysis, Scientific Computing and Applications, Eindhoven University of Technology, PO Box 513, 5600MB Eindhoven, The Netherlands
e-mail: n.kumar@tue.nl; j.h.m.tenthijeboonkkamp@tue.nl; b.koren@tue.nl

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_26

is computationally more expensive than the standard methods for computing the interface velocities. However, the accuracy gain allows us to use *coarser* grids as compared to the standard methods.

In the next section we give details of the underlying finite-volume method used for solving the incompressible Navier-Stokes equations. In this article we focus only on the two-dimensional case. In Sect. 3, we give the details for the integral representation of the interface velocities. In order to account for the nonlinear character of the two-point local BVP, the computation of the interface velocities is done iteratively, which is discussed in Sect. 4. The results for the proposed discretization scheme are presented in Sect. 5.

2 Convective Terms and Interface Velocities

Consider the incompressible Navier-Stokes equations,

$$\nabla \cdot \mathbf{u} = 0, \tag{1a}$$

$$\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{uu}) = -\nabla p + \frac{1}{\mathrm{Re}} \nabla^2 \mathbf{u}, \tag{1b}$$

where $\mathbf{u} = (u, v)$ is the velocity of the fluid, p the pressure and Re the Reynolds number. We discretize the above system of equations using a finite volume scheme on uniform staggered grid, as shown in Fig. 1. The discrete system of equations is

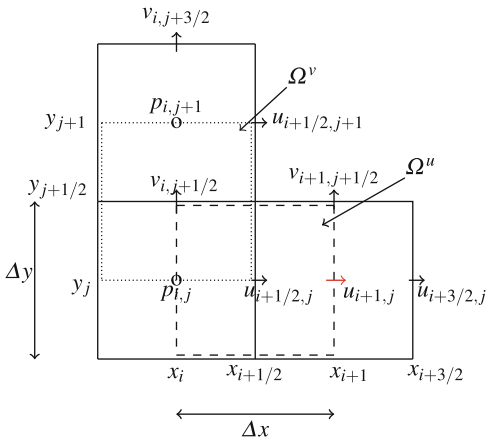


Fig. 1 Spatial discretization scheme on a uniform staggered grid, with pressure p defined in the cell center and the velocity components defined at the centers of the cell faces

written as

$$D\,\mathbf{u}(t) = \mathbf{r}_1(t), \tag{2a}$$

$$|\Omega|\mathbf{u}'(t) = -C(\mathbf{u}) + \frac{1}{\mathrm{Re}}L\mathbf{u}(t) - Gp(t) + \mathbf{r}_2(t), \tag{2b}$$

where D, C, L and G represent the discrete divergence, convection, diffusion and gradient operators, respectively, and where $|\Omega|$ represents the measure of the control volumes. The terms $\mathbf{r}_1(t)$ and $\mathbf{r}_2(t)$ include the boundary conditions for the system of equations, for details see [2].

Let us consider the u-component of the convective term $C(\mathbf{u})$ at $(x_{i+1/2}, y_j)$, i.e.,

$$\begin{aligned}\left(C^u(\mathbf{u})\right)_{i+1/2,\,j} =& \Delta y\left(u^2_{i+1,\,j} - u^2_{i,j}\right) + \\ & \Delta x\left(v_{i+1/2,\,j+1/2}u_{i+1/2,\,j+1/2} - v_{i+1/2,\,j-1/2}u_{i+1/2,\,j-1/2}\right).\end{aligned} \tag{3}$$

In order to compute the above term we need the interface velocities $u_{i+1,\,j}$, $u_{i+1/2,\,j+1/2}$ and $v_{i+1/2,\,j+1/2}$, the rest can be computed in similar way. We will focus on the computation of the interface velocity $u_{i+1,\,j}$ using the local momentum equation,

$$(u^2)_x - \epsilon u_{xx} = -p_x - \left((uv)_y - \epsilon u_{yy}\right) \qquad \left(x_{i+1/2} < x < x_{i+3/2};\;\; y = y_j\right), \tag{4}$$

where the flow is assumed to be locally steady ($u_t = 0$) and $\epsilon = 1/\mathrm{Re}$. Let $\mathsf{F}^{u,y} = (uv)_y - \epsilon u_{yy}$, then Eq. (4) becomes

$$(u^2)_x - \epsilon u_{xx} = -p_x - \mathsf{F}^{u,y}. \tag{5}$$

The above equation resembles a steady viscous Burgers equation. We assume that the pressure p is piecewise linear over $(x_{i+1/2}, x_{i+3/2})$, thus the pressure gradient p_x is piecewise constant with a jump at x_{i+1}. On the other hand, the cross-flux term $\mathsf{F}^{u,y}$ is constant over the interval. We now suppress the y-dependence of Eq. (5) and denote $u(x_i, y_j)$ by u_i. Thus we have to solve the equation for $x \in (x_{i+1/2}, x_{i+3/2})$ subject to the boundary conditions

$$u(x_{i+1/2}) = u_{i+1/2}, \quad u(x_{i+3/2}) = u_{i+3/2}, \tag{6}$$

in order to compute $u_{i+1} = u(x_{i+1})$ (indicated in red in Fig. 1).

Further, we linearize Eq. (5) by replacing the nonlinear term $(u^2)_x$ by $\mathsf{U}u_x$, where U is an estimate for the interface velocity $u_{i+1,\,j}$. The linearized equation is then solved iteratively, in order to account for the nonlinearity of the problem, for more details see [1], where it is assumed that $\mathsf{F}^{u,y} = 0$. In this paper, we briefly outline the method used in [1] and then extend it by including a constant cross flux term $\mathsf{F}^{u,y}$.

3 Integral Representation of the Interface Velocities

In the local BVP (5)–(6), we get the y-dependence of the velocity component u as a result of the inclusion of the cross-flux term $\mathsf{F}^{u,y}$. We introduce the following notation: $u' = u_x$, $p' = p_x$, $\mathsf{a} = \mathsf{U}/\epsilon$ and $\mathsf{P} = \mathsf{a}\Delta x$, with P being the local *Péclet number* for the control volume. Then Eq. (5) can be linearized and rewritten as

$$\epsilon(u' - \mathsf{a}u)' = p' + \mathsf{F}^{u,y}. \tag{7}$$

Using the integrating factor formulation $u' - \mathsf{a}u = e^{\mathsf{a}x}(e^{-\mathsf{a}x}u)'$ and integrating Eq. (7), we find

$$\epsilon(e^{-\mathsf{a}x}u)' = e^{-\mathsf{a}x}\big(I(x) + \mathsf{F}^{u,y}(x - x_{i+1}) + K\big), \qquad I(x) = \int_{x_{i+1}}^{x} p'(\xi)\mathrm{d}\xi.$$

Note that we begin the integration from x_{i+1}, as p' has a jump at $x = x_{i+1}$. Next integrating the equation from $x_{i+1/2}$ to x and using the boundary condition $u(x_{i+1/2}) = u_{i+1/2}$ yields

$$u(x) = e^{\mathsf{a}(x-x_{i+1/2})}u_{i+1/2} + \frac{1}{\epsilon}\int_{x_{i+1/2}}^{x} e^{\mathsf{a}(x-\xi)}I(\xi)\mathrm{d}\xi + \frac{1}{\epsilon}\mathsf{F}^{u,y}\int_{x_{i+1/2}}^{x} e^{\mathsf{a}(x-\xi)}(\xi - x_{i+1})\mathrm{d}\xi + \frac{1}{\epsilon}K\int_{x_{i+1/2}}^{x} e^{\mathsf{a}(x-\xi)}\mathrm{d}\xi.$$

We now introduce the scaled x-coordinate σ defined as

$$\sigma := \sigma(x) = \frac{x - x_{i+1/2}}{\Delta x}, \qquad \big(0 \le \sigma \le 1\big).$$

Using the scaled coordinate we get

$$u(\sigma) = e^{\mathsf{P}\sigma}u_{i+1/2} + \frac{1}{\epsilon}\Delta x\, J(\sigma) + \frac{1}{\epsilon}\mathsf{F}^{u,y}\Delta x^2\int_0^{\sigma} e^{\mathsf{P}(\sigma-\eta)}\Big(\eta - \frac{1}{2}\Big)\mathrm{d}\eta + \frac{K}{\mathsf{U}}(e^{\mathsf{P}\sigma} - 1),$$

$$J(\sigma) = \int_0^{\sigma} e^{\mathsf{P}(\sigma-\eta)}I(x_{i+1/2} + \eta\Delta x)\mathrm{d}\eta.$$

The integral in the RHS, which gives the contribution of the cross-flux term in the interface velocity, is given by

$$\int_0^{\sigma} e^{\mathsf{P}(\sigma-\eta)}\Big(\eta - \frac{1}{2}\Big)\mathrm{d}\eta = G(\sigma;\mathsf{P}), \quad \mathsf{G}(\sigma;\mathsf{P}) := \frac{1}{\mathsf{P}^2}\Big(\Big(1 - \frac{1}{2}\mathsf{P}\Big)(e^{\mathsf{P}\sigma} - 1) - \sigma\mathsf{P}\Big).$$

Thus we get

$$u(\sigma) = e^{\mathsf{P}\sigma} u_{i+1/2} + \frac{1}{\epsilon}\Delta x\, J(\sigma) + \frac{1}{\epsilon}\mathsf{F}^{u,y}\Delta x^2 \mathsf{G}(\sigma;\mathsf{P}) + \frac{K}{\mathsf{U}}(e^{\mathsf{P}\sigma} - 1).$$

Applying the boundary condition $u(x_{i+3/2}) = u_{i+3/2}$ we obtain

$$u(\sigma) = W(1-\sigma;-\mathsf{P})u_{i+1/2} + W(\sigma;\mathsf{P})u_{i+3/2} + \tfrac{1}{\epsilon}\Delta x\Big(J(\sigma) - W(\sigma;\mathsf{P})J(1)\Big) + \tfrac{1}{\epsilon}\mathsf{F}^{u,y}\Delta x^2\Big(\mathsf{G}(\sigma;\mathsf{P}) - W(\sigma;\mathsf{P})G(1;\mathsf{P})\Big),$$

where

$$W(\sigma;\mathsf{P}) = \frac{e^{\mathsf{P}\sigma} - 1}{e^{\mathsf{P}} - 1}, \quad \Big(0 \le W(\sigma;\mathsf{P}) \le 1; \quad W(1-\sigma;-\mathsf{P}) + W(\sigma;\mathsf{P}) = 1\Big).$$

The details for the computation of $J(\sigma)$ and $J(1)$ can be found in [1]. At this point we rewrite $u(\sigma)$ as a sum of components arising from terms in the RHS of Eq. (7), i.e,

$$u(\sigma) = u^{\mathrm{h}}(\sigma) + u^{\mathrm{p}}(\sigma) + u^{\mathrm{f}}(\sigma), \tag{10}$$

where

$$\begin{aligned} u^{\mathrm{h}}(\sigma) &:= W(1-\sigma;-\mathsf{P})u_{i+1/2} + W(\sigma;\mathsf{P})u_{i+3/2}, \\ u^{\mathrm{p}}(\sigma) &:= \tfrac{1}{\epsilon}\Delta x\Big(J(\sigma) - W(\sigma;\mathsf{P})J(1)\Big), \\ u^{\mathrm{f}}(\sigma) &:= \tfrac{1}{\epsilon}\mathsf{F}^{u,y}\Delta x^2\big(\tfrac{1}{\mathsf{P}}(W(\sigma;\mathsf{P}) - \sigma)\big). \end{aligned}$$

In case of no pressure gradient and no cross-flux, we get $u(\sigma) = u^{\mathrm{h}}(\sigma)$ on solving the homogeneous local BVP. Including the pressure gradient p' in the RHS of the homogeneous local BVP gives us $u(\sigma) = u^{\mathrm{h}}(\sigma) + u^{\mathrm{p}}(\sigma)$. Similarly, including the constant cross-flux term $\mathsf{F}^{u,y}$ gives us the additional component $u^{\mathrm{f}}(\sigma)$.

Finally the interface velocity $u_{i+1,j}$ can be computed as

$$u_{i+1,j} = u^{\mathrm{h}}_{i+1,j} + u^{\mathrm{p}}_{i+1,j} + u^{\mathrm{f}}_{i+1,j}.$$

For $x = x_{i+1}$, we have $\sigma = 0.5$, for which $W := W(0.5;\mathsf{P}) = (1 + e^{\mathsf{P}/2})^{-1}$. Now the velocity components are given by

$$u^{\mathrm{h}}_{i+1,j} = (1-W)u_{i+1/2,j} + W\,u_{i+3/2,j}, \tag{11a}$$

$$u^{\mathrm{p}}_{i+1,j} = -\frac{1}{4\epsilon}\Delta x^2\Big(A(-\mathsf{P}/2)(\delta_x p)_{i+1/2,j} + A(\mathsf{P}/2)(\delta_x p)_{i+3/2,j}\Big), \tag{11b}$$

$$u^{\mathrm{f}}_{i+1,j} = \frac{1}{\epsilon}\mathsf{F}^{u,y}_{i+1,j}\,\Delta x^2\big(W - \frac{1}{2}\big)\frac{1}{\mathsf{P}}, \tag{11c}$$

Algorithm 1: Iterative computation of the interface velocity $u_{i+1,j}$

Input: $u_{i+1/2,j}$, $u_{i+3/2,j}$, $(\delta_x p)_{i+1/2,j}$, $(\delta_x p)_{i+3/2,j}$, $\mathsf{F}^{u,y}_{i+1/2,j}$, $\mathsf{F}^{u,y}_{i+3/2,j}$, Δx and ϵ

Initialization: Set $u^{\mathrm{h}} = \frac{1}{2}(u_{i+1/2,j} + u_{i+3/2,j})$, $u^{\mathrm{p}} = 0$ and $u^{\mathrm{f}} = 0$

$$u_{i+1,j} = u^{\mathrm{h}} + u^{\mathrm{p}} + u^{\mathrm{f}}$$

Set, $u^{(\mathrm{k}-1)}_{i+1,j} = 0$ (the interface velocity from previous iteration)

Define *TOL*, as a control parameter for the convergence of the iterative procedure

Define $err := \left|u_{i+1,j} - u^{(\mathrm{k}-1)}_{i+1,j}\right|$

do

$u^{(\mathrm{k}-1)}_{i+1,j} = u_{i+1,j}$

$\mathsf{P} := \frac{1}{\epsilon} u_{i+1,j} \Delta x$ and $W = (1 + \exp(\mathsf{P}/2))^{-1}$

$u^{\mathrm{h}} = (1 - W)u_{i+1/2,j} + W u_{i+3/2,j}$

$u^{\mathrm{p}} = -\frac{\Delta x^2}{4\epsilon}\left(A(-\mathrm{P}/2)(\delta_x p)_{i+1/2,j} + A(\mathrm{P}/2)(\delta_x p)_{i+3/2,j}\right)$

$\mathsf{F}^{u,y}_{i+1,j} = (1 - W)\mathsf{F}^{u,y}_{i+1/2,j} + W\mathsf{F}^{u,y}_{i+3/2,j}$

$u^{\mathrm{f}} = \frac{1}{\epsilon}\mathsf{F}^{u,y}_{i+1,j}\Delta x^2\left(W - \frac{1}{2}\right)\frac{1}{\mathsf{P}}$

$u_{i+1,j} = u^{\mathrm{h}} + u^{\mathrm{p}} + u^{\mathrm{f}}$

$err = \left|u_{i+1,j} - u^{(\mathrm{k}-1)}_{i+1,j}\right|$

while $err \leq TOL$

where

$$(\delta_x p)_{i+1/2,j} = \frac{1}{\Delta x}(p_{i+1,j} - p_{i,j}) \quad \text{and} \quad A(z) = \frac{e^z - 1 - z}{z^2(e^z + 1)}.$$

4 Computation of the Interface Velocity

For solving the discretized momentum Eq. (2b), we need to compute $\mathsf{F}^{u,y} = (uv)_y - \epsilon u_{yy}$ at $(x_{i+1/2}, y_j)$ and $(x_{i+3/2}, y_j)$. Thus the terms $\mathsf{F}^{u,y}_{i+1/2,j}$ and $\mathsf{F}^{u,y}_{i+3/2,j}$ are computed at each time step. In order to compute $\mathsf{F}^{u,y}_{i+1,j}$ we take the weighted average of $\mathsf{F}^{u,y}_{i+1/2,j}$ and $\mathsf{F}^{u,y}_{i+3/2,j}$ analogous to Eq. (11a), i.e.,

$$\mathsf{F}^{u,y}_{i+1,j} = (1 - W)\mathsf{F}^{u,y}_{i+1/2,j} + W\mathsf{F}^{u,y}_{i+3/2,j}. \tag{12}$$

In order to account for the nonlinearity of Eq. (5), we iteratively compute the integral representation (Eq. (11)) of the interface velocities. Algorithm 1 describes the iterative computation of the interface velocity $u_{i+1,j}$.

For computing the convective term as given by Eq. (3), besides $u_{i+1,j}$ we also need $u_{i+1/2,j+1/2}$ and $v_{i+1/2,j+1/2}$. These velocities can also be computed using the

iterative local BVP method. Further details for the iterative computation of these interface velocities can be found in [1].

Note that the above discussion can be done analogously for the computation of the interface velocity $v_{i,\,j+1}$ involved in the convective term $C^v(\mathbf{u})$.

5 Numerical Results

We now use the proposed discretization scheme for computing the interface velocities involved in the convective terms of the incompressible Navier-Stokes equations applied to lid-driven cavity flow.

We present the results for Re = 100, on a hierarchy of rather coarse grids (8×8), (16×16) and (32×32). The results obtained using the present method are compared with those obtained using the standard average method, the upwind method and the 1-D local BVP method (absence of the cross flux term) described in [1]. We take the results from Ghia, Ghia and Shin [3] on a (128×128) grid as a reference. Figure 2 shows the velocity profiles for u along the vertical line passing through the geometric center of the cavity. It can be seen that the present method exhibits higher accuracy than the 1-D local BVP method for all grid sizes. Thus we have improved the 1-D local BVP method by including the cross-flux term, which provides a two-dimensional character to the interface velocities.

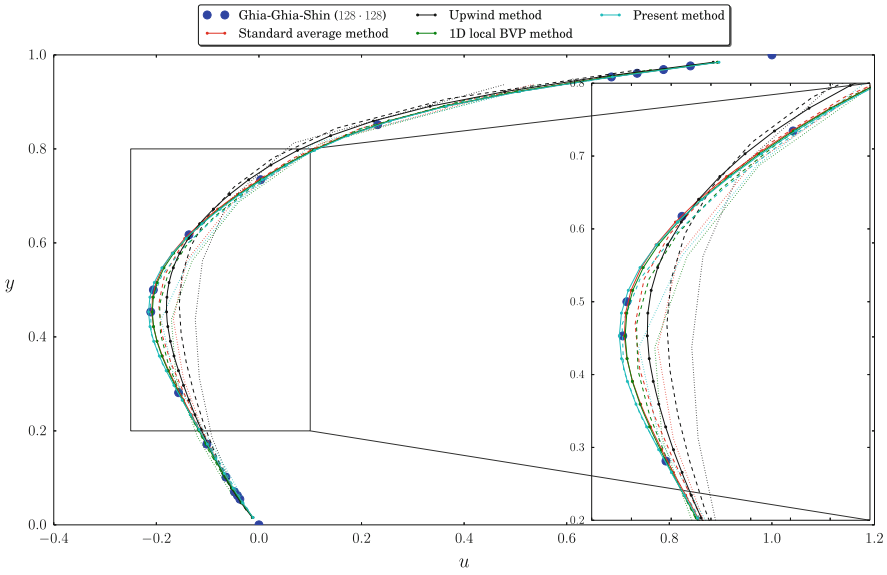


Fig. 2 Comparison of the velocity component u along the vertical centerline of the cavity for Re = 100. Grids used are 8×8 (*dotted lines*), 16×16 (*dashed lines*) and 32×32 (*solid lines*)

Next, Richardson extrapolation [4] was used to study the convergence of the present method. We observed that our method is second order convergent, just as the standard average method. Since the underlying FVM is second order, the overall accuracy of the scheme can never be higher than second order, even if the exact values of the interface velocities were known. It should be remarked though that conventional error analysis based on Taylor-series expansion is not very suitable here, since it leads to negative powers of the small diffusion coefficient, which might very well annihilate positive powers of the small mesh size. The computation of the interface velocities using the local two-point BVP approach, provides a more accurate estimate of the interface velocities, thereby leading to significantly smaller error constants (instead of higher order of accuracy), and hence allowing for the use of coarser grids.

Similar comparison of the velocity profiles along the geometric center of the cavity is also done for the case of Re $=$ 1000, which is shown in Fig. 3. The velocity profile obtained with the present method is remarkable; it shows a minimum of about the same magnitude and y-location as the Ghia-Ghia-Shin solution. For larger values of y though, it agrees well with the upwind method. For the Re $=$ 1000 case the flow separates from the flat cavity walls, leading to secondary and even smaller vortices. Flow separation from flat walls easily gives rise to non-unique solutions. Non-uniqueness seems to occur here indeed. In our opinion, the present results show that this high-Re test case can not be really relied upon in investigating the accuracy of discretization methods for the incompressible Navier-Stokes equations.

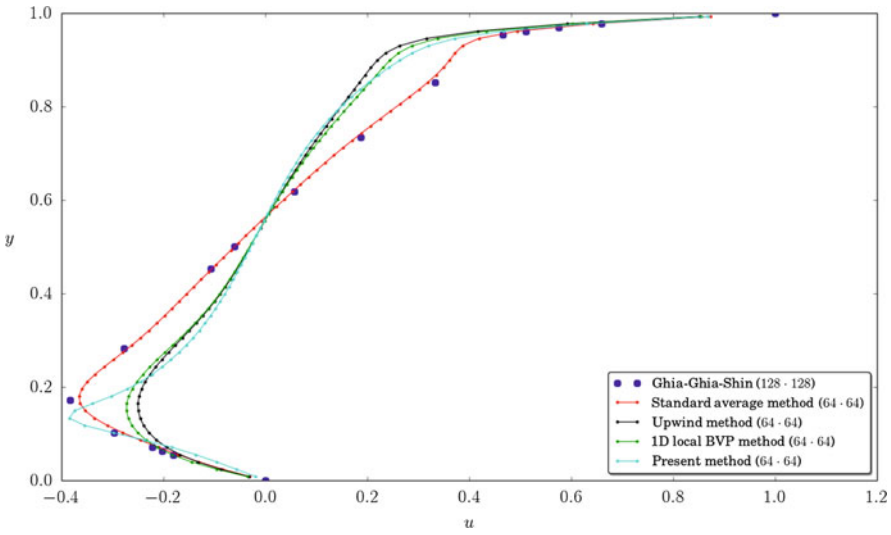


Fig. 3 Comparison of the velocity component u along the vertical centerline of the cavity for Re $=$ 1000

6 Conclusions

In the preceding sections we proposed a method for the sub-cell computation of interface velocities using local two-point BVPs. We presented an integral representation for the interface velocities as a sum of the components u^{h}, u^{p} and u^{f} given by Eq. (11). The integral representation is then solved iteratively to evaluate the interface velocities. As observed from Fig. 2, the present method exhibits higher accuracy than the 1-D local BVP method. Thus inclusion of the cross-flux term in the local BVP gives us higher accuracy by taking into account the two-dimensionality of the interface velocities.

Acknowledgements This work is part of the research programme of the Foundation for Fundamental Research on Matter (FOM), which is part of the Netherlands Organisation for Scientific Research (NWO).

References

1. N. Kumar, J.H.M. ten Thije Boonkkamp, B. Koren, A new discretization method for the convective terms in the incompressible Navier-Stokes equations, in *Finite Volumes for Complex Applications VII - Methods and Theoretical Aspects*, pp. 363–371 (Springer, Berlin, 2014)
2. B. Sanderse, Energy-conserving Runge-Kutta methods for the incompressible Navier-Stokes equations. J. Comput. Phys. **233**, 100–131 (2013)
3. U. Ghia, K.N. Ghia, C.T. Shin, High-Re solutions for the incompressible flow using the Navier-Stokes equations and a multigrid method. J. Comput. Phys. **48**, 387–411 (1982)
4. P.J. Roache, Quantification of uncertainty in computational fluid dynamics. Annu. Rev. Fluid. Mech. **28**, 123–160 (1997)

Localization in Spatial-Spectral Method for Water Wave Applications

R. Kurnia and E. van Groesen

Abstract In the description of water waves, dispersion is one of the most important physical properties; it specifies the propagation speed as function of the wavelength. Accurate modelling of dispersion is essential to obtain high-quality wave propagation results. The relation between speed and wavelength is given by a non-algebraic relation; for finite element/difference methods this relation has to be approximated and leads to restrictions for waves that are propagated correctly. By using a spectral implementation dispersion can be dealt with exactly above flat bottom using a pseudo-differential operator so that all wavelengths can be propagated correctly. However, spectral methods are most commonly applied for problems in simple domains, while most water wave applications need complex geometries such as (harbour) walls, varying bathymetry, etc.; also breaking of waves requires a local procedure at the unknown position of breaking. This paper deals with such inhomogeneities in space; the models are formulated using Fourier integral operators and include non-trivial localization methods. The efficiency and accuracy of a so-called spatial-spectral implementation is illustrated here for a few test cases: wave run-up on a coast, wave reflection at a wall and the breaking of a focussing wave. These methods are included in HAWASSI software (Hamiltonian Wave-Ship-Structure Interaction) that has been developed over the past years.

1 Introduction

As a simple introduction, consider the linear theory of water waves in one spatial direction x and time t above a flat bottom at depth D. With the elevation described by $\eta(x,t)$, denote by $\hat{\eta}(k,t)$ the spatial Fourier transform. Then the dynamics of each

R. Kurnia (✉)
University of Twente, Enschede, The Netherlands
e-mail: r.kurnia@utwente.nl

E. van Groesen
University of Twente, Enschede, The Netherlands
LabMath-Indonesia, Bandung, Indonesia
e-mail: E.W.C.vanGroesen@utwente.nl

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_27

mode (fixed k) is governed by a harmonic oscillator

$$\partial_t^2 \hat{\eta}(k,t) = -\hat{\Omega}^2(k,D)\hat{\eta}(k,t)$$

where $\Omega(k,D)$ defines the dispersion relation. The solutions $\exp\left(i(kx \pm \hat{\Omega}(k,D)t)\right)$ are harmonic in space and time, and constitute the building blocks of wave propagating to the right and left with phase speed given by $\hat{\Omega}(k,D)/k$. For linear water waves, the dispersion relation is explicitly given by $\hat{\Omega}(k,D) = \sqrt{gk\tanh(kD)}$, which is a skew symmetric and concave function of k, causing the problems for finite element and difference implementations. By defining the pseudo-differential operator Ω with symbol $\hat{\Omega}$, the equation above can be written in real space as

$$\partial_t^2 \eta(x,t) = -\Omega^2 \eta(x,t).$$

Spectral methods are widely used in water wave application, see for instance early contributions of [3, 4, 7], and [5, 6] for applications in simple domains. To deal with applications with complex geometries, in [16] a spatial-spectral implementation using Fourier integral operators was introduced for waves above varying bottom; [10] deals with localization for breaking mechanism in fully dispersive models. This paper illustrates methods to deal with such spatial inhomogeneities in the full nonlinear equations; using FFT for nonlinear terms, an efficient implementation with exact dispersion properties is obtained.

In Sect. 2, the full nonlinear dynamic equations are formulated as a Hamiltonian system in surface variables only by approximating the interior fluid motion through the kinetic energy, leading to a dimension reduction. In Sect. 3 we deal with inhomogeneous extensions, and Sect. 4 describes the test cases.

2 Hamiltonian Boussinesq Model

We briefly describe the basic equations and the spatial-spectral numerical implementation.

Waves in one horizontal direction x on the surface of incompressible, inviscid fluid under the influence of gravity can be described for irrotational internal fluid motion by a set of Hamilton equations in terms of the surface elevation $\eta(x,t)$ and the fluid potential $\phi(x,t)$ at the surface. This observation of [17] and [2] follows from Luke's variation principle [13] as was shown by Miles [15]. The dynamic equations are determined by partial variational derivatives with respect to η and to ϕ of the Hamiltonian which are written as $\delta_\eta \mathscr{H}(\phi,\eta)$ and $\delta_\phi \mathscr{H}(\phi,\eta)$ respectively. The dynamic equations can be compactly presented in the physical variables η and

the tangential velocity $u = \partial_x \phi$ as

$$\begin{aligned} \partial_t \eta &= -\partial_x \delta_u \mathscr{H}(u, \eta) \\ \partial_t u &= -\partial_x \delta_\eta \mathscr{H}(u, \eta). \end{aligned} \tag{1}$$

The Hamiltonian is the total energy, the sum of potential energy $P(\eta)$ and kinetic energy $K(u, \eta)$. The potential energy is given by $\int \frac{1}{2} g \eta^2 \, dx$. The kinetic energy is difficult to express in the variables at the surface; it requires to solve the interior fluid potential $\Phi(x, z, t)$ that satisfies the Laplace equation in the interior (representing the incompressible and irrotational fluid conditions), the surface condition $\Phi = \phi$ at the surface and the impermeable bottom boundary condition. Dirichlet's principle defines K through a minimization problem of which the potential is the solution

$$K(\phi, \eta) = \min_{\Phi} \left\{ \frac{1}{2} \int \int |\nabla \Phi|^2 \, dz dx \, | \Phi = \phi \text{ at } z = \eta \right\}. \tag{2}$$

By applying Green's theorem, the kinetic energy can be written as

$$K(\phi, \eta) = \frac{1}{2} \int \phi \partial_N \Phi \, dx = \frac{1}{2} \int \phi L \phi \, dx.$$

in which $\partial_N \Phi = L(\phi)$ is the Dirichlet-to-Neumann (DtN) operator.

The kinetic energy can also be expressed in u and η as

$$K(u, \eta) = \frac{1}{2g} \int (Cu)^2 \, dx.$$

where C can be interpreted as a phase velocity operator. Then the DtN operator is given by $L = -\frac{1}{g} \partial_x C^* C \partial_x$. In [10] the kinetic energy has been expanded up to 5th order nonlinearity; in this paper we only describe the 2nd order model.

The operator C can be obtained exactly in two limiting cases. The first case is that of linear equations above constant depth D mentioned in Sect. 1, that are obtained by taking for C the phase speed, i.e. the pseudo-differential operator with symbol

$$\hat{C}(k, D) = \hat{\Omega}(k, D)/k. \tag{3}$$

Another limit is obtained for long waves (all linear waves have the same velocity, no dispersion); above bathymetry with varying depth $D(x)$, the operator C is obtained as

$$C_{SW} = \sqrt{g(D(x) + \eta)}.$$

These two cases are obtained as the limits of the general Fourier integral operator with symbol $\hat{C}(k, H(x,t))$, where H is the total depth $H(x,t) = D(x) + \eta(x,t)$. The presence of $\eta(x,t)$ in H leads to equations that are second order accurate; as shown in [9].

The Fourier integral operator with symbol $\hat{C}(k,H)$ in the spatial-spectral implementation is given by

$$Cu = \frac{1}{2\pi}\int \hat{C}(k,H)\,\hat{u}(k)\,e^{ikx}dk = \mathscr{F}^{-1}\left[\hat{C}(k,H)\,\mathscr{F}(u)(k)\right]$$

where $\mathscr{F}$ and $\mathscr{F}^{-1}$ denote Fourier and inverse Fourier transformation. Note that the calculation of Cu (and the adjoint C^* needed in the DtN operator) requires a Fourier transformation for each value of x. The implementation of the Fourier integral operator is robust but quite time consuming; it can be made more efficient by a piecewise constant approximation using partition of unity of the interval of values of H, or by an interpolation method as described in [10, 16].

3 Spatial Localization in Spectral Implementations

In this section we describe localization methods in the spatial-spectral implementation of the wave equations to deal with wave generation, run up on a coast and reflection at a wall, and wave breaking.

3.1 Wave Generation

The dynamic equations for the numerical implementation are given as follows

$$\begin{aligned}\partial_t\eta &= -\partial_x\delta_u\mathscr{H}(u,\eta) + C\eta.\chi_d + S(x,t)\\ \partial_t u &= -\partial_x\delta_\eta\mathscr{H}(u,\eta) + Cu.\chi_d\end{aligned} \tag{4}$$

The second term in the right hand side (RHS) of both equations is a damping term that is localized using a smooth function χ_d to define a damping zone near the end points of the computation interval in order to prevent periodic looping in the spectral description.

The third term in the RHS of the continuity equation is a source term $S(x,t)$ that act as an embedded influx to generate waves. As described in [12], for a given influx signal $s_0(t)$ that defines the desired elevation at a given position in a specified direction, the source $S(x,t)$ is not unique: the spatial-temporal Fourier transform $\bar{S}(k,\omega)$) is unique only if k and ω are related by dispersion relation. This makes it possible to decide about the extent of the spatial generation area by modifying the given temporal function $s_0(t)$.

3.2 Run-Up on a Coast and Reflection at a Wall

When a wave is running-up a coast, the shoreline is moving and the governing dynamic equations hold on the wet side of the changing simulation interval. For a wave colliding at a wall, the simulation interval is restricted to the area in front of the wall. It turns out that for both cases a Heaviside function can be used to define the wet and dry domain; this corresponds to restricting the interval of the kinetic energy functional. The Heaviside function χ is inserted in the Hamiltonian as

$$\mathscr{H}(u,\eta) = \frac{1}{2}\int \left[g\eta^2 + \frac{1}{g}(Cu)^2\right]\chi\, dx. \tag{5}$$

with χ defined for the two cases as follows

$$\chi_{runup} = \begin{cases} 0 \text{ if } H(x,t) - H_{min} < 0 \\ 1 \text{ if } H(x,t) - H_{min} \geq 0 \end{cases} \qquad \chi_{wall} = \begin{cases} 0 \text{ if } x - x_{wall} > 0 \\ 1 \text{ if } x - x_{wall} \leq 0 \end{cases}$$

Here x_{wall} is the wall position; for the run-up case, $H(x,t)$ is the total depth and H_{min} the minimum depth that can be simulated depending on the maximal wave number used in the simulation. The dynamic equations accurate to second order are given by

$$\begin{aligned} \partial_t \eta &= -\partial_x \left[C^* \left(Cu.\chi\right)/g\right] \\ \partial_t u &= -\partial_x \left[\left(g\eta + C'u.Cu./g\right)\chi\right] \end{aligned} \tag{6}$$

in which $C' = \partial_\eta C$.

3.3 Wave Breaking

In [10] a breaking mechanism is described for the fully dispersive Hamiltonian equations. As trigger mechanism for the onset of breaking a kinematic criterion $U/C > b$, with $b \in (0.7; 1)$, U the particle speed at the crest and C the crest speed, is used. An eddy viscosity model with decay in tangential velocity is used as the dissipation mechanism; the dissipation conserves momentum and is localized in the front face of the wave.

4 Test Cases

In this section we illustrate the simulation capacity of the HAWASSI code for various cases.

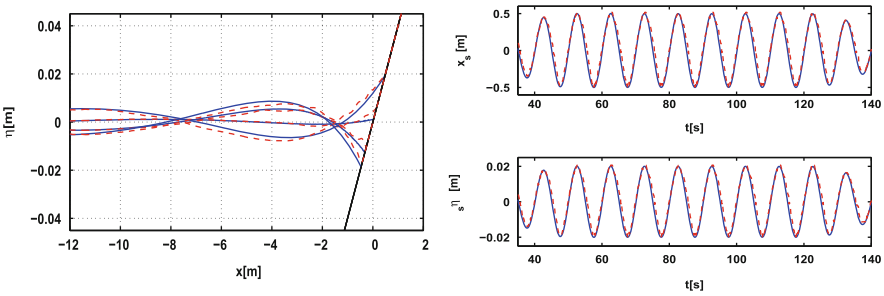


Fig. 1 Spatial wave profiles at four different times of the standing wave (*left*) and the horizontal (*top right*) and vertical movement (*bottom right*) of the shore-point; analytic approximation (*blue*, *solid-line*) and simulation with the HAWASSI code (*red*, *dashed-line*)

4.1 Harmonic Wave Run-Up on a Coast

A non-breaking long wave running up an impermeable slope is considered as in [8, 14]. The initial signal is harmonic with period 10 s, wave height 0.006 m; the bathymetry is 5 m deep at the flat area and decreases with a 1:25 slope to a shore. Wave run-up and run-down produces a standing wave. We compare the simulation result with an analytic approximate solution that is calculated based on [1]. This analytic approximation uses the nonlinear shallow water equations; a difference with the simulation that uses the fully dispersive model is visible in Fig. 1. At the left spatial profiles of the standing wave at four different times are shown for both methods; at the right the corresponding vertical and horizontal movement of the shore-point are plotted. The two result are qualitatively similar, with some quantitative differences. The relative computation time *Crel*, defined as the cpu-time divided by the total time of simulation, is about $Crel = 3.3$.

4.2 Reflection at a Wall

We study reflection at a wall of harmonic waves with initial amplitude 0.5 m above a flat bottom at depth 5 m. Three different periods are considered: 2, 4 and 6 s with related wavelengths of 6.2, 22.2 and 38.6 m, respectively. The wave is generated at $x = 0$ m and the wall is located at $x = 125$ m. Figure 2 shows spatial wave profile of simulations with the linear HAWASSI code and the analytic solution. The plots show that the simulation performs well; a small phase shift is observed for the simulation with period 6 s. The relative computation time *Crel* is less than 1.

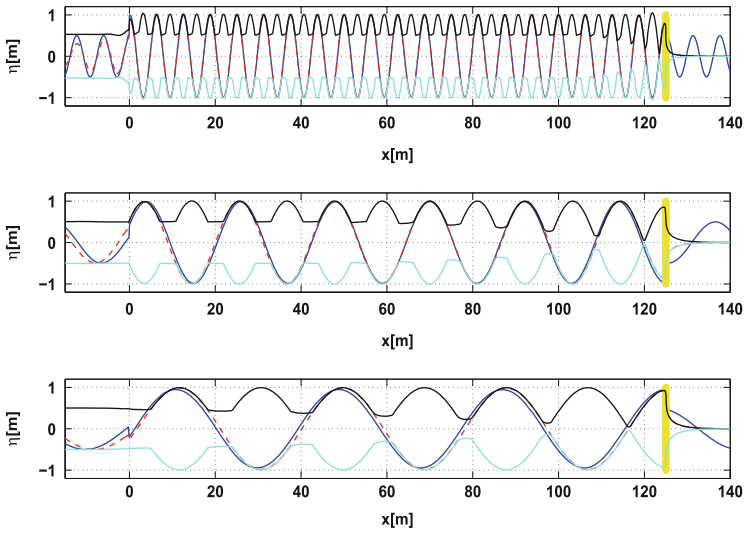


Fig. 2 Reflection at a wall positioned at 125 m for harmonic waves with periods 2 s (*top*), 4 s (*middle*) and 6 s (*bottom*). Plotted are the analytic solution (*blue*, *solid-line*), the simulation with the HAWASSI code (*red*, *dashed-line*), maximum temporal crest (*black*, *solid*), minimum temporal trough (*cyan*, *solid*) and the wall (*yellow*, *solid-line*)

4.3 Breaking of Focussing Wave Above a Flat Bottom

In this section we show simulation results for a focussing wave with initial steepness $kp.a = 0.13$, peak wave number kp, initial maximal amplitude $a = 0.12$ m, peak period $T = 1.96$ s, above a flat bottom at depth $D = 2.13$ m. This test case is one of a series of wave breaking experiments that have been conducted in the wave tank at TU Delft and registered as TUD1403Foc8 [11]. In the experiment the elevation is measured at six position: $W1, W2, \ldots, W6$ at $x = 10.31, 40.57, 60.83, 65.57, 70.31,$ and 100.57 m; the measured elevation at W1 is used as influx signal. For the simulation we use a third order Hamiltonian model with wave breaking mechanism. Figure 3 shows at the left a good agreement between the measurement and the simulation; the wave shape is well reproduced and a single breaking position is well predicted at $x = 58$ m (close to W3). The corresponding normalized amplitude spectra are shown at the right. Quantitatively, the correlation of the simulation and the measurement is at all positions larger than 0.97 of the maximal value 1. The relative computation time is $Crel = 0.56$.

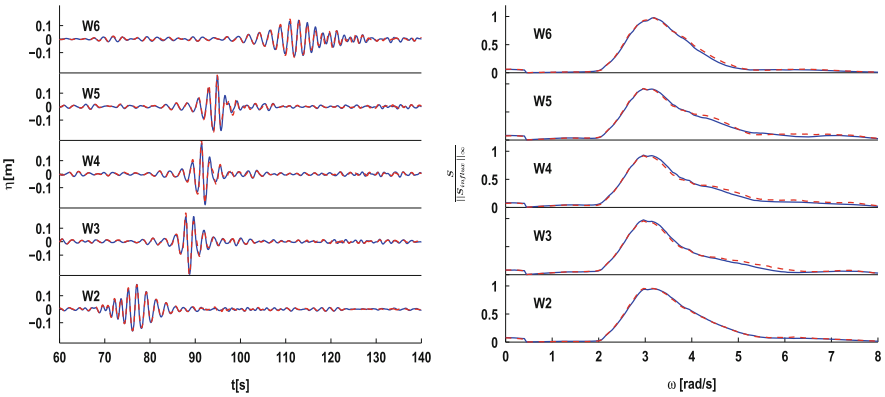


Fig. 3 Elevation time traces (*left*) and normalized spectra (*right*) at positions $W2$ to $W6$ for the breaking focussed wave (TUD1403Foc8); measurement (*blue*, *solid*) and simulation (*red*, *dashed-line*)

5 Conclusions

The spatial-spectral implementation using Fourier integral operator leads to consistent Hamiltonian modelling that preserves the exact dispersion relation. The non-trivial localization methods make it possible to deal with complex geometries. As test cases, we have shown simulation results of run-up on a coast, reflection at a wall and breaking of a focussed wave.

Acknowledgements This work is funded by the Netherlands Organization for Scientific Research NWO, Technical Science Division STW, project 11642.

References

1. M. Antuono, M. Brocchini, Solving the nonlinear shallow-water equations in physical space. J. Fluid Mech. **643**, 207–232 (2010)
2. L.J.F. Broer, On the Hamiltonian theory of surface waves. Appl. Sci. Res. **29**(1), 430–446 (1974)
3. W. Craig, C. Sulem, Numerical simulation of gravity waves. J. Comput. Phys. **108**(1), 73–83 (1993)
4. D.G. Dommermuth, D.K.P. Yue, A high-order spectral method for the study of nonlinear gravity waves. J. Fluid Mech. **184**, 267–288 (1987)
5. G. Ducrozet, F. Bonnefoy, D. Le Touzé, P. Ferrant, 3-D HOS simulations of extreme waves in open seas. Nat. Hazards Earth Syst. Sci. **7**(1), 109–122 (2007)
6. G. Ducrozet, F. Bonnefoy, D. Le Touzé, P. Ferrant, A modified high-order spectral method for wavemaker modeling in a numerical wave tank. Eur. J. Mech. B. Fluids **34**(0), 19–34 (2012)
7. J.D. Fenton, M.M. Rienecker, A Fourier method for solving nonlinear water-wave problems: application to solitary-wave interactions. J. Fluid Mech. **118**, 411–443 (1982)

8. A.B. Kennedy, Q. Chen, J.T. Kirby, R.A. Dalrymple, Boussinesq modeling of wave transformation, breaking, and runup. I: 1D. J. Waterw. Port Coast. Ocean Eng. **126**(1), 39–47 (2000)
9. R. Kurnia, E. van Groesen, Localization for spatial-spectral implementations of 1D Analytic Boussinesq equations (2015, submitted)
10. R. Kurnia, E. van Groesen, High order Hamiltonian water wave models with wave-breaking mechanism. Coast. Eng. **93**(0), 55–70 (2014)
11. R. Kurnia, T. van den Munckhov, C.P. Poot, P. Naaijen, R.H.M. Huijsmans, E. van Groesen, Simulation for design and reconstruction of breaking waves in a wavetank, in *Proceedings of the International Conference on Offshore Mechanics and Arctic Engineering - OMAE* (2015)
12. S.L. Lie, D. Adytia, E. van Groesen, Embedded wave generation for dispersive surface wave models. Ocean Eng. **80**(0), 73–83 (2014)
13. J.C. Luke, A variational principle for a fluid with a free surface. J. Fluid Mech. **27**, 395–397 (1967)
14. P.J. Lynett, T.R. Wu, P.L.F. Liu, Modeling wave runup with depth-integrated equations. Coast. Eng. **46**(2), 89–107 (2002)
15. J.W. Miles, On Hamiltons principle for surface waves. J. Fluid Mech. **83**(1), 153–158 (1977)
16. E. van Groesen, I. van der Kroon, Fully dispersive dynamic models for surface water waves above varying bottom, Part 2: hybrid spatial-spectral implementations. Wave Motion **49**(1), 198–211 (2012)
17. V.E. Zakharov, Stability of periodic waves of finite amplitude on the surface of a deep fluid. J. Appl. Mech. Tech. Phys. **9**(2), 190–194 (1968)

Sparse Modal Tau-Method for Helical Binary Neutron Stars

Stephen R. Lau and Richard H. Price

Abstract We sketch a modal tau approach for treating binary neutron stars, in particular a low-rank technique for dealing with the changing surface of a tidally distorted star.

1 Introduction and Preliminaries

We describe aspects of ongoing work toward solution of a specific problem: construction of initial data for relativistic binary neutron stars. Binary inspiral requires a starting configuration for the gravitational field and stellar structure. A promising way to provide such configurations is to retain the nonlinearities of relativistic gravitation, but suppress the radiation reaction that drives the inspiral. The resulting problem has helical symmetry and involves the corresponding reduction of the wave operator. Elsewhere [1–3], we have considered the helically reduced wave equation (HRWE) as a model problem. In terms of co-rotating Cartesian coordinates x, y, z (with the y-axis as the rotation axis; these are $\tilde{X}, \tilde{Y}, \tilde{Z}$ from [2]), the HRWE is

$$L\Psi = g(\mathbf{x}), \qquad L = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2} + \frac{\partial^2}{\partial z^2} - \Omega^2 \left(z\frac{\partial}{\partial x} - x\frac{\partial}{\partial z} \right)^2, \tag{1}$$

where Ω is the rotation rate and g a source. The equation is posed on a domain $\mathscr{D}$ with radiation conditions placed on (possibly part of) $\partial\mathscr{D}$. This problem is of mixed-type, although we solve it as a relaxation problem and in doing so have encountered no troubles. Here we describe coupling the HRWE to equations for stellar structure.

S.R. Lau (✉)
Mathematics and Statistics, University of New Mexico, Albuquerque, NM 87131, USA
e-mail: lau@math.unm.edu

R.H. Price
Center for Advanced Radio Astronomy, University of Texas at Brownsville, Brownsville, TX 78520, USA
e-mail: rprice.physics@gmail.com

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_28

Our work [1–3] thus far has been based on multidomain spectral methods, with several novel features. First, we adopt a modal approach (expansion coefficients as unknowns), and achieve sparse and banded systems through the use of integration matrices, as first described by Coutsias et al. [4]. In the context of iterative methods, the sparse formulation alone proves insufficient, and much of our work has focused on the preconditioning of such sparse modal systems. We have chiefly used block-Jacobi preconditioners, with the block sizes determined by the modal representations of the relevant operators. Recently [3], we have explored an interpolative decomposition technique to improve our preconditioners.

In this report we only touch on many of the key points underlying our approach, instead focusing on a new innovation in dealing with the interface of the stellar surface and exterior. Inclusion of the stellar surface adds significant computational complexity, since this surface, due to tidal deformation, is nonspherical. Moreover, it changes with each iteration in our relaxation scheme. Were the stellar surface realized as a boundary between subdomains, some form of bulk re-gridding (at least at the subdomain level) would be unavoidable. Realizing the stellar surface through tau conditions, we sketch how this can be avoided. While we have our specific target application in mind, the described technique could also be used for other problems in computational astrophysics and relativity (for example, the construction of Newtonian binaries or solution of the conformal thin sandwich equations [5]).

2 Modal-Tau Approximation of Nonspherical Stellar Surfaces

At each stellar surface regularity is lost both in the solution Ψ and right-hand side $g(\mathbf{x})$ of Eq. (1). Therefore, confining a stellar surface within a single spherical subdomain spoils spectral convergence. We describe here a treatment of stellar surfaces which should retain spectral convergence. Since working with the HRWE entails no further complications, for simplicity we here consider the Poisson problem

$$\nabla^2 \Phi = 4\pi G\rho(\mathbf{x}),\ \mathbf{x} \in D \quad \text{and} \quad \Phi(\mathbf{x}) = h(\mathbf{x}),\ \mathbf{x} \in \partial D, \tag{2}$$

where D is a 3d spherical ball with a 2d spherical boundary ∂D. Again for simplicity, here we consider an isolated Dirichlet problem associated with D, but in practice D is a subregion of an overall "two-center domain" $\mathscr{D}$ which is a larger 3d spherical ball containing both stars. Figure 2a depicts a two-center domain, with the closeup in Fig. 2b showing a configuration of subdomains surrounding one of the stars. D is the spherical region covered by this local configuration. Therefore, in practice interface conditions with the external subdomains would actually be specified on ∂D. The non-negative density ρ (stellar material) is compactly supported on D. That is, D contains the set $U = \{\mathbf{x} : \rho(\mathbf{x}) > 0\}$; however, the boundary ∂U is nonspherical and a priori unknown.

We typically partition subregion D into four subdomains, three concentric spherical shells and an center-filling rectangular block. The outermost shell (shell 3)

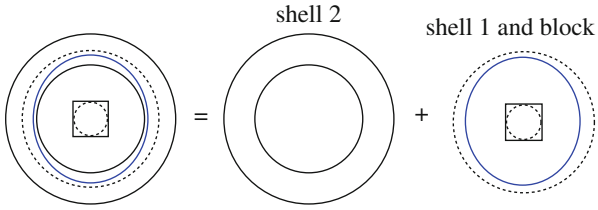


Fig. 1 Subdomain configuration surrounding a stellar surface depicted as an elliptical *bold curve*

is conforming with its neighbor (shell 2), i.e. the inner boundary of shell 3 and outer boundary of shell 2 are the same round 2d sphere. Shell 3 serves to couple D to the external cylindrical and block subdomains, and will not play a role in our description here. Figure 1 depicts the remaining subdomains which do play a role. An inner region consists of a spherical shell (shell 1) and the inner 3d block. For our purposes, we may consider these two subdomains as a single unit. An external region is another spherical shell (shell 2). Shells 1 and 2 overlap, and we assume that this overlap contains the boundary ∂U of the support of the stellar density ρ.

Let $\tilde{\Phi}^a_{\ell qn}$ represent the triply-indexed modal expansion coefficients on shell $a = 1, 2$. Here, the modal indices are $\ell = 0, \ldots, N_\theta$ dual to the polar angle, $q = 0, \cdots, N_\phi$ dual to the azimuthal angle, and $n = 0, \cdots, N^a_r$ dual to the radial coordinate. Throughout, shells 1 and 2 share the same angular resolution, so that N_θ and N_ϕ need not carry a superscript (subdomain index). We take $N_\phi = 2N_\theta$, although we enforce $\tilde{\Phi}^a_{\ell qn} = 0$ for $q > 2\ell$. As described in [2], we keep the nonphysical coefficients $\{\tilde{\Phi}^a_{\ell qn} : q > 2\ell\}$ to have the same data structure for the modal and nodal representations (convenient when using the spherical harmonic transform).

The representation of ∇^2 on a shell is block-diagonal. Precisely, for each (ℓ, q) we have an $(N_r + 1)$-by-$(N_r + 1)$ block. When $q > 2\ell$, each such block is the identity; however, the block corresponding to a physical mode $0 \le q \le 2\ell$ has the form

$$\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \\ \hline \mathsf{B}^{\ell q} \end{bmatrix} = I_{r[2]} A_r^2 - 2B_{r[2]} A_r - \ell(\ell+1) B^2_{r[2]}. \tag{3}$$

Here $\mathbf{0}$ represents a row of zeros, and $\mathsf{B}^{\ell q}$ is a sparse $(N_r - 1)$-by-$(N_r + 1)$ submatrix (here we use *superscripts* ℓ and q to label matrices). As described in [2], A_r represents multiplication by r in the Chebyshev basis and $B^2_{r[2]}$ double integration in this basis. A subscript [2] indicates two free rows of zeros. Therefore, the system sector corresponding to either shell 1 or 2 has $(N_\theta+1)^2$ free rows of zeros in which to enforce conditions at the stellar surface. We now describe how these rows are filled.

Each step of our iterative approach (see below) involves update of the density. We perform this update only on the inner region (shell 1 + block), and it yields

(modal coefficients for) the updated density ρ^1. This density is smooth and defined everywhere on the inner region. Moreover, $\rho^1(\mathbf{x}) < 0$ outside of the current nonspherical surface ∂U. On the external region (shell 2) we demand $\rho^2(\mathbf{x}) = 0$ for all points.

On shell $a = 1, 2$ the coefficients $\tilde{\Phi}^a_{\ell qn}$ determine the function (cf. Eq. (15) of [2])

$$\mathscr{P}_{N^a_r,N_\theta}\Phi^a(r,\theta,\phi) = \sum_{\ell=0}^{N_\theta}\sum_{q=0}^{2N_\theta}\sum_{n=0}^{N^a_r}\tilde{\Phi}^a_{\ell qn}\mathscr{E}^a_{\ell qn}(r,\theta,\phi). \tag{4}$$

Here the $\mathscr{P}$ merely indicates that the function arises as a finite expansion. Moreover, the basis functions $\mathscr{E}^a_{\ell qn}(r,\theta,\phi)$ are (with $m = 1,\ldots,N_\theta$)

$$\begin{aligned}
\mathscr{E}^a_{\ell 0n}(r,\theta,\phi) &= \overline{P}_{\ell 0}(\cos\theta)T_n(\xi^a(r)) \\
\mathscr{E}^a_{\ell,2m-1,n}(r,\theta,\phi) &= \overline{P}_{\ell m}(\cos\theta)\cos(m\phi)T_n(\xi^a(r)) \\
\mathscr{E}^a_{\ell,2m,n}(r,\theta,\phi) &= \overline{P}_{\ell m}(\cos\theta)\sin(m\phi)T_n(\xi^a(r)),
\end{aligned} \tag{5}$$

where the $\overline{P}_{\ell m}(u)$ are normalized associated Legendre functions (denoted by $\overline{P}^m_\ell(u)$ in [6]) and $\xi^a(r)$ maps the shell-a radial domain $[r^a_{\min}, r^a_{\max}]$ to $[-1, 1]$.

Conditions which enforce continuity of the numerical solution and its normal derivative across the stellar surface ∂U are then represented by

$$\mathscr{P}_{N^1_r,N_\theta}\Phi^1(\mathbf{x}_{jk}) = \mathscr{P}_{N^2_r,N_\theta}\Phi^2(\mathbf{x}_{jk}),\ \mathbf{n}\cdot\left(\nabla\mathscr{P}_{N^1_r,N_\theta}\Phi^1\right)(\mathbf{x}_{jk}) = \mathbf{n}\cdot\left(\nabla\mathscr{P}_{N^2_r,N_\theta}\Phi^2\right)(\mathbf{x}_{jk}), \tag{6}$$

where $\mathbf{x}_{jk} = \mathbf{x}(r_{jk},\theta_j,\phi_k)$ are Cartesian points on and $\mathbf{n}$ is the normal to ∂U. In practice the points $\mathbf{x}_{jk}$ are determined by the angular collocation points (θ_j,ϕ_k) corresponding the (discrete) spherical harmonic transform [7] and the corresponding radial values r_{jk}. The preceding equations determine $2(N_\theta+1)(2N_\theta+1)$ linear relationships between the modal coefficients $\tilde{\Phi}^1_{\ell qn}$ and $\tilde{\Phi}^2_{\ell qn}$. Indeed, there are $(N_\theta+1)(2N_\theta+1)$ physical points $\mathbf{x}_{jk}$. Among these relationships are, for example,

$$\sum_{\ell=0}^{N_\theta}\sum_{q=0}^{2N_\theta}\sum_{n=0}^{N^1_r}\tilde{\Phi}^1_{\ell qn}\mathscr{F}^1_{\ell qn}(\mathbf{x}_{jk}) = \sum_{\ell=0}^{N_\theta}\sum_{q=0}^{2N_\theta}\sum_{n=0}^{N^2_r}\tilde{\Phi}^2_{\ell qn}\mathscr{F}^2_{\ell qn}(\mathbf{x}_{jk}), \tag{7}$$

where $\mathscr{F}^a_{\ell qn}(\mathbf{x}) \equiv (\mathbf{n}\cdot\nabla\mathscr{E}^a_{\ell qn})(\mathbf{x})$. Evidently, this is a linear relationship expressible in terms of the vector direct sum of the modal coefficients $\tilde{\Phi}^1_{\ell qn}$ and $\tilde{\Phi}^2_{\ell qn}$ as well as a matrix $F_{2,1:2}$ which has $\mathscr{F}^1_{\ell qn}(\mathbf{x}_{jk})$ and $\mathscr{F}^2_{\ell qn}(\mathbf{x}_{jk})$ as entries. The lead index 2 on $F_{2,1:2}$ indicates that these relationships as intended for filling zero rows associated with the shell 2 row sector of the linear system, and the trailing 1:2 (colon notation)

Algorithm I. Computation of matching conditions across a stellar surface.

INPUT: Modal coefficients $\{\tilde{\rho}^1_{\ell qn}\}$ determining density $\rho^1(\mathbf{x})$ on shell 1.

OUTPUT: Matrices $\tilde{E}_{1,1:2}$ and $\tilde{F}_{2,1:2}$ defining tau conditions.

1: Find surface ∂U on which $\rho^1 = 0$. Precisely, for each angular collocation direction (θ_j, ϕ_k) compute radius $r_{jk} = r(\theta_j, \phi_k)$ corresponding to $\rho^1(\mathbf{x}_{jk}) = 0$.
2: Using the spherical harmonic transform, from the r_{jk} obtain the modal coefficients $\hat{r}_{\ell m}$ which define the stellar surface ∂U as $r(\theta, \phi) = \sum_{\ell m} \hat{r}_{\ell m} Y_{\ell m}(\theta, \phi)$.
3: Obtain the components $\mathbf{n} = (n^1, n^2, n^3)$ of the normal to ∂U. Here we use

$$(1-u^2)\frac{dP^m_\ell}{du} = (\ell+1)uP^m_\ell - (\ell-m+1)P^m_{\ell+1},$$

where $P^m_\ell(u)$ is an associated Legendre function with $u = \cos\theta$. This identity determines $\partial Y_{\ell m}/\partial x^k$ for $x^k = (x, y, z)$. Then $n^k \propto r^{-1}x^k - \sum_{\ell m} \hat{r}_{\ell m} \partial Y_{\ell m}/\partial x^k$.
4: For each shell $a = 1, 2$ compute and store the factors

$$\mathscr{E}^a_{\ell qn}(\mathbf{x}_{jk}), \qquad \mathscr{F}^a_{\ell qn}(\mathbf{x}_{jk}).$$

The angular factors defining these expressions may be computed once and stored. This step and the previous one defines the matrices $E_{1,1:2}$ and $F_{2,1:2}$.
5: Compute column-by-column spherical harmonic transforms $\tilde{E}_{1,1:2}$ and $\tilde{F}_{2,1:2}$.

that they will stretch across the shell 1 and shell 2 column sectors. The other set of matching conditions similarly determine a matrix $E_{1,1:2}$.

The matrices $E_{1,1:2}$ and $F_{2,1:2}$ have too many rows, since, as mentioned above, there are only $2(N_\theta + 1)^2$ free rows of zeros, whereas both $E_{1,1:2}$ and $F_{2,1:2}$ have $(N_\theta + 1)(2N_\theta + 1)$ rows. We reduce the number of equations as follows. Using the spherical harmonic transform, we compute the column-by-column transforms $\tilde{E}_{1,1:2}$ and $\tilde{F}_{2,1:2}$. The rows of these matrices which correspond to physical index pairs then define the tau conditions. The procedure is summarized in Algorithm II.

3 Numerical Experiments

This section describes two experiments. The first from [2] tests the accuracy of the basic linear solve. The second is a proof-of-concept experiment testing our method for treating stellar surfaces. Throughout, the two-center domain $\mathscr{D}$ and truncations are from [2]. Some modifications of this setup are necessary for the second experiment.

First, we consider the *retarded* solution to

$$\left(\nabla^2 - \partial_t^2\right)\Psi = -4\pi\delta^{(3)}(\mathbf{x} - \boldsymbol{\xi}(t)), \quad \boldsymbol{\xi}(t) = a\cos(\Omega t)\mathbf{e}_x + a\sin(\Omega t)\mathbf{e}_y, \tag{8}$$

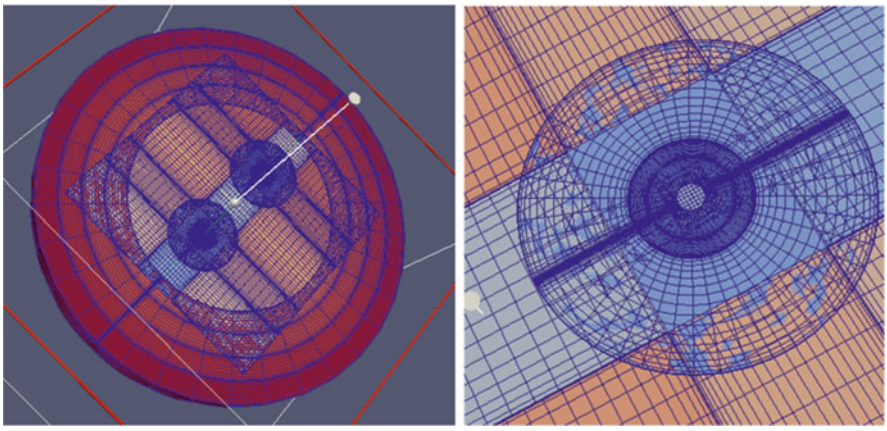


Fig. 2 Two-center domain decomposition for neutron star problem. (**a**) Domain decomposition, (**b**) close up of decomposition

where $a\Omega < 1$ so that the source point moves subluminally. When expressed in terms of co-rotating coordinates, this equation takes the form (1) with an inhomogeneity g determined by a Dirac-delta function excited at a fixed-location. An exact solution to this problem is expressible through Liénard-Wiechert potentials.

We consider two such Dirac-delta forcings for the HRWE (1), one of "charge" $Q_A = 0.5$ located on the z-axis at $z_A = -0.9$, and the other with $Q_B = 1.0$ and $z_B = 1.0$. The domain shown in Fig. 2 then needs modification. More precisely, the inner blocks and shells are excised, so that Dirac-delta sources lie outside of $\mathscr{D}$. We then use the Liénard-Wiechert solution to fix inner Dirichlet boundary conditions. As described in [2], at the outer boundary we enforce exact nonlocal radiation boundary conditions [8]. In all, $\mathscr{D}$ is comprised of 11 subdomains: 2 inner spherical shells (one around each source), 1 outer spherical shell, 3 blocks, and 5 cylinders.

We approximate the problem using our sparse modal-tau method, and then solve iteratively with preconditioned GMRES [9]. As described in [2], preconditioning plays a crucial role. Results for the full solve appear in Table 1. A numerical solution is a collection of modal expansion coefficients (one for each subdomain); however, comparisons with the exact solution are computed in physical space using the nodal grid dual to the modal expansion on each subdomain. These nodal grids are coarse, and the norms reported in the table do not settle down quickly. In the table each solve serves as the initial guess for the next; therefore, the iteration count drops.

We now turn to our second experiment involving the Newtonian equations for stellar structure with ∇^2 replaced by L from (1). We consider the simplest possible equation of state corresponding to an $n = 1$ polytrope. The system to solve is then

$$\begin{aligned} &L\Psi = 4\pi G\rho(\mathbf{x}), \qquad \mathscr{B}(\Psi) = 0, \qquad \rho(r_A\boldsymbol{\theta}_A) = 0 = \rho(r_B\boldsymbol{\theta}_B) \\ &\kappa_{A,B} = 2K_{A,B}\rho(\mathbf{x}) + \Psi(\mathbf{x}) - \frac{1}{2}\varpi^2(\mathbf{x})\Omega^2 \quad \text{for } \mathbf{x} \in U_A, U_B \end{aligned} \tag{9}$$

Table 1 Solution of the HRWE on $\mathscr{D}$

$\Omega = 0.1$						
MPSPD	L_2 error	L_2 norm	L_∞ error	L_∞ norm	Iterations	Tolerance
15.7	3.7532E−06	7.0509E−01	9.9579E−05	3.6556E+00	5	1.0000E−05
23.9	4.2440E−08	7.8382E−01	5.8222E−07	3.6563E+00	3	1.0000E−07
31.0	2.6333E−10	8.3492E−01	4.0406E−09	3.6564E+00	3	1.0000E−09
37.2	4.1855E−12	9.3982E−01	8.6696E−11	3.6565E+00	3	1.0000E−11
37.9	4.7733E−13	9.5252E−01	8.2254E−12	3.6565E+00	2	1.0000E−12

MPSPD stands for *modes per subdomain per dimension*. Note that an MPSPD of 37.9 corresponds to (11 subdomains) × $(37.9^3) \simeq$ 599,000 unknowns

Algorithm II. One iteration of equal-mass binary SCF method.

INPUT/OUTPUT: ρ, Ψ, and envelope functions $r_{A,B}$ for stars.

1: Solve $L\Psi = 4\pi\rho$ with interface conditions determined by envelope functions.
2: Let $\mathbf{x}^\pm$ be the north/south pole of one star, so $\rho(\mathbf{x}^\pm) = 0$. To keep $\mathbf{x}^\pm$ fixed throughout the iteration, choose κ and Ω to solve $\kappa = \Psi(\mathbf{x}^\pm) - \frac{1}{2}\varpi^2(\mathbf{x}^\pm)\Omega^2$.
3: On each star's inner shell/block update $\rho(\mathbf{x}) \leftarrow \frac{1}{2}K^{-1}\big(\kappa - \Psi(\mathbf{x}) + \frac{1}{2}\varpi^2(\mathbf{x})\Omega^2\big)$.
4: Via bisection, find $r_{A,B}(\cdot)$ from $\rho^1_{A,B}(\mathbf{x})$ in each angular direction (see above).

Here $\mathscr{B}(\Psi) = 0$ is the radiation boundary condition, $\varpi(\mathbf{x})$ is the distance from the rotation axis, and now U from before is the union of two sets, U_A and U_B, one for each star. The $\boldsymbol{\theta}_{A,B}$ are direction cosines relative to star A, B, and $r_{A,B}(\boldsymbol{\theta}_{A,B})$ are envelope functions for the (a priori unknown) free surfaces $\partial U_{A,B}$. The envelope functions are part of the solution. $K_{A,B}$ and $\kappa_{A,B}$ are constants.

We have considered the self consistent field (SCF) method for solving this problem. SCF is essentially a fixed-point method, and it is provably convergent for single stars [10]. However, its implementation for binary stars is complicated [11]. Here we consider its simplest form [12] for equal mass binaries; see Algorithm III. For this case $\kappa_A = \kappa_B$ and $K_A = K_B$. Although to date all methods that we have considered are experimental (and our results therefore tentative), the experiment here suggests that our modal treatment of stellar surfaces is viable.

To generate an initial configuration for the iteration, we consider the Newtonian Lane-Emden solution corresponding to a single $n = 1$ polytropic star:

$$\Phi = \begin{cases} -2K\rho^c[1 + (\pi r/R)^{-1}\sin(\pi r/R)] \\ -2K\rho^c R/r \end{cases}, \qquad \rho = \begin{cases} \rho^c(\pi r/R)^{-1}\sin(\pi r/R) & \text{for } r \le R \\ 0 & \text{for } r \ge R. \end{cases} \tag{10}$$

where the stellar radius is $R = \sqrt{K\pi/(2G)}$. In terms of the central density ρ^c the mass is $M = 4\pi^2\rho^c(R/\pi)^3$. As an initial configuration, we superpose two Lane-Emden stars, with $\rho^c_A = 100, R_A = 1.875, z_A = -5.0$ and $\rho^c_B = 100, R_B = 1.875$, $z_B = 5.0$. This configuration is roughly a dilation by 5 of the previous "Liénard-Wiechert" configuration. In (9) we fix $\Omega = 0.03125$ in L, but change Ω in $\kappa_{A,B}$. Kepler's law $\Omega^2 = (M_A + M_B)G/a^3$ with the L rate fixes G. Table 2 indicates

Table 2 Root-mean-square errors

Iteration	$\|r_A^{\text{new}}(\cdot) - r_A^{\text{old}}(\cdot)\| \Big/ \|r_A^{\text{new}}(\cdot)\|$	$\|L\Psi - 4\pi G\rho\|$
0	–	1.4473E−02
1	1.2481E−02	2.1790E−03
2	3.3602E−03	1.2159E−03
3	1.3015E−03	7.6345E−04
4	5.2798E−04	6.1734E−04
5	1.9848E−04	5.4127E−04
6	7.9286E−05	4.9521E−04

The middle column lists relative surface errors; in the last column the residual $L\Psi - 4\pi G\rho$ uses the previous Ψ and updated ρ from Algorithm III

convergence of the successive stellar surfaces for subdomain truncations similar to the lowest resolution run in the first "Liénard-Wiechert" experiment. Computation of the middle column errors uses Spherepack [7] quadrature weights. Convergence is lost for some truncations/domain decompositions.

4 Conclusion

We have described a modal tau approach for solving the structure equations for binary stars, in particular focusing on treatment of stellar surfaces. Our approach avoids re-computation of the bulk operators around the stars, although interface-tau conditions are altered as a stellar surface evolves. Such change involves low-rank modification of the previous linear system, and we have exploited the Woodbury identity to avoid full re-computation of subdomain preconditioners. We are currently working to implement the approach for helically symmetric binaries using Newton-Raphson iteration. In our experience the SCF method described here is sensitive to instabilities arising from the chosen truncations and domain decomposition.

Acknowledgements We gratefully acknowledge support through NSF grant No. DMS 1216866. Additionally, we thank Daniel Appelö for comments.

References

1. S.R. Lau, R.H. Price, Multidomain spectral method for the helically reduced wave equation. J. Comput. Phys. **227**, 1126–1161 (2007)
2. S.R. Lau, R.H. Price, Sparse spectral-tau method for the three-dimensional helically reduced wave equation on two-center domains. J. Comput. Phys. **231**, 7695–7714 (2012)
3. M. Beroiz, T. Hagstrom, S.R. Lau, R.H. Price, Multidomain, sparse, spectral-tau method for helically symmetric flow. Comput. Fluids **102**, 250–265 (2014)

4. E.A. Coutsias, T. Hagstrom, J.S. Hesthaven, D. Torres, Integration preconditioners for differential operators in spectral τ-methods, in *Proceedings of ICOSAHOM 1995, Spec. Issue Houston J. of Mathematics*, 1996, pp. 21–38
5. J.W. York Jr., Conformal "thin sandwich" data for the initial-value problem of general relativity. Phys. Rev. Lett. **82**, 1350–1353 (1999); H.P. Pfeiffer, J.W. York, Extrinsic curvature and the Einstein constraints. Phys. Rev. D **67**, Article ID 044022 (2003); F. Foucart, L.E. Kidder, H.P. Pfeiffer, S.A. Teukolsky, Initial data for black hole-neutron star binaries: a flexible, high-accuracy spectral method, Phys. Rev. D **77**, Article ID 124051 (2008)
6. M. Abramowitz, I.A. Stegun, *Handbook of Mathematical Functions* (Dover Publishing Inc, New York, 1970)
7. J.C. Adams, P.N. Swarztrauber, SPHEREPACK 3.0: a model development facility. Mon. Weather Rev. **127**, 1872–1878 (1999)
8. B. Alpert, L. Greengard, T. Hagstrom, Rapid evaluation of nonreflecting boundary Kernels for time-domain wave propagation. SIAM J. Numer. Anal. **37**, 1138–1164 (2000)
9. V. Frayssé, L. Giraud, S. Gratton, J. Langou, A set of GMRES routines for real and complex arithmetics on high performance computers. CERFACS Technical Report TR/PA/03/3, July 2007. http://www.cerfacs.fr/algor/
10. R.H. Price, C. Markakis, J.L. Friedman, Iteration stability for simple Newtonian stellar systems. J. Math. Phys. **50**, Article ID 073505 (2009)
11. I. Hachisu, Y. Eriguchi, K. Nomoto, Fate of merging double white dwarfs II. numerical method. Astrophys. J. **311**, 214–225 (1986)
12. I. Hachisu, A versatile method for obtaining structures of rapidly rotating stars. II. Three-dimensional self-consistent field method. Astrophys. J. Suppl. Ser. **62**, 461–499 (1986)

Uniformly Best Wavenumber Approximations by Spatial Central Difference Operators: An Initial Investigation

Viktor Linders and Jan Nordström

Abstract A characterisation theorem for best uniform wavenumber approximations by central difference schemes is presented. A central difference stencil is derived based on the theorem and is compared with dispersion relation preserving schemes and with classical central differences for a relevant test problem.

1 Introduction

Modelling wave propagation over sizeable intervals using finite differences is a common problem in fields ranging from aeroacoustics to seismology. For high frequency problems the numerical error may over time be dominated by inaccurate approximations of the dispersion relation, leading to errors in phase and group velocity, unless the spatial increment, Δx is very small.

A remedy, presented in [1] for central differences, is to perturb the classical schemes by an extra parameter but decreasing the formal accuracy. The new parameter is used to minimise the dispersion error in the $L^2[0, \pi/2]$ norm. Such schemes are known as Dispersion Relation Preserving (DRP). For other approaches based on similar ideas, see e.g. [2–4].

The DRP approach is disadvantageous in that it provides no means of obtaining wavenumber-specific error bounds. For problems involving a range of wavenumbers it is more convenient to minimise the dispersion error in the L^∞-norm. In this paper we present a characterisation theorem for *best* uniform wavenumber approximations. New central difference schemes are derived and compared with their classical and DRP counterparts.

V. Linders (✉) • J. Nordström
Division of Computational Mathematics, Department of Mathematics, Linköping University, SE-581 83 Linköping, Sweden
e-mail: viktor.linders@liu.se; jan.nordstrom@liu.se

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_29

2 Central Difference Schemes

We begin by demonstrating why classical central differences are suboptimal for wavenumber approximation. Consider a central difference stencil of order $2p$,

$$(u_x)_j = \frac{1}{\Delta x}\sum_{k=1}^{p} c_k^{(p)}(u_{j+k} - u_{j-k}) + \mathcal{O}(\Delta x^{2p}).$$

The numerical wavenumber of this scheme is (see e.g. [1])

$$\bar{\xi}_c = 2\sum_{k=1}^{p} c_k^{(p)} \sin(k\xi) \tag{1}$$

where $\xi = \Delta x\kappa$ and κ is the exact wavenumber of the propagating solution. Here we let $\xi \in [0, \xi_{max}] \subseteq [0, \pi]$.

A Taylor expansion reveals that to obtain desired accuracy, $c_k^{(p)}$ must satisfy

$$\begin{pmatrix} 1 & 2 & \dots & p \\ 1 & 2^3 & \dots & p^3 \\ \vdots & \vdots & & \vdots \\ 1 & 2^{2p-1} & \dots & p^{2p-1} \end{pmatrix} \begin{pmatrix} c_1^{(p)} \\ c_2^{(p)} \\ \vdots \\ c_p^{(p)} \end{pmatrix} = \begin{pmatrix} \frac{1}{2} \\ 0 \\ \vdots \\ 0 \end{pmatrix}. \tag{2}$$

The following observation is useful. For the proof, see [5].

Lemma 1 *Consider the function*

$$f_p(x) = 1 - 2\sum_{k=1}^{p} c_k^{(p)} k T_k(x)$$

where $c_k^{(p)}$ satisfies (2) and $T_k(x)$ is the k^{th} order Chebyshev polynomial of the first kind, uniquely defined through the relation $T_k(\cos(\phi)) = \cos(k\phi)$. Then

$$f_p(x) = d_p(1-x)^p$$

for some d_p that depends exclusively on p.

Theorem 1 *Classical central difference stencils underestimate the speed of propagating solutions.*

Proof Let

$$E_c(\xi) \equiv \xi - \bar{\xi}_c = \xi - 2\sum_{k=1}^{p} c_k^{(p)} \sin(k\xi)$$

be the error function associated with a classical central difference stencil of order $\mathcal{O}(\Delta x^{2p})$. Note from the definition of f_p and the Chebyshev polynomials that

$$\frac{\mathrm{d}E_c}{\mathrm{d}\xi} = f_p(\cos(\xi)).$$

It follows that $\frac{\mathrm{d}E_c}{\mathrm{d}\xi} = 0$ only when $\cos(\xi) = 1$. It is thus clear that $E_c(\xi)$ has an inflection point at $\xi = 0$ and no other extrema in the domain of interest. Consequently $E_c(\xi)$ is monotonic and since $E_c(0) = 0$ and $E_c(\pi) = \pi$ it is also increasing. We thus have $\bar{\xi}_c \leq \xi$ with equality only at $\xi = 0$. Therefore the classical central difference stencils underestimate the analytic wavenumber. It follows now that the relative error in the phase speed is

$$\frac{\bar{v}_p - v_p}{v_p} = \frac{\bar{\xi}_c}{\xi} - 1 \leq 0$$

and so the phase speed is underestimated. □

Let us now, like for DRP schemes, perturb the stencil by adding an additional coefficient, a_{p+1} without increasing the accuracy. The coefficients of the new stencil must solve the system (2) for each a_{p+1}, though this system is now underdetermined. Calling the coefficients of the new system a_k, $k = 1, \ldots, p+1$, we must have a linear dependence of the first p coefficients on the added parameter, a_{p+1}. We write $a_k = c_k^{(p)} + c'_k a_{p+1}$, $k = 1, \ldots, p$.

In view of (3) the numerical wavenumber of the perturbed stencil is

$$\bar{\xi} = 2\sum_{k=1}^{p+1} a_k \sin(k\xi), \qquad 0 \leq \xi \leq \xi_{max} \leq \pi. \tag{3}$$

Let us define the error function of this scheme as

$$E(\xi) \equiv \xi - \bar{\xi} = \xi - 2\sum_{k=1}^{p+1} a_k \sin(k\xi). \tag{4}$$

Our goal is to choose a_{p+1} such as to minimise the magnitude of any extrema of $E(\xi)$. In order to do so we will extend Lemma 1 to the perturbed stencil. For a detailed proof, see [5].

Lemma 2 *Let*

$$g_p(x) = 1 - 2\sum_{k=1}^{p+1} a_k k T_k(x)$$

where $a_k = c_k^{(p)} + c_k' a_{p+1}$ *are the coefficients of the perturbed central difference stencil as defined previously, and* $T_k(x)$ *is the* k^{th} *order Chebyshev polynomial. Then*

$$g_p(x) = (1-x)^p \left[(1-x)\frac{d_{p+1}}{c_{p+1}^{(p+1)}} a_{p+1} + d_p \left(1 - \frac{a_{p+1}}{c_{p+1}^{(p+1)}} \right) \right]$$

where d_p *and* $c_k^{(p)}$ *are defined as before.*

Corollary 1 $E(\xi)$ *has at most one extremum in the open interval* $(0, \xi_{max}]$ *and it is located at*

$$\xi = \xi_r = \arccos\left(1 - \frac{d_p}{d_{p+1}} \left[1 - \frac{c_{p+1}^{(p+1)}}{a_{p+1}} \right] \right).$$

For good approximations ξ_r *is a minimum. This occurs only when* a_{p+1} *and* $c_{p+1}^{(p+1)}$ *have the same sign and* $|a_{p+1}| \geq \left| c_{p+1}^{(p+1)} \right|$*, where equality holds only for classical stencils.*

Again, for the proof we refer to [5]. From the above corollary we conclude that we can have $|E(\xi)| = \|E\|_\infty$ only at two possible points, namely at ξ_r or ξ_{max}, i.e.

$$\|E\|_\infty = \max\{|E(\xi_{max})|, |E(\xi_r)|\}. \tag{5}$$

Of course $E(\xi_r)$ and $E(\xi_{max})$ depend on how we choose a_{p+1} and in view of (5) it is of interest to investigate this dependency. Our goal is to find the choice of a_{p+1} that minimises (5). In fact we have

Theorem 2 *Consider a* $2p+3$ *point central difference scheme of order* $\mathcal{O}(\Delta x^{2p})$ *with numerical wavenumber* $\bar{\xi}$ *defined as in (3), and a corresponding error function* $E(\xi) = \xi - \bar{\xi}$*. The stencil that uniformly minimises the error function, i.e.* $\min_{\bar{\xi}} \|E\|_\infty = \min_{a_{p+1} \in \mathbb{R}} \|\xi - \bar{\xi}\|_\infty$*, is uniquely characterised by the property*

$$E(\xi_r) + E(\xi_{max}) = 0. \tag{6}$$

The proof is found in [5] where it is also demonstrated how this result generalises to an arbitrary number of free parameters, $a_{p+1}, \ldots, a_{p+n}$.

3 A Numerical Example

For a given ξ_{max} solving (6) is a simple matter. Even for general ξ_{max} good estimates may be found by replacing $E(\xi_r)$ by a suitable polynomial approximation. For the case $p = 1$ this is shown in Fig. 1 for $\xi_{max} \in [0, \pi/2]$. Here $e = -\min_{\tilde{\xi}} \|E\|$.

To illustrate the strength of Theorem 2 we consider a profoundly polychromatic solution to the advection equation over a periodic domain:

$$u_t + u_x = 0, \quad 0 \le x \le 3, \quad t \ge 0$$
$$u(x, 0) = \exp(-3200(x - 1/2)^2).$$

This pulse is narrow and its Fourier transform is wide resulting in a significant contribution from a broad range of wavenumbers. The dominating wavenumber is $\kappa = 0$. Contributions from larger wavenumbers decay exponentially but slowly. It makes sense to use a scheme that accurately approximates the dispersion relation near $\xi = \kappa \Delta x = 0$ and for some suitably chosen region of larger wavenumbers.

From Fig. 1 we see that if we choose $\xi_{max} = \pi/4 \approx 0.785$ we will have an error in the dispersion relation of around 10^{-3}. Solving (6) gives the scheme

$$a_1 = 0.683345936919182, \qquad a_2 = -0.091672968459591.$$

In Fig. 2 the dispersion error of the new scheme is plotted as a function of ξ. The approximation starts to deviate from the exact result for $\xi \ge \xi_{max}$. The errors of a fourth order classical central difference scheme and of a five-point DRP scheme [1] are also included. As expected from Theorem 1 the classical stencil underestimates

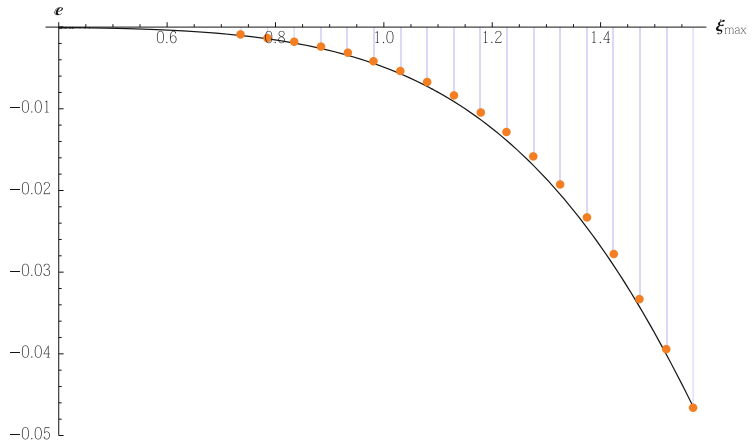


Fig. 1 Error of best uniform wavenumber approximation for given $\xi_{max} \in [0, \pi/2]$ (*orange dots*) and general solution using third degree polynomial approximation (*black line*)

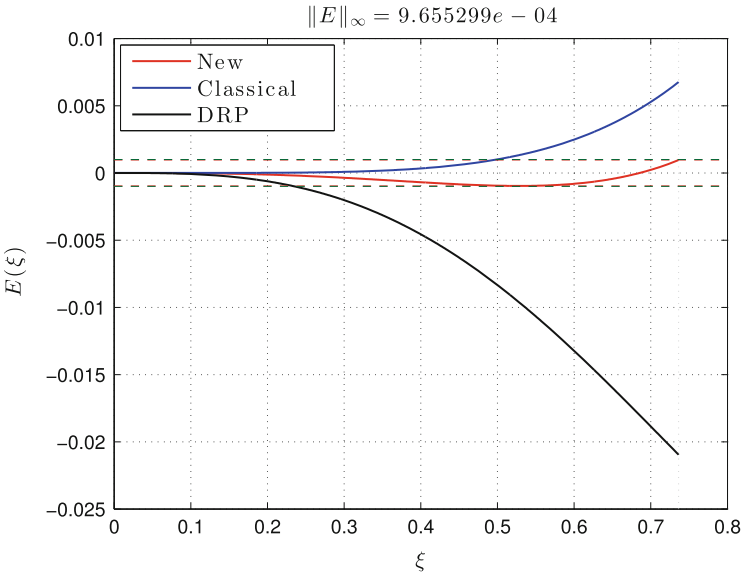


Fig. 2 Dispersion error for the new scheme, the classical 4th order stencil, and a five-point DRP scheme

the dispersion relation, seen by the positive sign of the error. For the shown range of ξ it seems that the DRP scheme overestimates the dispersion relation whereas our new scheme stays within tight error bounds.

We set $\Delta x = 1/120$ and integrate in time using the classical fourth order Runge-Kutta scheme with time step $\Delta t = 10^{-3}$ so the contribution from the temporal discretisation is small. The exact and numerical solutions are shown in Fig. 3 (top) together with the error as a function of time (bottom). All numerical solutions quickly disperse into a train of pulses of decaying amplitude trailing behind the main peak. As expected from Fig. 2, the DRP scheme overestimates the speed of some pulses. Our new scheme does this as well but to a much reduced extent. The smaller pulse train behind the DRP solution with respect to the new scheme may be attributed to a better approximation for very high wavenumbers. However, since the contribution of these wavenumbers are comparably small, the resulting error remains larger for the DRP scheme as compared with the classical and the new stencil.

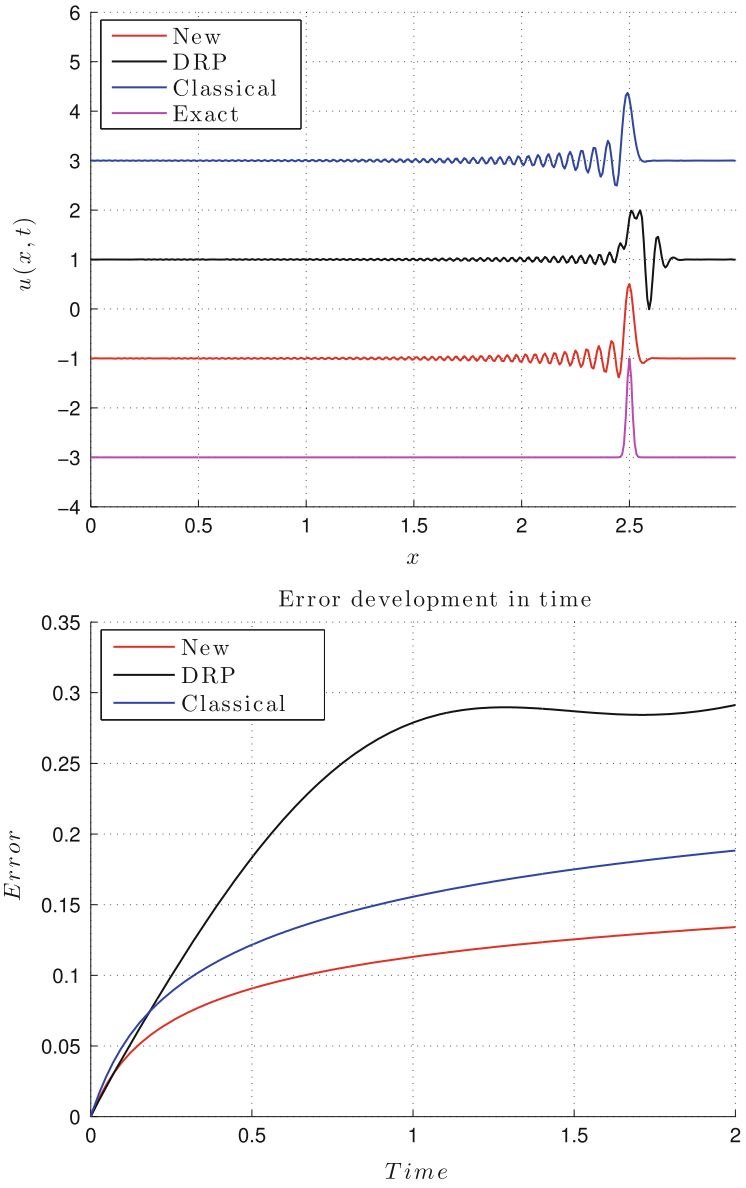


Fig. 3 (*Top*) Exact solution and numerical approximations after 2000 time steps. (*Bottom*) Corresponding errors

4 Extension to Multiple Dimensions

Extending the new stencil to multiple dimensions is in principle straight forward. As an example, for the advection problem in 2D we may, after discretising in space, write

$$\mathbf{v}_t + (D_x \otimes I_y)\mathbf{v} + (I_x \otimes D_y)\mathbf{v} = 0$$

where $\mathbf{v}$ is a grid vector approximating the true solution, $D_{x,y}$ are periodic operators containing the central difference stencil and operating on a given cartesian grid in the x and y direction respectively. Here $I_{x,y}$ are identity matrices of appropriate dimensions and $\otimes$ denotes the Kronecker product.

For this situation, the solution propagates at an angle θ with respect to the x-axis. It should be noted that the stencils presented here are optimal for the one-dimensional problem, that is for the cases when $\theta = n\pi/4$, $n = 0, 1, 2, 3$. For any other angle the stencils will be suboptimal since the numerical dispersion relation depends on the direction of propagation. In other words, these stencils may be sensitive to *numerical anisotropy*. For a comprehensive overview of methods that handles this issue, see e.g. [6]. At present we shall not consider this phenomenon further.

5 Conclusion

We have proved a characterisation theorem for best uniform wavenumber approximations by central difference stencils with one free parameter. The best approximation is unique and may be easily obtained numerically for a given range of wavenumbers. This allows for accurate approximations of problems of high frequency waves, or multi-frequency solutions, with a relatively coarse spatial mesh.

References

1. C.K.W. Tam, J.C. Web, Dispersion-relation-preserving finite difference schemes for computational acoustics. J. Comput. Phys. **107**, 262–281 (1993)
2. D.W. Zing, H. Lomax, H. Jurgens, High-accuracy finite-difference schemes for linear wave propagation. SIAM J. Sci. Compute. **17**, 328–346 (1996)
3. D.W. Zingg, H. Lomax, H. Jurgens, An optimized finite-difference scheme for wave propagation problems. AIAA paper 93(0459) (1993)

4. C. Bogey, C. Bailly, A family of low dispersive and low dissipative explicit schemes for flow and noise computations. J. Comput. Phys. **194**, 194–214 (2004)
5. V. Linders, J. Nordström, Uniformly best wavenumber approximations by spatial central difference operators. LiTH-MAT-R, 2014:17, Department of Mathematics, Linköping University, 2014
6. A. Sescu, Numerical anisotropy in finite differencing. Adv. Differ. Equ. **2015**(9) (2015). doi:10.1186/s13662-014-0343-0

Development of Unstructured Curved Meshes with G^1 Surface Continuity for High-Order Finite Element Simulations

Qiukai Lu and Mark S. Shephard

Abstract This paper presents a curved meshing technique for unstructured tetrahedral meshes where G^1 surface continuity is maintained for the triangular element faces representing the curved domain surfaces. A bottom-up curving approach is used to support geometric models with multiple surface patches where either C^0 or G^1 geometry continuity between patches is desired. Specific parametrization approaches based on Bézier forms and blending functions are used to define the mapping for curved element faces and volumes between parametric and physical coordinate systems. A preliminary result demonstrates that using G^1-continuity meshes can improve the solution results obtained.

1 Introduction

It is well known that high-order finite element methods are among the most powerful methods for simulating complex engineering problems [2]. In order to fully realize the benefits of the high-order methods, the mesh entities representing curved portions of the domain geometry must be curved and provide an high-enough order of geometry approximation [11, 12]. The ability to provide such a higher order of geometric approximation is facilitated by the use of greater than C^0 geometric shape continuity between elements [8, 14–16]. Although such higher order geometric continuity is being increasingly used with tensor product representations over quadrilaterals (see [6, 10]), there is also the desire to have higher than C^0 geometry continuity between elements on unstructured meshes where curved triangular finite element faces are used. The current work is intended to investigate and address the technical difficulties with developing curved meshing techniques for unstructured meshes where G^1 surface geometry continuity is maintained for the triangular element faces representing the curved domain surfaces. A preliminary result is also

Q. Lu (✉) • M.S. Shephard
Scientific Computation Research Center, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY, USA
e-mail: luq3@rpi.edu; shephard@rpi.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_30

included that shows improved solution results when G^1 surface triangulations are used.

2 Procedure to Create G^1 Curved Meshes

Many triangular patches have been developed in the CAGD community to construct G^1 continuous surface interpolations [8, 9, 14–16]. The techniques can in general be categorized into one of the two sets—polynomial based patches or rational blend based patches. The schemes using polynomials address the problem by either using single patch with relatively high polynomial degree or creating piecewise parametrization using sub-patches [13], both of which lead to more control points to be determined for the patch. The schemes using rationals are able to keep a patch complete by using blending functions [3]. Rational patches achieve G^1 with relatively low degree and require fewer control points. For the study in this paper, the rational blend based scheme is chosen because of its relatively straight-forward to construct and the data structure is similar to a regular Bézier triangle. The procedure to create G^1 curved meshes from C^0 straight-sided meshes using rational triangular patches is introduced in the following subsections. It is assumed that a straight-sided mesh is given with the set of boundary mesh entities correctly classified on a CAD model. Each mesh vertex on the model boundary is able to obtain its position and surface normal data by interacting with the CAD model.

2.1 Rational Triangular G^1 Patch

The essential part of the procedure to create G^1 curved meshes is the scheme to construct triangular G^1 patches for mesh faces that interpolate the position $x_i(\xi)$ and normal data n_j at their bounding vertices. The scheme used in this work to represent the curved geometry is based on an extension of the Gregory patch proposed by Walton et al. [16]. For each individual mesh face, each of the three bounding edges is assigned with a geometric representation of a cubic Bézier curve $B^{(3)}(\xi)$. Tangent vectors are obtained along the curve direction by taking the derivatives of the Bézier curve parametrization $t^{(2)} = \frac{\partial B}{\partial \xi}$. Cross-boundary tangent vectors are calculated by taking cross product with the surface normal given at mesh vertices $g^{(2)} = n_j \times t^{(2)}$. In order to obtain the required G^1 continuity, the cross-boundary tangent fields associated with the three mesh edges have to be satisfied simultaneously, thus requiring more degrees of freedom than a typical triangular Bézier patch. As a result, the order of the polynomials representing the surface patch is increased from cubic to at least quartic $B^{(4)}(\xi)$, which leads to a set of three surface control points. Each of the three surface control point is subsequently split into two and related together using linear blending functions. The rational blend degree-4 triangular Bézier patch

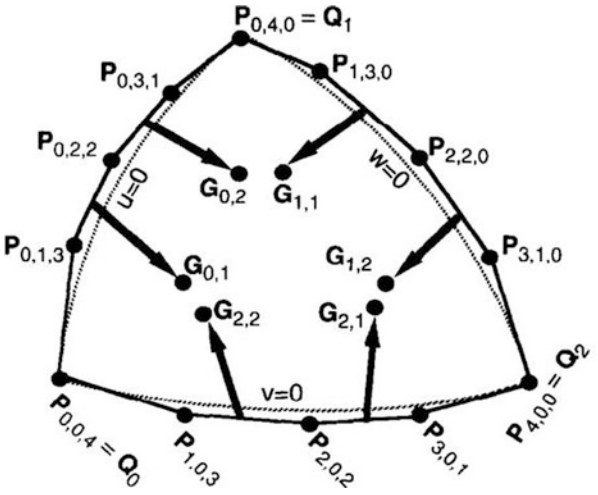


Fig. 1 Triangular Gregory patch and its control points

is defined by

$$B^{(4)}(\xi) = P_{ijk} b^4_{ijk}(\xi) \tag{1}$$

where P_{ijk} are the control points and $b^4_{ijk}(\xi)$ are 4*th* order Bernstein basis functions. The surface control points $P_{112}, P_{121}, P_{211}$ are affine combinations of the split surface control points $G_{i,j}$ and are calculated using $P_{1,1,2} = \frac{1}{\xi_1+\xi_2}(\xi_1 G_{2,2} + \xi_2 G_{0,1}), P_{1,2,1} = \frac{1}{\xi_3+\xi_1}(\xi_3 G_{0,2} + \xi_1 G_{1,1}), P_{2,1,1} = \frac{1}{\xi_2+\xi_3}(\xi_2 G_{1,2} + \xi_3 G_{2,1})$.

Figure 1 shows an example patch and its control point set.

2.2 Surface Mesh with Mixed C^0 and G^1 Continuity

The procedure introduced in Sect. 2.1 serves the purpose of creating G^1 surface meshes for models with a single model face. In the mean time, most 3D models with challenging geometric features consist of more than one model face. Any procedure aiming to create proper surface meshes for such multi-patch models has to account for the mixture of C^0 and G^1 continuity. In this work, a bottom-up approach is adopted based on the different topological types of model entities on which a mesh entity is classified. Specifically, the mesh edges that represent the model edges where model faces join with C^0-continuity are curved first to be G^1 along the model edge direction while maintaining C^0 in the cross-edge direction. After that, the remaining surface mesh entities that represent the rest of the model boundary are curved using the procedure discussed in Sect. 2.1. As a result, a piecewise G^1 surface mesh is created where it is G^1 within each model face as well as along the

```
for each edge M_i^1 in the mesh do
    if M_i^1 represents model edge with C^0 continuity then
        determine edge control points to interpolate model edge tangent;
    end
    if M_i^1 represents model face or edge with G^1 continuity then
        determine edge control points to interpolate model face normal;
    end
end
for each face M_i^2 in the mesh do
    if M_i^2 ⊏ G_j^2 then
        compute edge tangent vector t;
        compute cross-edge tangent vector g;
        determine face control points G_{i,j};
    end
end
```

Algorithm 1: Algorithm for creating G^1 meshes for multi-patch CAD models

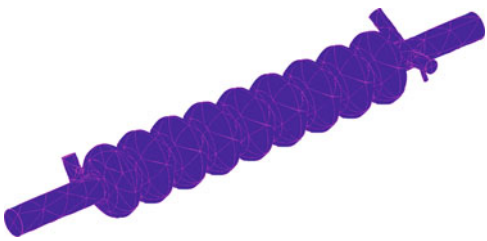


Fig. 2 Curved G^1 mesh of a linear accelerator model

bounding model edges and C^0 in the cross-boundary direction at the model edges where model faces join together. Note that in the case where two model faces join with G^1 continuity in the first place, G^1 continuity is maintained by curving the mesh edges representing the model edge in the same way as those representing model faces. The pseudo code for the overall procedure is given in Algorithm 1. Figure 2 shows an example mesh created using the algorithm.

3 Integration with Finite Element Analysis Solver

With the conventional isoparametric approach with C^0 meshes, the volumetric mapping between a standard parametric space and the physical space is constructed based on the same polynomial basis functions used for the finite element space. However, the basis functions used to represent the rational G^1 curved mesh are generally not the same as the finite element shape functions used for analysis. Therefore, a more general approach is adopted to construct the volumetric mapping in order to account for the G^1 surface geometry. The approach taken in this work is based on blending [7]. More specifically, the shapes of lower dimensional mesh

entities bounding the element volume are multiplied with linear blending functions, and the contributions are summed together to get the complete volume mapping. The equation to calculate the mapping is given in Eq. (2).

$$\begin{aligned} x_i(\xi_j) = &(1-\xi_1)E_1(\xi') + (1-\xi_2)E_2(\xi') + (1-\xi_3)E_3(\xi') + (1-\xi_4)E_4(\xi') \\ &-(1-\xi_1-\xi_2)F_1(\xi') - (1-\xi_1-\xi_3)F_2(\xi') - (1-\xi_1-\xi_4)F_3(\xi') \\ &-(1-\xi_2-\xi_3)F_4(\xi') - (1-\xi_2-\xi_4)F_5(\xi') - (1-\xi_3-\xi_4)F_6(\xi') \\ &+\xi_1 V_1(1,0,0,0) + \xi_2 V_2(0,1,0,0) + +\xi_3 V_3(0,0,1,0) + \xi_4 V_4(0,0,0,1) \end{aligned} \quad (2)$$

Here, $E_j, j = 1,2,3,4$ represent the four edge parametrization. Similarly, $F_j, j = 1,2,3,4,5,6$ represent face parametrization. V_j are the vertices. It is worth noting that the blending approach is independent of the chosen face and edge parametrization, therefore can be used with other types of parametric representations of mesh faces.

With the blending based volume parametrization, coordinate mapping can be easily evaluated. Derivatives quantities $\frac{\partial x_i}{\partial \xi_j}$ can also be evaluated by applying chain rule to Eq. (2) to obtain the analytic express of the derivatives of the blending mapping. With calculated derivatives, Jacobian of the mapping and its determinant can be easily evaluated.

4 Geometric Interpolation Accuracy

To study and quantify the geometric interpolation properties of the quartic G^1 patch discussed in Sect. 2.1, a set of numerical experiments have been conducted. A series of uniformly refined meshes are generated on a CAD model representing a cylinder. The distance between the mesh faces and CAD model faces is measured for each of the uniformly refined meshes. The distance is measured in terms of the Hausdorff norm which is commonly used to measure the distance between two parametric faces [1]. The definition of Hausdorff distance is given by Eq. (3).

$$d(S, S') = \max_{p \in S} \min_{p' \in S'} \left\| p - p' \right\|_2 \quad (3)$$

As a comparison, the measurement is done for both the G^1 meshes and a set of C^0 meshes using quartic Lagrange basis functions with optimal point distribution scheme proposed by Chen and Babuska [5]. Figure 3 shows the convergence plot generated from the distance data. For the quartic G^1 meshes, $4th$ order interpolation accuracy is observed, and for quartic C^0 meshes, it shows $5th$ order interpolation accuracy. It is a well known result in 1D that the order of accuracy for polynomial interpolation is $p + 1$, where p is the highest complete polynomial order [8]. The one order difference in interpolation accuracy between the G^1 and C^0 is due to the

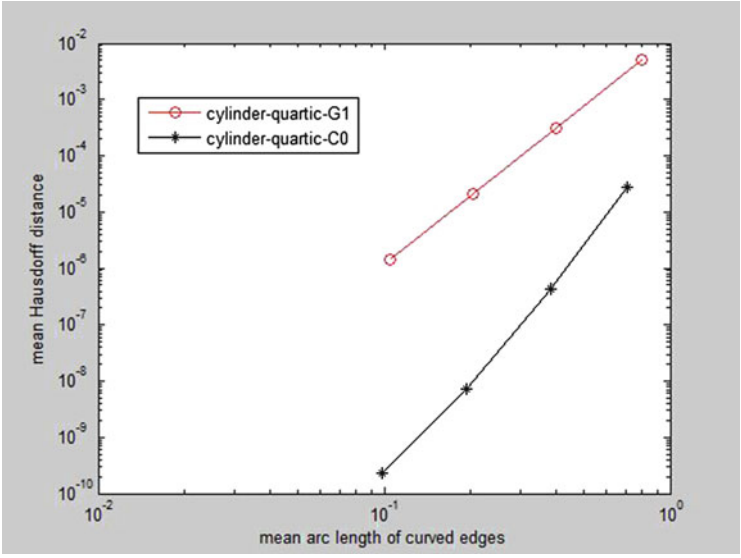


Fig. 3 Convergence of geometric approximation error

fact that certain portion of the control points of the G^1 patch have to be constrained to ensure the higher surface continuity.

5 Impact on Finite Element Solution Accuracy

The primary interest for using G^1 meshes is to see if they produce better finite element simulation results. The test problem chosen is the Poiseuille flow, which models viscous flow inside a pipe of constant circular cross-section. Governing Equation for the Poiseuille flow is defined as:

$$\frac{1}{r}\frac{\partial}{\partial r}(r\frac{\partial u_z}{\partial r}) = \frac{1}{\mu}\frac{\partial p}{\partial z} \tag{4}$$

The fully developed flow is assumed to be incompressible, steady, laminar and has a closed form exact solution which indicates a velocity profile of a parabola. The analytic expression is given as:

$$u_z = -\frac{1}{4\mu}\frac{\partial p}{\partial z}(R^2 - r^2) \tag{5}$$

In the numerical test, it is of interest to solve for the fully developed velocity profile and compare it with the exact solution. A CAD model is constructed to represent the flow domain of a cylinder with radius $r = 0.5$. No-slip condition is set for the wall. At inlet, the velocity profile is set to be fully developed: $u_z = 0.25 - r^2$. The pressure at the outlet is set to be constant.

The finite element solver package being used to perform the analysis is *Nektar++* [4], which is a spectral/hp element framework being developed by research groups at the University of Utah and Imperial College London. It has a set of flow solvers that use high-order finite element methods. Specific modifications are made to *Nektar++* to account for the G^1 mesh construction including elemental mapping evaluation, derivatives and Jacobian calculation procedures. Two types of meshes with the same number of elements, the same order of polynomial degree, but different order of geometric continuity are used, namely, quartic C^0 curved and quartic G^1 curved meshes (See Fig. 4). A series of simulations are performed with each type of the meshes using 4th and 5th order Legendre polynomial shape functions. The error of finite element solution of the velocity field against the exact analytic solution is measured in terms of the L_2 norm and is shown in Table 1. It is observed in this test case that meshes with G^1 surface continuity achieve better solution accuracy compared with C^0 meshes for the same order of shape functions.

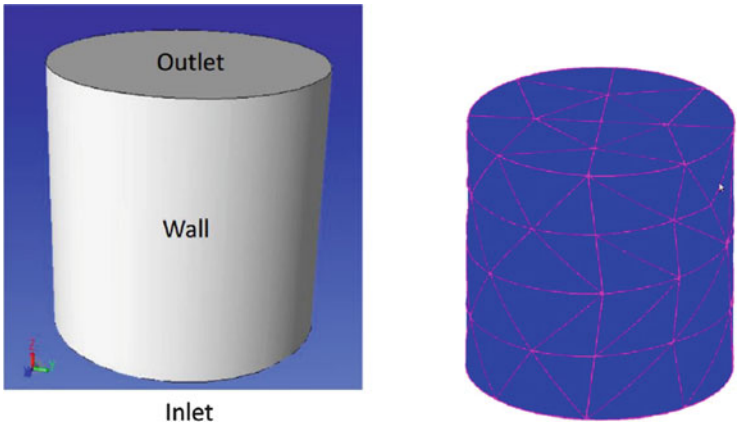


Fig. 4 CAD model and quartic G^1 mesh

Table 1 Finite element solution error for different types of curved meshes

Shape func order	Quartic C^0	Quartic G^1
4	1.29207e−3	4.33327e−4
5	5.98625e−4	9.67477e−5

6 Closing Remarks

This paper presented a procedure to create G^1 curved surface meshes for high-order finite element simulations. A method to create G^1-continuous surface patches is introduced and an approach to integrate a G^1 mesh with existing finite element solver is presented. A preliminary test result shows the advantage of using G^1 continuous meshes, compared with conventional C^0 meshes, in terms of finite element solution accuracy of a standard integral norm. Additional studies to examine the influence of G^1 continuity on more problems and for other solution norms must be carried out. There is particular interest to examine solution parameters more local to the surface. For future developments, it is of interest to study other types of high-order surface patches. Furthermore, the capability of using the CAD model surface parametrization to define exact geometric mapping is to be developed. In order to support adaptive simulations, extensions of existing mesh modification operations and mesh adaptation procedure [11] will be needed to account for high-order curved meshes.

References

1. N. Aspert, D. Santa-Cruz, T. Ebrahimi, Mesh: measuring errors between surfaces using the Hausdorff distance, in *IEEE International Conference on Multimedia and Expo*, vol. 1, pp. 705–708 (2002)
2. I. Babuska, B.A. Szabo, I.N. Katz, The p-version of the finite element method. SIAM J. Numer. Anal. **18**(3), 515–545 (1981)
3. M. Boschiroli, C. Funfzig, L. Romani, G. Albrecht, G1 rational blend interpolatory schemes: a comparative study. Graph. Model. **74**(1), 29–49 (2012)
4. C.D. Cantwell, S. Yakovlev, R.M. Kirby, N.S. Peters, S.J. Sherwin, High-order spectral/hp element discretisation for reaction-diffusion problems on surfaces: application to cardiac electrophysiology. J. Comput. Phys. **257, Part A**(0), 813–829 (2014)
5. Q. Chen, I. Babuška, Approximate optimal points for polynomial interpolation of real functions in an interval and in a triangle. Comput. Methods Appl. Mech. Eng. **128**(3), 405–417 (1995)
6. L. Demkowicz, P. Gatto, W. Qiu, A. Joplin, G1-interpolation and geometry reconstruction for higher order finite elements. Comput. Methods Appl. Mech. Eng. **198**(13), 1198–1212 (2009)
7. S. Dey, M.S. Shephard, J.E. Flaherty, Geometry representation issues associated with p-version finite element computations. Comput. Methods Appl. Mech. Eng. **150**(1–4), 39–55 (1997). Symposium on Advances in Computational Mechanics
8. G.E. Farin, Triangular bernstein-bezier patches. Comput. Aided Geom. Des. **3**, 83–127 (1986)
9. S. Hahmann, G.P. Bonneau, Triangular g1 interpolation by 4-splitting domain triangles. Comput. Aided Geom. Des. **17**(8), 731–757 (2000)
10. T.J.R. Hughes, J. Cottrell, Y. Bazilevs, Isogeometric analysis: cad, finite elements, nurbs, exact geometry and mesh refinement. Comput. Methods Appl. Mech. Eng. **194**(39–41), 4135–4195 (2005)
11. Q. Lu, M. Shephard, S. Tendulkar, M. Beall, Parallel mesh adaptation for high-order finite element methods with curved element geometry. Eng. Comput. **30**(2), 271–286 (2014)
12. X. Luo, M.S. Shephard, J.F. Remacle, R.M. O'bara, M.W. Beall, B. Szabo, R. Actis, p-version mesh generation issues, in *Proceedings of the 11th Meshing Roundtable, Ithaca, NY*, pp. 343–354, Ithaca, NY (2002)

13. S. Mann, C. Loop, M. Lounsbery, D. Meyers, J. Painter, T. DeRose, K. Sloan, A survey of parametric scattered data fitting using triangular interpolants, Curve and Surface Design, vol. 29, pp. 145–172 (1992)
14. J. Peters, Biquartic c1-surface splines over irregular meshes. Comput. Aided Des. **27**(12), 895–903 (1995)
15. B.R. Piper, Visually smooth interpolation with triangular bezier patches, in *Geometric Modelling: Algorithms and New Trends*, ed. by G. Farin (SIAM, Philadelphia, 1987), pp. 221–234
16. D. Walton, D. Meek, A triangular g1 patch from boundary curves. Comput. Aided Des. **28**(2), 113–123 (1996)

Efficient Fully Discrete Summation-by-Parts Schemes for Unsteady Flow Problems: An Initial Investigation

Tomas Lundquist and Jan Nordström

Abstract We make an initial investigation into the temporal efficiency of a fully discrete summation-by-parts approach for stiff unsteady flows with boundary layers. As a model problem for the Navier–Stokes equations we consider a two-dimensional advection-diffusion problem with a boundary layer. The problem is discretized in space using finite difference approximations on summation-by-parts form together with weak boundary conditions, leading to optimal stability estimates. For the time integration part we consider various forms of high order summation-by-parts operators, and compare the results to an existing popular fourth order diagonally implicit Runge-Kutta method. To solve the resulting fully discrete equation system, we employ a multi-grid scheme with dual time stepping.

1 Introduction

Based on finite difference operators on summation-by-parts (SBP) form and the simultaneous-approximation-term (SAT) technique for imposing boundary conditions, the SBP-SAT technique constitutes a robust framework for implementing high order finite difference schemes on complex geometries. By construction, it leads to discrete energy estimates that perfectly imitates the corresponding continuous estimates. This technique was recently extended to initial value problems [10, 12], making it possible to formulate fully discrete SBP-SAT approximations with the same optimal energy estimates. The purpose of this work is to make an initial efficiency study of these new temporal schemes for a stiff model problem with a boundary layer. A more detailed description of this study can be found in [9].

The numerical treatment of unsteady flow problems has gained increased attention in later years due to increased computer resources making realistic calculations of this type more viable. However, the construction of efficient algorithms still remains a significant computational challenge. The two basic methods most

T. Lundquist (✉) • J. Nordström
Department of Mathematics, Computational Mathematics, Linköping University, SE-581 83 Linköping, Sweden
e-mail: tomas.lundquist@liu.se; jan.nordstrom@liu.se

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_31

commonly used employ Newton iteration and dual time stepping. While Newton iterations are typically better for deep convergence, dual time stepping is more reliable, at least for the initial iterations, and it can also be used for preconditioning purposes [8]. Some studies indicate that a combination of both these techniques can be the most fruitful approach [1, 2, 8].

To illustrate the SBP-SAT technique for time integration, we consider the test equation $u_t + \lambda u = 0$ with initial condition $u(0) = f$. The corresponding SBP-SAT approximation of this problem is

$$D\mathbf{U} + \lambda\mathbf{U} = P^{-1}\sigma(U_0 - f)\mathbf{e_0}. \tag{1}$$

The SAT penalty treatment on the right hand side of (1) forces the solution at $t = 0$ to initial data, and the first derivative operator D satisfies the SBP property given by the decomposition $D = P^{-1}Q$, where $Q + Q^T = \text{Diag}(-1, 0, \ldots, 0, 1)$, and P is a positive definite matrix that defines a numerical quadrature. This formulation leads in an automatic way to a clean, optimally sharp energy estimate. With the choice $\sigma = -1$, we get after multiplying (1) with $\mathbf{u}^*P$ and adding the conjugate transpose:

$$|u_N|^2 + 2Re(\lambda)||\mathbf{u}||_P^2 = |f|^2 - |u_0 - f|^2,$$

where the norm is defined as $||\mathbf{u}||_P^2 = u^*Pu$. This mimics the continuous energy estimate $|u(T)|^2 + 2Re(\lambda)\|u\|^2\mathrm{d}t = |f|^2$, where $\|u\|^2 = \int_0^T |u|^2\mathrm{d}t$ (obtained by multiplying the test equation with u^* and then integrating).

As an alternative to the global formulation (1), we may also consider a multi-stage version with $r + 1$ stages:

$$\begin{aligned}(P^{-1}Q + \lambda I)\mathbf{V}^{n+1} &= P^{-1}\sigma(V_0^{n+1} - U^n)\mathbf{e_0}\\ U^{n+1} &= V_r^{n+1},\end{aligned}$$

where $\mathbf{V}^{n+1} = (V_0^{n+1}, V_1^{n+1}, \ldots, V_r^{n+1})$. The size of the matrix operator $P^{-1}Q$ in the multi-stage formulation is given by the number of intermediate stages $r + 1$ used for each subinterval, and thus remains constant also for long time calculations. Conversely, the minimum number of stages stages depends on how small $P^{-1}Q$ can be made. The classical SBP operators are based on a repeated central finite difference stencil together with boundary closures. An example is the second order operator given by

$$P = \Delta t \begin{bmatrix} \frac{1}{2} & & \\ & 1 & \\ & & \ddots \end{bmatrix} \quad Q = \begin{bmatrix} -\frac{1}{2} & \frac{1}{2} & & \\ -\frac{1}{2} & 0 & \frac{1}{2} & \\ & \ddots & \ddots & \ddots \end{bmatrix}.$$

Higher order operators have more extensive boundary closures, thus increasing the minimum number of stages required in the multi-stage approach. Other operators on SBP form, e.g. based on Legendre spectral collocation [5], may alternatively be

used to decrease the number of stages necessary, as demonstrated in [3, 4]. See also [6] for more details on the construction non-classical SBP operators based on any type of quadrature.

We summarize the most important advantages of the SBP-SAT technique below.

- The schemes are always A-stable and L-stable. If P is diagonal they are also B-stable and preserve energy stability. Moreover, they lead to optimally sharp fully discrete energy estimates.
- The order of convergence is given by the order of accuracy of the quadrature P. For classical SBP operators this coincides with the order of the interior scheme.
- The stage order, and thus the order of stiff convergence, is given by the local order of consistency of the operator $P^{-1}Q$. Classical operators are thus limited by the accuracy of the boundary closures.

2 A Stiff Flow Model in Two Dimensions

As a model of the Navier–Stokes equation, we study a viscous fluid undergoing advective flow past a plate with fixed temperature.

$$\begin{aligned} u_t + u_x &= \varepsilon(u_{xx} + u_{yy}) + \psi && 0 \le x, y \le 1 \quad t \ge 0 \\ u(0, x, y) &= f(x, y) && t = 0 \\ u(t, 0, y) - \varepsilon u_x(t, 0, y) &= g_1(t, y) && \partial\Omega_1 = \{(x, y) : x = 0\} \\ \varepsilon u(t, 1, y) &= g_2(t, y) && \partial\Omega_2 = \{(x, y) : x = 1\} \\ u(t, x, 0) &= 0 && \partial\Omega_3 = \{(x, y) : y = 0\} \\ u_y(t, x, 1) &= 0 && \partial\Omega_4 = \{(x, y) : y = 1\} \end{aligned} \tag{2}$$

where $\varepsilon = 0.01$. The solid boundary $\partial\Omega_3$ is associated with a stiff boundary layer of width $\sqrt{\varepsilon}$, the inflow and outflow boundaries are $\partial\Omega_1$ and $\partial\Omega_2$ respectively, while $\partial\Omega_4$ is a far-field boundary. An exact manufactured solution can be imposed by appropriately specifying the forcing function ψ. The energy method yields the estimate

$$\|u\|_t^2 + 2\varepsilon(\|u_x\|^2 + \|u_y\|^2) = \int_{\partial\Omega_1} (g_1^2 - (u - g_1)^2) \mathrm{d}S + \int_{\partial\Omega_2} (g_2^2 - (u - g_2)^2) \mathrm{d}S.$$

which shows that the problem (2) is well-posed.

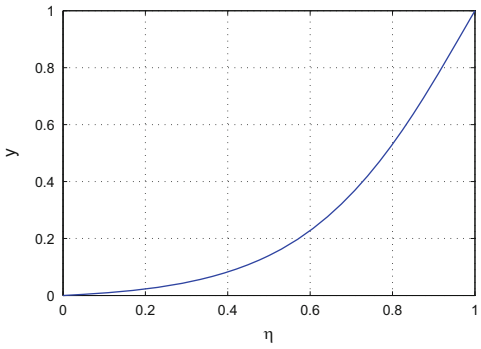


Fig. 1 Stretching of vertical coordinate to resolve the boundary layer

In order to resolve the boundary layer around $\partial\Omega_3$, we introduce a stretching function η of the vertical coordinate, given by

$$y = 1 + \frac{tanh(B(\eta - 1))}{tanhB},$$

where $B = 9/4$. This gives $y_\eta(0) = \sqrt{\varepsilon}$, and the full stretching function is shown in Fig. 1. After this change of coordinate, the model problem (2) becomes

$$\begin{aligned}
u_t + u_x &= \varepsilon(u_{xx} + \eta_y(\eta_y u_\eta)_\eta) + \psi && 0 \le x, \eta \le 1 \quad t \ge 0 \\
u(0, x, \eta) &= f(x, \eta) && t = 0 \\
u(t, 0, \eta) - \varepsilon u_x(t, 0, \eta) &= g_1(t, \eta) && \partial\Omega_1 = \{(x, \eta) : x = 0\} \\
\varepsilon u(t, 1, \eta) &= g_2(t, \eta) && \partial\Omega_2 = \{(x, \eta) : x = 1\} \\
u(t, x, 0) &= 0 && \partial\Omega_3 = \{(x, \eta) : \eta = 0\} \\
\eta_y u_\eta(t, x, 1) &= 0 && \partial\Omega_4 = \{(x, \eta) : \eta = 1\}
\end{aligned} \tag{3}$$

We now use the techniques outlined in [11, 13] to discretize (3) in space using SBP-SAT:

$$\begin{aligned}
U_t + (P_x^{-1} Q_x \otimes I_\eta)U &= \epsilon(((P_x^{-1} Q_x)^2 \otimes I_\eta)U + (I_x \otimes (H_y P_\eta^{-1} Q_\eta)^2)U) \\
&\quad + (P_x^{-1} \otimes P_\eta^{-1} H_y)(\Sigma_x(t) + \Sigma_\eta(t)) + \Psi(t) \\
U(0) &= F,
\end{aligned} \tag{4}$$

where

$$
\begin{aligned}
H_y &= \mathrm{Diag}(\boldsymbol{\eta}_y), \quad \Psi(t) = \psi(t, (\mathbf{x} \otimes \mathbf{1}_y), (\mathbf{1}_x \otimes \boldsymbol{\eta})) \\
\Sigma_x(t) &= \sigma_{0x}(\mathbf{e}_{0x} \otimes H_y^{-1} P_\eta(\mathbf{u}|_{x=0} - \epsilon \mathbf{u}_x|_{x=0} - \mathbf{g}_1(t))) \\
&\quad + \sigma_{1x}(\mathbf{e}_{1x} \otimes H_y^{-1} P_\eta(\epsilon \mathbf{u}_x|_{x=1} - \mathbf{g}_2(t))) \\
\Sigma_\eta(t) &= \sigma_{0\eta}(P_x \mathbf{u}|_{\eta=0} \otimes \mathbf{e}_{1\eta}) + \sigma_{1\eta}(P_x \mathbf{u}_\eta|_{\eta=1} \otimes \mathbf{e}_{1\eta}) \\
\mathbf{g}_1(t) &= g_1(t, (\mathbf{e}_{x0} \otimes \boldsymbol{\eta})), \quad \mathbf{g}_2(t) = g_2(t, (\mathbf{e}_{x1} \otimes \boldsymbol{\eta})), \quad F = f((\mathbf{x} \otimes \mathbf{1}_\eta), (\mathbf{1}_x \otimes \boldsymbol{\eta}).
\end{aligned}
$$

After analyzing (4) using the energy method, we obtain a stable scheme with the following set of penalty parameters: $\sigma_{0x} = \sigma_{1x} = -1$, $\sigma_{1\eta} = -1/2$ and $\sigma_{0\eta} = -\varepsilon \eta_y(0)^2/(P_\eta)_{11}$.

3 SBP-SAT in Time with Dual Time Stepping

We now consider the semi-discrete problem (4) written in a compact way as

$$
\begin{aligned}
U_t + BU &= R(t),\ 0 < t \leq T \\
U(0) &= F.
\end{aligned}
$$

The semi-discrete spectrum of a fifth order discretization with $N_x = N_\eta = 95$ is shown in Fig. 2. The spectral radius of almost 10^5 indicates that an explicit time marching scheme would be an inefficient way to solve this problem. Instead we

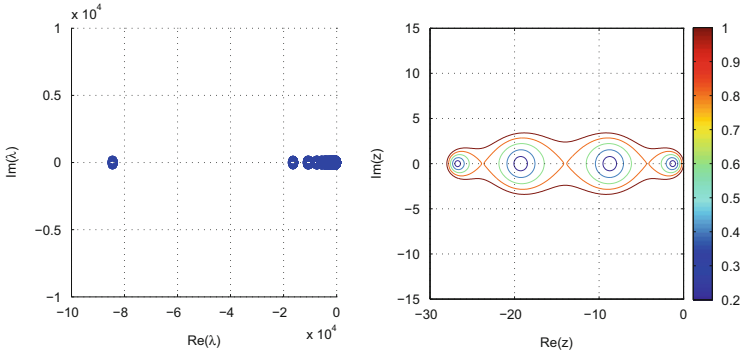


Fig. 2 *Left*: the semi-discrete spectrum of (4). *Right*: Stability region of Runge–Kutta smoother

employ an implicit SBP-SAT time integration scheme with $r+1$ stages:

$$\begin{aligned}(P^{-1}Q \otimes I_B)\mathbf{V}^{n+1} + (I_t \otimes B)\mathbf{V}^{n+1} &= (P^{-1}\sigma \mathbf{e}_0) \otimes (V_0^{n+1} - U^n) + R \\ U^{n+1} &= V_r^{n+1},\end{aligned} \tag{5}$$

where $R = (R(t^n), R(t^n + \Delta t/r), \ldots, R(t^n + \Delta t))$. Consider the compact form of (5):

$$\begin{aligned}\tilde{B}\mathbf{V}^{n+1} &= \tilde{R} \\ U^{n+1} &= V_r^{n+1},\end{aligned} \tag{6}$$

where $\tilde{B} = P^{-1}(Q - \sigma e_0 e_0^T) \otimes I_B + I_t \otimes B$. Using the dual time stepping technique, we now employ a multi-grid cycle for solving (6), where the smoothing step consists of stepping forward in pseudo-time toward steady-state. Thus, we add a pseudo time derivative to (6):

$$\frac{\mathrm{d}\mathbf{V}^{n+1}}{\mathrm{d}\tau} + \tilde{B}\mathbf{V}^{n+1} = \tilde{R}.$$

To march forward in pseudo-time, we use an explicit s-stage low storage Runge-Kutta smoother:

$$\begin{aligned}\mathbf{W}_0^{n+1,m+1} &= \mathbf{V}^{n+1,m} \\ \mathbf{W}_p^{n+1,m+1} &= \mathbf{V}^{n+1,m} + \Delta\tau\alpha_p(\tilde{R} - \tilde{B}\mathbf{W}_{p-1}^{n+1,m+1}),\ p = 1, \ldots, s \\ \mathbf{V}^{n+1,m+1} &= \mathbf{W}_s^{n+1,m+1}\end{aligned}$$

The stability function of this scheme is $S(z) = (1 + \alpha_s z(1 + \alpha_{s-1} z(\ldots(1 + \alpha_1 z)\ldots)))$. To match the semi-discrete spectrum to the left in Fig. 2, we use the 4-stage smoother $\alpha = (0.0178571, 0.0568106, 0.174513, 1)$ proposed in [7]. The stability region of this scheme is shown to the right in Fig. 2.

4 Numerical Results

We employ the manufactured solution $u = sin(2\pi(x-t))e^{\frac{1-y}{\sqrt{\varepsilon}}}$ to (2), and compare the numerical results for a selection of high order temporal schemes. We use both classical diagonal norm operators, denoted SBP($2s$,s), as well as spectral element operators based on Gauss-Lobatto quadrature, denoted by GL($2s$,s). In both cases, $2s$ denotes the order of the scheme, and s the stage order. For the classical operators we always use the minimum number of stages possible. We use the following operators of order four and eight: SBP(4,2) with 8 implicit stages, SBP(8,4) with 16 implicit stages, GL(4,2) with 3 implicit stages, and GL(8,4) with 5 implicit stages. For comparison we also consider a fourth order diagonally implicit Runge-Kutta scheme ESDIRK4, with a stage order of 2, and 5 implicit stages.

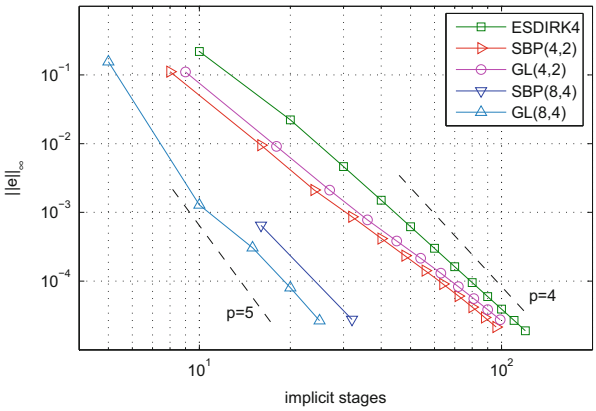


Fig. 3 Accuracy of the high order temporal schemes at $t = 1$

In order to minimize the spatial error component we use the fifth order discretization with $N_x = N_\eta = 95$ with spectrum shown in Fig. 2. Note that the spectral radius of almost 10^5 is a result both of the boundary layer and of overresolving in space. The time integration is carried out using a multi-grid V-cycle on three grid levels, with refinement in the vertical coordinate only. On each grid, 10 steps of the explicit Runge-Kutta scheme is used as smoother, with a pseudo-time step restriction of $Re(z) = -25$ to make the Runge–Kutta scheme stable on each respective grid, see Fig. 2. The number of pseudo time iterations is set to make the iteration error less than 10 % compared with the error from the physical time discretization.

In Fig. 3 we measure the accuracy at $t = 1$ of the different temporal schemes as a function of the total number of implicit stages. In all cases we observe a small level of order reduction, with convergence rates slightly less than the order of the scheme (but higher than the stage order). The number of multi-grid iterations required to converge each implicit stage on average is shown in Fig. 4. Figure 5 finally shows the total efficiency, where work is defined as the total number of multi-grid cycles summed over all implicit stages. With this measure the results are comparable between methods using different numbers of implicit stages in each implicit solve. We note that there is no advantage of the diagonally implicit ESDIRK4 method over the Gauss-Lobatto SBP schemes. The classical SBP operators on the other hand, using more implicit stages in each implicit solve, are clearly less efficient here. Current work however indicate that this drawback might be possible to correct by modifying the multi-grid scheme in an appropriate way. Finally, we note that the increased accuracy of the eighth order schemes is counterbalanced by the reduced multi-grid efficiency due to the coarser discretizations in physical time, resulting in very similar results as for the fourth order schemes.

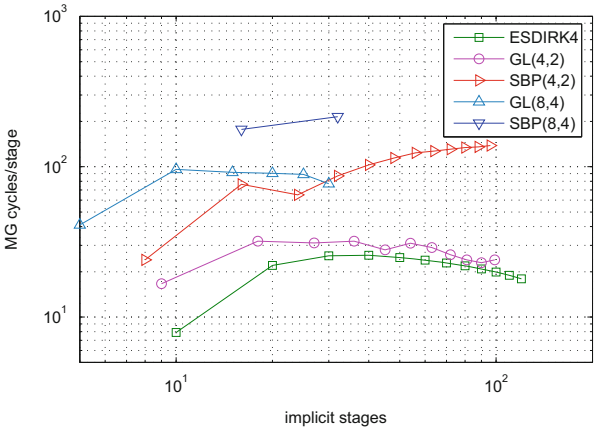


Fig. 4 The amount of work required to resolve each implicit stage

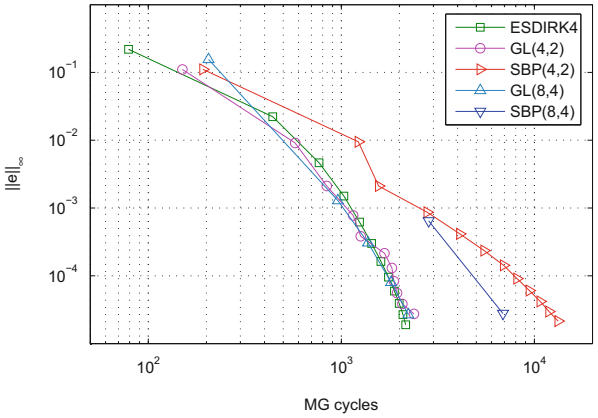


Fig. 5 Accuracy versus the total amount of work required to solve up to $t = 1$

5 Conclusions and Further Work

We have investigated the temporal efficiency of fully discrete SBP-SAT discretizations for unsteady flow calculations. A stiff linear model problem was considered, and a basic dual time-stepping scheme was employed with no attempt made at optimizing the smoother. The numerical results indicate that some of the SBP-SAT time stepping schemes can compete with ESDIRK4 for efficiency already in this basic setting, even though the classical SBP schemes using a larger number of stages did not perform as well. Current work indicate that this disadvantage can be overcome with a more suitable choice of multi-grid scheme that works more efficiently for fully implicit methods.

Future work will aim at developing more efficient multi-grid schemes, including optimization of the Runga Kutta smoother used for pseudo-time stepping. Non-linear model problems will also be considered, as well as combining the dual time stepping technique with Newton iteration.

References

1. H. Bijl, M. Carpenter, Iterative solution techniques for unsteady flow computations using high order time integration schemes. Int. J. Numer. Meth. Fluids **47**, 857–862 (2005)
2. P. Birken, A. Jameson, On nonlinear preconditioners in Newton-Krylov methods for unsteady flows. J. Comput. Phys. **62**, 565–573 (2010)
3. P. Boom, D. Zingg, Runge-Kutta characterization of the generalized summation-by-parts approach in time. arXiv:1410.0202 (2014)
4. P. Boom, D. Zingg, High-order implicit time-marching methods based on generalized summation-by-parts operators. arXiv:1410.0201 (2014)
5. M. Carpenter, D. Gottlieb, Spectral methods on arbitrary grids. J. Comput. Phys. **129**, 74–86 (1996)
6. D. Del Rey Fernandez, P. Boom, D. Zingg, A generalized framework for nodal first derivative summation-by-parts operators. J. Comput. Phys. **266**, 214–239 (2014)
7. W.L. Kleb, Efficient multi-stage time marching for viscous flows via local preconditioning. AIAA J. **99**, 181–194 (1999)
8. D. Knoll, D. Keyes, Jacobian free Newton-Krylov methods: a survey of approaches and applications. J. Comput. Phys. **193**, 357–397 (2004)
9. T. Lundquist, J. Nordström, Efficient fully discrete summation-by-parts schemes for unsteady flow problems. LiTH- MAT-R, 2014:18, Department of Mathematics, Linköping University, 2014
10. T. Lundquist, J. Nordström, The SBP-SAT technique for initial value problems. J. Comput. Phys. **270**, 86–104 (2014)
11. J. Nordström, M. Carpenter, High order finite difference methods, multidimensional linear problems and curvilinear coordinates. J. Comput. Phys. **173**, 149–174 (2001)
12. J. Nordström, T. Lundquist, Summation-by-parts in time. J. Comput. Phys. **251**, 487–499 (2013)
13. M. Svärd, J. Nordström, A stable high-order finite difference scheme for the compressible Navier-Stokes equations No-slip wall boundary conditions. J. Comput. Phys. **227**, 4805–4824 (2008)

Physics-Based Stabilization of Spectral Elements for the 3D Euler Equations of Moist Atmospheric Convection

Simone Marras, Andreas Müller, and Francis X. Giraldo

Abstract In the context of stabilization of high order spectral elements, we introduce a dissipative scheme based on the solution of the compressible Euler equations that are regularized through the addition of a residual-based stress tensor. Because this stress tensor is proportional to the residual of the unperturbed equations, its effect is close to none where the solution is sufficiently smooth, whereas it increases elsewhere. This paper represents a first extension of the work by Nazarov and Hoffman (Int J Numer Methods Fluids 71:339–357, 2013) to high-order spectral elements in the context of low Mach number atmospheric dynamics. The simulations show that the method is reliable and robust for problems with important stratification and thermal processes such as the case of moist convection. The results are partially compared against a Smagorinsky solution. With this work we mean to make a step forward in the implementation of a stabilized, high order, spectral element large eddy simulation (LES) model within the Nonhydrostatic Unified Model of the Atmosphere, NUMA.

1 Introduction

Recently [18], a numerically stable and computationally inexpensive large-eddy simulation (LES) model for compressible flows was designed for adaptive finite elements. It is a close relative of the entropy-viscosity method by Guermond and co-workers (see, e.g. [7]), although no entropy equation is used to construct the dynamic viscosity coefficient of the stress tensor.

In the current paper, we explore the capabilities of the aforementioned LES model to act as a stabilization method for the spectral element solution of the Euler equations at the low Mach number regimes typical of atmospheric flows.

S. Marras (✉) • A. Müller • F.X. Giraldo
Department of Applied Mathematics, Naval Postgraduate School, 833 Dyer Rd., Monterey, CA, USA
e-mail: smarras1@nps.edu; amueller@nps.edu; fxgirald@nps.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_32

This effort is justified by the fact that, within the community of atmospheric modelers, there is still a widespread concern about the most proper stabilization scheme to be used with either Galerkin or other approximation methods of the equations of atmospheric dynamics. Although the use of residual-based stabilizing schemes has been largely assessed for the finite element method during the past thirty years (e.g. Streamline-Upwind/Petrov-Galerkin (SUPG) [3], Galerkin/Least-Squares (GLS) [10], Variational Multiscale (VMS) [2, 8, 9, 15]), hyper viscosity is still today the most classical approach in spite of its important drawbacks and mathematical inconsistency.

This work is a first step toward the implementation of a stabilized high order spectral element LES model (LES-SEM) for the *Nonhydrostatic Unified Model of the Atmosphere* (NUMA) developed by the authors [6, 11]. The rest of the paper is organized as follows. The set of equations and the LES model are described in Sect. 2. Some basics on the space and time discretization of these equations is reported in Sect. 3, which is followed by the numerical tests and results in Sect. 4. Some conclusions are given in Sect. 5.

2 Equations for Wet Dynamics

Let $\Omega \in \mathbb{R}^3$ be a fixed three dimensional domain with boundary $\partial\Omega$ and Cartesian coordinates $\mathbf{x} = (x, y, z)$. Let us identify the dry air density, the velocity vector, and the potential temperature with the symbols $\rho, \mathbf{u}$, and θ. Let us also define the mixing ratios of water vapor, cloud water, and rain as $q_v = \rho_v/\rho, q_c = \rho_c/\rho$ and $q_r = \rho_r/\rho$, where $\rho_{v,c,r}$ are the densities of vapor, cloud, and rain. Furthermore, let $\rho'(t, \mathbf{x}) = \rho(t, \mathbf{x}) - \rho_0(z)$, $\theta'(t, \mathbf{x}) = \theta(t, \mathbf{x}) - \theta_0(z)$, and $p'(t, \mathbf{x}) = p(t, \mathbf{x}) - p_0(z)$ be the perturbations of density, potential temperature, and pressure with respect to a hydrostatically balanced background state indicated by the subscript *0*. Then, the strong form of the time-dependent Euler equations with gravity, g, can be written as:

$$\begin{aligned}
\rho'_t + \mathbf{u} \cdot \nabla\rho + \rho\nabla \cdot \mathbf{u} &= 0, \\
\mathbf{u}_t + \mathbf{u} \cdot \nabla\mathbf{u} + \tfrac{1}{\rho}\nabla \cdot (\mathbf{I}p') &= g(1 + \epsilon q_v - q_c - q_r)\mathbf{k}, \\
\theta'_t + \mathbf{u} \cdot \nabla\theta &= S_\theta(\rho, \theta, q_v, q_c, q_r), \\
q_{i_t} + \mathbf{u} \cdot \nabla q_i &= S_{q_i}(\rho, \theta, q_v, q_c, q_r), \qquad \text{for } i = v, c, r,
\end{aligned} \tag{1}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{k}$ is the unit vector $[0\,0\,1]^{\mathcal{T}}$, and $\epsilon = R/R_v$ is the ratio of the gas constant of dry air, R and the constant of water vapor, R_v. Because moist air contributes to the buoyancy of the flow, the right hand side of the momentum

equation is corrected with total buoyancy $\mathbf{B} = g(1 + \epsilon q_v - q_c - q_r)\mathbf{k}$. Due to the microphysical processes that involve phase change in the water content, the source/sink term S at the right-hand side of the equations of potential temperature and water tracers must be computed. These processes are modeled by the Kessler parameterization [12]. Equations (1) must be solved in $\Omega\ \forall t \in (0, T)$. Initial and boundary conditions will be assigned. θ, ρ, and p are related through the equation of state for a perfect gas.

2.1 Dynamic Dissipation in an LES Sense

In the absence of any type of either physical or artificial viscosity, the high-order SEM[1] approximation of (1) is characterized by numerical instabilities that may cause the solution to break if not stabilized in some way. Furthermore, in the case of the transport equations for water tracers, where the water quantities are often characterized by sharp gradients, unphysical Gibbs oscillations may compromise the stability of the solution even more (see, e.g., [14] and citations therein). To stabilize the problem, the Euler equations are corrected to include an artificial diffusion whose viscosity coefficients are given by a residual-based approximation that leads the problem to converge to the entropy solution, as proved in [17].

Remark 1 Because a saturation adjustment scheme [20] is used to treat the moist thermodynamics, the source terms are set to zero in the main step of the solution, and are only computed within the Kessler sub-step. For this reason, the sources will not appear in the regularized version of Eqs. (1).

We write:

$$\begin{aligned}
\rho'_t + \mathbf{u}\cdot\nabla\rho + \rho\nabla\cdot\mathbf{u} &= \nabla\cdot(\nu_n\nabla\rho)\\
\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} + \tfrac{1}{\rho}\nabla\cdot(\mathbf{I}p) &= \tfrac{1}{\rho}\nabla\cdot\left(\mu_n\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}}\right)\right) + \mathbf{B}\\
\theta_t + \mathbf{u}\cdot\nabla\theta &= \nabla\cdot(\kappa_n\nabla\theta)\\
q_{i_t} + \mathbf{u}\cdot\nabla q_i &= \nabla\cdot(\nu_c\nabla\theta)\,.
\end{aligned} \tag{2}$$

Except for ν_c that, for the time being, is set to a constant, the viscosity coefficients that appear in the first five equations are computed dynamically as a function of the solution. They are calculated element-wise on every high order element Ω_e. More

[1]The high-order spectral elements used for this study are built using Legendre-Gauss-Lobatto (LGL) integration and interpolation points.

specifically, given the sensible temperature $T = \theta(p/p_0)^{R/c_p}$ and one element with equivalent length $\bar{h}_{\Omega_e}$, we start by defining the dynamic viscosities

$$\mu_{\max}|_{\Omega_e} = 0.5\bar{h}_{\Omega_e} \| |\mathbf{u}| + \sqrt{c_p(\gamma-1)T} \|_{\infty,\Omega_e}, \tag{3}$$

and

$$\mu_1|_{\Omega_e} = \bar{h}^2_{\Omega_e} \max\left(\frac{\|R(\rho)\|_{\infty,\Omega_e}}{\|\rho - \bar{\rho}\|_{\infty,\Omega}}, \frac{\|R(\mathbf{u})\|_{\infty,\Omega_e}}{\|\mathbf{u} - \bar{\mathbf{u}}\|_{\infty,\Omega}}, \frac{\|R(\theta)\|_{\infty,\Omega_e}}{\|\theta - \bar{\theta}\|_{\infty,\Omega}} \right), \tag{4}$$

where $\bar{\cdot}$ indicates the space average of the quantity at hand over Ω and the $\| \cdot \|_{\infty,\Omega}$ terms at the denominator are used for normalization for a consistent dimension of the resulting equation. Having $\mu_{\max}$ and μ_1 constructed, we can compute the dynamics coefficients of the viscosity terms in Eqs. (2) as:

$$\mu_n|_{\Omega_e} = \min\left(\mu_{\max}|_{\Omega_e}, \mu_1|_{\Omega_e}\right), \quad \kappa_n|_{\Omega_e} = \frac{\Pr}{\gamma - 1}\mu_n|_{\Omega_e}, \quad \nu_n|_{\Omega_e} = \frac{\Pr}{\|\rho^n\|_{\infty,\Omega_e}}\mu_n|_{\Omega_e}, \tag{5}$$

where $Pr = 0.7$ is the Prandtl number of dry air.

Remark 2 To keep the discussion brief, the details of the derivation of the equations is not reported and the notation is somewhat abused. A proper formulation will be reported in a subsequent paper.

3 Space and Time Discretization

Equations (2) are approximated in space by high order spectral elements using LGL points and by an Implicit-Explicit (IMEX) method in time. Details can be found in, e.g. [5] (SEM) and [6] (IMEX).

4 Numerical Tests

The SEM-LES method is tested against benchmarks of ubiquitous use when testing the dynamical core of new atmospheric codes. First, the model is verified in dry mode. We perturb a neutrally stable atmosphere with a cold thermal anomaly that triggers the development of a density current. Once we have verified the ability of the model to handle dry dynamics, we solve a fully three-dimensional supercell triggered by the thermal perturbation of a realistic, moist, partially unstable background state.

4.1 Density Current in a Pseudo-3D Domain

The density current is a standard benchmark in the development of atmospheric codes [21]. The inviscid version of [1] is used for our analysis. This is because we are interested in assessing the current LES-like approach as a stabilizing tool that does not require further viscosity. The background state is characterized by a neutral atmosphere at uniform potential temperature $\theta = 300$ K and hydrostatically balanced pressure. Due to the symmetry of the original problem with respect to the plane center line of the $x - z$ plane, the solution is computed in the region $\Omega = 25.6 \times \infty \times 6.4\,\text{km}^3$ The perturbation θ' centered in $(x_c, z_c) = (0, 3)$ km has radii $(r_x, r_y, r_z) = (4, \infty, 2)$ km and is given by $\theta' = 0.5\Delta\theta\,(1 + \cos(\pi_c R))$ for $R \leq 1$, with amplitude $\Delta\theta = -15$ K and section $R = \sqrt{(x - x_c)/r_x^2 + (z - z_c)/r_z^2}$. Periodic boundary conditions are used along y whereas no-flux conditions are set in x and z. The initial velocity is zero everywhere. Figure 1 shows the fully developed current at time $t = 900$ s on two grids with uniform resolutions $\Delta x = \Delta z = 50$ m and $\Delta x = \Delta z = 25$ m. To measure the front position at $t_f = 900$ s, we take the node on the ground where $\theta' = -1$ K. A comparison of the front position and $\theta'_{max,min}$ with respect to previous work is reported in Table 1. As the resolution decreases, the front appears slower; this fact is also observed in Fig. 5 of [21].

We are aiming at using the current stabilizing scheme as a Large Eddy Simulation scheme. As a first analysis in this direction, we compare how the current model compares with the classical model by Lilly and Smagorinsky [13, 19]. The Smagorinsky solution (implemented within NUMA as well) is plotted in Fig. 2. A more thorough

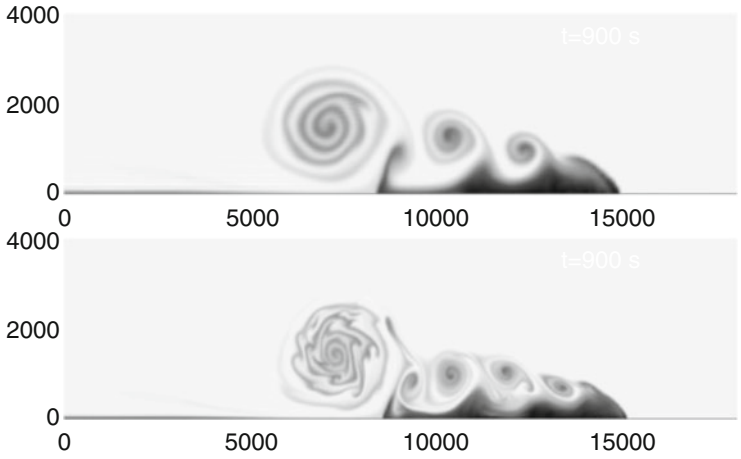


Fig. 1 Density current: θ' at 900 s. *Top*: $128 \times 1 \times 32$ el. ($\overline{\Delta z} = \overline{\Delta x} \approx 50$ m). *Bottom*: $256 \times 1 \times 64$ el. ($\overline{\Delta z} = \overline{\Delta x} \approx 25$ m). 4th-order elements

Table 1 Case 3. Comparative results of front location at 900 s

Model	N_{el}	Order	$\mu = 75\,\mathrm{m^2\,s^{-1}}$	Front location (m)
LES (25 m)	$256 \times 1 \times 64$	4th	NO	15,080
LES (50 m)	$128 \times 1 \times 32$	4th	NO	14,888
LES (100 m)	$64 \times 1 \times 16$	4th	NO	14,546
LES (200 m)	$32 \times 1 \times 8$	4th	NO	13,736
LES	$32 \times 1 \times 8$	6th	NO	14,568
LES	$32 \times 1 \times 8$	8th	NO	14,754
VMS [16] (25 m)			NO	14,890
VMS [16] (50 m)			NO	14,629
VMS [16] (75 m)			NO	14,487
VMS [16] (100 m)			NO	14,355
WRF-ARW 50 m			YES	14,470
SE [5] 50 m			YES	14,767
DG [5] 50 m			YES	14,767
f-wave (FV) [1] 50 m			YES	14,975
REFC [21] 50 m			YES	14,437
PPM [21] 50 m			YES	15,027

LES (SEM), VMS (FE), WRF-ARW V2.2 (FD), f-wave (FV), filtered Spectral Elements (SE), filtered Discontinuous Galerkin (DG), REFC, REFQ and PPM results are compared. All models but LES and VMS used artificial diffusion with constant $\mu = 75\,\mathrm{m^2\,s^{-1}}$

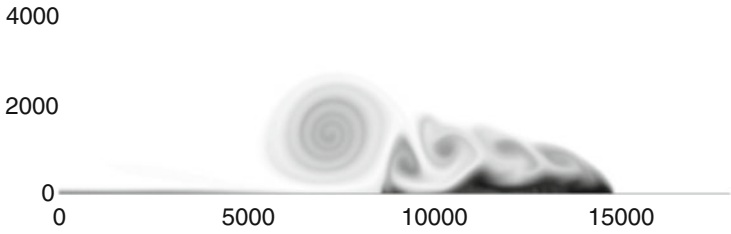


Fig. 2 Density current using a classical Smagorinsky SGS scheme with constant $C_s = 0.14$: θ' at 900 s. $256 \times 1 \times 64$ el. ($\overline{\Delta z} = \overline{\Delta x} \approx 25$ m) 4th-order elements

and quantitative analysis is currently being carried out by the authors. At a resolution $\overline{\Delta z} = \overline{\Delta x} \approx 25$ m and by plotting comparable contours (values not shown in the plot), the two models are highly comparable, although the degree of dissipation of the current scheme seems lower than Smagorinsky's using a Smagorinsky constant $C_s = 0.14$. Significantly more sub-grid structures are resolved using the current model. Further analysis is though required.

Remark 3 Throughout this paper we have discussed an LES approach to stabilization. Nevertheless, it must be pointed out that the simulations that we have presented are not necessarily to be viewed as LES simulations unless finer grids are used.

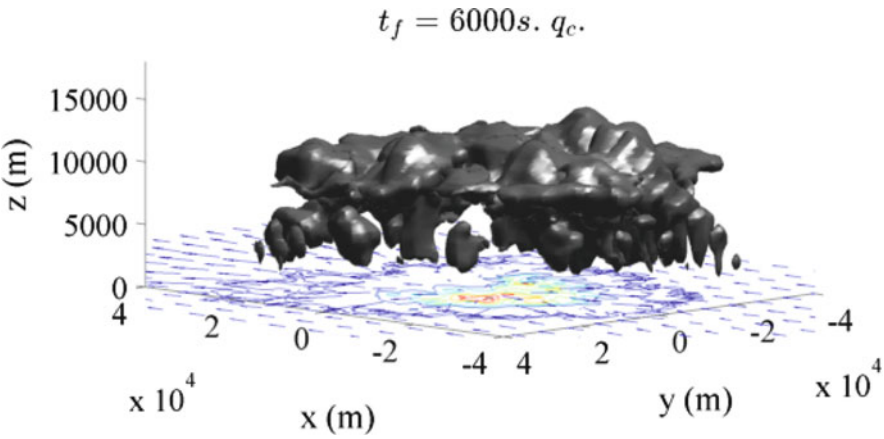


Fig. 3 Supercell: 3d view (using az = -135 and el = 8) of q_c (*grey surface*), surface velocity (vectors), and the instantaneous distribution of q_r on the ground (contours)

4.2 3D Moist Convection

The three-dimensional simulation of a convective cell is defined in the domain $160 \times 120 \times 24\,\text{km}^3$. The initial field is perturbed by a temperature anomaly θ' 3 K warmer than the surrounding environment, which is given by the sounding of [4]. The domain Ω^h is subdivided into $40 \times 30 \times 24$ elements of order 4. A stretched grid along z is used to make the resolution higher in the lower atmosphere where convection is triggered. The domain is crossed by a horizontal wind along the x-direction with a $12\,\text{m}\,\text{s}^{-1}$ shear at $z = 2000\,\text{m}$. A no-slip condition is applied on the surface boundary while periodic boundaries are defined along x and y. A Rayleigh type absorbing layer is included at $z \geq 19{,}000\,\text{m}$. The cloud first forms at approximately 500 s, and is fully develop after 4500 s. A 3D instantaneous view of q_c is plotted in Fig. 3. Qualitatively, it is comparable to previous results on a similar case. A quantitative evaluation of the instantaneous rain on the ground is plotted in Fig. 4a, whereas the cloud content obtained by averaging q_c along the $y-$direction is plotted in Fig. 4b.

5 Conclusions

We extended to high order spectral elements the LES-based stabilization method first introduced in [18] for the finite element solution of fully compressible flows. We explored the capabilities of this inexpensive technique to solve the Euler equations of stratified flows at the low-Mach regimes encountered in atmospheric flows. When applied to dry and moist simulations, the current implementation appears

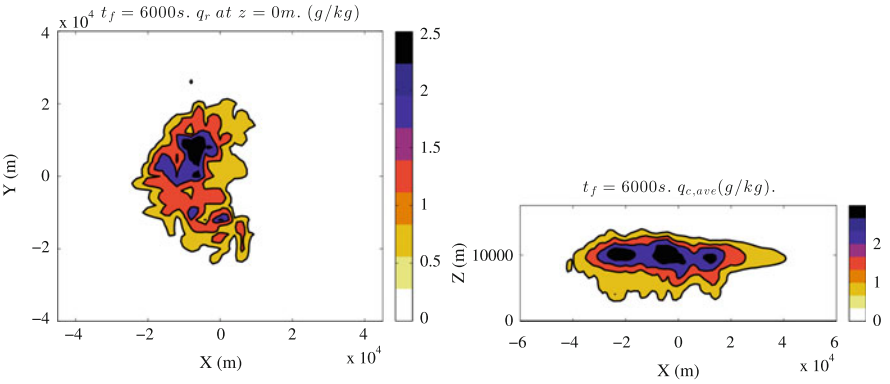


Fig. 4 3D supercell: horizontal slice of q_r at $z = 0$ m and y-averaged q_c at $t = 6000$ s. (**a**) Instantaneous rain distribution on the ground at $t = 6000$ s (**b**) Vertical slice of the distribution of q_c averaged along the y direction

to give satisfactory results that are comparable to others presented in the literature. Without the need for any additional viscosity, this dynamic LES scheme proved to be sufficient to stabilize the spectral element solution of the Euler equations in atmospheric applications. However, since a thorough analysis was not carried out to evaluate this approach in terms of its turbulence modeling properties, much additional work is necessary to fully assess it in its applicability as a turbulence closure for atmospheric simulations.

Acknowledgements The authors are thankful to Dr. Murtazo Nazarov for his clarifications about the original method. They also gratefully acknowledge the support of the Office of Naval Research through program element PE-0602435N, the National Science Foundation (Division of Mathematical Sciences) through program element 121670, and the Air Force Office of Scientific Research through the Computational Mathematics program. The first and second authors were supported by the National Academies through a National Research Council fellowship.

References

1. N. Ahmad, J. Lindeman, Euler solutions using flux-based wave decomposition. Int. J. Numer. Method Fluids **54**, 47–72 (2007)
2. M. Avila, R. Codina, J. Principe, Large eddy simulation of low mach number flows using dynamic and orthogonal subgrid scales. Comput. Fluids **99**, 44–66 (2014)
3. A.N. Brooks, T.J.R Hughes, Streamline upwind/Petrov-Galerkin formulations for convective dominated flows with particular emphasis on the incompressible Navier-Stokes equations. Comput. Methods Appl. Mech. Eng. **32**, 199–259 (1982)
4. S. Gaberšek, F.X. Giraldo, J. Doyle, Dry and moist idealized experiments with a two-dimensional spectral element model. Mon. Weather Rev. **140** 3163–3182 (2012)

5. F.X. Giraldo, M. Restelli, A study of spectral element and discontinuous Galerkin methods for the Navier-Stokes equations in nonhydrostatic mesoscale atmospheric modeling: equation sets and test cases. J. Comput. Phys. **227**, 3849–3877 (2008)
6. F.X. Giraldo, J.F. Kelly, E. Constantinescu, Implicit-explicit formulations of a three-dimensional Nonhydrostatic Unified Model of the Atmosphere (NUMA). SIAM J. Sci. Comput. **35**, 1162–1194 (2013)
7. J.L. Guermond, R. Pasquetti, Entropy-based nonlinear viscosity for Fourier approximations of conservation laws. C. R. Acad. Sci. Ser. I **346**, 801–806 (2008)
8. G. Houzeaux, J. Principe, A variational subgrid scale model for transient incompressible flows. Int. J. Comput. Fluid Dyn. **22**, 135–152 (2008)
9. T. Hughes, Multiscale phenomena: Green's functions, the Dirichlet-to-Neumann formulation, subgrid scale models, bubbles and the origins of stabilized methods. Comput. Methods Appl. Mech. and Eng. **127**, 387–401 (1995)
10. T.J.R Hughes, L.P Franca, G.M. Hulbert, A new finite element formulation for computational fluid dynamics: III. The Galerkin/least-squares method for advection-diffusive equations. Comput. Methods Appl. Mech. Eng. **73**, 329–336 (1989)
11. J.F. Kelly, F.X. Giraldo, Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: limited-area mode. J. Comput. Phys. **231**, 7988–8008 (2012)
12. E. Kessler, On the distribution and continuity of water substance in atmospheric circulation. Meteorol. Monogr. **10**, 32 (1969)
13. D.K. Lilly, On the numerical simulation of buoyant convection. Tellus **14**, 148–172 (1962)
14. S. Marras, F.X. Giraldo, A parameter-free dynamic alternative to hyper-viscosity for coupled transport equations: application to the simulation of 3D squall lines using spectral elements. J. Comput. Phys. **283**, 360–373 (2015)
15. S. Marras, M. Moragues, M R. Vázquez, O. Jorba, G. Houzeaux. Simulations of moist convection by a variational multiscale stabilized finite element method. J. Comput. Phys. **252**, 195–218 (2013)
16. S. Marras, M. Moragues, M R. Vázquez, O. Jorba, G. Houzeaux, A variational multiscale stabilized finite element method for the solution of the Euler equations of nonhydrostatic stratified flows. J. Comput. Phys. **236**, 380–407 (2013)
17. M. Nazarov, Convergence of a residual based artificial viscosity finite element method. Comput. Math. Appl. **65**(4), 616–626 (2013)
18. M. Nazarov, J. Hoffman, Residual-based artificial viscosity for simulation of turbulent compressible flow using adaptive finite element methods. Int. J. Numer. Methods Fluids **71**, 339–357 (2013)
19. J. Smagorinsky, General circulation experiments with the primitive equations: I. the basic experiment. Mon. Weather Rev. **91**, 99–164 (1963)
20. S. Soong, Y. Ogura, A comparison between axisymmetric abd slab0symmetric cumulus cloud models. J. Atmos. Sci. **30**, 879–893 (1973)
21. J. Straka, R. Wilhelmson, L. Wicker, J. Anderson, K. Droegemeier. Numerical solution of a nonlinear density current: a benchmark solution and comparisons. Int. J. Numer. Methods Fluids **17**, 1–22 (1993)

High-Order Finite-Differences on Multi-threaded Architectures Using OCCA

David Medina, Amik St-Cyr, and Timothy Warburton

Abstract High-order finite-difference methods are commonly used in wave propagator for industrial subsurface imaging algorithms. Computational aspects of the reduced linear elastic vertical transversely isotropic propagator are considered. Thread parallel algorithms suitable for implementing this propagator on multi-core and many-core processing devices are introduced. Portability is addressed through the use of the OCCA runtime programming interface. Finally, performance results are shown for various architectures on a representative synthetic test case.

1 Introduction

High-order finite-differences are used in seismic imaging and many other industrial applications primarily because of their computational efficiency. A high-order wave propagator for vertical transversely isotropic media (VTI), at the heart of numerous seismic imaging applications such as full waveform inversion and reverse time migration, is studied with respect to its multi-threaded performance on various current and emerging computing architectures. OCCA, a recently developed library for handling multi-threading is employed. The latter is a C++ library making use of run-time compilation and macro expansions which results in a novel and simple single kernel language that expands to multiple threading languages. OCCA supports device kernel expansions for the OpenMP, OpenCL, pThreads, Intel COI and CUDA APIs. In the following we describe the reduced elastic VTI model for isotropic media together with a typical finite-difference discretization employed in industry and present performance characteristics for implementations built on top of the OCCA API. Using the unified OCCA

D. Medina (✉) • T. Warburton
Computational and Applied Mathematics, Rice University, Houston, TX, USA
e-mail: dsm5@rice.edu; timwar@rice.edu

A. St-Cyr
Seismic Applications Team, Royal Dutch Shell, Rijswijk, Netherlands
e-mail: amik.st-cyr@shell.com

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_33

programming approach allows customized kernels optimized for CPU and GPU architectures.

The VTI propagator introduced in [3] is given by

$$\frac{\partial^2 p}{\partial t^2} = v_x^2 \left[\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right] + v_z^2 \frac{\partial^2 q}{\partial z^2} + s(t)\delta(\mathbf{x} - \mathbf{x_i}), \tag{1}$$

$$\frac{\partial^2 q}{\partial t^2} = v_n^2 \left[\frac{\partial^2 p}{\partial x^2} + \frac{\partial^2 p}{\partial y^2} \right] + v_z^2 \frac{\partial^2 q}{\partial z^2}. \tag{2}$$

In the preceding equations, p is an approximation for the P-wave while q is and auxiliary wavefield variable. ϵ and δ are the anisotropic parameters. The vertical P-wave velocity is represented with v_z and its horizontal component is $v_x = v_z\sqrt{1+2\epsilon}$ while the normal move-out velocity is $v_n = v_z\sqrt{1+2\delta}$. For this approximation to be relevant $\epsilon - \delta \leq 0$ is necessary. The forcing considered in our benchmark is the Ricker wavelet $s = (1 - 2\pi^2 f^2 t^2)e^{-\pi^2 f^2 t^2}$ with $f = 15Hz$.

We consider a centered finite-difference discretization in time and space in second order form on infinite domains. For $\mathbf{u}(\mathbf{x}, t) = (p, q)^T$ and $\mathbf{F}(\mathbf{u}, \mathbf{x}, t)$ set as the right and side of (1) and (2) the centered in time approximation reads

$$\mathbf{u}^{n+1} - 2\mathbf{u}^n + \mathbf{u}^{n-1} \approx \Delta t^2 \mathbf{F}(\mathbf{u}^n) \tag{3}$$

where $\mathbf{u}^k \equiv \mathbf{u}(\mathbf{x}, t^n)$ with $t^n = n\Delta t$. High-order finite-difference stencils are of practical importance for the efficient numerical solutions of wave propagation problems [1, 14]. Indeed, for a similar number of points composing the computational grid, the number of points required to resolve the shortest wavelength (as defined by Nyquist) decreases and gets close to the spectral or pseudo-spectral limit of two points per wavelength [4]. Most propagators used in seismic applications use two different flavors of high-order finite-differences. The earth subsurface is geologically horizontally layered. Since depth, represented by the z coordinate, will experience the most changes in the rock properties, while in the $x - y$ planes the properties will remain constant within a layer. Therefore, a common strategy is to have a symmetric stencil in the $x - y$ direction, while handling a variable spacing in z. The weights and spacings can be optimized to handle a variety of physical and numerical properties [5, 6]. For simplicity, we suppose a domain $\Omega = [0, Lx] \times [0, Ly] \times [0, Lz]$ where $\Delta x = \Delta y = h$ and Δz_k result from the discretization in space using $N_{d=\{x,y,z\}}$ points in each direction respectively. The mesh size in the z direction varies per grid point belonging to a different $x-y$ plane. Adopting the convention $p(x_i, y_j, z_k) = p_{i,j,k}$, the differentiation stencil in the $x - y$ plane is

$$h^2\left(\frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}\right)p_{i,j,k} \approx w_0^{xy} p_{i,j,k} + \sum_{l=1}^{R_{xy}} w_l^{xy} \left(p_{i+l,j,k} + p_{i-l,j,k} + p_{i,j+l,k} + p_{i,j-l,k}\right) \tag{4}$$

where the w_l^{xy} are the $R_{xy} + 1$ weights for approximating the two dimensional Laplacian. The differentiation is a bit simpler in the z direction:

$$\frac{\partial^2}{\partial z^2} q_{i,j,k} \approx \sum_{l=-R_z+}^{R_z} w_{k,l}^z q_{i,j,k+l}. \tag{5}$$

Again, the $w_{k,l}^z$ are the weights for approximated the second derivative. However, for each position z_k where the value of the derivative is sought, $2R_z + 1$ weights are needed instead of $R_z + 1$ as in the symmetric case due to the asymmetry in the z direction. The grid size Δz_k is absorbed into the $w_{k,l}^z$ weights in practice and therefore are not appearing above. The domain Ω is embedded into a larger domain where a damping formula is applied as in [2]. Outside the damping region, the solution is assumed to be zero for R_{xy} points in the x and y directions and R_z points in z.

In the following sections, we describe the reduced elastic VTI model for isotropic media together with a typical finite-difference discretization employed in industry.

2 Computational Efficiency of High-Order Finite Differences

The peak parallel floating point operations per second (flops) available on modern CPUs and GPUs have followed the trend set by Moores law. Unfortunately, the available memory bandwidth lagged this trend. This gap in bandwidth currently favors algorithms generating lots of flops per byte of data moved [7]. For VTI, using this type of stencil, a pessimistic computational intensity is

$$CI \approx (1/4)(5R_{xy} + 4R_z)/(4R_{xy} + 2R_z) \approx 0.4 \ \text{ flops/byte} \tag{6}$$

where most of the loads are assumed to be not in cache. An idealized version is to consider the least loads as possible (assumes most of the data in cache). This is done by assuming three single precision loads per point for the model properties (v_x^2, v_n^2 and v_z^2) as well as the two pairs of loads and stores for $\mathbf{u}^n$ and, respectively, $\mathbf{u}^{n+1}$.

$$CI \approx (1/28)(5R_{xy} + 4R_z) \approx 0.3(R_{xy} + R_z) \ \text{ flops/byte} \tag{7}$$

Therefore increasing the order of the stencil augments the intensity since the low order case is close to the pessimistic estimate. In practice, better approximations can be obtained [15]. Performing those measurements automatically using hardware counters is still in development [12]. Moreover in [4], the effectiveness of finite-differences for wave propagation problems is shown to increase with order. Indeed for a fixed number of Fourier modes "M", $\tilde{N}_d$ points are required to guarantee their resolution according to Nyquist. The relation is approximated with $\tilde{N}_d = c_p M^{1+(2R)^{-1}}$ and therefore doubling the polynomial order for a fixed number of

modes, leads to $M^{\frac{1}{4}}$ times more points in each direction. Since the method is explicit in time, the total increase in computational cost is M in 3D.

3 OCCA: Portable Multi-threading

OCCA, a recently developed C++ library for handling multi-threading is employed. The OCCA library is an API providing a kernel language and an abstraction layer to back-ends APIs such as OpenMP, OpenCL and CUDA see [10, 11, 13] amongst others. It uses run-time compilation and macro expansions which results in a novel and simple single kernel language that expands to multiple threading languages. OCCA currently supports device kernel expansions for the OpenMP, OpenCL, pThreads, Intel COI and CUDA languages. Performance characteristics are given for our implementations built on top of the OCCA API. Using the unified OCCA programming approach allows customized kernels optimized for CPU and GPU architectures with a single "host" code.

OCCA host API: Aside from language-based libraries from OpenMP, OpenCL or CUDA, the OCCA host API is a stand-alone library.This independence allows OCCA to be combined with other libraries without conflict, as shown in Fig. 1. The three key components that influenced the OCCA host API development: the platform device, device memory and device kernels. Presenting the entire OCCA API is not feasible in this paper. For the complete details see [8] and the git repository for the latest developments.[1] We try here to expose the minimal knowledge required to write the VTI kernel.

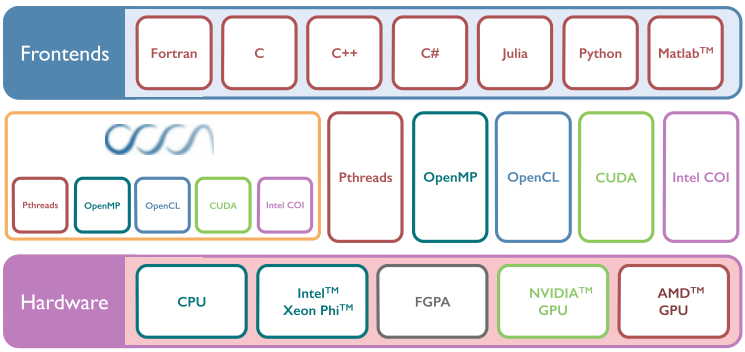


Fig. 1 OCCA wraps different language APIs and is non-conflicting with external libraries in either platform

[1] http://www.github.com/tcew/OCCA.

```
for(int bZ = 0; bZ < gridDim.z; ++bZ; outer2){        // (1)
  for(int bY = 0; bY < gridDim.y; ++bY; outer1){
    for(int bX = 0; bX < gridDim.x; ++bX; outer0){
      // Shared memory is initialized here
      for(int tZ = 0; tZ < blockDim.z; ++tZ; inner2){  // (2)
        for(int tY = 0; tY < blockDim.y; ++tY; inner1){
          for(int tX = 0; tX < blockDim.x; ++tX; inner0){
            // Work here,  initialize register memory
}}}}}}
```

Listing 1 The expansion of the implicit for-loops found in CUDA and OpenCL kernels is displayed. The OCCA outer-loops (1) map to multi-dimensional work-groups [[blocks]] and OCCA inner-loops (2) map to multi-dimensional work-items [[threads]].

```
Partition the top plane of the grid into B_x × B_y blocks of size w × h

For time-step n = 0, 1, ..., time-Steps
    For each block (b_i, b_j)                                          (1)
        For n = 0, 1, ..., N_z
            For each point (i, j, k) such that
                (b_i ≤ i < b_i + w) and (b_j ≤ j < b_j + h)             (2)
                Update p^{n+1}(i, j, k) and q^{n+1}(i, j, k)
            End For // Point Update
        End For     // Traversing depth
    End For         // Iterating over blocks
End For             // Computing a time-step update
```

Listing 2 For each time-step, the 2D blocks at the top of the structured grid sweep in the z direction and update all points in the current z plane.

OCCA kernel language: GPU computing involves many threads and the thread-space is logically decomposed into thread-blocks. Thread blocks are queued for execution onto the available multiprocessors. In general a GPU chip has more than a single multiprocessor and the choices for number of blocks and threads per blocks are dependent on the algorithm, resources available and the developer. he resulting kernel language is shown in Listing 1, where outer-loops and inner-loops, denoted by the 4th clause in the for-loops, map to work-groups [[blocks]] and work-items [[threads]] respectively.

The use of shared memory is still available in OCCA since it is essential for many GPU optimized codes. Shared memory still acts as a scratchpad cache for GPU architectures but can be seen as a prefetch buffer for CPU-modes in OCCA.

The OCCA:OpenMP code performs the VTI steps uses the classic technique of cache blocking as seen in code Listing 2. The best performing kernel had 2D cache blocking with the Z-block first followed by the Y-block. The innermost loop would be x (stride-1) then z and finally y. The z-blocks were handled in OCCA with `occaOuterFor2`, the y one with `occaOuterFor1` and so on. The vectorization was handled directly by the Intel compiler by placing a pragma `#pragma ivdep` in the `occaInnerFor0` (x) and making sure the data was correctly padded. The size of the

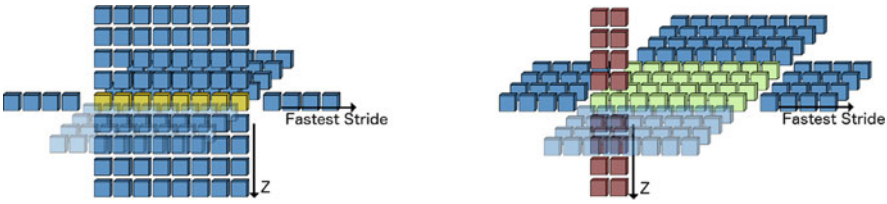


Fig. 2 The *left panel* represents a 3D finite-difference stencil vectorized with AVX. The fast stride is in the x direction and 8 single precision stencil evaluation are performed simultaneously. The *right panel* represents thread block with the large 2D subdomain the information loaded into shared (fast) memory. The register rolling in the z-direction is shown and for a "two-elements" Kernel, each thread handles two columns

blocking in $z-y$ was determined as $(28, 20)$ by running the code over a set of grids and possible block ranges and comparing throughput times see Sect. 4. The OpenMP first-touch policy was critical in obtaining performance across dual sockets as well as the correct thread affinity. Finally, to make sure the compiler was optimizing as depicted in Fig. 2, a hand written kernel with explicit register blocking was written: 5 % increase in performance was observed.

A single implementation encompasses OCCA:OpenCL and OCCA:CUDA follows directly the work of [9]. As depicted in the right panel of Fig. 2, for a given thread block, the 2D $x-y$ stencil executes into fast shared memory while the z direction is handled by register rolling. If each thread handles one such column per thread block then this is a *one-element* approach while a *two-elements* approach consists of having two such columns per thread. Care was taken to align the data to enable coalescing loads to shared memory.

4 Performance

The VTI kernel is integrated in time for a thousand time steps. A metric of performance used in seismic is the throughput: number of sweeps through the entire grid block per second. The precision is set at $R_{xy} = 12$ and $R_z = 8$ and yields approximatively 92 flops per point. The CI optimistic model derived in Sect. 2 yields a factor of 3.3.

Results on a dual socket node with E5-2670 are reported in Table 1. The dual socket node is capable of 666 single precision GFlops while the bandwidth is 102.4 GB/s. The optimistic CI predicts a maximal peak of 47 %. The results show the fastest OCCA kernel achieving 21 % and good scalability as compared to the native OpenMP code (without OCCA). The difference stems from the added knowledge at compile time for OCCA, where all loop-bounds are known at compile time.

Table 1 Multithreading scaling with OpenMP using alternative thread distributions on different number of cores (using two Xeon E5-2640 Processors)

Project	Distribution	1 thread	2 threads	4 threads	8 threads	16 threads	% Peak
Native	Compact	92	183 (98 %)	360 (96 %)	668 (89 %)	1226 (82 %)	17
Native	Scatter	92	183 (98 %)	356 (95 %)	686 (92 %)	1191 (80 %)	16
OCCA	Compact	115	229 (99 %)	448 (97 %)	820 (89 %)	1548 (84 %)	21
OCCA	Scatter	115	230 (100 %)	454 (98 %)	884 (96 %)	1411 (76 %)	19

Table 2 Performance comparisons on the VTI update kernels tailored for GPU architectures

Project	Kernel language	K10 (1-chip)	K20x
Native	CUDA	1068	1440
Native (2)	CUDA	1296	2123
OCCA	OCCA:CUDA	1241	1934
OCCA (2)	OCCA:CUDA	1579	2431
OCCA	OCCA:OpenCL	1303	1954
OCCA (2)	OCCA:OpenCL	1505	2525

Update kernels use 1-point updates per work-item/thread or are labeled with (2) to represent 2-point update kernels. One K10 chip runs at 745 MHz and contains 1536 floating point units with 160 GB/s bandwidth. By comparison, the K20x runs at 732 MHz and contains 2496 floating point units with 250 GB/s bandwidth

Table 3 Performance comparisons between combinations of OpenMP, CUDA and OpenCL running on the CPU and GPU tailored kernels

	CPU-tailored Kernel	GPU-tailored Kernel
OpenMP	1548	364 (23 %)
CUDA (1 K10 core)	515 (41 %)	1241
OpenCL (1 K10 core)	665 (51 %)	1302

Table 2 contains performance on GPU architectures that were based on optimized CUDA code and translated to OCCA. We note that performance seen in Table 2 was on par with native code due to optimizations that can be done with run-time compilation including manual unrolling and manual bounds on OpenMP-loops.

Table 3 contains results from two optimized kernels, a CPU-tailored code and a GPU-tailored code, run on OpenMP, OpenCL and CUDA to note performance portability. Although it was expected that optimal CPU-tailored algorithms would not give optimal performance for GPU architectures, we see 40–50 % of optimal performance by just running the OCCA kernels in GPU-modes. The GPU-tailored algorithm ran on CPU-modes ended running on 20 % performance compared with optimal CPU code, mainly due to the lack of direct control over shared memory as seen on GPU architectures.

5 Conclusion and Future Work

We have studied a vertical transverse isotropic propagator discretized with centered finite-differences in time and space. Finite-differences are extensively used in seismic modeling. We have justified the advantage of using high-order stencils both in terms of computational efficiency and points needed per wavelength. To enable the study on various compute architectures, a multi-threaded gateway API to many multi-threading APIs was employed: OCCA. The performance results obtained with the library are generally faster than with the codes written using the best API for the hardware, thanks to the just-in-time compilation. For now, it seems a single OCCA kernel solution performing well for two types of architecture is impossible. The main factor preventing portable optimization is due to the lack of direct control over cache on CPU architectures which can be done on GPU architectures through shared memory. This level of control is currently only available for GPGPUs and unavailable for traditional CPUs. Having such control on the next generation of CPUs would most certainly re-open possibilities of a single code performing efficiently on both architectures.

Acknowledgements This work funded partly by Royal Dutch Shell, ONR award number N00014-13-1-0873, and sub-contract to the CESAR Exascale Co-design Center at Argonne National Lab award number ANL 1F-32301.

References

1. R.M. Alford, K.R. Kelly, D.M. Boore, Accuracy of finite-difference modeling of the acoustic wave equation. Geophysics **39**(6), 834–842 (1974)
2. C. Cerjan, D. Kosloff, R. Kosloff, M. Reshef, A nonreflecting boundary condition for discrete acoustic and elastic wave equations. Geophysics **50**(4), 705–708 (1985)
3. X. Du, R.P. Fletcher, P.J. Fowler, A new pseudo-acoustic wave equation for vti media, in *70th EAGE Conference & Exhibition*, 2008
4. B. Fornberg, The pseudospectral method: Comparisons with finite differences for the elastic wave equation. Geophysics **52**(4), 483–501 (1987)
5. B. Fornberg, Classroom note: calculation of weights in finite difference formulas. SIAM Rev. **40**(3), 685–691 (1998)
6. O. Holberg, Computational aspects of the choice of operator and sampling interval for numerical differentiation in large-scale simulation of wave phenomena. Geophys. Prospect. **35**(6), 629–655 (1987)
7. J.D. McCalpin, Stream: sustainable memory bandwidth in high performance computers. Technical report, University of Virginia, Charlottesville, Virginia, 1991–2007. A Continually Updated Technical Report. http://www.cs.virginia.edu/stream/
8. D.S. Medina, A. St.-Cyr, T. Warburton, OCCA: a unified approach to multi-threading languages. CoRR, abs/1403.0968 (2014)
9. P. Micikevicius, Gpu performance analysis and optimization, in *GPU Technology Conference*, 2012. http://www.developer.download.nvidia.com/GTC/PDF/GTC2012/PresentationPDF/S0514-GTC2012-GPU-Performance-Analysis.pdf

10. J. Nickolls, I. Buck, M. Garland, K. Skadron, Scalable parallel programming with cuda. Queue **6**(2), 40–53 (2008)
11. OpenMP Architecture Review Board, OpenMP application program interface version 3.0, May 2008
12. F. Rubio, M. Hanzich, J. de la Puente, A. Farrés, M. Ferrer, P. Thierry, Roofline-based optimizations for elastic propagation on xeon, in *77th EAGE Conference and Exhibition 2015* (2015)
13. J.E. Stone, D. Gohara, G. Shi, Opencl: a parallel programming standard for heterogeneous computing systems. IEEE Des. Test **12**(3), 66–73 (2010)
14. D. Vishnevsky, V. Lisitsa, V. Tcheverda, G. Reshetova, Numerical study of the interface errors of finite-difference simulations of seismic waves. Geophysics **79**(4), T219–T232 (2014)
15. S. Williams, A. Waterman, D. Patterson, Roofline: an insightful visual performance model for multicore architectures. Commun. ACM **52**(4), 65–76 (2009)

Modified Equation Analysis for the Discontinuous Galerkin Formulation

Rodrigo Costa Moura, Spencer Sherwin, and Joaquim Peiró

Abstract In this paper we present an assessment of the discontinuous Galerkin (DG) formulation through modified equation analysis (MEA). When applied to linear advection, MEA can help to clarify wave-propagation properties previously observed in DG. In particular, a connection between MEA and dispersion-diffusion (eigensolution) analysis is highlighted. To the authors' knowledge this is the first application of MEA to DG schemes, and as such this study focuses only on element-wise constant and linear discretizations in one dimension. For the linear discretization, we found that the physical mode's accuracy can be increased via upwinding. MEA's application to higher order solutions and non-linear problems is also briefly discussed. In special, we point out that MEA's applicability in the analysis of DG-based implicit large eddy simulations seems infeasible due to convergence issues.

1 Introduction

The so-called modified equation analysis (MEA) technique is arguably one of the most fundamental tools one can apply to analyse a numerical scheme. By using Taylor series to rewrite discrete derivative expressions, MEA can reveal which PDE is actually governing a numerical solution. Generally, due to the presence of truncation terms, the resulting PDE (referred to as the modified equation) differs from the physical PDE being discretized. As a result, dispersion and diffusion errors are quantified and numerical aspects such as accuracy and stability can be assessed.

To the authors' knowledge, this is the first successful application of MEA to the discontinuous Galerkin (DG) formulation, although previous attempts have been reported [7]. The study focuses on linear advection in one dimension and sheds more light on wave-propagation characteristics previously verified for DG [3, 6], specially the super-convergence properties [1, 2]. Due to the exploratory character

R.C. Moura (✉) • S. Sherwin • J. Peiró
Aeronautics, Imperial College London, London SW7 2BY, UK
e-mail: r.moura13@imperial.ac.uk; rodrigomoura.t10@gmail.com

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_34

of the study, only element-wise constant and linear approximations are considered in detail, though the path to higher order discretizations is discussed.

In addition, we highlight a connection between MEA and eigensolution analysis (ESA), sometimes simply called dispersion-diffusion analysis, and argue that both techniques provide essentially the same information when applied to linear advection. Still, while ESA is restricted to linear problems, MEA's usage usually extends to non-linear problems. For such problems, an interesting application of MEA would be the analysis of DG-based implicit LES. In [5], this approach was chosen to assess the suitability of a specific finite volume scheme for implicit LES. However, as we point out further on, our results do not encourage such approach for DG.

2 MEA for Linear Advection with DG

We consider the 1D advection equation within an infinite or periodic domain Ω,

$$\frac{\partial u}{\partial t} + a\frac{\partial u}{\partial x} = 0\,, \tag{1}$$

being a the advection speed. In the DG framework, the solution is approximated by a weighted sum of basis functions ϕ_i within each element Ω_e ($\Omega = \bigcup_e \Omega_e$), namely

$$u|_{\Omega_e} \cong \sum_{i=0}^{P} c_i(t)\,\phi_i(\xi)\,, \tag{2}$$

in which ϕ_i is chosen in this study to be the orthonormal Legendre polynomial (of degree i) where ξ is defined in the standard domain $\Omega_{st} = [-1, 1]$, see [4] for details.

The discrete residue is then required to vanish at the element level via projection, while inter-element communication is enforced by a numerical flux. Hence, when using a polynomial basis of degree P, one is left with $P + 1$ PDEs per element:

$$\frac{h}{2a}\frac{\partial c_i}{\partial t} = \sum_{j=0}^{P} c_j \mu_{ij} \;-\; (\tilde{u}\,\phi_i)|_{\Omega_e^R} \;+\; (\tilde{u}\,\phi_i)|_{\Omega_e^L}\,, \tag{3}$$

where the left (or right) boundary of Ω_e is referred to by L (or R), h is the (constant over Ω) mesh spacing, while the constants μ_{ij} and the numerical flux $\tilde{u}$ are given by

$$\mu_{ij} = \int_{\Omega_{st}} \phi_j \frac{\partial \phi_i}{\partial \xi}\, d\xi \quad \text{and} \quad \tilde{u}(u_\ominus, u_\oplus) = \frac{u_\ominus + u_\oplus}{2} + \beta S_a \frac{u_\ominus - u_\oplus}{2}\,, \tag{4}$$

being $\beta \in [0, 1]$ an upwinding parameter, $S_a = |a|/a$ the sign of a, and $u_\ominus$ (or $u_\oplus$) is simply the local solution at the left (or right) side of each considered interface.

Finally, in order to obtain the modified equations, the coefficients c_i^L and c_i^R of the neighbouring elements must be expressed through quantities related to the (central) element Ω_e. This is done by considering c_i (for $i = 0, \ldots, P$) as functions of x and t, since they are the solution of the PDE in Eq. (3), and then by relating $c_i(x \pm h, t)$ to $c_i(x, t)$ via Taylor series, namely

$$c_i(x \pm h, t) = \sum_{k=0}^{\infty} (\pm 1)^k \frac{h^k}{k!} \partial_x^k c_i|_{(x,t)} \,, \tag{5}$$

which translates into

$$c_i^L = c_i - h\frac{\partial c_i}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 c_i}{\partial x^2} - \cdots \quad \text{and} \quad c_i^R = c_i + h\frac{\partial c_i}{\partial x} + \frac{h^2}{2!}\frac{\partial^2 c_i}{\partial x^2} + \cdots \,. \tag{6}$$

3 Analysis of $P = 0$ and $P = 1$ Discretizations

Deriving the modified equation for the element-wise constant ($P = 0$) discretization is straightforward because only one PDE stems from Eqs. (3) and (6):

$$\frac{\partial c_0}{\partial t} + a\frac{\partial c_0}{\partial x} = \frac{\beta|a|}{2}\frac{\partial^2 c_0}{\partial x^2}h - \frac{a}{6}\frac{\partial^3 c_0}{\partial x^3}h^2 + \frac{\beta|a|}{24}\frac{\partial^4 c_0}{\partial x^4}h^3 - \frac{a}{120}\frac{\partial^5 c_0}{\partial x^5}h^4 + \cdots \,, \tag{7}$$

where one can clearly see that the leading error term is $O(h)$ and of diffusive nature, provided that $\beta \neq 0$. If $\beta = 0$, i.e. for centred numerical flux, the leading error is $O(h^2)$ and of dispersive nature, as expected.

We note that the correct coefficients' PDE for any value of P is known (a priori) to be $\partial c_i/\partial t + a\,\partial c_i/\partial x = 0$. Once the analytical solution of the advection PDE is the exact propagation of a given signal, it is only natural to expect that the solution's coefficients should follow the same rule.

For the element-wise linear ($P = 1$) case, two PDEs stem from Eq. (3):

$$\begin{aligned} \frac{h}{2a}\frac{\partial c_0}{\partial t} = {} & -\tfrac{1}{2}\beta S_a c_0 + \tfrac{1}{4}c_0^R(\beta S_a - 1) + \tfrac{1}{4}c_0^L(\beta S_a + 1) + \\ & -\tfrac{\sqrt{3}}{2}c_1 - \tfrac{\sqrt{3}}{4}c_1^R(\beta S_a - 1) + \tfrac{\sqrt{3}}{4}c_1^L(\beta S_a + 1) \,, \end{aligned} \tag{8}$$

$$\begin{aligned} \frac{h}{2a}\frac{\partial c_1}{\partial t} = {} & +\tfrac{\sqrt{3}}{2}c_0 + \tfrac{\sqrt{3}}{4}c_0^R(\beta S_a - 1) - \tfrac{\sqrt{3}}{4}c_0^L(\beta S_a + 1) + \\ & -\tfrac{3}{2}\beta S_a c_1 - \tfrac{3}{4}c_1^R(\beta S_a - 1) - \tfrac{3}{4}c_1^L(\beta S_a + 1) \,. \end{aligned} \tag{9}$$

Before resorting to Taylor series, it is worth noting that these PDEs are coupled: $\partial_t c_0$ and $\partial_t c_1$ are both functions of c_0 and c_1. There is however a way of obtaining

separate PDEs, one for each coefficient. By defining the identity operator $\mathbf{I}[c_i] = c_i$, as well as the operators $\mathbf{M}$ and $\mathbf{N}$ through the relations in Eq. (6) as

$$\mathbf{M}[c_i] = \frac{c_i^R + c_i^L}{2} = c_i + \frac{h^2}{2!}\frac{\partial^2 c_i}{\partial x^2} + \frac{h^4}{4!}\frac{\partial^4 c_i}{\partial x^4} + \cdots , \tag{10}$$

$$\mathbf{N}[c_i] = \frac{c_i^R - c_i^L}{2} = h\frac{\partial c_i}{\partial x} + \frac{h^3}{3!}\frac{\partial^3 c_i}{\partial x^3} + \frac{h^5}{5!}\frac{\partial^5 c_i}{\partial x^5} + \cdots , \tag{11}$$

one can rewrite Eqs. (8) and (9) as:

$$\left[\partial_t + \frac{a}{h}\mathbf{N} - \frac{\beta|a|}{h}(\mathbf{M} - \mathbf{I})\right] c_0 = \left[\frac{\sqrt{3}a}{h}(\mathbf{M} - \mathbf{I}) - \frac{\sqrt{3}\beta|a|}{h}\mathbf{N}\right] c_1 , \tag{12}$$

$$\left[\partial_t - \frac{3a}{h}\mathbf{N} + \frac{3\beta|a|}{h}(\mathbf{M} + \mathbf{I})\right] c_1 = \left[-\frac{\sqrt{3}a}{h}(\mathbf{M} - \mathbf{I}) + \frac{\sqrt{3}\beta|a|}{h}\mathbf{N}\right] c_0 . \tag{13}$$

Noting that all the bracketed operators in Eqs. (12) and (13) are linear, one can perform a Gauss-like elimination by applying the left-hand side operator of Eq. (12) over Eq. (13) and vice versa. Then, after simple substitution, this procedure yields exactly the same PDE for both coefficients, namely

$$\begin{aligned} &\left[\partial_t + \frac{a}{h}\mathbf{N} - \frac{\beta|a|}{h}(\mathbf{M} - \mathbf{I})\right]\left[\partial_t - \frac{3a}{h}\mathbf{N} + \frac{3\beta|a|}{h}(\mathbf{M} + \mathbf{I})\right] c_i = \\ &= \left[\frac{\sqrt{3}a}{h}(\mathbf{M} - \mathbf{I}) - \frac{\sqrt{3}\beta|a|}{h}\mathbf{N}\right]\left[-\frac{\sqrt{3}a}{h}(\mathbf{M} - \mathbf{I}) + \frac{\sqrt{3}\beta|a|}{h}\mathbf{N}\right] c_i . \end{aligned} \tag{14}$$

We verified that these manipulations (including the Gauss-like elimination) can be used for higher values of P to provide a single PDE which governs the evolution of all coefficients. Accordingly, each coefficient evolves independently of the others. Also, the highest time derivative of such PDE will be of order $P + 1$. The role of high-order time derivatives can be better understood by recalling the wave equation,

$$\frac{\partial^2 u}{\partial t^2} - a^2\frac{\partial^2 u}{\partial x^2} = 0 \iff \left(\frac{\partial}{\partial t} + a\frac{\partial}{\partial x}\right)\left(\frac{\partial}{\partial t} - a\frac{\partial}{\partial x}\right) u = 0 , \tag{15}$$

whose solution is the sum of two signals travelling in opposite directions. Recognizing such behaviour in DG is not surprising, since the capability of supporting multiple solution modes is a common feature for spectral element discretizations.

Following this idea, one can factor out the single PDE derived above (consider Eq. (14) without c_i) by "solving" it for ∂_t. For the simpler case $\beta = 0$, the roots are

$$\partial_t = \frac{a}{h}\mathbf{N} \pm \frac{a}{h}\sqrt{4\mathbf{N}^2 - 3(\mathbf{M} - \mathbf{I})^2} , \tag{16}$$

which, by using the right-hand side of Eqs. (10) and (11), and then the expansion

$$f(z) = (y+z)^n = y^n + ny^{n-1}z + n(n-1)y^{n-2}\frac{z^2}{2!} + \cdots , \tag{17}$$

with $n = 1/2$, constant y and $z = z(h)$, yields

$$\partial_t = a\partial_x + \frac{ah^2}{6}\partial_x^3 + \frac{ah^4}{120}\partial_x^5 + \cdots \pm \left(2a\partial_x + \frac{7ah^2}{48}\partial_x^3 + \frac{121ah^4}{15360}\partial_x^5 + \cdots\right) . \tag{18}$$

Separating these two roots leads to the desired PDE in factored form, namely

$$\left(\frac{\partial}{\partial t} - 3a\frac{\partial}{\partial x} - \frac{5ah^2}{16}\frac{\partial^3}{\partial x^3} - \frac{83ah^4}{5120}\frac{\partial^5}{\partial x^5} - \frac{313ah^6}{688128}\frac{\partial^7}{\partial x^7} + \cdots\right)$$
$$\left(\frac{\partial}{\partial t} + a\frac{\partial}{\partial x} - \frac{ah^2}{48}\frac{\partial^3}{\partial x^3} - \frac{7ah^4}{15360}\frac{\partial^5}{\partial x^5} + \frac{599ah^6}{10321920}\frac{\partial^7}{\partial x^7} + \cdots\right) c_i = 0 . \tag{19}$$

The numerical solution can then be seen as the sum of two modes, one physical (lower bracket) and another one unphysical (upper bracket). These can be distinguished by the sign and magnitude of their advection speed. Regarding numerical accuracy, for this case ($\beta = 0$), all the truncation terms are of a dispersive nature, and the leading error term of the physical mode is of second order. Moreover, there is no point in discussing order of accuracy for the unphysical mode, since it simply does not approximate the advection equation being discretized.

Let us now consider the case $\beta \neq 0$. For this more general case, the same steps taken before can be adopted (though with much greater algebraic manipulation) to provide the modified equation, which can be compacted in factored form as

$$\left(\frac{\partial}{\partial t} - 3a\frac{\partial}{\partial x} + \frac{6\beta|a|}{h} + \beta|a|h\frac{\partial^2}{\partial x^2} - \frac{ah^2}{3}\frac{\partial^3}{\partial x^3} - \frac{|a|h^3}{\beta_3^{\ominus}}\frac{\partial^4}{\partial x^4} - \frac{ah^4}{\beta_4^{\ominus}}\frac{\partial^5}{\partial x^5} + \cdots\right)$$
$$\left(\frac{\partial}{\partial t} + a\frac{\partial}{\partial x} + \frac{|a|h^3}{72\beta}\frac{\partial^4}{\partial x^4} + \frac{ah^4}{\beta_4^{\oplus}}\frac{\partial^5}{\partial x^5} + \frac{|a|h^5}{\beta_5^{\oplus}}\frac{\partial^6}{\partial x^6} + \cdots\right) c_i = 0 \tag{20}$$

by using the constants $\beta_3^{\ominus} = [(72\beta)^{-1} - \beta/12]^{-1}$, $\beta_4^{\ominus} = [(108\beta^2)^{-1} + 1/90]^{-1}$, $\beta_4^{\oplus} = [(108\beta^2)^{-1} - 1/180]^{-1}$ and $\beta_5^{\oplus} = [(162\beta^3)^{-1} - (216\beta)^{-1}]^{-1}$.

Now, the leading error term for the physical mode is of a diffusive nature, has the expected sign (stabilizing for $\beta > 0$) and is of third order. Surprisingly, here the accuracy of the physical mode is increased via upwinding, though the opposite happens for the unphysical mode. This is advantageous since in the unphysical PDE,

$$\frac{\partial c_i}{\partial t} - 3a\frac{\partial c_i}{\partial x} = -\frac{6\beta|a|}{h}c_i - \beta|a|h\frac{\partial^2 c_i}{\partial x^2} + \frac{ah^2}{3}\frac{\partial^3 c_i}{\partial x^3} + \cdots , \tag{21}$$

the leading error term is proportional to c_i and plays the role of a singular energy drain (for $\beta > 0$). This term is also inversely proportional to h, growing without bound as $h \to 0$ while other truncation terms vanish. This grants exponential decay of the unphysical mode through mesh refinement alone. For example, by considering only the leading error term, the exact solution of the unphysical PDE would be

$$c_i(x,t) = c_i^o(x + 3at)\, \exp\left(-\frac{6\beta|a|}{h}\, t\right) , \tag{22}$$

where $c_i^o(x)$ is the initial condition for the unphysical mode. The above expression states that, for a given mesh, the unphysical mode will decay exponentially in time. Also, at a given instant t, the magnitude of the unphysical mode will decay exponentially as the mesh spacing is reduced.

4 Validation and Connections to ESA

To validate the results obtained so far, one can compare the information contained in truncation error terms with dispersion and diffusion curves provided by eigensolution analysis. Here, for the sake of brevity, these comparisons will be shown only for $\beta = 1$, which is the representative value for practical (stabilized) simulations.

When wave-like solutions in the form $c_j(x,t) = e^{ikx}e^{(r+is)t}$ are assumed for the modified PDEs, the real and imaginary parts of the modified wavenumber can be evaluated as $Real(k^*) = -s/a$ and $Imag(k^*) = r/a$, where a is the advection speed. After usual normalization, these can be compared directly with ESA results. However, only a finite number of truncation terms can be taken into account. The case $P = 0$ is shown in Fig. 1, where the relation between modified (k^*) and actual (k) wavenumbers is depicted for an increasing number of truncation terms considered.

It is observed that MEA-based curves approach the exact numerical eigencurves as more truncation terms are taken into account. Exact dispersion/diffusion curves for this case ($P = 0$, $\beta = 1$) are derived analytically in [6] and given by

$$\overline{k^*} = i\left[\exp\left(-i\,\overline{k}\right) - 1\right] , \tag{23}$$

where $\overline{k} = k\,h/(P+1)$ and $\overline{k^*} = k^*h/(P+1)$.

Now for the case $P = 1$ (with $\beta = 1$), one has the formula (again from [6])

$$\overline{k^*} = \frac{1}{2i}\left[2 + \exp\left(-2i\,\overline{k}\right)\right] \pm \frac{1}{2}\sqrt{2 - 10\exp\left(-2i\,\overline{k}\right) - \exp\left(-4i\,\overline{k}\right)} , \tag{24}$$

altogether for the physical (positive root sign) and unphysical (negative root sign) modes. The comparison for these modes are given respectively in Figs. 2 and 3.

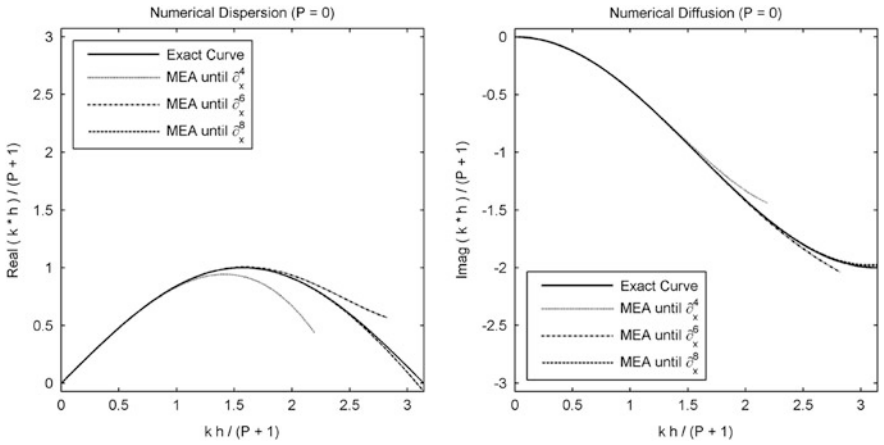


Fig. 1 Comparison between exact and MEA-based eigencurves (for $P = 0$, $\beta = 1$)

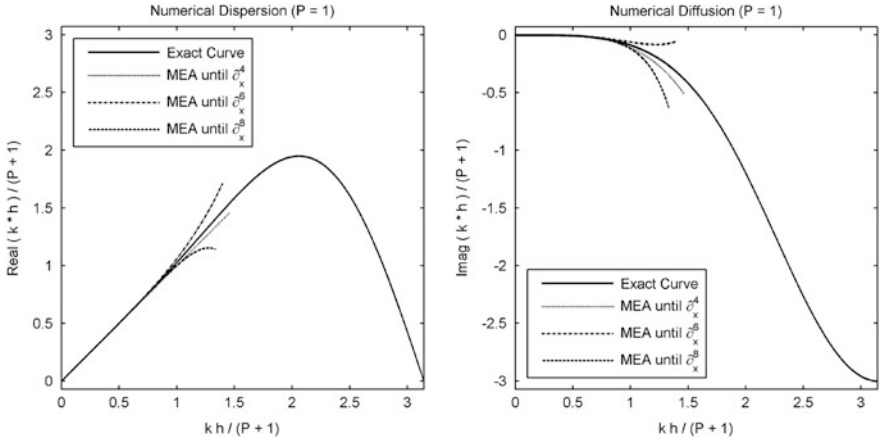


Fig. 2 Comparison between exact and MEA-based eigencurves (physical mode, $P = 1$, $\beta = 1$)

In this case, MEA-based eigencurves also converge to the exact numerical curves but only within a limited radius around $\bar{k} = 0$. It was then verified that the (analytically obtained) Taylor series expansion of Eq. (24) exhibits the same behaviour. Moreover, we found a one-to-one correspondence between the terms of such Taylor series and the truncation terms of the associated modified equation. This equivalence between MEA and ESA results (for linear advection) can be justified mathematically from ESA-based relations, but is omitted here for space considerations.

We remark that these are not issues of the MEA technique as applied here, but stem from a limited convergence radius of the Taylor expansion of the exact numerical dispersion-diffusion relations. Moreover, for well-resolved simulations

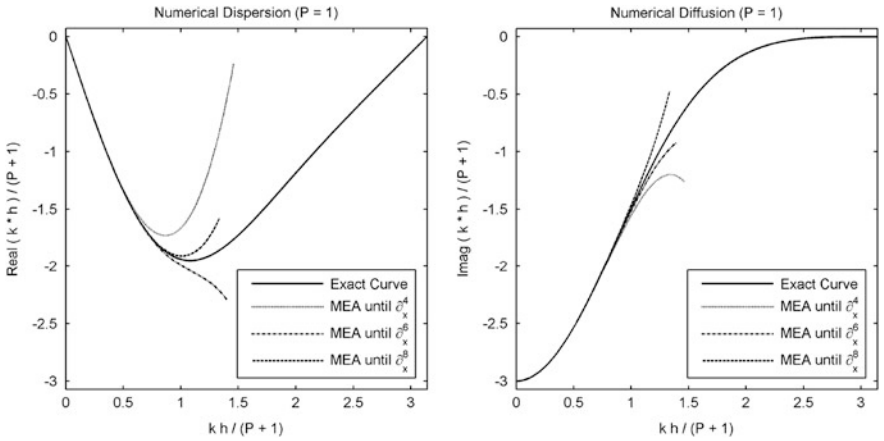


Fig. 3 Comparison between exact and MEA-based eigencurves (unphysical mode, $P = 1, \beta = 1$)

($\overline{k} \ll \pi$), these issues are not really important since only the first few truncation terms are of interest. If these are predicted correctly, the eigencurves' behaviour near $\overline{k} = 0$ is being captured, and that is sufficient to analyse stability, order of accuracy and even exponential damping of unphysical modes. On the other hand, for under-resolved simulations, convergence issues are indeed important and should not be neglected.

5 Concluding Remarks

The present study discussed how to apply the modified equation analysis (MEA) technique to the discontinuous Galerkin (DG) formulation. While focusing on linear advection, this paper presented a complementary view on wave-propagation properties for DG methods. For stabilized simulations ($\beta > 0$) in particular, orders of accuracy of 1 and 3 were verified for the modified equations of physical modes respectively for $P = 0$ and $P = 1$. Such results are consistent with previous works on wave propagation for DG [1, 2], where order of accuracy of $2P + 1$ has been shown.

A connection between MEA and eigensolution analysis (ESA) was also pointed out, so that both approaches can be considered to provide essentially the same information (for linear advection). However, while ESA is restricted to linear problems, the usage of MEA normally extends to non-linear problems. An interesting application of MEA to such problems would be the assessment of DG's suitability for implicit LES. In e.g. [5], MEA's application to a specific finite volume scheme revealed similarities between its truncation terms and mixed subgrid-scale models. However, the results obtained in Sect. 4 do not encourage this approach for DG. The

convergence issues verified for under-resolved simulations are unlikely to disappear for discretizations of higher order or when considering non-linear problems.

References

1. M. Ainsworth, Dispersive and dissipative behaviour of high order discontinuous Galerkin finite element methods. J. Comp. Phys. **198**, 106–130 (2004)
2. H.L. Atkins, Super-convergence of discontinuous Galerkin method applied to the Navier-Stokes equations. AIAA Paper 2009–3787 (2009)
3. F.Q. Hu, M.Y. Hussaini, P. Rasetarinera, An analysis of the discontinuous Galerkin method for wave propagation problems. J. Comp. Phys. **151**, 921–946 (1999)
4. G.E. Karniadakis, S.J. Sherwin, *Spectral/hp Element Methods for Computational Fluid Dynamics*, 2nd edn. (Oxford University Press, Oxford, 2005)
5. L.G. Margolin, W.J. Rider, A rationale for implicit turbulence modelling. Int. J. Numer. Meth. Fluids **39**, 821–841 (2002)
6. S.J. Sherwin, Dispersion analysis of the continuous and discontinuous Galerkin formulations, in *Discontinuous Galerkin Methods: Theory, Computation and Applications*, ed. by B. Cockburn, G. Karniadakis, C.W. Shu (Springer, New York, 2000), pp. 425–431
7. M. Zhang, C.W. Shu, An analysis of and a comparison between the discontinuous Galerkin and the spectral finite volume methods. Comp. Fluids **34**, 581–592 (2005)

Fully Discrete Energy Stable High Order Finite Difference Methods for Hyperbolic Problems in Deforming Domains

Samira Nikkar and Jan Nordström

Abstract A time-dependent coordinate transformation of a constant coefficient hyperbolic system of equations is considered. We use the energy method to derive well-posed boundary conditions for the continuous problem. Summation-by-Parts (SBP) operators together with a weak imposition of the boundary and initial conditions using Simultaneously Approximation Terms (SATs) guarantee energy-stability of the fully discrete scheme. We construct a time-dependent SAT formulation that automatically imposes the boundary conditions, and show that the numerical Geometric Conservation Law (GCL) holds. Numerical calculations corroborate the stability and accuracy of the approximations. As an application we study the sound propagation in a deforming domain using the linearized Euler equations.

1 Introduction

High order SBP-SAT schemes, can efficiently and reliably handle large problems on structured grids for reasonably smooth geometries [7, 11]. The developments within this framework, have so far dealt mostly with steady meshes while computing flow-fields around moving and deforming objects involves time-dependent meshes [3, 12]. In this paper (and the full paper [5]) we treat the time-dependent transformations in a SBP-SAT framework. To guarantee stability of the fully discrete approximation we employ the recently developed SBP-SAT technique in time [4, 8].

S. Nikkar (✉) • J. Nordström
Linköping University, SE-581 83 Linköping, Sweden
e-mail: samira.nikkar@liu.se; jan.nordstrom@liu.se

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_35

2 The Continuous Problem

The following hyperbolic symmetrized constant coefficient system,

$$V_t + (\hat{A}V)_x + (\hat{B}V)_y = 0,\ (x,y) \in \Phi(t),\ t \in [0,\ T], \tag{1}$$

can, with the use of the GCL [3], be rewritten as

$$\begin{aligned} (JV)_\tau + (AV)_\xi + (BV)_\eta &= 0, && (\xi,\eta) \in \Omega,\ \ \tau \in [0,\ T], \\ LV &= g(\tau,\xi,\eta), && (\xi,\eta) \in \delta\Omega,\ \tau \in [0,\ T], \\ V &= f(\xi,\eta), && (\xi,\eta) \in \Omega,\ \ \tau = 0, \end{aligned} \tag{2}$$

through a time-dependent transformation from the Cartesian coordinates into curvilinear coordinates as

$$x(\tau,\xi,\eta) \rightleftharpoons \xi(t,x,y),\ y(\tau,\xi,\eta) \rightleftharpoons \eta(t,x,y),\ t = \tau. \tag{3}$$

In (2), $A = J\xi_t I + J\xi_x\hat{A} + J\xi_y\hat{B},\ B = J\eta_t I + J\eta_x\hat{A} + J\eta_y\hat{B}$, and $\Omega = [0,\ 1] \times [0,\ 1]$. Moreover, L is the boundary operator, g is the boundary data, f is the initial data, and $J = x_\xi y_\eta - x_\eta y_\xi > 0$ is the determinant of the Jacobian of the transformation.

2.1 Well-Posedness

The energy method (multiply (2) with the transpose of the solution and integrate over the domain Ω and time-interval $[0,\ T]$) is applied to (2), and the term $V_\tau^T JV + V_\xi^T AV + V_\eta^T BV = 0$ is added to the integral argument. Then, integration together with the use of Green-Gauss theorem gives

$$||V(T,\xi,\eta)||_J^2 = ||f(\xi,\eta)||_J^2 - \int_0^T \oint_{\delta\Omega} V^T[(A,B)\cdot n]\ V\ ds\ d\tau, \tag{4}$$

where the norm is defined by $||V||_J^2 = \iint_\Omega V^T J\ V\ d\xi\ d\eta$. In (4), n is the unit normal pointing outward from Ω, and ds is an infinitesimal element along the boundary, $\delta\Omega$.

In order to bound the energy of the solution, boundary conditions must be applied when the matrix $C = (A,B)\cdot n$ is negative definite. We decompose $C = X\Lambda_C X^T = X\Lambda_C^+ X^T + X\Lambda_C^- X^T = C^+ + C^-$ where Λ_C^+ and Λ_C^- are diagonal matrices with positive and negative eigenvalues of C, respectively. We choose the characteristic boundary conditions, in order to bound the energy of the solution as

$$(X^T V)_i = (X^T V_\infty)_i,\ (\Lambda_C)_{ii} < 0, \tag{5}$$

where V_∞ is the solution at $\delta\Omega$.

The continuous energy, using (5) is estimated as

$$||V(T,\xi,\eta)||_J^2 = ||f(\xi,\eta)||_J^2 - \int_0^T \oint_{\delta\Omega} V_\infty^T C^- V_\infty \, ds \, d\tau - \int_0^T \oint_{\delta\Omega} V^T C^+ V \, ds \, d\tau. \tag{6}$$

The estimate (6) guarantees uniqueness of the solution and existence is given by the fact that we use the correct number of boundary conditions. Hence we can summarize the results obtained so far in the following proposition.

Proposition 1 *The continuous problem (2) with the boundary condition in (5) is strongly well-posed and has the bound (6).*

3 The Discrete Problem

The spatial domain, Ω, is a square in ξ, η coordinates, and discretized using N and M nodes in ξ and η directions respectively. In time we use L time levels from 0 to T.

The first derivative u_ξ is approximated by $D_\xi \mathbf{u}$, where D_ξ is a so-called SBP operator, see [10]. A multi-dimensional finite difference approximation (including the time discretization [4, 8]), on SBP-SAT form, is constructed by extending the one-dimensional SBP operators in a tensor product fashion as

$$D_\tau = P_\tau^{-1} Q_\tau \otimes I_\xi \otimes I_\eta \otimes I, \; D_\xi = I_\tau \otimes P_\xi^{-1} Q_\xi \otimes I_\eta \otimes I, \; D_\eta = I_\tau \otimes I_\xi \otimes P_\eta^{-1} Q_\eta \otimes I \tag{7}$$

where $\otimes$ represents the Kronecker product [14]. In (7), I denotes the identity matrix with a size consistent with its position in the Kronecker product. In [5] it is shown that the operators in (7) commute.

The SBP-SAT approximation of (2) including the penalty terms for the weak boundary conditions (we only consider the boundary along which $\eta = 0$, namely the south boundary, denoted by subscript s), and a weak initial condition, is constructed as

$$\begin{aligned} &\tfrac{1}{2}[D_\tau(\mathbf{JV}) + \mathbf{J}D_\tau\mathbf{V} + \mathbf{J}_\tau\mathbf{V}] + \tfrac{1}{2}[D_\xi(\mathbf{AV}) + \mathbf{A}D_\xi\mathbf{V} + \mathbf{A}_\xi\mathbf{V}] + \\ &\tfrac{1}{2}[D_\eta(\mathbf{BV}) + \mathbf{B}D_\eta\mathbf{V} + \mathbf{B}_\eta\mathbf{V}] = \tilde{P}_i^{-1}\Sigma_i(\mathbf{V}-\mathbf{f}) + \tilde{P}_s^{-1}\Sigma_s\mathbf{X}_s^T[\mathbf{V} - V_\infty], \end{aligned} \tag{8}$$

in which the bold face of the variables corresponds to the approximated values. Σ_i and Σ_s are the penalty matrices for the weak initial condition and the south boundary procedure. Furthermore $\tilde{P}_i^{-1} = P_\tau^{-1}E_0 \otimes I_\xi \otimes I_\eta \otimes I$, $\tilde{P}_s^{-1} = I_\tau \otimes I_\xi \otimes P_\eta^{-1}E_0 \otimes I$, and $\mathbf{X}_s = (I_\tau \otimes I_\xi \otimes E_0 \otimes X)$. Also, the vectors V_∞ and $\mathbf{f}$ contain the boundary data at $\eta = 0$ and initial data at $\tau = 0$ respectively. Note that in (8), the splitting

technique described in [6] is used prior to the discretizations, in order to get similar energy estimate as the one in the continuous case.

3.1 Stability

The energy method (multiplying from the left with $\mathbf{V}^T(P_\tau \otimes P_\xi \otimes P_\eta \otimes I)$) is applied to (8) and the equation is added to its transpose. The result is

$$\begin{aligned}&\mathbf{V}^T(\tilde{B}_\tau \mathbf{J} + \tilde{B}_\xi \mathbf{A} + \tilde{B}_\eta \mathbf{B})\mathbf{V} + \mathbf{V}^T\tilde{P}(\mathbf{J}_\tau + \mathbf{A}_\xi + \mathbf{B}_\eta)\mathbf{V} = \\ &\mathbf{V}^T(E_0 \otimes P_\xi \otimes P_\eta \otimes I)\Sigma_i(\mathbf{V}-\mathbf{f}) + (\mathbf{V}-\mathbf{f})^T\Sigma_i^T(E_0 \otimes P_\xi \otimes P_\eta \otimes I)\mathbf{V} + \\ &\mathbf{V}^T(P_\tau \otimes P_\xi \otimes E_0 \otimes I)\Sigma_s\mathbf{X}_s^T[\mathbf{V}-\mathbf{V}_\infty] + [\mathbf{V}-\mathbf{V}_\infty]^T\mathbf{X}_s\Sigma_s^T(P_\tau \otimes P_\xi \otimes E_0 \otimes I)\mathbf{V},\end{aligned} \tag{9}$$

where $\tilde{P} = (P_\tau \otimes P_\xi \otimes P_\eta \otimes I)$, $\tilde{B}_\tau = [(Q + Q^T)_\tau \otimes P_\xi \otimes P_\eta \otimes I]$, $\tilde{B}_\xi = [P_\tau \otimes (Q + Q^T)_\xi \otimes P_\eta \otimes I]$, and $\tilde{B}_\eta = [P_\tau \otimes P_\xi \otimes (Q + Q^T)_\eta \otimes I]$. The following Lemma is proved in [2].

Lemma 1 *The numerical GCL holds:* $\mathbf{J}_\tau + \mathbf{A}_{\boldsymbol{\xi}} + \mathbf{B}_{\boldsymbol{\eta}} = 0$.

In (9), by using Lemma 1 we get

$$\begin{aligned}&\mathbf{V}_T^{\mathbf{T}}\mathbf{J}(E_L \otimes P_{\xi\eta} \otimes \otimes I)\mathbf{V} = \mathbf{V}^T(E_0 \otimes P_{\xi\eta} \otimes I)(\mathbf{J} + 2\Sigma_i)\mathbf{V} - \mathbf{f}^T(E_0 \otimes P_{\xi\eta} \otimes I)\Sigma_i\mathbf{V} - \\ &\mathbf{V}^T(E_0 \otimes P_{\xi\eta} \otimes I)\Sigma_i\mathbf{f} + \mathbf{V}^T(P_{\tau,\xi} \otimes E_0 \otimes I)(\mathbf{B_s} + \Sigma_s\mathbf{X}_s^T + \mathbf{X}_s\Sigma_s^T)V - \\ &\mathbf{V}^T(P_{\tau\xi} \otimes E_0 \otimes I)\Sigma_s\mathbf{X}_s^T(\mathbf{V}_\infty)_s - (\mathbf{V}_\infty)_s^T\mathbf{X}_s\Sigma_{bs}^T(P_{\tau\xi} \otimes E_0 \otimes I)\mathbf{V},\end{aligned} \tag{10}$$

where $P_{\xi\eta} = P_\xi \otimes P_\eta$, $P_{\tau\xi} = P_\tau \otimes P_\xi$, $\mathbf{B}_s = (I_\tau \otimes I_\xi \otimes E_0 I_\eta \otimes I)\mathbf{B}$, and E_0, E_L are zero matrices except at the one entry corresponding to the initial and final time, respectively.

Proposition 2 *The problem (8) is stable if* $\mathbf{J} + 2\Sigma_i \leq 0,\ \Sigma_s\mathbf{X}_s^T + \mathbf{X}_s\Sigma_s^T + \mathbf{B}_s \leq 0$.

Proof With zero boundary and initial data the solution at the final time is clearly bounded. □

4 Numerical Experiments

We consider the two-dimensional linearized symmetrized Euler equations in a deforming domain described by (1), where $V = [\bar{c}\rho/(\sqrt{\gamma}\bar{\rho}), u, v, T/(\bar{c}\sqrt{\gamma(\gamma-1)})]^T$, and ρ, u, v, T and γ are respectively the density, the velocity components in x and y directions, the temperature and the ratio of specific heats [1, 13]. An equation of

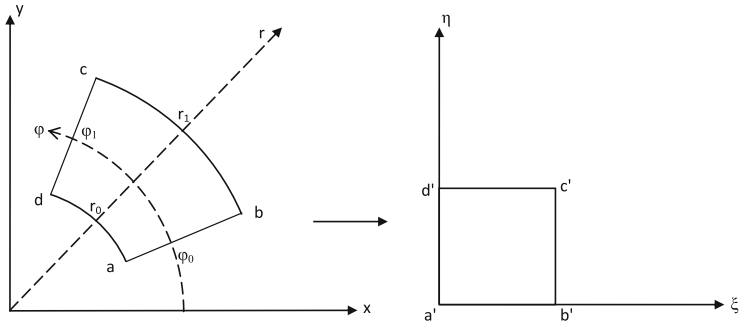


Fig. 1 A schematic of the Cartesian-polar transformation, and illustrations of r_0, r_1, ϕ_0 and ϕ_1; Also boundary definitions as West: $ad \rightarrow a'd'$, East: $bc \rightarrow b'c'$, South: $ab \rightarrow a'b'$, North: $dc \rightarrow d'c'$

state in form of $\gamma p = \bar{\rho} T + \rho \bar{T}$ closes the system (1), in which the bar denotes the state around which we have linearized. Moreover the matrices in (1) are

$$\hat{A} = \begin{pmatrix} \bar{u} & \bar{c}/\sqrt{\gamma} & 0 & 0 \\ \bar{c}/\sqrt{\gamma} & \bar{u} & 0 & \sqrt{\frac{\gamma-1}{\gamma}}\bar{c} \\ 0 & 0 & \bar{u} & 0 \\ 0 & \sqrt{\frac{\gamma-1}{\gamma}}\bar{c} & 0 & \bar{u} \end{pmatrix}, \hat{B} = \begin{pmatrix} \bar{v} & 0 & \bar{c}/\sqrt{\gamma} & 0 \\ 0 & \bar{v} & 0 & 0 \\ \bar{c}/\sqrt{\gamma} & 0 & \bar{v} & \sqrt{\frac{\gamma-1}{\gamma}}\bar{c} \\ 0 & 0 & \sqrt{\frac{\gamma-1}{\gamma}}\bar{c} & \bar{v} \end{pmatrix}. \tag{11}$$

The deforming domain is chosen to be a portion of a ring-shaped geometry where the boundaries are moving while always coinciding with a coordinate line in the corresponding polar coordinate system. We transform the deforming domain from Cartesian coordinates, x, y, into polar coordinates, r, ϕ, and scale the polar coordinates such that $\Omega = [0,\ 1] \times [0,\ 1]$, see Fig. 1, as

$$\xi(x, y, t) = \frac{r(x,y,t) - r_0(t)}{r_1(t) - r_0(t)},\ \eta(x, y, t) = \frac{\phi(x,y,t) - \phi_0(t)}{\phi_1(t) - \phi_0(t)}. \tag{12}$$

4.1 Order of Accuracy

We move the boundaries by the transformation

$$\begin{aligned} r_0(t) &= 1 - \tfrac{0.1}{2\pi}\sin(2\pi t),\ \phi_0(t) = -\tfrac{0.5}{2\pi}\sin(2\pi t), \\ r_1(t) &= 2 + \tfrac{0.2}{2\pi}\sin(2\pi t),\ \phi_1(t) = \tfrac{\pi}{2} + \tfrac{0.5}{2\pi}\sin(2\pi t), \end{aligned} \tag{13}$$

Table 1 Convergence rates at T = 1, for a sequence of mesh refinements, SBP63 in space, SBP84 in time (L = 201)

N, M	21	31	41	51	61	71
ρ	5.780	4.681	4.502	4.379	4.320	4.296
u	6.120	4.531	4.585	4.588	4.575	4.558
v	6.138	4.300	4.179	4.215	4.249	4.268
p	5.701	4.124	4.267	4.340	4.380	4.402

and construct the matrices $\hat{A}$ and $\hat{B}$ for a state where $\bar{u} = 1$, $\bar{v} = 1$, $\bar{\rho} = 1$, $\bar{\gamma} = 1.4$ and $\bar{c} = 2$. To verify the order of accuracy of our method, we use the method of manufactured solution [9], and impose the characteristic boundary conditions as derived in (5).

The numerical solution for a scheme with *SBP*63 in space and *SBP*84 in time, converges to the exact solution at T = 1 with the convergence rate presented in Table 1. Moreover, the scheme is tested with *SBP*21 and *SBP*42 and the convergence rates are quantified as 2 and 3 respectively [5].

4.2 The Sound Propagation Application

We consider a deforming domain where the west boundary is moving, see Figs. 2 and 3. Note that these schematics are for illustration purposes only, the numerical experiments are carried out on finer meshes. The movements are defined by

$$\begin{aligned} &r_0(t) = 1 + \sin(4\pi t)/(4\pi),\ \phi_0(t) = \pi/4, \\ &r_1(t) = 5,\ \phi_1(t) = 3\pi/4. \end{aligned} \tag{14}$$

We choose $\gamma = 1.4$, $\bar{c} = 2$, $\bar{\rho} = 1$ and manufacture $\bar{u}$ and $\bar{v}$ such that the mean flow satisfies the solid wall no-penetration condition at the moving boundary by

$$\bar{u} = x_\tau / \exp(\xi),\ \bar{v} = y_\tau / \exp(\xi). \tag{15}$$

Consider the eigenvalue matrix, $C = X\Lambda X^T$ at the west boundary, in which $\Lambda = \mathscr{R}_1 \operatorname{diag}(\hat{\omega},\ \hat{\omega},\ \hat{\omega} - \bar{c},\ \hat{\omega} + \bar{c})$, where $\hat{\omega} = -(J\xi_t + J\xi_x\bar{u}_b + J\xi_y\bar{v}_b)/\mathscr{R}_1$ and $\mathscr{R}_1 = \sqrt{(J\xi_x)^2 + (J\xi_y)^2}$. The no-penetration condition for the mean flow at the moving boundary results in $\hat{\omega} = 0$, which takes (6) to

$$||V(T,\ \xi,\ \eta)||_J^2 = ||f(\xi,\ \eta)||^2 - \int_0^T \int_0^1 \bar{c}(\tilde{v}_4^2 - \tilde{v}_3^2)\, d\eta + \text{BT}. \tag{16}$$

In (16), $\tilde{V} = X^T V = [\tilde{v}_1,\ \tilde{v}_2,\ \tilde{v}_3,\ \tilde{v}_4]^T$, and *BT* is the contribution at the other boundaries. Any boundary condition of the form $\tilde{v}_3 = \pm\tilde{v}_4$ is well-posed. We choose $\tilde{v}_3 + \tilde{v}_4 = 0$, which is the no-penetration boundary condition. Also we impose characteristic boundary conditions with zero data at the other boundaries,

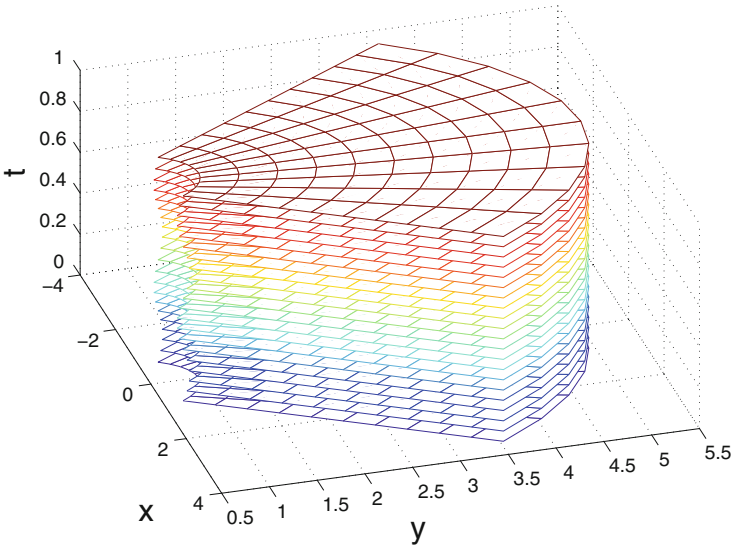


Fig. 2 A schematic of the deforming mesh at different times, sound propagation

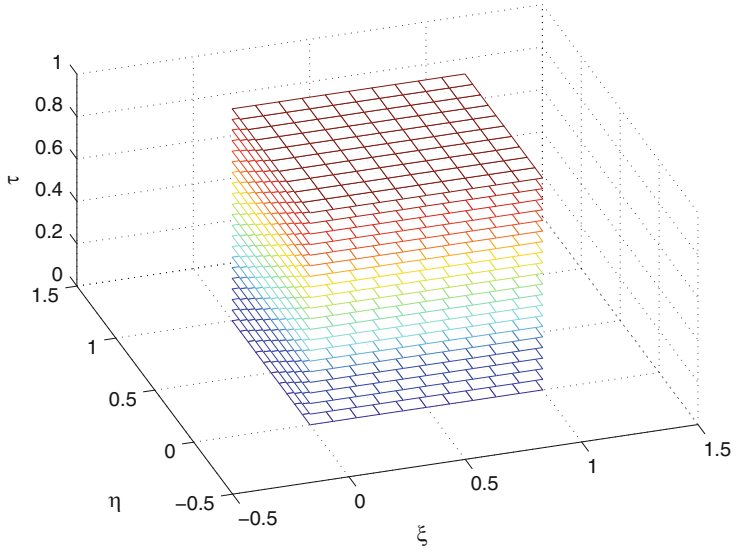


Fig. 3 A schematic of the fixed mesh at different times, sound propagation

and initialize the solution with zero data for density and velocities, together with an initial pressure pulse centered at $(-1.5,\ 3.5)$. We have used $N = M = 50, L = 100$ and SBP42 in space and time. The velocity field at two different time levels, with non-penetrating flow close to the solid wall, are presented in Figs. 4, 5, 6, and 7.

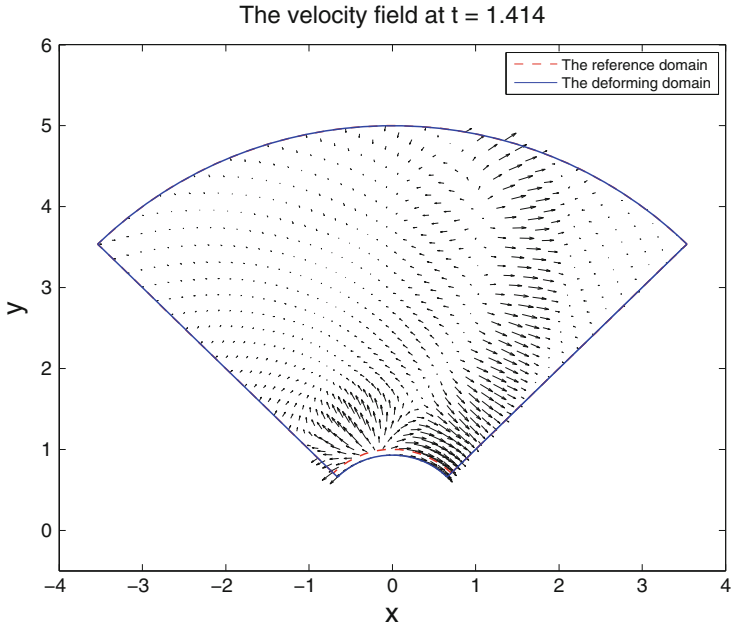


Fig. 4 The global velocity field

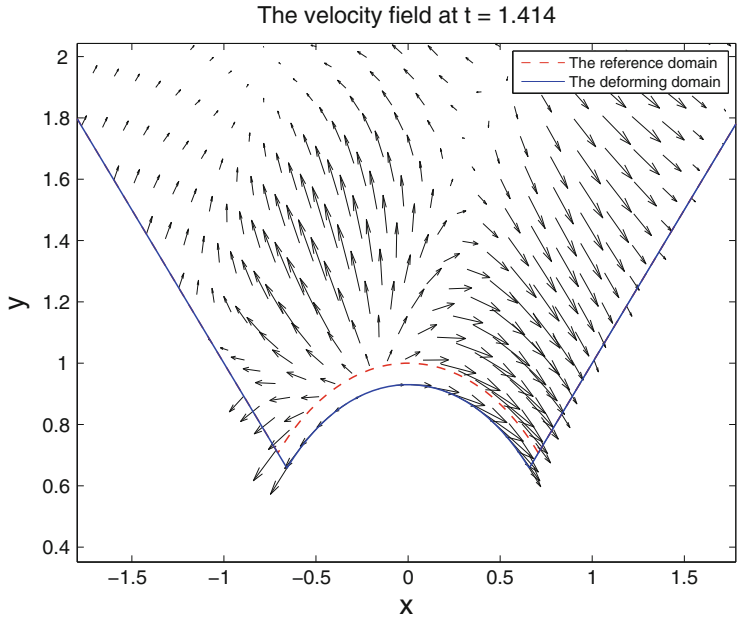


Fig. 5 A blow-up of the velocity field.

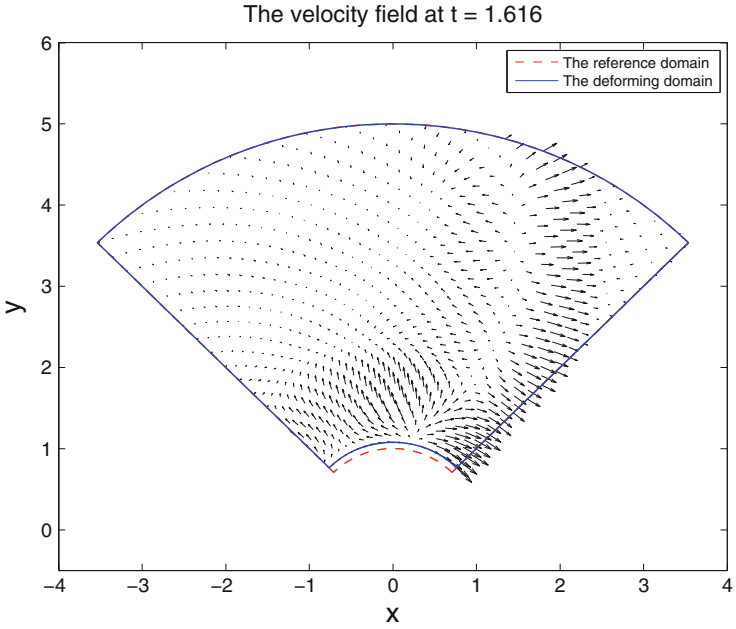


Fig. 6 The global velocity field

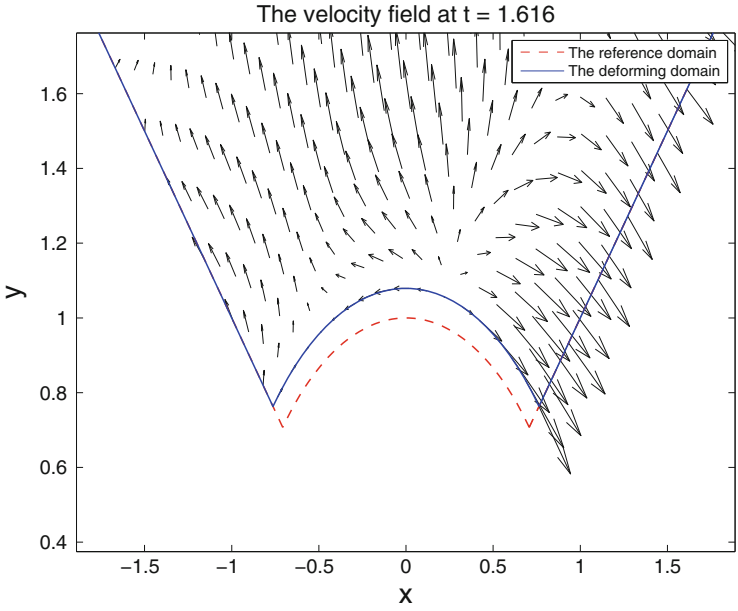


Fig. 7 A blow-up of the velocity field

The reference domain in Figs. 4, 5, 6, and 7 illustrate the movements of the south boundary relative to its initial location. As seen in the figures, the flow stays tangential to the moving solid boundary all the time, as it should for an Euler solution.

5 Summary and Conclusions

We have considered a constant coefficient hyperbolic system of equations in time-dependent curvilinear coordinates. The system is transformed into a fixed coordinate frame, resulting in variable coefficient system. We show that the energy method applied to the transformed systems together with time-dependent appropriate boundary conditions leads to strongly well-posed problem.

By using a special splitting technique, summation-by-parts operators in space and time, weak imposition of the boundary and initial conditions and the discrete energy method, a fully-discrete strongly stable and high order accurate numerical scheme is constructed. The fully-discrete energy estimate is similar to the continuous one with small added damping terms. Furthermore, by the use of SBP operators in time, the Geometric Conservation Law is shown to hold numerically.

We have tested the scheme for high order accurate SBP operators in space and time using the method of manufactured solution. Numerical calculations corroborate the stability and accuracy of the fully-discrete approximations. Finally, as an application, sound propagation by the linearized Euler equations in a deforming domain is illustrated.

References

1. S. Abrabanel, D. Gottleib, Optimal time splitting for two- and three-dimensional Navier-Stokes equations with mixed derivatives. J. Comput. Phys. **41**, 1–43 (1981)
2. Y. Abe, N. Izuka, T. Nonomura, K. Fuji, Symmetric-conservative metric evaluations for higher-order finite difference scheme with the GCL identities on the three-dimensional moving and deforming mesh, in *ICCFD7*, (2012)
3. C. Farhat, P. Geuzaine, C. Grandmont, The discrete geometric conservation law and the nonlinear stability of ALE schemes for the solution of flow problems on moving grids. J. Comput. Phys. **174**, 669–694 (2001)
4. T. Lundquist, J. Nordström, The SBP-SAT technique for initial value problems. J. Comput. Phys. **270**, 86–104 (2014)

5. S. Nikkar, J. Nordström, Fully discrete energy stable high order finite difference methods for hyperbolic problems in deforming domains, LiTH- MAT-R, 2014:15, Department of Mathematics, Linköping University, 2014
6. J. Nordström, Conservative finite difference formulations, variable coefficients, energy estimates and artificial dissipation. J. Sci. Comput. **29**, 375–404 (2006)
7. J. Nordström, H. Carpenter, High-order finite difference methods, multidimensional linear problems and curvilinear coordinates. J.Comput. Phys. **173**, 149–174 (2001)
8. J. Nordström, T. Lundquist, Summation-by-parts in time. J. Comput. Phys. **251**, 487–499 (2013)
9. K. Salari, Code verification by the method of manufactured solutions. doi:10.2172/759450
10. B. Strand, Summation by parts for finite difference approximations of d/dx. J. Comput. Phys. **110**, 47–67 (1994)
11. M. Svärd, J. Nordström, A stable high-order finite difference scheme for the compressible Navier Stokes equations: no-slip wall boundary conditions. J. Comput. Phys. **227**(10), 4805–4824 (2008)
12. P.D. Thomas, C.K. Lombard, Geometric conservation law and its application to flow computations on moving grids. AIAA J. **17**, 1030–1037 (1979).
13. E. Turkel, Symmetrization of the fluid dynamics matrices with applications. Math. Comput. **27**, 729–736 (1973)
14. C.F. Van Loan, The ubiquitous Kronecker product. J. Comput. Appl. Math. **123**, 85–100 (2000)

Stabilized Spectral Element Approximation of the Saint Venant System Using the Entropy Viscosity Technique

R. Pasquetti, J.L. Guermond, and B. Popov

Abstract We consider the Saint Venant system (shallow water equations), i.e. an approximation of the incompressible Euler equations widely used to describe river flows, flooding phenomena or erosion problems. We focus on problems involving dry-wet transitions and propose a solution technique using the Spectral Element Method (SEM) stabilized with a variant of the Entropy Viscosity Method (EVM) that is adapted to treat dry zones.

1 Introduction

Because high-order methods are known to produce spurious oscillations in shocks, solving non-linear hyperbolic systems of conservation equations with high accuracy is a challenging task. Assuming that an entropy does exist for the considered physical problem, the Entropy Viscosity Method (EVM) offers an elegant way to stabilize various numerical discretizations, including the standard Finite Element Method or Spectral Element Method (SEM) and even Fourier expansions [4]. The basic idea consists of introducing in the governing equations a nonlinear viscous term based on the residual of the Partial Differential Equation (PDE) that governs the evolution of the entropy and to bound from above this term by a first order viscosity.

We consider in the present paper the Saint Venant system, i.e. a simplified form of the incompressible Euler equations well adapted to describe free surface flows like rivers or flooding phenomena. We especially focus on problems involving dry-wet transitions, e.g. the classical dram break problem. This class of problems is generally addressed in the finite volume literature by using Godunov-type methods, i.e. Riemann solvers together with flux or slope limiters, see e.g. [6] for a review.

R. Pasquetti (✉)
Laboratoire J.A. Dieudonné, UMR CNRS 7351 & INRIA Project CASTOR, University of Nice-Sophia Antipolis, Parc Valrose, 06108 Nice Cedex 2, France
e-mail: richard.pasquetti@unice.fr

J.L. Guermond • B. Popov
Department of Mathematics, Texas A&M University, College Station, 77843 TX, USA
e-mail: guermond/popov@math.tamu.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_36

We introduce a new ingredient in the EVM that enables the method to handle the dry-wet transition problem satisfactorily. The numerical discretization is based on the SEM in space and on a standard forth order Runge-Kutta (RK4) scheme in time. Although all the numerical simulations shown in the paper are one-dimensional, the method is a priori multi-dimensional. Finally, the proposed approach can be used to treat problems in gas dynamics with vacuum.

The paper is organized as follows. We introduce the Saint Venant system and recall its basic properties in Sect. 2. The SEM approximation and the EVM stabilization are described and discussed in Sect. 3. Some examples of applications, all of them involving dry-wet transitions, are presented in Sect. 4.

2 The Saint Venant System

The Saint-Venant system (shallow water equations) is an approximation of the incompressible Euler equations assuming that the pressure is hydrostatic and the free surface perturbations are small compared to the water height. The one-dimensional version of this system is

$$\partial_t h + \partial_x(hu) = 0 \tag{1}$$

$$\partial_t(hu) + \partial_x(hu^2 + gh^2/2) + gh\partial_x z = 0, \tag{2}$$

where $h(x,t)$ is the water height, $u(x,t)$ the horizontal velocity, g the gravity acceleration, $z(x)$ the topography, for which it is assumed that $\partial_x z \ll 1$. The independent variables are time $t \in (0, t_F)$ and space $x \in D = (x_{\text{inf}}, x_{\text{sup}})$. These PDEs are obtained by integrating the mass and momentum conservation equations in the Euler system over the vertical direction. This nonlinear two equations system has the following properties:

- The system is hyperbolic, which means that discontinuities may develop;
- Assuming that the inlet flow-rate equals the outlet flow-rate, the total mass is preserved: $d_t \int_D h\, dx = 0$;
- The height h is nonnegative: $\forall x, t,\ h(x,t) \geq 0$;
- Rest solutions are stable: $u = 0, \quad h(x,t) + z(x) = constant$;
- There exists a convex entropy (actually the energy E) such that:

$$\partial_t E + \partial_x((E + gh^2/2)u) \leq 0, \quad E = hu^2/2 + gh^2/2 + ghz. \tag{3}$$

3 Stabilized SEM Approximation

The EVM-stabilization is obtained via the introduction of nonlinear viscous terms in the governing equations. The *entropy viscosity* is computed from the residual of the entropy inequality and bounded from above by a first order viscosity. In case

of a scalar conservation law, with δx for the grid size, we generally set, see [4] for details:

$$\nu = \mathcal{S}(\min(\nu_{max}, \nu_E)) \quad \text{where} \tag{4}$$

$$\nu_{max} = \alpha \max_{loc} |\boldsymbol{f}'(u)| \delta x \tag{5}$$

$$\nu_E = \beta \delta x^2 |r_E| / \Delta E \tag{6}$$

where r_E is the residual of the entropy inequality; $\boldsymbol{f}(u)$,$\boldsymbol{f}'(u)$ are the flux and derivative of the flux; α and β are user defined parameters; ΔE is a scaling parameter equal to the amplitude of variations of the entropy. The local maximum is generally based on the computational cell. $\mathcal{S}$ is a smoothing operator required by the fact that at the discrete level the residual r_E is oscillatory. For hyperbolic systems $\boldsymbol{f}'(u)$ is the Jacobian matrix of $\boldsymbol{f}$, and $|\boldsymbol{f}'|$ is defined to be the absolute value of the largest eigenvalue of $\boldsymbol{f}'(u)$.

Discretization of the Saint Venant system: Set $q = hu$ and, for any t, let $h_N(x,t)$ (resp. $q_N(x,t)$) to be the continuous piecewise polynomial approximation of degree N of $h(x,t)$ (resp. $q(x,t)$) built on a discretization of $D = (x_{\text{inf}}, x_{\text{sup}})$; i.e. we use the standard SEM for the space approximation, see e.g. [5] . Then we propose the following EVM-stabilized weak formulation of the Saint Venant system:

$$\int_D (\partial_t h_N + \partial_x q_N) v_N = -\int_D \nu \partial_x h_N \, \partial_x v_N \tag{7}$$

$$\int_D (\partial_t q_N + \partial_x (q_N^2/h_N + g h_N^2/2) + g h_N z_x) w_N = -\int_D \nu \partial_x q_N \, \partial_x w_N, \tag{8}$$

where v_N, w_N are test functions spanning the approximation space and ν is the entropy viscosity, still to be defined. As usual, the viscous (stabilization) terms have been integrated by parts. Note that a viscous stabilization is added to the mass equation and that the stabilization is done on q instead of u in the momentum equation. This differs from the physically and mathematically well justified viscous form of the Saint-Venant system, which makes only use of $\partial_x(h\nu\partial_x u)$ in the momentum equation [2]. In [3], where the Euler system is addressed, it is however outlined that the physical stabilization may not be the best suited one for numerical purposes.

Time is approximated using an explicit RK4 scheme.

Entropy viscosity for the Saint-Venant system: First we define the viscosity ν_E associated to the residual of the entropy equation. Using the expression (3) leads to a viscosity ν_E that depends on z, i.e. on the choice of the coordinate system. To avoid this arbitrariness, we take into account the mass conservation equation in (3) to derive an expression that only depends on $\partial_x z$ and governs the evolution of an entropy $\tilde{E}$ which satisfies:

$$\partial_t \tilde{E} + \partial_x((\tilde{E} + g h^2/2)u) + g h u \partial_x z \leq 0, \quad \tilde{E} = h u^2/2 + g h^2/2. \tag{9}$$

The evaluation of the entropy viscosity is done at each time step before entering the RK explicit time scheme. This is done at time t_n by using a Backward Difference Formula (e.g. BDF2) for the approximation of $\partial_t \tilde{E}_N$; more precisely, denoting by $\Delta\tilde{E}_N/\Delta t$ the approximation of $\partial_t \tilde{E}_N$, we compute

$$r_E = \Delta\tilde{E}_N/\Delta t + \partial_x((\tilde{E}_N + gh_N^2/2)q_N/h_N) + gq_N\partial_x z \tag{10}$$

with $\tilde{E}_N = q_N^2/(2h_N) + gh_N^2/2$, and we set

$$\nu_E = \beta|r_E|/\Delta E_N \delta x^2\,, \quad \Delta E_N = \max_D E_N - \min_D E_N \tag{11}$$

where the grid size δx is that of the Gauss-Lobatto-Legendre (GLL) mesh.

The first order viscosity $\nu_{\max}$ for the Saint Venant system must be based on a wave speed that should be larger than $\lambda_\pm = u \pm \sqrt{gh}$. We set

$$\nu_{\max} = \alpha \max_D(|q_N/h_N| + \sqrt{gh_N})\delta x \tag{12}$$

where again δx is the GLL grid-size.

The viscosity is then defined by $\nu = \min(\nu_{\max}, \nu_E)$. This viscosity is additionally smoothed by using a two-step procedure:

- first locally (in each element), e.g. $(\nu_{i-1} + 2\nu_i + \nu_{i+1})/4 \to \nu_i$
- then globally, by projection onto the space of the C^0 piecewise polynomial of degree N. Note that this is easy to do, since the SEM mass matrix is diagonal.

We now finally recall how to adjust the values of the EVM control parameters:

- First, one solves the problem with the viscosity $\nu_{\max}$ and adjust α to obtain a smooth solution.
- Second, one solves with the entropy viscosity ν and adjust β.

Properties of the approximation: The following properties are expected from the SEM/EVM approximation:

- Mass conservation: Setting $v_N = 1$ in the equation for h_N yields

$$\int_D (\partial_t h_N + \partial_x q_N)\,dx = \int_D \partial_t h_N\,dx + 0 = d_t \int_D h_N\,dx = 0 \tag{13}$$

  if $q_N(x_{\sup}) - q_N(x_{\inf}) = 0$, which means that the total mass is preserved. Indeed, the GLL quadratures are here exact.
- Conservation of energy for smooth solutions. There is no guaranty here, because the equation for the energy involves non-linear terms that are approximated by the GLL quadratures.
- Positivity of h. Here again, one may expect difficulties as soon as $N > 1$, i.e. when the space approximation is not simply piecewise linear. For problems in which we are interested in, i.e. involving dry-wet transitions, numerical difficulties systematically occur when using the standard form of the EVM. To

overcome these difficulties, we suggest to use the first order viscosity as soon as the fluid height becomes small. We thus supplement the EVM with the following step:

$$\nu = \nu_{\max} \quad \text{if} \quad h_N < h_{\text{thres}} \tag{14}$$

where the threshold height h_{thres} is small, i.e. typically 10^{-3} of the mean fluid height. Moreover, we have not based $\nu_{\max}$ on a local but on a global maximum of the wave speed, see Eq. (12).

4 Examples of Applications

The following test-cases have been considered: (1) Lake at rest with an emerged bump. The surface water should remain flat. This is what one usually expects of a *well balanced scheme*. (2) Oscillations in a parabolic cup. The solution to this problem being smooth, the energy should remain constant over time. (3) Dam break on a dry domain. The main problem here is to get the right velocity at the front of the water wave. (4) Dam break on a sinusoidal topography. This problem combines different aspects previously mentioned. It should be remarked that all these test-cases have dry-wet transitions. The first three test cases have analytical solutions, see e.g. [1].

Lake at rest with an emerged bump: In this test the free surface should remain flat and the velocity must be zero at all times. Figure 1 shows the EVM solution as well as the entropy viscosity. As desired, the viscosity is maximal in the dry part of the bump. The result is satisfactory, even if one observes (on an animation) some traveling waves with very low amplitude.

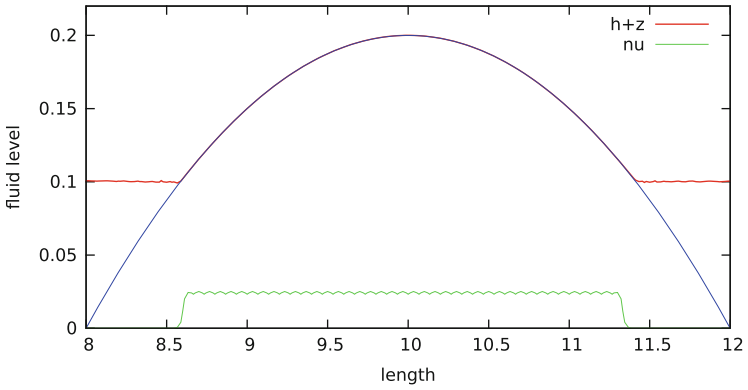


Fig. 1 Bump problem: $D = (8, 12)$, $t_F = 400$, 60 elements, $N = 4$, $\alpha = 1$, $\beta = 10$, $h_{\text{thres}} = 10^{-4}$. EVM solution and entropy viscosity at time $t_F = 400$

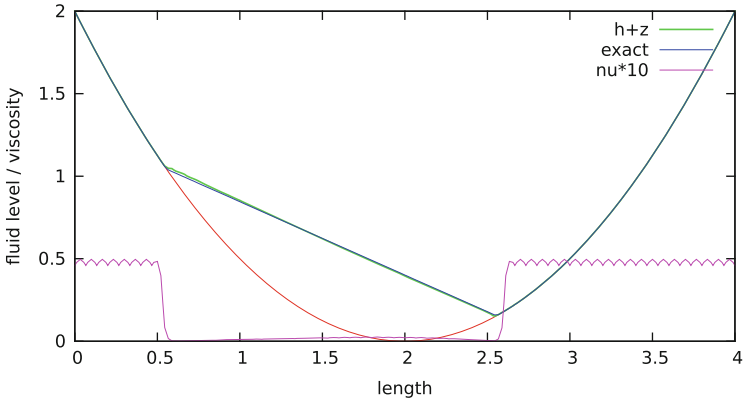


Fig. 2 Cup problem: $D = (0, 4)$, $t_F = 50$, 60 elements, $N = 4$, $\alpha = 1$, $\beta = 10$, $h_{\text{thres}} = 10^{-3}$. EVM and exact solutions, entropy viscosity at time $t_F = 50$

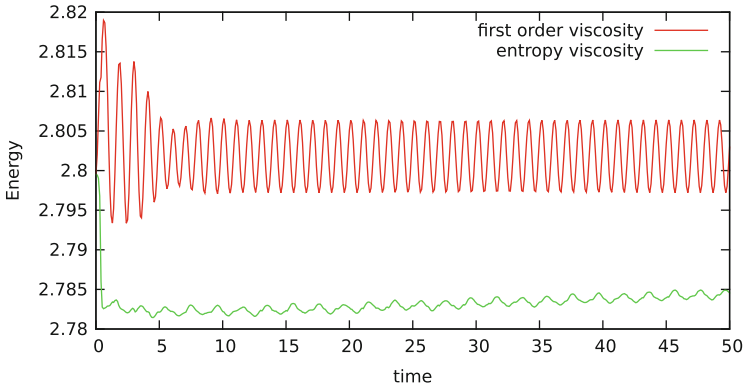


Fig. 3 Cup problem: Time-variations of the total energy for the solutions obtained with the entropy viscosity and with the first order viscosity

Oscillations in a cup: The topography is a parabolic bowl. The fluid level, $h + z$, at the initial time is defined by an inclined line. Since the solution to the problem is smooth there is no dissipation and the fluid oscillates indefinitely. Figure 2 compares the exact solution with the computed one at the final time, $t_F = 50$. The entropy viscosity is also shown.

It is interesting for this problem to verify how well the energy is conserved. Figure 3 shows the time evolution of the total energy for both the EVM and the first order viscosity solutions. One observes some oscillations, especially for the first order viscosity solution, and there is a slight increase in energy for the EVM solution. The result is however satisfactory since the oscillatory motion is well maintained, i.e. there is no significant artificial dissipation of the energy.

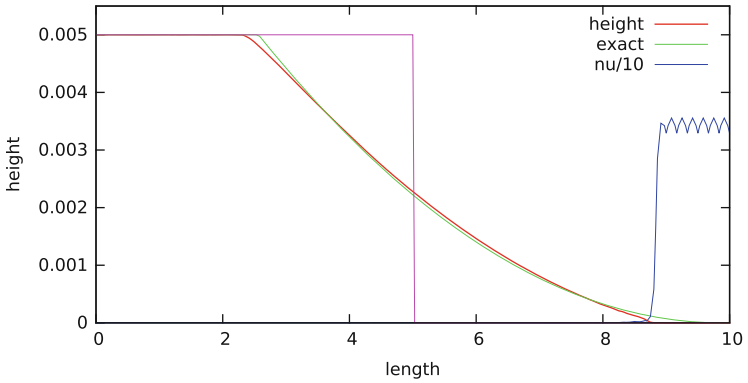


Fig. 4 Dam break problem: $D = (0, 10)$, $t_F = 120$, 60 elements, $N = 4$, $\alpha = 2$, $\beta = 20$, $h_{\text{thres}} = 10^{-6}$. EVM and exact solutions, entropy viscosity at $t_F = 11$. The initial condition is also shown

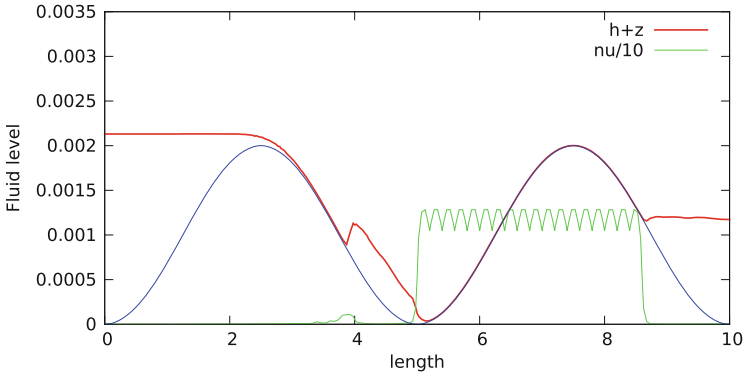


Fig. 5 Dam/bump problem: $D = (0, 10)$, $t_F = 600$, 60 elements, $N = 4$, $\alpha = 1.5$, $\beta = 30$, $h_{\text{thres}} = 10^{-5}$. EVM solution and entropy viscosity at $t = 129$. Initial condition: $h + z = 0.003$ if $x < 2$, $h = 0$ if $x > 2$

Dam break: The dam break on dry domain is a classical test-case. It is especially of interest to verify whether the velocity of the leading wave is correct. Figure 4 shows that the results from the EVM are satisfactory, even if some slight differences can be observed at the upper left and bottom right parts of the expansion wave.

Dam break over bumps: We now solve the dam break problem on a dry domain with a sinusoidal topography. Figure 5 shows a snapshot of the solution. At the end of the computation one recovers the situation met previously for the cup problem, i.e. the fluid oscillates between the two bumps and remains trapped therein indefinitely.

Acknowledgements This work is partly supported by the National Science Foundation grants DMS-1015984 and DMS-1217262 and by the Air Force Office of Scientific Research, USAF,

under grant/contract number FA99550-12-0358. It is also supported by grants of the French research agency (ANR project MEDIMAX).

References

1. O. Delestre, C. Lucas, P.-A. Ksinant, F. Darboux, C. Laguerre, T.-N.-T. Vo, F. James, S. Cordier, SWASHES: a compilation of shallow water analytic solutions for hydraulic and environmental studies. Int. J. Numer. Methods Fluids **72**(3), 269–300 (2013)
2. J.-F. Gerbeau, B. Perthame, Derivation of the viscous Saint-Venant system for laminar shallow water; numerical validation. Discrete Contin. Dyn. Syst. Ser. B 1 **1**, 89–102 (2001)
3. J.L. Guermond, B. Popov, Viscous regularization of the Euler equations and entropy principles. SIAM J. Appl. Math. **74**(2), 284–305 (2014)
4. J.L. Guermond, R. Pasquetti, B. Popov, Entropy viscosity method for non-linear conservation laws, J. Comput. Phys. **230**(11), 4248–4267 (2011)
5. G.E. Karniadakis, S.J. Sherwin, *Spectral HP Element Methods for CFD* (Oxford University Press, London, 1999)
6. R.J. Leveque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, 2007)

A Windowed Fourier Method for Approximation of Non-periodic Functions on Equispaced Nodes

Rodrigo B. Platte

Abstract A windowed Fourier method is proposed for approximation of non-periodic functions on equispaced nodes. Spectral convergence is obtained in most of the domain, except near the boundaries, where polynomial least-squares is used to correct the approximation. Because the method can be implemented using partition of unit and domain decomposition, it is suitable for adaptive and parallel implementations and large scale computations. Computations can be carried out using fast Fourier transforms. Comparisons with Fourier extension, rational interpolation and least-squares methods are presented.

1 Introduction

The recovery of a function from a finite set of its values is a common problem in scientific computing and is one of the main underlying problems in the numerical solution of partial differential equations. This manuscript focuses on the special case of approximating functions from values sampled at evenly distributed points.

It is known that polynomial interpolants of smooth functions at equally spaced points do not necessarily converge, even if the function is analytic. Instead one may see wild oscillations near the endpoints, an effect known as the Runge phenomenon. Associated with this phenomenon is the exponential growth of the condition number of the interpolation process. Several other methods have been proposed for recovering smooth functions from uniform data, such as polynomial least-squares, rational interpolation, and radial basis functions; to name but a few. It is now known that these methods cannot converge at geometric (exponential) rates and remain stable for large data sets [11]. In practice, however, some methods perform remarkably well.

In this work we present a hybrid method based on *windowed Fourier* (WF) approximations combined with polynomial least-squares corrections near boundaries. The algorithm is an adaptation of the method presented in [10], in which

R.B. Platte (✉)
School of Mathematical and Statistical Sciences, Arizona State University, Tempe, AZ, USA
e-mail: rbp@asu.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_37

a hybrid grid was used in the approximations—uniform nodes in the interior of the domain combined with Chebyshev points near the endpoints. In contrast, the algorithm proposed here relies strictly on equispaced grids. Besides describing a criterium for choosing parameters in the algorithm (window size, boundary layer correction, and polynomial degree), a generalized version of Hermite's error formula is used to compare the accuracy of the proposed algorithm with other known methods for the approximation of analytic functions.

The WF scheme can be closely related with Fourier continuation (or extension/embedding) methods [4, 6], which have been extensively explored recently. It is important to point out that Fourier extension requires a periodic continuation of the target function outside the domain of interest. The extension is not unique and different strategies have been presented in the literature to generate them. In [6], for example, an SVD based least squares approximation is used, while in [3] polynomials are used to periodically extend the function. Although an FFT based implementation is available for the SVD approach [8], it is restricted to extended domains that are at least twice as large (in 1D) as the domain of interest, a limitation that has implications on the oversampling rate for stable approximations. A detailed study of the tradeoffs between amount of oversampling and numerical stability has been recently presented in [1, 2]. The WF method, on the other hand, does not require least-squares approximations on the interior of the domain, with Fourier coefficients being computed by interpolation.

Along these lines, several other methods have been proposed to approximate functions from equispaced nodes with spectral-like accuracy. Examples can be found, for instance, in [5, 11]. Here we focus on describing the WF method and providing numerical experiments to demonstrate its performance.

2 Background and Algorithm

For simplicity, we describe the scheme for approximations on a bounded interval. Computations in higher dimensions are carried out using tensor products. The WF method for equispaced points is motivated by Platte and Gelb [10], where a similar strategy was proposed as an alternative to traditional spectral methods. In that paper, polynomial approximations near the edges of the domain were computed on Chebyshev nodes, as the main focus was the solution of partial differential equations. In the present work, we replace polynomial interpolation with least-squares, and relax the restriction on the node distribution near the boundary.

A windowed Fourier approximation is illustrated in Fig. 1. To approximate a non-periodic function u, using Fourier expansions, a smooth window function w is used. The window and its derivatives are close to zero at boundary points and the product uw can be accurately approximated by a truncated trigonometric series. The function u can be recovered from this approximation by dividing it by w. Since w is close to zero near the boundaries, errors are amplified in that region. To correct the

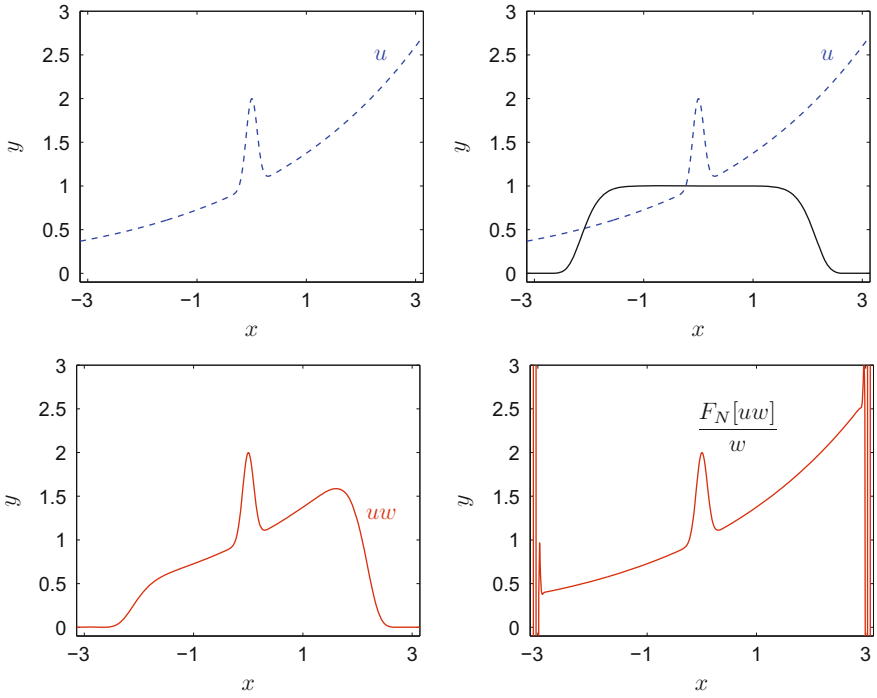


Fig. 1 From *left* to *right*: the function $u(x) = \exp(x/\pi) + \exp(-50x^2)$ (*dashed*), the window function w (*solid*), the product uw, and the Fourier approximation of uw divided by w

approximation near the ends of the interval, local polynomial approximations are used.

Although not pursued here, it is also possible to take advantage of the windowing process to decompose the domain if variable resolution is required. There are many possible window functions. As in [10], the method proposed here uses super-Gaussian window functions, $w(x) = \exp\left(-\alpha(x/\pi)^{2\lambda}\right)$, $x \in [-\pi, \pi]$, where λ is a positive integer and $\alpha \approx 52 \ln 2$ is used in double precision. This choice of α ensures that $w(\pm\pi) \approx 2^{-52}$, which is the machine epsilon.

The difficulty in finding a suitable window is that a nonzero function that has all derivatives vanishing at a point cannot be analytic on any neighborhood of the interval in consideration. There must be a compromise between enforcing periodicity and the stiffness of the product uw. This aspect is investigated in detail in [10]. Figure 2 shows super-Gaussian window functions for $\lambda = 2$, 5, and 20. Notice that $\lambda = 20$ gives large support for the Fourier approximation but also stiff gradients near the boundaries. Although super-Gaussians are not compactly supported in infinite precision, we found that their Fourier sums have better convergence properties than C^∞ compactly supported window functions, such as $\exp(-1/(1-(x/\pi)^2)^\lambda)$. The lack of exact periodicity in derivatives in the

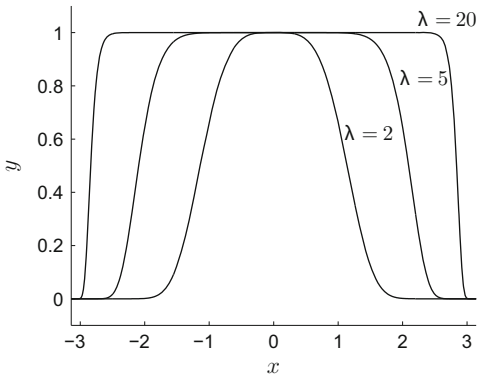


Fig. 2 Super-Gaussian window functions: $\exp(-32(x/\pi)^{2\lambda})$

case of super-Gaussians only affects convergence rates once the error falls below machine precision and their Fourier sums converge geometrically (exponentially) for all practical purposes.

Throughout this work we consider the truncated Fourier series,

$$F_N[u](x) = \sum_{n=-N}^{N}{}'\hat{u}_n \exp(inx), \qquad -\pi \le x \le \pi, \tag{1}$$

where the coefficients $\hat{u}_n$ are computed so that $u(x_j) = F_N[u](x_j)$ at $2N$ equally spaced nodes. The prime indicates that the terms $n = \pm N$ are multiplied by $\frac{1}{2}$. It is well known that the series converges exponentially fast, as $N \to \infty$, to smooth periodic functions.

The accuracy and performance of the WF method depends on how well the product uw is approximated. Analysis presented in [10] shows that the number of modes in (1) required to approximate w, to a fixed accuracy, is linearly proportional to λ. In particular, approximations of w accurate to almost machine precision can be obtained with $N > 12\lambda$. This linear dependence can be explained using standard error estimates for Fourier interpolation and we refer to [10] for details.

Once the product uw is approximated, the recovery of u can be obtained by a point-wise division by w. Notice that the magnitude of the error in this process is inversely proportional to the values of $w(x)$, i.e.

$$|u(x) - F_N[uw](x)/w(x)| = |u(x)w(x) - F_N[uw](x)|\,/w(x).$$

Therefore, if corrections are made to the approximation in the regions where $w(x) < 0.05$, the error at the cutoff points would be about twenty times larger than at the center of the interval. Choosing the cutoff points, x_a and x_b, to satisfy $w(x_a) = w(x_b) = 0.05$, gives

$$x_a = -\pi\left((\ln 20)/\alpha\right)^{1/2\lambda} \text{ and } x_b = \pi\left((\ln 20)\alpha\right)^{1/2\lambda}. \tag{2}$$

Asymptotically, this means that the number of nodes in the correction regions remains nearly constant as $N \to \infty$.

For the 1-D case, the **algorithm** can be summarized as follows.

- Given $2N$ equispaced nodes on $[-\pi, \pi]$, choose λ to be $N/12$.
- Set the cutoff points according to (2).
- Approximate the product wu using (1). The approximation in $[x_a, x_b]$ is then given by $(F_N[uw](x))/w(x)$.
- Correct approximations in $[-\pi, x_a]$ and $[x_b, \pi]$ using polynomial least-squares. Here we choose the polynomial degree to be half the number of nodes in the correction regions.

The rate of convergence of scheme is limited by the least-squares polynomial correction near the ends of the domain. That region, however, shrinks as N is increased. Spectral accuracy is attained in most of the domain and fast convergence is expected for functions free of singularities or steep gradients near the boundary.

3 Accuracy for Analytic Functions

In this section we use a generalization of Hermite's error formula to compare the accuracy of different methods. To this end, we consider a general linear approximation of an analytic function f from its data values as

$$\mathcal{L}_{f,N}(x) := \sum_{j=1}^{N} f(x_j) L_j(x), \tag{3}$$

where the L_j are bounded functions. In the case of interpolation, L_j would be a cardinal functions.

Theorem 1 *Suppose f is analytic in a closed simply connected region R and C is a simple, closed, rectifiable curve that lies in R and encloses the interpolation points x_j, $j = 1, \ldots, N$. The error at x in the approximation (3) is*

$$f(x) - \mathcal{L}_{f,N}(x) = \frac{1}{2\pi i} \int_C \frac{f(z)}{z - x} r_N(z, x) dz, \tag{4}$$

$$r_N(z, x) = 1 - (z - x) \sum_{j=1}^{N} \frac{L_j(x)}{z - x_j}. \tag{5}$$

From (5) we can see that $r_N(z, x)$ can be interpreted as the relative error at x in approximating the function $1/(z - x)$ with (3). Additional details, including the proof, can be found in [9].

Using (4) we can bound the error. Under the assumptions of Theorem 1, assume that x and all interpolation points are in $[-1, 1]$, then

$$\left\| f - \mathcal{L}_{f,N} \right\|_{[-1,1]} \leq M_f \max_{x\in[-1,1]} |r_N(z,x)|, \quad \text{where } M_f = \frac{\text{arclength}(C) \max_{z\in C} |f(z)|}{2\pi \min_{x\in[-1,1],\, z\in C} |z-x|}.$$

Notice that M_f depends only on the function being approximated and the accuracy of the approximating scheme, including node distribution, is captured in r_N.

Figure 3 shows the contour levels of $R_N(z) := \max_{x\in[-1,1]} |r_N(z,x)|$. Results are shown for $N = 200$ and are qualitatively similar for other values of N. The top panel in Fig. 3 shows R_{200} for the WF method. It shows that approximations are very accurate in the interior of the domain. As could be expected, R_N decays more slowly near the end points due to the local polynomial correction in that region.

For reference, the bottom panel of Fig. 3 shows the corresponding values of R_N for three other methods: polynomial least-squares, rational interpolation, and Fourier continuation. These methods are known to work well on equispaced nodes. For the least-squares approximation, the degree of the polynomial was chosen to be approximately $4\sqrt{N}$ to ensure stable results (see e.g. [11, 12]). The rational interpolation approximation is computed using the method presented Floater and

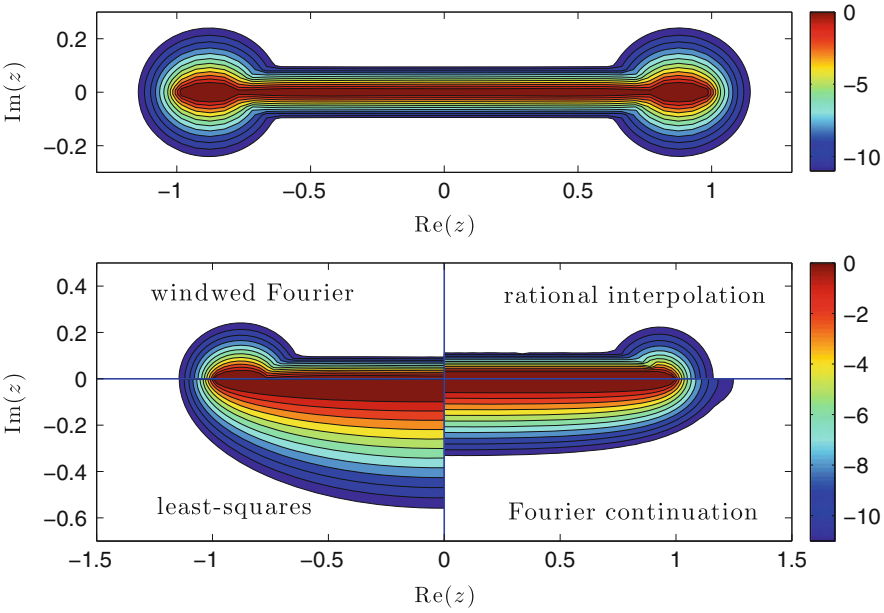


Fig. 3 *Top*: Level curves of $R_N(z) = \max_{x\in[-1,1]} |r_N(z,x)|$ in a log scale for WF with $N = 200$. Contours shown correspond to $10^{-11}, 10^{-10}, \ldots, 1$. *Bottom*: Same as the top plot, but with three other methods for comparison

Hormann in [7] with degree 15. The Fourier continuation scheme was computed using the SVD approach and oversampled least-squares [8]. Extended domains were twice as large as each subdomain (see [2, 8] for details). Notice that the WF method compares favorably to all three methods as R_N takes smaller values in a larger part of the complex plane. This figure also shows that R_N is similar for the WF and rational interpolation methods.

Finally, we use four functions to test the performance of the WF method. Figure 4 shows the error as a function of N, the number of equispaced nodes, for each function. For reference, the error in polynomial interpolation on N Chebyshev nodes is also included. As predicted by the contours in Fig. 3, WF and rational interpolation present similar accuracy. Due to better resolution in the interior of the domain, the WF approximation of f_2 converges faster than polynomial interpolation on Chebyshev nodes. While Fourier continuation outperforms WF approximations for the oscillatory function f_1, WF is significantly more accurate for f_2 and f_3. Notice that f_4 has a singularity close to a boundary point, at $x = 1.05$, and as a consequence all methods converge at sub-geometric rates (and are much less accurate than interpolation at Chebyshev points). This result is also in good agreement with Fig. 3.

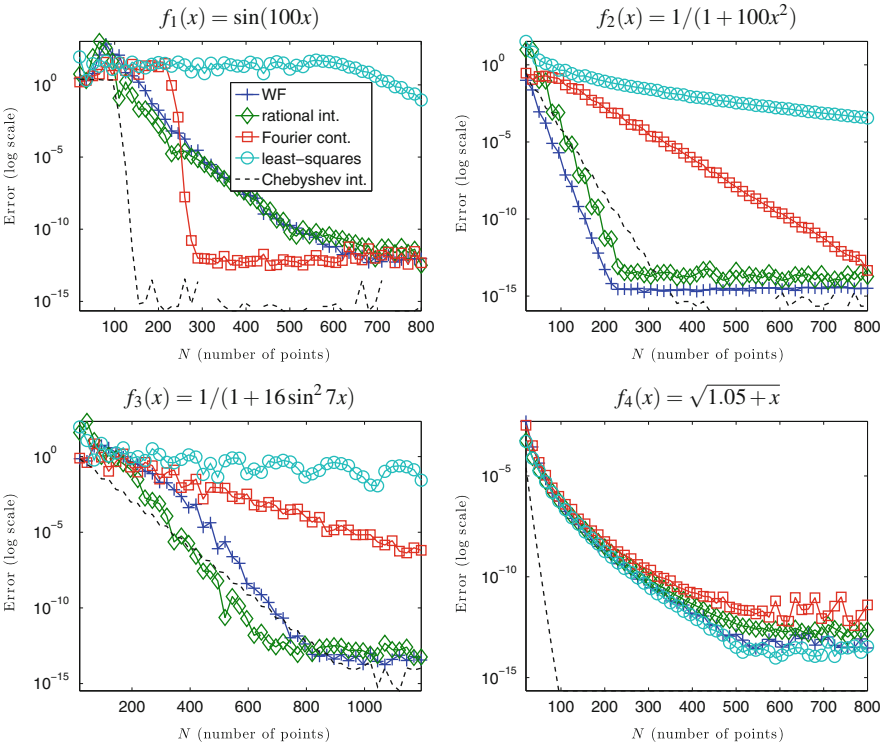


Fig. 4 Error decay in the approximation of four functions on the interval $[-1, 1]$

4 Concluding Remarks

It is important to point out that the WF method is not exponentially convergent for all analytic functions. The results presented in Fig. 4 do not contradict the theorem in [11], which asserts that stable methods cannot converge at geometric rates for all functions that analytic in regions enclosing the interval of approximation. This is evident in the approximation of $f_4(x) = \sqrt{1.05 + x}$, where the error plot shows sub-geometric decay for all methods (except for Chebyshev interpolation).

The approximation order of the method presented here is dominated by the polynomial least-squares approximation near the endpoints. Because the number of nodes on these regions is nearly constant (the size of boundary layer shrinks as the overall number of points is increased), the polynomial degree remains nearly constant as $N \to \infty$. For the parameters used to generate the plots in Fig. 4, the degree for the polynomial corrections is 21 when $N \geq 150$. The algebraic convergence near the endpoints of the interval is reflected in the lobes of Fig. 3. Although the formal convergence of the method is sub-geometric, in many practical cases, the error decays exponentially fast for practical values of N . In the case of $f_2(x) = 1/(1 + 100x^2)$, for instance, the approximation error is dictated by the singularity near $x = 0$ and the correction regions are very accurately approximated by a high order polynomial.

In contrast to the other three numerical schemes used to obtain Fig. 4, the polynomial correction step in the algorithm being proposed here leads to an approximation that is not continuous. The jump size in the resulting approximation is of the order of the approximation error and can be a source of instability when solving PDEs. This issue has been partially addressed in [10], when the polynomial correction is carried out using Chebyshev interpolation, but has not yet been addressed when equispaced nodes are used.

Acknowledgements This work was supported in part by AFOSR FA9550-12-1-0393.

References

1. B. Adcock, J. Ruan, Parameter selection and numerical approximation properties of Fourier extensions from fixed data. J. Comput. Phys. **273**, 453–471 (2014)
2. B. Adcock, D. Huybrechs, J. Martín-Vaquero, On the numerical stability of Fourier extensions. Found. Comput. Math. **14**(4), 653–687 (2012)
3. N. Albin, O.P. Bruno, A spectral FC solver for the compressible Navier–Stokes equations in general domains i: explicit time-stepping. J. Comput. Phys. **230**(16), 6248–6270 (2011)
4. J.P. Boyd, A comparison of numerical algorithms for Fourier extension of the first, second, and third kinds. J. Comput. Phys. **178**(1), 118–160 (2002)
5. J.P. Boyd, J.R. Ong, Exponentially-convergent strategies for defeating the Runge phenomenon for the approximation of non-periodic functions, part I: single-interval schemes. Commun. Comput. Phys. **5**(2–4), 484–497 (2009)

6. O.P. Bruno, Y. Han, M.M. Pohlman, Accurate, high-order representation of complex three-dimensional surfaces via Fourier continuation analysis. J. Comput. Phys. **227**(2), 1094–1125 (2007)
7. M.S. Floater, K. Hormann, Barycentric rational interpolation with no poles and high rates of approximation. Numer. Math. **107**, 315–331 (2007)
8. M. Lyon, A fast algorithm for Fourier continuation. SIAM J. Sci. Comput. **33**(6), 3241–3260 (2011)
9. R.B. Platte, How fast do radial basis function interpolants of analytic functions converge? IMA J. Numer. Anal. **31**(4), 1578–1597 (2011)
10. R.B. Platte, A. Gelb, A hybrid Fourier-Chebyshev method for partial differential equations. J. Sci. Comput. **39**(2), 244–264 (2009)
11. R.B. Platte, L.N. Trefethen, A.B.J. Kuijlaars, Impossibility of fast stable approximation of analytic functions from equispaced samples. SIAM Rev. **53**, 308–318 (2011)
12. E.A. Rakhmanov, Bounds for polynomials with a unit discrete norm. Ann. Math. (2) **165**(1), 55–88 (2007)

Smoothness-Increasing Accuracy-Conserving (SIAC) Filters in Fourier Space

Liangyue Ji and Jennifer K. Ryan

Abstract It has been noted in the past that discontinuous Galerkin methods can be viewed as a low order multi-domain Spectral method with penalty term (Hesthaven et al., Spectral methods for time-dependent problems, Cambridge University Press, Cambridge, 2007). It is then logical to first ask how to relate filters in Spectral Methods to Smoothness-Increasing Accuracy-Conservin (SIAC) filters, which are typically applied to approximations obtained via the discontinuous Galerkin methods. In this article we make a first effort to relate Smoothness-Increasing Accuracy-Conserving filtering to filtering for Spectral Methods. We frame this discussion in the context of Vandeven (J Sci Comput 6:159–192, 1991).

1 Background

In Fourier Spectral methods [5], we expect that the approximation to a given partial differential equation will have exponential accuracy if the solution is analytic. However, the convergence deteriorates if the solution is less smooth, with a discontinuous solution leading to Gibbs phenomenon. We can overcome the deteriorated convergence with the use of a filter. In general, a filter can reduce oscillations in the vicinity of a discontinuity and recover the appropriate accuracy order. In this article we analyze filters from the perspective of Vandeven [11] and apply this analysis to Smoothness-Increasing Accuracy-Conserving Filters that are typically applied to discontinuous Galerkin approximations [3], which were developed based on [1–4, 10, 12].

L. Ji
University of Minnesota, Minneapolis, MN, USA
e-mail: jil@umn.edu

J.K. Ryan (✉)
University of East Anglia, Norwich, UK
e-mail: Jennifer.Ryan@uea.ac.uk

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_38

We frame our discussion in the context of a one-dimensional time-dependent PDE with periodic boundary conditions,

$$u_t = \mathcal{L}u, \qquad 0 \le x \le 2\pi, \tag{1}$$

$$u(x,0) = u_0(x), \tag{2}$$

$$u(0,t) = u(2\pi,t). \tag{3}$$

In the Fourier Spectral method, we seek an approximate solution of the form

$$v(x,t) = \sum_{\ell=-N}^{N} C_\ell(t) e^{i\ell x}, \tag{4}$$

where $v(x,t)$ is an approximation to the exact solution of Eq. (1). There are a few methods of determining the coefficients $C_\ell(t)$. For example, if we apply the Fourier-Galerkin method, $v(x,t)$ satisfies Eq. (1) weakly. If we apply the Fourier-Collocation method, $v(x,t)$ satisfied Eq. (1) strongly at the grid points.

2 The Fourier Spectral Approximation

The unfiltered Fourier Spectral approximation is given by

$$u_N(x,t) = \sum_{\ell=-N}^{N} u^{(\ell)}(t) e^{i\ell x}, \qquad u^{(\ell)}(t) = \frac{1}{2\pi}\int_0^{2\pi} u(y,t) e^{-i\ell y}\, dy. \tag{5}$$

From [11], we know that we can write the filtered approximation as

$$u_N(x,t) = \sum_{\ell=-N}^{N} \sigma\left(\frac{\ell}{N}\right) u^{(\ell)}(t) e^{i\ell x}, \tag{6}$$

where $\sigma\left(\frac{\ell}{N}\right) u^{(\ell)}(t)$ are the filtered coefficients. Note that this allows us to rewrite the filtered approximation as

$$u_N^\sigma(x,t) = \frac{1}{2\pi}\int_0^{2\pi} u(y,t) K_N(x-y)\, dy, \tag{7}$$

where K_N is a convolution kernel given as

$$K_N(x) = 1 + \sum_{\ell=1}^{N} \sigma\left(\frac{\ell}{N}\right)\cos(\ell x). \tag{8}$$

In [11] a generalised definition of filters is given:

Definition 1 A *filter* σ *of order* p is a smooth even function whose support is $[-1, 1]$ and such that

$$\sigma(0) = 1 \quad \text{and} \quad \sigma^{(\alpha)}(0) = 0, \quad 1 \le \alpha \le p-1, \tag{9}$$

$$\sigma^{(\alpha)}(1) = 0, \quad 1 \le \alpha \le p-1. \tag{10}$$

Some classical filters given in [5] are the Lanczos filter, raised cosine, sharpened raised cosine and exponential cutoff. They are defined in the following manner:

Lanczos

$$p = 1, \qquad \sigma_1 = \frac{\sin(\pi x)}{\pi x}.$$

Raised cosine

$$p = 2, \qquad \sigma_2 = \frac{1 + cos(\pi x)}{2}.$$

Sharpened raised cosine

$$p = 7, \qquad \sigma_3 = \sigma_2^4(35 - 84\sigma_2 + 70\sigma_2^2 - 20\sigma_2^3).$$

Exponential cutoff

$$f(x) = \begin{cases} 1, & x \le x_c \\ e^{-\beta(|x|-x_c)^4}, & x_c \le x \le 1. \end{cases} \tag{11}$$

In Fig. 1 the difference in the Fourier expansion of a saw-tooth function and the filtered solution is shown. We can see that the filter removes the majority of the oscillations, with a few oscillations remaining at the discontinuity.

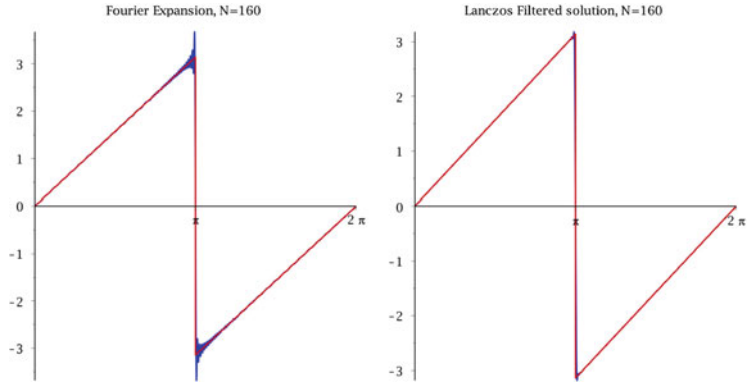


Fig. 1 *Left*: Fourier expansion of a saw-tooth function. *Right*: Filtered Fourier expansion using the Lanczos filter. The exact function is given in *red* and the approximation is given in *blue*

3 Smoothness-Increasing Accuracy-Conserving (SIAC) Filters

We now focus on the SIAC filter [6–9], which is typically implemented for approximations obtained using the discontinuous Galerkin (DG) method. The DG method can be viewed as a low-order, multi-domain Spectral method [5].

3.1 The DG Approximation and SIAC Filters in Physical Space

The discontinuous Galerkin approximation is obtained much like the Spectral Galerkin approximation. That is by multiplying Eq. (1) by a test function and integrating by parts. The difference is that the DG approximation is formed over one element and the Spectral approximation is formed globally, over the entire domain. Additionally, the approximation space for the DG solution is defined as piecewise polynomials of degree less than or equal to k on each element,

$$\phi_j^{(\ell)}(x) \in V_h^k = \{v \in L^2(\Omega) : v \in \mathbb{P}^k(I_j), j = 1, \ldots, N\}, \tag{12}$$

where I_j are the elements. The DG approximation can then be written as

$$u_h(x,t) = \sum_{\ell=0}^{k} u_j^{(\ell)}(t)\phi_j^{(\ell)}(x), \qquad x \in I_j,\ j = 1, \ldots, N. \tag{13}$$

The SIAC filtered DG solution is then given by

$$u^*(x,t) = \frac{1}{h}\int_{\mathbb{R}} K\left(\frac{x-y}{h}\right) u_h(y,t)\,dy, \tag{14}$$

with the convolution kernel having the form

$$K^{r+1,\ell}(x) = \sum_{\gamma=0}^{r} c_\gamma^{r+1,\ell}\psi^{(\ell)}(x-x_\gamma), \tag{15}$$

where x_γ depends on the location of the point being post-processed. More details are given in [6]. Note that Bramble and Schatz [1] and Cockburn et al. [4] originally introduced this as a post-processor to enhance the accuracy of finite element and discontinuous Galerkin solutions respectively. Typically, $r = 2k$, $\ell = k+1$ and $x_\gamma = \gamma - k$. The coefficients, $c_\gamma^{r+1,\ell}$, satisfy $K * x^p = x^p$, $p = 0, 1, \cdots, r$, and $\psi^{(\ell)}$ is a central B-spline obtained by convolving the $\chi_{[-\frac{1}{2},\frac{1}{2}]}$ with itself $\ell - 1$ times.

3.2 SIAC Filters in Fourier Space

An interesting question to ask is what is the Fourier transform of the SIAC filter? Interestingly enough, it is very similar to Eq. (8) and is given by

$$\hat{K}(\xi) = \left(\frac{\sin(\xi/2)}{(\xi/2)}\right)^{k+1}\left(c_0 + 2\sum_{\gamma=1}^{k} c_\gamma^{2(k+1),k+1}\cos(\gamma\xi)\right). \tag{16}$$

More specifically, for $k = 1$ we have

$$\hat{K}(\xi) = \left(\frac{\sin(\xi/2)}{(\xi/2)}\right)^{2}\left(\frac{7}{6} - \frac{1}{6}\cos(\xi)\right)$$

and for $k = 2$,

$$\hat{K}(\xi) = \left(\frac{\sin(\xi/2)}{(\xi/2)}\right)^{3}\left(\frac{437}{320} - \frac{97}{240}\cos(\xi) + \frac{37}{960}\cos(2\xi)\right).$$

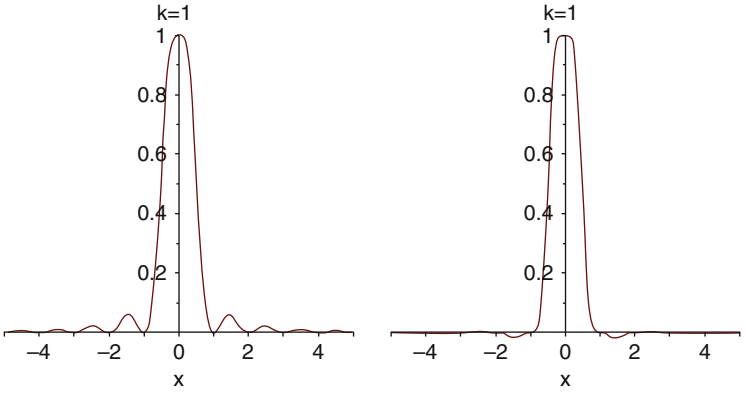


Fig. 2 SIAC filters for $k = 1$ (*left*) and $k = 2$ (*right*) written in Fourier space

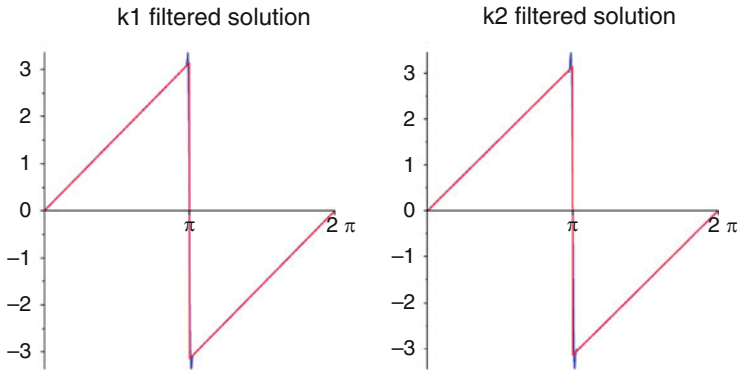


Fig. 3 The Fourier SIAC filter for $k = 1$ (*left*) and $k = 2$ (*right*) applied to the saw-tooth function

Plots of these kernels in Fourier space are shown in Fig. 2. We can see as k increases the oscillations away from zero decrease. Further, the Fourier SIAC filter for $k = 1$ and $k = 2$ applied to the saw-tooth function is shown in Fig. 3. Similar to the Spectral filters, it reduces the oscillations away from the discontinuity. A comparison of the errors with the Lanczos filter is given in Fig. 4. We can see that for increasing k, the errors for the Fourier SIAC filtered solution decay faster away from the discontinuity.

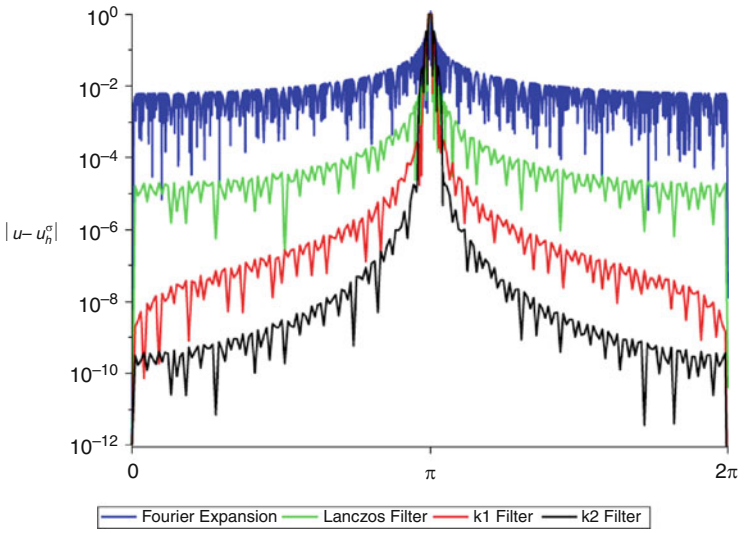


Fig. 4 A comparison of errors for the Lanczos filter and the Fourier SIAC filter for $k = 1, 2$ applied to the saw-tooth function

4 Fourier SIAC Filter

Theorem 1 *The Fourier SIAC Filter is a filter of order $p = 2k + 2$ and is around one,*

$$\hat{K}(\xi) = 1 + \mathcal{O}(\xi^{2k+2}). \tag{17}$$

Further, we have

$$\frac{d}{dx}\hat{K}(0) = \cdots = \frac{d^{2k+1}}{dx^{2k+1}}\hat{K}(0) = 0. \tag{18}$$

Proof In order to show that the Fourier SIAC filter is a filter of order $k+1$, we must show that the kernel is around one and that the derivatives up to order k vanish.

To show the Fourier form of the SIAC filter is around one we begin by noting that the Fourier form of the filter was studied by Thomeé in [10]. He in fact suggested that

$$\hat{K}(\xi) = 1 + \mathcal{O}(\xi^{2k+2}).$$

We can confirm this by writing polynomial reproduction property of SIAC filters (that it reproduces polynomials up to order $2k+1$) in Fourier space:

$$\hat{K}(\xi)\mathcal{F}\{x^p\} = \mathcal{F}\{x^p\}. \tag{19}$$

with

$$\mathcal{F}\{x^p\} = \left(\frac{i}{2\pi}\right)^p \delta_0^{(p)}(\xi), \tag{20}$$

where $\delta_0^{(p)}(\xi)$ the p-th derivative of Dirac delta function δ_0. Hence we have

$$\hat{K}(\xi)\delta_0^{(p)}(\xi) = \delta_0^{(p)}(\xi), \qquad p = 0, 1, \cdots, 2k. \tag{21}$$

We note that for any element $g \in S(\mathbb{R})$ in the Schwartz space

$$\left(\hat{K}\delta_0^{(p)}, g\right) = \left(\delta_0^{(p)}, g\hat{K}\right) = \left(\delta_0^{(p)}, g\right). \tag{22}$$

Therefore,

$$(-1)^p(g\hat{K})^{(p)}|_{\xi=0} = (-1)^p g^{(p)}|_{\xi=0}. \tag{23}$$

Now let g be a smooth function on $\mathbb{R}$ that is equal to one when $|x| \leq 1$ and equal to zero for $|x| > 2$, then we obtain

$$\hat{K}(0) = 1 \quad \text{and} \quad \hat{K}^{(p)}|(0) = g^{(p)}|(0) = 0, \qquad p = 1, 2, \cdots, 2k. \tag{24}$$

Consider the Taylor expansion of $\hat{K}(\xi)$,

$$\hat{K}(\xi) = \sum_{n=0}^{\infty} d_n \xi^n = d_0 + d_1\xi + d_2\xi^2 + \cdots + d_n\xi^n + \cdots. \tag{25}$$

By (24), $d_0 = 1$ and $d_n = 0,\ n = 1, \cdots, 2k$. Additionally, $\hat{K}$ is even function, so all the odd terms vanish, which means $d_{2k+1} = 0$. Hence

$$\hat{K}(\xi) = 1 + \mathcal{O}(\xi^{2k+2}).$$

and

$$\frac{d}{dx}\hat{K}(0) = \cdots = \frac{d^{2k+1}}{dx^{2k+1}}\hat{K}(0) = 0.$$

Using Definition 1 of the filter, we have now demonstrated that the Fourier SIAC filter is a filter of order $p = 2k+2$.

5 Conclusion and Future Work

This is a first attempt at relating Spectral filters to SIAC filters. We have demonstrated that the Fourier SIAC filter is indeed a filter of order $2k + 1$ by the definition of Vandeven [11]. We hope that this will give more incite into the properties of SIAC filters and filters in general. Equally, instead of investigating the use of SIAC filters for accuracy enhancement, we will explore whether they are suitable to apply to approximations where there are discontinuities for removing oscillations in those regions.

References

1. J.H. Bramble, A.H. Schatz, Higher order local accuracy by averaging in the finite element method. Math. Comput. **31**, 94–111 (1977)
2. J.H. Bramble, J.A. Nitsche, A.H. Schatz, Maximum-norm interior estimates for Ritz-Galerkin methods. Math. Comput. **29**, 677–688 (1975)
3. B. Cockburn, C. Johnson, C.-W. Shu, E. Tadmor, *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations*. Lecture Notes in Mathematics, vol. 1697 (Springer, Berlin, 1998)
4. B. Cockburn, M. Luskin, C.-W. Shu, E. Süli, Enhanced accuracy by post-processing for finite element methods for hyperbolic equations. Math. Comput. **72**, 577–606 (2003)
5. J.S. Hesthaven, S. Gottlieb, D. Gottlieb, *Spectral Methods for Time-Dependent Problems* (Cambridge University Press, Cambridge, 2007)
6. L. Ji, P. van Slingerland, J.K. Ryan, C.W. Vuik, Superconvergent Error estimates for a position-dependent smoothness-increasing accuracy-conserving filter for DG solutions. Math. Comput. **83**, 2239–2262 (2014)
7. J.K. Ryan, B. Cockburn, Local Derivative Post-processing for the discontinuous Galerkin method. J. Comput. Phys. **228**, 8642–8664 (2009)
8. J.K. Ryan, C.-W. Shu, H. Atkins, Extension of a post-processing technique for the discontinuous Galerkin method for hyperbolic equations with application to an aeroacoustic problem. SIAM J. Sci. Comput. **26**, 821–843 (2005)
9. M. Steffan, S. Curtis, R.M. Kirby, J.K. Ryan, Investigation of smoothness enhancing accuracy-conserving filters for improving streamline integration through discontinuous fields. IEEE-TVCG, **14**, 680–692 (2008)
10. V. Thomée, High order local approximations to derivatives in the finite element method. Math. Comput. **31**, 652–660 (1977)
11. H. Vandeven, Family of spectral filters for discontinuous problems. J. Sci. Comput. **6**, 159–192 (1991)
12. L.B. Wahlbin, *Superconvergence in Galerkin Finite Element Methods* (Springer, Berlin, 1995)

Algorithms for Higher-Order Mimetic Operators

Eduardo Sanchez, Christopher Paolini, Peter Blomgren, and Jose Castillo

Abstract We present an algorithm that reformulates existing methods to construct higher-order mimetic differential operators. Constrained linear optimization is the key idea of this resulting algorithm. The authors exemplified this algorithm by constructing an eight-order-accurate one-dimensional mimetic divergence operator. The algorithm computes the weights that impose the mimetic condition on the constructed operator. However, for higher orders, the computation of valid weights can only be achieved through this new algorithm. Specifically, we provide insights on the computational implementation of the proposed algorithm, and some results of its application in different test cases. Results show that for all of the proposed test cases, the proposed algorithm effectively solves the problem of computing valid weights, thus constructing higher-order mimetic operators.

1 Methods and Algorithms to Construct Mimetic Differential Operators

The construction of discrete differential operators that satisfy the mathematical properties of their continuous counterparts is a topic of intense research. These types of discrete differential operators are said to be **mimetic** [1–4].

The **Castillo–Grone (CGM)** is a method for the construction of mimetic operators that yield approximations with the same order of accuracy at the boundary and the interior of the domain [5]. The construction of higher-order accurate CGM-based mimetic operators has been studied thus far. Specifically, second-, fourth-, and sixth-order-accurate mimetic gradient and divergence operators have been fully tested and implemented in diverse problems [3, 4]. In this work, we base our study

E. Sanchez (✉) • C. Paolini • P. Blomgren • J. Castillo
Computational Science Research Center, San Diego State University,
5500 Campanile Drive, San Diego, CA 92182-1245, USA
e-mail: ejspeiro@gmail.com

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_39

on a variant of the CGM first presented in [2, 5]. We will refer to this variant as the **Castillo–Runyan Method (CRM)**. Theoretical aspects of mimetic finite differences have also been studied in [6, 7].

Previous works from the authors presented the theory of an algorithm that reformulates the CRM. In [8], the authors address the construction of an algorithm implementing the CRM. Specifically, important general concepts are presented that generalize the CRM to construct mimetic differential operators as a function of any required (even) order of numerical accuracy. We will refer to this algorithm implementing the CRM as the **Castillo–Runyan–Sanchez (CRS)** algorithm. Further theoretical details of the CRM and the CRS algorithm are given in [9].

Numerical experiments on the CRS algorithm revealed a problem on both the CGM and the CRM [9]. The algorithm computes the weights that impose the mimetic condition on the constructed operator. However, for orders higher or equal than eight, the CRS algorithm yields negative weights. This violates the definition of the weighted norms used to impose the mimetic conditions on the constructed operators. The solution to this problem implies a modification on the original methods thus yielding a new algorithm. We will refer to this second and improved algorithm as the **Castillo–Blomgren–Sanchez (CBS)** algorithm. The authors detail the CBS algorithm as well as its benefits on a particular test case in [9].

In this work, we first explain the main objective of the CRS algorithm, and we explain the core of its problem. We then review the math of the CBS algorithm. We continue the work presented in [9]. Specifically, we provide insights on the computational implementation of the CBS algorithm and some results of its application in different test cases.

2 The Castillo–Runyan–Sanchez (CRS) Algorithm

A detailed explanation of this algorithm is given in [8]. The general purpose of both the CGM and the CRM, and therefore of both the CRS and the CBS algorithms is to construct a matrix, implementing a k-th order mimetic operator (Algorithm 1), over a one-dimensional uniform staggered grid. We denote such matrix as $\breve{\mathbf{D}}_x^k$, and, in general, it has the following structure:

$$\breve{\mathbf{D}}_x^k = \begin{bmatrix} & 0 & \cdots & & & & & & & \cdots & 0 \\ \mathbf{A} & 0 & \cdots & & & & & & & \cdots & 0 \\ & 0 & \cdots & & & & & & & \cdots & 0 \\ 0 & \cdots & 0 & s_1 & s_2 & \cdots & s_k & 0 & 0 & 0 \cdots & 0 \\ & & & & & & & & & & \\ 0 & \cdots & 0 & 0 & \ddots & \ddots & \cdots & \ddots & 0 & 0 \cdots & 0 \\ 0 & \cdots & 0 & 0 & 0 & s_1 & s_2 & \cdots & s_k & 0 \cdots & 0 \\ 0 & \cdots & & & & & & \cdots & 0 & & \\ 0 & \cdots & & & & & & \cdots & 0 & \mathbf{A}' & \\ 0 & \cdots & & & & & & \cdots & 0 & & \end{bmatrix}, \tag{1}$$

Algorithm 1: Common approach of the CRS and the CBS algorithms to construct a 1D, k-th-order mimetic operator, $\breve{\mathbf{D}}_x^k$

begin
- Compute the k coefficients approximating at the interior of the grid with a numerical accuracy of k-th order: $\{s_i\}_{i=1}^k$
- Compute the coefficients approximating at the west boundary of the one-dimensional grid. These coefficient are the elements of a submatrix, $\mathbf{A}$, of $\breve{\mathbf{D}}_x^k$.
- Exploit the center-skew-symmetry property of the resulting operator, to compute the coefficients approximating at the east boundary of the grid. These coefficient are the elements of a submatrix, $\mathbf{A}'$, of $\breve{\mathbf{D}}_x^k$.

end

where $\{s_i\}_{i=1}^k$ are the values for an stencil vector approximating the divergence at the interior cells, and $\mathbf{A}$ is a sub-matrix approximating the values at the west boundary. In this work, we will focus our attention to divergence operator, however, the technique can easily be applied to compute gradient operators. The matrices $\mathbf{A}$ and $\mathbf{A}'$ are related by the center-skew-symmetry property of the operator. A thorough explanation can be found in [3]. Therefore, our explanations only refer to the computation of the values for $\mathbf{A}$.

Numerical experiments on the CRS algorithm have successfully computed mimetic matrix operators (see [3] for more details) for lower orders. For example, for $k = 6$:

$$\breve{\mathbf{D}}_x^6 = \begin{bmatrix} d_{11} & d_{12} & d_{13} & d_{14} & d_{15} & d_{16} & d_{17} & d_{18} & d_{19} & 0 & \cdots \\ d_{21} & d_{22} & d_{23} & d_{24} & d_{25} & d_{26} & d_{27} & d_{28} & d_{29} & 0 & \cdots \\ -\frac{9}{1920} & \frac{125}{1920} & -\frac{2250}{1920} & \frac{2250}{1920} & -\frac{125}{1920} & \frac{9}{1920} & 0 & 0 & 0 & 0 & \cdots \\ 0 & -\frac{9}{1920} & \frac{125}{1920} & -\frac{2250}{1920} & \frac{2250}{1920} & -\frac{125}{1920} & \frac{9}{1920} & 0 & 0 & 0 & \cdots \\ 0 & 0 & -\frac{9}{1920} & \frac{125}{1920} & -\frac{2250}{1920} & \frac{2250}{1920} & -\frac{125}{1920} & \frac{9}{1920} & 0 & 0 & \cdots \\ 0 & 0 & 0 & -\frac{9}{1920} & \frac{125}{1920} & -\frac{2250}{1920} & \frac{2250}{1920} & -\frac{125}{1920} & \frac{9}{1920} & 0 & \cdots \\ & & & & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \end{bmatrix}, \tag{2}$$

where, for the first row, we have:

$$\begin{aligned} d_{11} &= -\tfrac{1077397}{1273920} & d_{12} &= \tfrac{15668474643803}{32472850116480} & d_{13} &= \tfrac{49955527}{39491520} \\ d_{14} &= -\tfrac{25369793}{19745760} & d_{15} &= \tfrac{12220145}{15796608} & d_{16} &= -\tfrac{21334421}{78983040} \\ d_{17} &= \tfrac{460217}{9872880} & d_{18} &= -\tfrac{101017}{39491520} & d_{19} &= \tfrac{3369}{26327680}, \end{aligned}$$

and for the second row, we have:

$$d_{21} = \tfrac{31}{960}\ d_{22} = -\tfrac{687}{640}\ d_{23} = \tfrac{129}{128}\ d_{24} = \tfrac{19}{192}\ d_{25} = -\tfrac{3}{32}$$
$$d_{26} = \tfrac{21}{640}\ d_{27} = -\tfrac{3}{640}\ d_{28} = 0 \quad d_{29} = 0.$$

Those weights will be computed, in the CRS algorithm, as part of the solution to a system, $\mathbf{\Pi q} = \mathbf{h}$, where the construction of the $\mathbf{\Pi}$ matrix, as well as the solution approach, are the main difference between the CRS and the CBS algorithm. The solution for this system has the form $\mathbf{q} = [q_1, \ldots q_k, \lambda_1, \ldots \lambda_{(k/2)-1}]^T$, where $\{q_i\}_{i=1}^{k}$ are the weights we require, and $\{\lambda_i\}_{i=1}^{(k/2)-1}$ are the scalars that arise as a consequence of the CRS algorithm's attempt to impose the mimetic condition, i.e. complying with an extended version of Gauss Divergence Theorem, as we shall briefly described on [9].

However, when the CRS algorithm is used to generate an eight-order mimetic divergence operator, the attained collection of weights, includes a negative one:

$$\mathbf{q}_{k=8} = [q_1\ q_2\ q_3\ q_4\ q_5\ q_6\ q_7\ q_8 \mid \lambda_1\ \lambda_2\ \lambda_3] = \tag{3}$$
$$= \left[\frac{29059}{23224}\ \frac{13735}{23224}\ \frac{71826}{25805} - \frac{\mathbf{7678657}}{\mathbf{6635520}}\ \frac{24991643}{9289728}\ \frac{4301443}{25804800}\right.$$
$$\left.\frac{286984471}{232243200}\ \frac{225451487}{232243200}\ \frac{7621}{107520}\ \frac{159}{17920}\ \frac{5}{7168}\right]$$

The existence of negative weights violates the mimetic conditions, since these conditions stem from a discrete version of the Extended Gauss' Theorem, described in [3]. This discrete version reads as follows: $\langle \breve{\mathbf{G}}\tilde{f}, \tilde{\mathbf{v}}\Delta x\rangle_{\mathbf{P}} + \langle \tilde{f}, \breve{\mathbf{D}}\tilde{\mathbf{v}}\Delta x\rangle_{\mathbf{Q}} = \langle \tilde{f}, \breve{\mathbf{B}}\tilde{\mathbf{v}}\rangle$. Here, matrices $\mathbf{P}$ and $\mathbf{Q}$ are the weighting, strictly diagonal, matrices, that approximate the inner products as numerical . It is clear then that these matrices should be positive-definite (because they are strictly diagonal). This is the same as requesting for their values (the weights) to be strictly positive.

3 The Castillo–Blomgren–Sanchez (CBS) Algorithm

The CBS algorithm is a modification on the CRS algorithm that focuses on computing the weights as the solution to a constrained linear optimization (CLO) problem, rather than as a solution to a system of linear algebraic equations.

In the CBS algorithm, we propose to exploit the logical equivalence, between a system of linear algebraic equations and a CLO problem. This equivalence is mathematically described in [9].

Specifically, given the relatively small size of this problem's potential instances, as well as its linearity, we use the Simplex Method, to try to solve a modified form of the system producing the weights. However, as already discussed, the solution vector in the original $\mathbf{q}$-system built by the CRM is not exclusively comprised of the target weights, but it includes the $\{\lambda_i\}_{i=1}^{(k/2)-1}$ scalars.

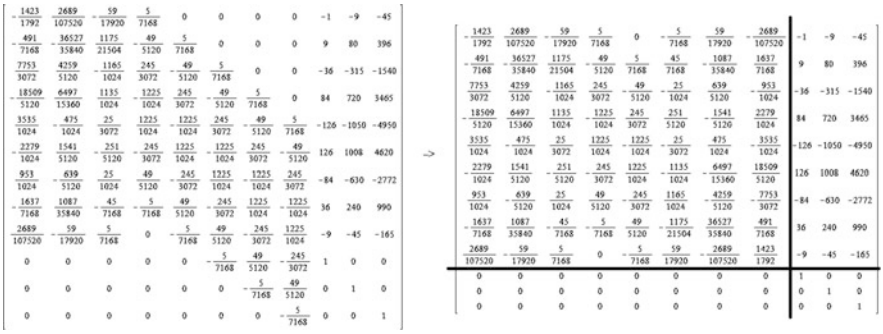


Fig. 1 Proposed modification to the CRS algorithm yielding the CBS algorithm. The CRM algorithm constructs a matrix $\mathbf{\Pi}$ in order to compute the weights (*left*). However, in the CBS algorithm, this matrix is constructed differently, as in the *right*. We call this matrix $\mathbf{\Phi}$, instead

The CBS algorithm proposes a modification of this system, based on the permutation of the elements of kernels of the boundary and near-the-boundary nodes, to construct a different $\mathbf{\Pi}$ matrix, and a different RHS vector, thus reducing the dimensionality of this system, by getting rid of the $\{\lambda_i\}_{i=1}^{(k/2)-1}$. Figure 1 renders the idea of this modification.

We have replaced the columns that were preventing the decoupling of the $\{\lambda_i\}_{i=1}^{(k/2)-1}$, by an arrangement of the approximating coefficients near and at the boundary. We denote this new matrix, or better stated, set of rows, as $\mathbf{\Phi}$.

Once the matrix has been modified like this, we define the following CLO problem, Let

$$\tilde{\mathbf{h}} = \left[-1\ 0\ 0\ 0\ 0 - \frac{5}{7168}\ \frac{159}{17920} - \frac{7621}{107520}\ \frac{30251}{26880}\right]^T. \tag{4}$$

Let $\mathbf{K}$ be the matrix whose columns are the computed elements for a rational basis of the kernel of the Vandermonde matrices that are different than 1 or 0. That is, for $k = 8$, let:

$$\mathbf{K} = \begin{bmatrix} -1 & -9 & -45 \\ 9 & 80 & 396 \\ -36 & -315 & -1540 \\ 84 & 720 & 3465 \\ -126 & -1050 & -4950 \\ 126 & 1008 & 4620 \\ -84 & -630 & -2772 \\ 36 & 240 & 990 \\ -9 & -45 & -165 \end{bmatrix} \tag{5}$$

Also, let $\boldsymbol{\lambda} \in \mathbb{R}^{\dim(\ker(\mathbf{V}_i))}$, where $\mathbf{V}_i$ is the i-th Vandermonde matrix to approximate the operator near and at the boundary, and $i \in [1, k/2-1]$ to be defined as: $\boldsymbol{\lambda} = [\lambda_1 \lambda_2 \lambda_3]^T$. Finally, define the new RHS for the modified system, as:

$$\boldsymbol{\Lambda} \triangleq (-1)\mathbf{K}\boldsymbol{\lambda} + \tilde{\mathbf{h}}. \tag{6}$$

In the CBS algorithm, the CLO problem we have just built can be written as:

$$\text{Find } \check{\tilde{\mathbf{q}}} \text{ such that (minimize) } \mathbf{r}_i^T \check{\tilde{\mathbf{q}}} = \min_{\tilde{\mathbf{q}} \in \mathbb{R}^k} r_i(\tilde{\mathbf{q}}) = \min_{\tilde{\mathbf{q}} \in \mathbb{R}^k} \mathbf{r}^T \tilde{\mathbf{q}} \tag{7}$$

$$\text{subject to } \tilde{\boldsymbol{\Phi}}_i \check{\tilde{\mathbf{q}}} \geq \boldsymbol{\Lambda}_i, \tag{8}$$

$$\text{with } \check{\tilde{\mathbf{q}}} \geq \mathbf{0}, \tag{9}$$

with $\mathbf{r}_i, \check{\tilde{\mathbf{q}}} \in \mathbb{R}^{k \times 1}$, $\tilde{\boldsymbol{\Phi}}_i \in \mathbb{R}^{k \times k}$, $\boldsymbol{\Lambda}_i \in \mathbb{R}^{k \times 1}$, and $i \in [3, k+1]$.

Specifically, the objective function will be the difference between any rows of the resulting matrix and its correspondent value in the RHS. If we consider row 1, for example, our **objective residual function** will be defined as:

$$\begin{aligned} r_1(\mathbf{q}) = &-\frac{1423}{1792} q_1 + \frac{2689}{107520} q_2 - \frac{59}{17920} q_3 + \frac{5}{7168} q_4 - \frac{5}{7168} q_6 + \\ &\frac{59}{17920} q_7 - \frac{2689}{107520} q_8 + \lambda_1 + 9\lambda_2 + 45\lambda_3 - 1, \end{aligned} \tag{10}$$

where, as previously stated, the $\{\lambda_i\}_{i=1}^{(k/2)-1}$ values are known. The tilde symbolize the new matrix, without the objective function.

4 Results

The results were computed by integrating the CBS algorithm into the Mimetic Methods Toolkit (MTK), a C++11 software library for mimetic numerical methods [10]. For this stage we used the GLPK to solve the linear programming problems. The GLPK (GNU Linear Programming Kit) package is a software library used to solve large-scale linear programming problems. Figure 2 visually renders the first set of results.

These results are computed considering the concept of a mimetic tolerance. Let τ denote the **mimetic threshold** which can be interpreted as a measure of how mimetic the operator can get while preserving a uniform order of numerical accuracy. Specifically, we let τ be used as a surplus quantity in the linear programming problem.

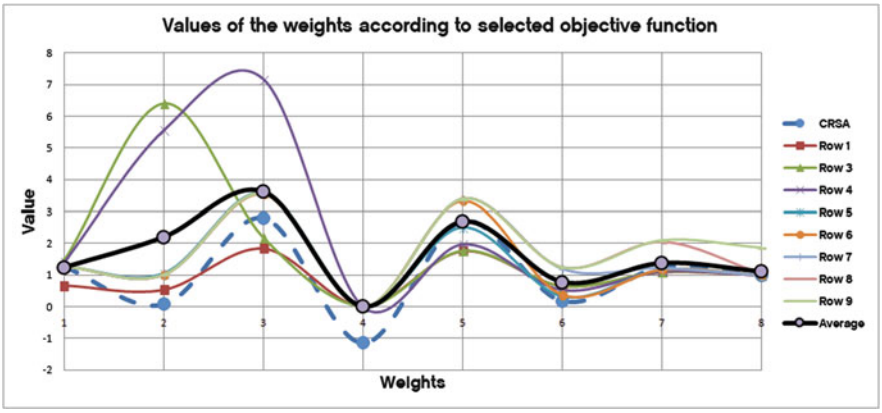


Fig. 2 Computed value of the weights according to the selected objective function. For this figure, an eight-order divergence was built. We also plot the average of all of the values, and the values using the CRS algorithm. It can be seen that q_4 is negative for this case, but through the CBS algorithm is then made equal to τ

Figure 2 renders the values of the weights according to the selected objective function. These values are given in Table 1. Table 1 shows, at the bottom row, the values produced by executing the CBS algorithm with the constraint of $\mathbf{q} > 0$ excluded. These are claimed to be the real mimetic values since these satisfy Gauss' Extended Theorem the best. However, these are not positive-definite, which is why we must include this constraint on the CBS algorithm. We then select different objective functions and compute the weights. The relative error is included in the last column. We can see that in the case of the row number 2, we can not compute any feasible set. On the other cases, the negative weights become equal to the mimetic threshold, τ, which for this case was set to 1.00E−06. Table 2 generalizes these results for higher orders of accuracy. We set $\tau = 1.00\text{E}{-}06$. We executed the CRS algorithm to construct operators of order 8, 10, and 12. Computationally speaking, the construction of higher orders involves a multi-scale problem since the involved Vandermonde matrices include terms that spam k orders of numerical magnitude. We can see that the CRS algorithm yields more negative values as we increase k. However, through the CBS algorithm, we are capable to make the negative weight with the highest numerical value equal to the mimetic threshold, and from there, other weights turn to a positive value with a numerical magnitude inversely proportional to that of its negative counterpart from the CRS algorithm.

Table 1 Computed weights according to the selected objective function. The minimal relative error is boldfaced and it is achieved when using row 5 to compute the positive-definite weights through the CBS. These weights are rendered in Fig. 2

Row	q_1	q_2	q_3	q_4	q_5	q_6	q_7	q_8	Relative error
1	0.664988	0.526825	1.84592	1.00E−06	1.76226	0.647214	1.08873	0.995654	0.436981174
2				0					
3	1.45181	6.39792	2.17188	1.00E−06	1.75818	0.643264	1.09129	0.995373	1.453166033
4	1.40466	5.55803	7.16349	1.00E−06	1.97418	0.532991	1.11934	0.994014	1.588625361
5	1.27771	1.05619	3.58569	1.00E−06	2.509	0.32137	1.1698	0.99173	**0.386115352**
6	1.27639	1.0083	3.54084	1.00E−06	3.33234	0.370455	1.16703	0.991558	0.40073651
7	1.27647	1.01122	3.54322	1.00E−06	3.38597	1.19434	1.21835	0.988425	0.462710997
8	1.27647	1.01131	3.54334	1.00E−06	3.38281	1.24567	2.04211	1.04293	0.501305752
9	1.27647	1.01126	3.54331	1.00E−06	3.38267	1.2424	2.09333	1.86018	0.541913298
Average	1.238121	2.197631875	3.61721125	0.000001	2.68592625	0.774713	1.3737475	1.107483	0.586613303
Constraint off	1.251	0.05914	2.783	−1.157	2.69	0.1667	1.236	0.9708	–

Table 2 Results of the CBS algorithm versus the CRS algorithm to construct a higher-order divergence operator. Boldfaced quantity highlight the occurrences of negative weights when using the CRM and how the CBS corrects these values them. For this second set of results, $\tau = 1.00\text{E}{-}06$

$k = 12$	CBS	CRS	$k = 10$	CBS	CRS	$k = 8$	CBS	CRS
q_1	1.58778	1.3534	q_1	1.39352	1.30472	q_1	1.27647	1.25125
q_2	**9.67127**	**−0.90544**	q_2	**3.23107**	**−0.3948**	q_2	1.01126	0.0591387
q_3	13.4287	6.91854	q_3	7.1187	4.49029	q_3	3.54331	2.78342
q_4	**5.09271**	**−11.7959**	q_4	**1.00E−06**	**−4.8847**	q_4	**1.00E−06**	**−1.15721**
q_5	20.2708	20.9737	q_5	9.63071	7.89854	q_5	3.38267	2.69024
q_6	**1.00E−06**	**−21.9175**	q_6	**0.31868**	**−4.6581**	q_6	1.2424	0.166692
q_7	20.0599	20.4101	q_7	6.69451	4.1963	q_7	2.09333	1.23571
q_8	**5.36898**	**−11.0229**	q_8	**3.70077**	**−0.1882**	q_8	1.86018	0.970756
q_9	13.1848	6.30627	q_9	4.60898	1.26227			
q_{10}	**10.2919**	**−0.58300**	q_{10}	4.25651	0.973915			
q_{11}	11.0256	1.28651						
q_{12}	10.3	0.976215						

5 Summary, Concluding Remarks, and Directions of Future Work

We have discussed further results of the CBS algorithm, which is an algorithm created to circumvent a limitation on previous methods (CGM, CRM) and algorithms (CRS) to construct higher order mimetic operators. Results were positive, and the CBS algorithm can successfully compute positive weights where other methods can not. If we chose the **natural lexicographical order** to index the elements of the 2D staggered grid, we can build the 2D counterparts to higher-order mimetic operators, as follows [9]: $\breve{\mathbf{G}}^k_{xy} = [\mathbf{G}_x\ \mathbf{G}_y]^T$, and $\breve{\mathbf{D}}^k_{xy} = \begin{bmatrix}\mathbf{D}_x\ \mathbf{D}_y\end{bmatrix}$, where each auxiliary discretization matrix along each spatial dimension can be computed from the 1D mimetic operator, as follows $\mathbf{G}_x = \hat{\mathbf{I}}^T_n \otimes \breve{\mathbf{G}}^k_x$, $\mathbf{G}_y = \breve{\mathbf{G}}^k_y \otimes \hat{\mathbf{I}}^T_m$, $\mathbf{D}_x = \hat{\mathbf{I}}_n \otimes \breve{\mathbf{D}}^k_x$, $\mathbf{D}_y = \breve{\mathbf{D}}^k_y \otimes \hat{\mathbf{I}}_m$. Similarly, we can compute the 3D operators: $\breve{\mathbf{G}}^k_{xyz} = [\mathbf{G}_x\ \mathbf{G}_y\ \mathbf{G}_z]^T$, where: $\mathbf{G}_x = \hat{\mathbf{I}}^T_n \otimes \hat{\mathbf{I}}^T_m \otimes \breve{\mathbf{G}}^k_x$, $\mathbf{G}_y = \hat{\mathbf{I}}^T_n \otimes \breve{\mathbf{G}}^k_y \otimes \hat{\mathbf{I}}^T_k$, and $\mathbf{G}_z = \breve{\mathbf{G}}^k_z \otimes \hat{\mathbf{I}}^T_m \otimes \hat{\mathbf{I}}^T_k$. Finally, $\breve{\mathbf{D}}^k_{xyz} = [\mathbf{D}_x\ \mathbf{D}_y\ \mathbf{D}_z]$, with: $\mathbf{D}_x = \hat{\mathbf{I}}_n \otimes \hat{\mathbf{I}}_m \otimes \breve{\mathbf{D}}^k_x$, $\mathbf{D}_y = \hat{\mathbf{I}}_n \otimes \breve{\mathbf{D}}^k_y \otimes \hat{\mathbf{I}}_k$, and $\mathbf{D}_z = \breve{\mathbf{D}}^k_z \otimes \hat{\mathbf{I}}_m \otimes \hat{\mathbf{I}}_k$. Our immediate future work is to implement the construction of higher-order operators using the CBS to explore their accuracy in solving problems of a physical nature. These implementations will be integrated in the MTK.

References

1. J.E. Castillo, J.M. Hyman, M.J. Shashkov, S. Steinberg, The sensitivity and accuracy of fourth order finite difference schemes on nonuniform grids in one dimension. Comput. Math. Appl. **30**(8), 41–55 (1995)
2. J.B. Runyan, A novel higher order finite difference time domain method based on the Castillo-Grone mimetic curl operator with application concerning the time-dependent Maxwell equations. Master's thesis, San Diego State University, San Diego, CA, 2011
3. J.E. Castillo, G.F. Miranda, *Mimetic Discretization Methods*, 1st edn. (CRC Press, Boca Raton, 2013)
4. J. De la Puente, M. Ferrer, M. Hanzich, J.E. Castillo, J.M. Cela, Mimetic seismic wave modeling including topography on deformed staggered grids. Geophysics **79**, T125–T141 (2014)
5. J.E. Castillo, R.D. Grone, A matrix analysis approach to higher-order approximations for divergence and gradients satisfying a global conservation law. SIAM J. Matrix Anal. Appl. **25**, 128–142 (2003)
6. H.O. Kreiss, G. Scherer, Finite element and finite difference methods for hyperbolic partial differential equations, in *Mathematical Aspects of Finite Elements in Partial Differential Equations*, ed. by C. De Boor (Academic Press, New York, 1974), pp. 195–212
7. P. Olsson, Summation by parts, projections, and stability i. Math. Comput. **64–211**, 1035–1065 (1995)
8. E. Sanchez, J. Castillo, An algorithmic study of the construction of higher-order one-dimensional Castillo-Grone mimetic gradient and divergence operators. Technical Report, San Diego State University, San Diego, 2013
9. E.J. Sanchez, C.P. Paolini, J.E. Castillo, The Mimetic Methods Toolkit: an object-oriented API for Mimetic Finite Differences. J. Comput. Appl. Math. **270**, 308–322 (2014). ISSN 0377-0427. http://dx.doi.org/10.1016/j.cam.2013.12.046; http://www.sciencedirect.com/science/article/pii/S037704271300719X

Exponential Convergence of Simplicial *hp*-FEM for H^1-Functions with Isotropic Singularities

Christoph Schwab

Abstract For functions $u \in H^1(\Omega)$ in an open, bounded polyhedron $\Omega \subset \mathbb{R}^d$ of dimension $d = 1, 2, 3$, which are analytic in $\overline{\Omega}\backslash\mathscr{S}$ with point singularities concentrated at the set $\mathscr{S} \subset \overline{\Omega}$ consisting of a finite number of points in $\overline{\Omega}$, the exponential rate $\exp(-b\sqrt[d+1]{N})$ of convergence of *hp*-version continuous Galerkin finite element methods on families of regular, simplicial meshes in Ω can be achieved. The simplicial meshes are assumed to be geometrically refined towards $\mathscr{S}$ and to be shape regular, but are otherwise unstructured.

1 Introduction

Many nonlinear PDEs admit solutions with are analytic but exhibit isolated point singularities at a set $\mathscr{S}$. We mention only nonlinear Schrödinger equations with self-focusing, density functional models in electron structure calculations (e.g. [2, 4, 10] and the references there), nonlinear parabolic PDEs with critical growth (e.g. [19] and the references there), or continuum models of crystalline solids with isolated point defects. (e.g. [17] and the references there).

The *hp*-version of the Finite Element Method ("*hp*-FEM" for short) is known to deliver exponential convergence for such problems; we refer to [8, 12, 20] for such results in space dimension $d = 1$, to [23] and the references there for theory in $d = 1, 2$ space dimensions, to [21] for exponential convergence of conforming *hp*-FEM on geometric meshes of hexahedra, and to [9, 11] for details on implementational aspects and numerical experiments.

In the present note, we state an exponential convergence result for C^0-conforming *hp*-FEM on regular, simplicial mesh families with *isotropic, geometric refinement* towards the singular point(s) $c \in \mathscr{S}$. These meshes are in addition required to be shape-regular. This type of mesh arises for example in adaptive bisection-tree refinements. Specifically, for singular solutions $u \in H^1(\Omega)$ where $\Omega \subset R^d$, $d = 2, 3$ belonging to a countably normed space with radial weights introduced in [7],

C. Schwab (✉)
SAM, ETH, CH-8092 Zürich, Switzerland
e-mail: schwab@math.ethz.ch

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_40

we construct a continuous, piecewise polynomial interpolant $I^{hp}u$ which exhibits exponential convergence: there exist constants $b, C > 0$ which depend on Ω and on u, in general, such that

$$\|u - I^{hp}u\|_{H^1(\Omega)} \leq C \exp(-bN^{1/(d+1)}) \ . \tag{1}$$

Here, $d = 2, 3$ denotes the space dimension and N denotes the number of degrees of freedom in the hp-FE approximation. This rate coincides, in space dimensions $d = 1, 2$, with the bounds obtained in [12, 13] for corner singularities on structured geometric meshes, and in [25] on unstructured, simplicial geometric meshes. In space dimension $d = 3$, this generalizes the hp-approximations in [22, Sect. 5.2.2] in the case of vertex singularities to unstructured, tetrahedral meshes with geometric refinement towards $\mathscr{S}$.

The structure of the note is as follows: in Sect. 2, we introduce a model problem, the geometric assumptions on the singularities, and precise the analytic regularity in countably normed, weighted Sobolev spaces with radial weight functions. In Sect. 3, we introduce the hp-version FEM; we specify in particular the assumptions on the simplicial, geometric meshes, on the elemental polynomial degrees, and on the definition of the hp FE spaces. Section 4 outlines a proof of the exponential convergence bound in $H^1(\Omega)$ on regular, simplicial geometric mesh families, with details given in [24].

2 Analytic Regularity

Analytic regularity is characterized in countably normed weighted Sobolev spaces which have been introduced and used in exponential convergence estimates in a number of references; we only mention [1, 7, 12–15] and the references there. Here, we denote by $\mathscr{S} \subset \overline{\Omega}$ the set of singular points c; we consider solutions $u \in H^1(\Omega)$ which are smooth in $\overline{\Omega}\backslash\mathscr{S}$ so that the singular support of u coincides with $\mathscr{S}$. We work under the following separation assumption on $\mathscr{S}$.

$$\text{The singular set } \mathscr{S} \text{ consist of a finite number of isolated points } c \in \overline{\Omega}. \tag{2}$$

Assumption (2) implies $\varepsilon(\Omega, \mathscr{S}) := \min\{\text{dist}(c, c') : c, c' \in \mathscr{S}, c \neq c'\} > 0$, and allows to partition the set Ω into $|\mathscr{S}|$ many disjoint neighborhoods ω_c of the singularities $c \in \mathscr{S}$. We set $\Omega_{\mathscr{S}} := \bigcup_{c\in\mathscr{S}} \omega_c$ and denote $\Omega_0 := \Omega\backslash\overline{\bigcup_{c\in\mathscr{S}} \omega_c}$.

We characterize analytic regularity of singular solutions by weighted Sobolev spaces. To define these, we introduce distance functions:

$$r_c(x) = \text{dist}(x, c) \ , \qquad x \in \Omega \ , \quad c \in \mathscr{S} \ . \tag{3}$$

With $c \in \mathscr{S}$ we collect all singular exponents $\beta_c \in \mathbb{R}$ in the "multi-exponent"

$$\underline{\beta} = \{\beta_c : c \in \mathscr{S}\} \in \mathbb{R}^{|\mathscr{S}|} . \tag{4}$$

We assume ($\underline{\beta} > s$ and $\underline{\beta} \pm s$ being understood componentwise for $s \in \mathbb{R}$) that in space dimension $d = 3$ (the results and ranges of weight exponents in space dimension $d = 2$ are analogous; cp. [7])

$$\underline{b} := -1 - \underline{\beta} \in (0, 1/2) , \quad \text{i.e.} \quad -1 > \underline{\beta} > -3/2 . \tag{5}$$

Consider the semi-norms (cp. [7, Definition 6.2 and Eq. (6.9)], [1, 14]),

$$|u|^2_{M^k_{\underline{\beta}}(\Omega)} = |u|^2_{H^k(\Omega_0)} + \sum_{c\in\mathscr{S}} \sum_{\substack{\alpha\in\mathbb{N}_0^d \\ |\alpha|=k}} \left\| r_c^{\beta_c+|\alpha|} \mathsf{D}^\alpha u \right\|^2_{L^2(\omega_c)} , \quad k \in \mathbb{N}_0 . \tag{6}$$

We define the norm $\|u\|_{M^m_{\underline{\beta}}(\Omega)}$ by $\|u\|^2_{M^m_{\underline{\beta}}(\Omega)} = \sum_{k=0}^m |u|^2_{M^k_{\underline{\beta}}(\Omega)}$. Here, $|u|_{H^m(\Omega_0)}$ is the usual Sobolev semi-norm of integer order m on Ω_0, and D^α denotes the partial derivative of order $\alpha \in \mathbb{N}_0^d$. The space $M^m_{\underline{\beta}}(\Omega)$ is the weighted Sobolev space obtained as the closure of $C_0^\infty(\Omega)$ with respect to the norm $\|\cdot\|_{M^m_{\underline{\beta}}(\Omega)}$. Under (5), for $\Omega \subset \mathbb{R}^3$ holds $M^2_{\underline{\beta}}(\Omega) \subset H^{1+\theta}(\Omega)$ for some $\theta > 1/2$: choose $\theta(\underline{\beta}) = 1 - \beta_m - \varepsilon$ in [14, Theorem 3.5] with $\beta_m := -1 - \beta_c \in (0, 1/2)$, and $0 < \varepsilon < 1/2 - \beta_m = 3/2 + \beta_c$. In dimension $d = 2$, i.e. for $\Omega \subset \mathbb{R}^2$, we find under (5) that $M^2_{\underline{\beta}}(\Omega) \subset H^{1+\theta}(\Omega)$ for some $\theta > 0$, so that for $d = 2$ holds $M^2_{\underline{\beta}}(\Omega) \subset C^0(\overline{\Omega})$ with continuous embedding. With $M^k_{\underline{\beta}}(\Omega)$ in (6), the analytic class in [7, Definition 6.3] reads

$$A_{\underline{\beta}}(\mathscr{S};\Omega) = \left\{ u \in \bigcap_{k\geq 0} M^k_{\underline{\beta}}(\Omega) : \exists\, C_u > 0 \text{ s.t. } |u|_{M^k_{\underline{\beta}}(\Omega)} \leq C_u^{k+1} k! \;\forall\, k \in \mathbb{N}_0 \right\} . \tag{7}$$

Several application problems have solutions in this class, cp. [10] for electron structure models, [1, 7] for elliptic problems in polyhedral domains.

3 *hp*-Finite Element Spaces

For two parameters $0 < \kappa, \sigma < 1$, we consider families $\mathfrak{M}_{\kappa,\sigma} = \{\mathscr{M}^{(\ell)}\}_{\ell\geq 1}$ of geometric meshes $\mathscr{M}^{(\ell)} \in \mathfrak{M}_{\kappa,\sigma}$. The meshes $\mathscr{M} \in \mathfrak{M}_{\kappa,\sigma}$ are regular partitions of the polyhedron Ω into a finite number of open simplices (triangles in space dimension $d = 2$, tetrahedra in space dimension $d = 3$) $T \in \mathscr{M}^{(\ell)}$. Here, regular

means that for every $\mathscr{M} \in \mathfrak{M}_{\kappa,\sigma}$, the intersections of closures of any two distinct $T, T' \in \mathscr{M}$ are either empty, a vertex v, an entire edge e or an entire face f. We assume the family $\mathfrak{M}_\sigma$ to be *uniformly κ-shape regular*: for a simplex $T \in \mathscr{M}^{(\ell)}$, we denote by $h_T = \mathrm{diam}(T)$ its diameter and by $\rho_T = \sup\{\rho > 0 | B_\rho \subset T\}$, the radius of the largest ball B_ρ that can be inscribed into T. For a regular, simplicial mesh $\mathscr{M}$, the (nondimensional) shape parameter $\kappa(\mathscr{M}) = \max\{h_T/\rho_T | T \in \mathscr{M}\}$ is well defined. A collection $\{\mathscr{M}^{(\ell)}\}_{\ell\geq 1}$ of regular, simplicial meshes is called *κ-shape regular*, if $\sup_{\ell\geq 1} \kappa(\mathscr{M}^{(\ell)}) \leq \kappa < \infty$.

Each simplex $T \in \mathscr{M}_\ell$ is the image of the reference simplex, defined by $\hat{T} := \{\hat{x} \in \mathbb{R}^d : \hat{x}_i > 0, \sum_{i=1}^d \hat{x}_i < 1\}$, under the affine element map F_T, i.e.

$$T = F_T(\hat{T}), \quad T \ni x = F_T(\hat{x}) = B_T\hat{x} + b_T, \ \ \hat{x} \in \hat{T} \, . \tag{8}$$

For a regular, simplicial triangulation $\mathscr{M}$ of Ω with $\kappa(\mathscr{M}) < \infty$, the affine element maps are nondegenerate: the Jacobians $B_T = DF_T$ in (8) are nonsingular, and $\|B_T\|_F \leq \kappa(\mathscr{M})$, see, e.g., [3, Sect. II]. The reference simplex $\hat{T}$ is contained in the unit cube $\hat{K} = (0,1)^d$; with each $T \in \mathscr{M}$, we associate a parallelepiped via $K_T = F_T(\hat{K})$ and *assume* that $K_T \subset \Omega$. Here, for $T \in \mathscr{M}$ the local polynomial approximation space $\mathbb{P}^p(T) = \mathrm{span}\{x^\alpha : |\alpha| \leq p\}$ is the linear space of all multivariate polynomials on $T \in \mathscr{M}$ whose total degree does not exceed p. The space $\mathbb{P}^p(T)$ is invariant under the affine mapping F_T, i.e. $u \in \mathbb{P}^p(T)$ if and only if $\hat{u} := u \circ F_T \in \mathbb{P}^p(\hat{T})$. On parallelepipeds K, $\mathbb{Q}^p(K)$ is the affine image of $\mathbb{Q}^p(\hat{K})$, $\hat{K} = \hat{I}^d$ with $\hat{I} = (0,1)$,

$$\mathbb{Q}^p(\hat{K}) = \mathrm{span}\,\{\,\hat{x}^\alpha \ : \ 0 \leq \alpha_i \leq p, \ 1 \leq i \leq d\,\} \, . \tag{9}$$

For each parallelepiped K_T associated with a tetrahedron $T \in \mathscr{M}$ (resp. a triangle if $\Omega \subset \mathbb{R}^2$), with associated affine element mapping $F_T : \hat{K} \to K_T$ and polynomial degree $p \geq 0$, we set

$$\mathbb{Q}^p(K_T) = \left\{ v \in L^2(K_T) : (v|_{K_T} \circ F_T) \in \mathbb{Q}^p(\hat{K}) \right\} \, . \tag{10}$$

For polynomial degree $p \geq 1$, and for a family of regular, simplicial triangulations $\mathscr{M}^{(\ell)} \in \mathfrak{M}_{\kappa,\sigma}$ of Ω, we introduce the finite element spaces

$$S^p(\mathscr{M}^{(\ell)}) = \left\{ u \in H^1(\Omega) \ : \ u|_T \in \mathbb{P}^p(T), \ T \in \mathscr{M}^{(\ell)} \right\} \, . \tag{11}$$

hp-FEM are obtained when the level ℓ of geometric mesh refinement is tied to the polynomial degree p.

Mesh Layers A key ingredient in exponential convergence proofs of *hp*-FEM is *geometric mesh refinement* towards the set $\mathscr{S}$ of singularities. We call a regular, simplicial mesh family $\mathfrak{M}_{\kappa,\sigma} = \{\mathscr{M}^{(\ell)}\}_{\ell\geq 1}$ *σ-geometrically refined towards $\mathscr{S}$* $\subset$

Ω if there exists $0 < \sigma < 1$ such that for every $T \in \mathscr{M}^{(\ell)} : \overline{T} \cap \mathscr{S} = \emptyset, \ell = 1, 2, \ldots$ holds

$$0 < \sigma < \rho(T; \mathscr{S}) := \frac{\operatorname{diam}(T)}{\operatorname{dist}(T, \mathscr{S})} < \frac{1}{\sigma} . \tag{12}$$

We tag members of a σ-geometric family $\mathfrak{M}_{\kappa,\sigma}$ by a subscript σ, i.e. we write $\mathscr{M}_\sigma^{(\ell)}$.

Proposition 1 *Consider a regular, nested and σ-geometrically refined, κ-shape regular simplicial mesh family $\mathfrak{M}_{\kappa,\sigma}$ in Ω. Then, all elements $T \in \mathscr{M}_\sigma^{(\ell)}$ for every $\ell \geq 1$, can be grouped in* mesh-layers*: there exists a partition*

$$\bigcup_{\ell \geq 1} \mathscr{M}_\sigma^{(\ell)} = \mathfrak{L}_1 \,\dot\cup\, \mathfrak{L}_2 \,\dot\cup \ldots . \tag{13}$$

and a constant $c(\mathfrak{M}_{\kappa,\sigma}) \geq 1$ with

$$\forall k \geq 1 : \qquad \#(\mathfrak{L}_k) \leq c(\mathfrak{M}_{\kappa,\sigma}) \tag{14}$$

and such that, for every $T \in \mathfrak{L}_k$ and every $k \geq 1$,

$$0 < \frac{1}{c(\mathfrak{M}_{\kappa,\sigma})} \leq \frac{\operatorname{diam}(T)}{\sigma^k} \leq c(\mathfrak{M}_{\kappa,\sigma}) . \tag{15}$$

Proof The proof is by induction over ℓ.

Based on Proposition 1, for ℓ sufficiently large, there exists a constant $c_{\mathfrak{T}}(\kappa, \sigma) > 0$ independent of ℓ, so that every mesh $\mathscr{M}_\sigma^{(\ell)} \in \mathfrak{M}_{\kappa,\sigma}$ may be partitioned into

$$\mathscr{M}_\sigma^{(\ell)} = \mathfrak{O}_\sigma^{(\ell)} \,\dot\cup\, \mathfrak{T}_\sigma^{(\ell)} , \tag{16}$$

where

$$\mathfrak{O}_\sigma^{(\ell)} := \mathfrak{O}_\sigma^{(\ell-1)} \,\dot\cup\, \mathfrak{L}_\ell = \mathfrak{L}_1 \,\dot\cup\, \mathfrak{L}_2 \,\dot\cup \ldots \,\dot\cup\, \mathfrak{L}_\ell ,$$

and such that for all ℓ holds

$$\mathscr{S} \subset \bigcup_{T \in \mathfrak{T}_\sigma^{(\ell)}} \overline{T} , \qquad \operatorname{dist}(\mathscr{S}, \mathfrak{O}^{(\ell)}) \geq c_{\mathfrak{T}} \sigma^\ell . \tag{17}$$

The terminal mesh layers $\mathfrak{T}_\sigma^{(\ell)} \subset \mathscr{M}_\sigma^{(\ell)}$ in (16) satisfy the following properties.

Proposition 2 *There exists a constant $c_{\mathfrak{T}}(\kappa, \sigma) > 0$ such that for every $\mathscr{M}_\sigma^{(\ell)} \in \mathfrak{M}_{\kappa,\sigma}$, the set $\mathfrak{T}_\sigma^{(\ell)}$ has the following properties: for all $\ell \geq 1$ holds* (1) $\#(\mathfrak{T}_\sigma^{(\ell)}) \leq$

$c_{\mathfrak{T}}(\kappa,\sigma)$, (2) $\forall c \in \mathscr{C} : |\mathfrak{T}_\sigma^{(\ell)} \cap \omega_c| \le c_{\mathfrak{T}}(\kappa,\sigma)\sigma^{d\ell}$, (3) $\forall T \in \mathfrak{T}_\sigma^{(\ell)} : h_T \le c_{\mathfrak{T}}(\kappa,\sigma)\sigma^{\ell}$.

4 Exponential Convergence

4.1 Statement of the Exponential Convergence Result

Theorem 1 *Let $u \in M^2_{-1-\underline{\beta}}(\Omega)$ with weight vector $\underline{\beta}$ as in (5) in a bounded polyhedron $\Omega \subset \mathbb{R}^d$, $d = 2, 3$.*

Then, for every sequence $\mathfrak{M}_{\kappa,\sigma}(\mathscr{S})$ of nested, regular simplicial meshes in Ω which are σ-geometrically refined towards $\mathscr{S}$ and which are κ shape-regular, there exist continuous projectors $\Pi^p_{\kappa,\sigma} : M^2_{-1-\underline{\beta}}(\Omega) \to S^p(\mathscr{M}_\sigma^{(p)})$ and constants $b, C > 0$ (depending on κ, C_u, d_u in (7) and on σ) such that there holds the error bound

$$\left\| u - \Pi^p_{\kappa,\sigma} u \right\|_{H^1(\Omega)} \le C \exp(-b \sqrt[d+1]{N}) \,. \tag{18}$$

Here, $N = \dim(S^p(\mathscr{M}_\sigma^{(p)})) = O(p^{d+1})$.

4.2 Outline of Proof

The proof of the *approximation result* Theorem 1 is based on constructing the projectors $\Pi^p_{\kappa,\sigma}$; the construction in [24] consists in several steps and is detailed there for $d = 3$, the case $d = 2$ being a (minor) modification. First, from [22, Sect. 5] we obtain a family of univariate *hp*-projections with error bounds which are explicit in the polynomial degree as well as in the regularity of the functions to be approximated. A corresponding family of polynomial projectors on the unit cube $\hat{K} = (0,1)^3$ with analogous consistency error bounds is then obtained as in [22, Sect. 5] by tensorization and scaling. We use these bounds for a tetrahedron $T \in \mathfrak{O}_\sigma^{(\ell)} \subset \mathscr{M}_\sigma^{(\ell)} \in \mathfrak{M}_{\kappa,\sigma}$ as follows. By Proposition 1, $T \in \mathfrak{L}_k$ for some $1 \le k \le \ell - 1$. The (up to orientation) unique parallelepiped $K_T = F_T(\hat{K})$ associated with $T \in \mathfrak{L}_k$ has the same scaling properties as T, in particular (15) also holds for K_T. For u belonging to the analytic class (7) with weight vector satisfying (5), $u \in C^0(\overline{\Omega}) \cap C^\infty(\overline{\Omega}\backslash\mathscr{S})$. For $T \in \mathfrak{O}_\sigma^{(\ell)}$, the pullback $\hat{u}_T = u|_{K_T} \circ F_T$ satisfies on $\hat{K}$ the same analytic derivative bounds as $u|_T \circ F_T$ on $\hat{T}$ (with possibly larger constant C_u, depending on κ, but independent of ℓ and of T). The tensorized *hp* interpolation operator from [22], [24, Proposition 3] on $\hat{K}$ is therefore well-defined and allows to construct a polynomial approximation $\hat{u}^p_T \in \mathbb{Q}^p(\hat{K})$ with analytic consistency error bounds on $\hat{K}$; since $\hat{T} \subset \hat{K}$, and since $\mathbb{Q}^p(\hat{T}) \subset \mathbb{P}^{pd}(\hat{T})$, the pushforwards of the restrictions $\hat{u}^p_T|_{\hat{T}}$ under the affine mapping $F_T : \hat{T} \to T$ will be local polynomial

approximations of degree pd with exponential convergence estimates in $H^1(T)$. Moreover, since the tensorized interpolant is nodally exact in the vertices of $\hat{K}$, and since the set of vertices of $\hat{T}$ is a subset of the set of vertices of $\hat{K}$, *the pushforwards of* $\hat{u}_T^p|_{\hat{T}}$ *under* F_T *are nodally exact in the vertices of* T. By the continuity of $u \in A_{\underline{\beta}}(\mathscr{S};\Omega)$ on $\Omega\backslash\mathscr{S}$, the resulting global, piecewise polynomial interpolant is nodally exact (and, in particular, continuous) in all vertices of $T \in \mathfrak{O}_\sigma^{(p)}$, but has polynomial jump discontinuities across edges and (in space dimension $d = 3$) faces of $T \in \mathfrak{O}_\sigma^{(p)}$ which we remove by *polynomial trace liftings*, preserving the exponential convergence estimates. We refer to [18] and [24, Sect. 4.2] for details.

5 Concluding Remarks

We presented an exponential convergence rate (18) estimate for continuous *hp*-FE approximations on κ shape-regular, simplicial meshes with geometric refinement to analytic functions with isolated point singularities at a finite set $\mathscr{S}$ in a bounded domain $D \subset \mathbb{R}^d$, of dimension $d = 1, 2, 3$. Apart from κ-shape regularity and σ-geometric mesh refinement the proof did not assume further structural assumptions on the triangulations. In particular, simplicial partitions which are obtained by successive bisection tree refinement in the course of adaptive subdivisions are admissible. The approximation results imply the exponential convergence rate $\exp(-b\sqrt[3]{N})$ for second order, elliptic PDEs in polygons $D \subset \mathbb{R}^2$ (where $\mathscr{S}$ denotes the set of corners of D) which are considered, for example, in [1, 6, 15]. Theorem 1 also implies the exponential convergence rate $\exp(-b\sqrt[4]{N})$ for *hp*-approximations of electron densities in DFT, due to the quasioptimality of Galerkin approximations shown, for example, in [2, 4] and the references there. In this application, $\mathscr{S}$ denotes the set of nuclei, whose centers $c \in \mathscr{S}$ are assumed known. Unlike other approaches such as plane waves, *hp*-approximations do not, apriori, impose any specific functional form of the electron densities. Due to the locality of approximation and the separation (2) of the points $c \in \mathscr{S}$, we may apply Theorem 1 in each neighborhood ω_c implying that the total number of degrees of freedom to achieve accuracy $\varepsilon > 0$ in the norm $H^1(D)$ scales as $O(\#(\mathscr{S})|\log\varepsilon|^4)$, i.e. *linear scaling* in the number $\#(\mathscr{S})$ of nuclei and *polylogarithmic scaling* in the target accuracy ε. This is analogous to what is reported recently for discontinuous Galerkin discretizations in [16]: [24, Proposition 4] can be used as starting point of proof of an exponential convergence result on tetrahedral meshes; for geometric meshes of hexahedra, analogous results can be found in [22, Sect. 5.2.2]. Exponentially convergent quadrature algorithms for the (singular) electron-pair integrals are available in [5]. The results in the present note are confined to space dimension $d \leq 3$. The approach generalizes, however, directly to *hp*-approximations of point singularities in any dimension d with exponential rate $\exp(-b\sqrt[d+1]{N})$. Likewise, the result remains true for *linear polynomial degree vectors* (with larger constant $b > 0$

in the exponent) and, more generally, for polynomial degree vectors of bounded variation as introduced in [22]. The details will be reported elsewhere.

Acknowledgements This work is supported by grant ERC AdG STAHDPDE 247277.

References

1. I. Babuška, B.Q. Guo, Regularity of the solution of elliptic problems with piecewise analytic data. I. Boundary value problems for linear elliptic equation of second order. SIAM J. Math. Anal. **19**(1), 172–203 (1988)
2. G. Bao, G. Hu, D. Liu, An *h*-adaptive Finite Element solver for the calculation of the electronic structures. J. Comput. Phys. **231**, 4967–4979 (2012)
3. D. Braess, *Finite Elements*, 5th edn. (Cambridge University Press, Cambridge, 2011)
4. E. Cances, R. Chakir, Y. Maday, Numerical analysis of the planewave discretization of some orbital-free and Kohn-Sham models. ESAIM: Math. Model. Numer. Anal. **46**(02), 341–388 (2012)
5. A. Chernov, T. von Petersdorff, C. Schwab, Exponential convergence of *hp* quadrature for integral operators with Gevrey kernels. ESAIM: Math. Model. Numer. Anal. **45**, 387–422 (2011)
6. M. Costabel, M. Dauge, C. Schwab, Exponential convergence of *hp*-FEM for Maxwell's equations with weighted regularization in polygonal domains. Math. Models Methods Appl. Sci. **15**(4), 575–622 (2005)
7. M. Costabel, M. Dauge, S. Nicaise, Analytic regularity for linear elliptic systems in polygons and polyhedra. Math. Models Methods Appl. Sci. **22**(8), 12500–12515 (2012)
8. W. Dahmen, K. Scherer, Best approximation by piecewise polynomials with variable knots and degrees. J. Approx. Theory **26**(1), 1–13 (1979)
9. L. Demkowicz, J.J. Kurtz, D. Pardo, M. Paszynski, W. Rachowicz, A. Zdunek, Computing with hp-adaptive finite elements, in *Frontiers: Three Dimensional Elliptic and Maxwell Problems with Applications*. Chapman and Hall/CRC Applied Mathematics and Nonlinear Science Series, vol. 2 (Chapman and Hall/CRC, Boca Raton, FL, 2007)
10. S. Fournais, T.O. Sørensen, M. Hoffmann-Ostenhof, T. Hoffmann-Ostenhof, Non-isotropic cusp conditions and regularity of the electron density of molecules at the nuclei. Ann. Inst. Henri Poincaré **8**, 731–748 (2007)
11. P. Frauenfelder, hp-finite element methods on anisotropically, locally refined meshes in three dimensions with stochastic data. Ph.D. thesis, Swiss Federal Institute of Technology. (2004) http://e-collection.library.ethz.ch/
12. W. Gui, I. Babuška, The *h*, *p* and *h-p* versions of the finite element method in 1 dimension. II. The error analysis of the *h*- and *h-p* versions. Numer. Math. **49**(6), 613–657 (1986)
13. B.Q. Guo, I. Babuška, The *hp*-version of the finite element method. Part I: the basic approximation results, Part II: general results and applications. Comput. Mech. **1**, 21–41; 203–220 (1986)
14. B.Q. Guo, I. Babuška, Regularity of the solutions for elliptic problems on nonsmooth domains in $\mathbb{R}^3$. I. Countably normed spaces on polyhedral domains. Proc. R. Soc. Edinb. Sect. A **127**(1), 77–126 (1997)
15. B.Q. Guo, C. Schwab, Analytic regularity of Stokes flow on polygonal domains in countably weighted Sobolev spaces. J. Comput. Appl. Math. **119**, 487–519 (2006)
16. L. Lin, J. Lu, L. Ying, E. Weinan, Adaptive local basis set for Kohn-Sham density functional theory in a discontinuous Galerkin framework I: total energy calculation. J. Comput. Phys. **231**(4), 4515–4529 (2012)
17. M. Luskin, C. Ortner, Atomistic-to-continuum coupling, Acta Numerica **22**, 397–508 (2013)

18. R. Munoz-Sola, Polynomial liftings on a tetrahedron and applications to the *hp*-FEM in three dimensions. SIAM J. Numer. Anal. **34**, 282–314 (1997)
19. A.A. Samarskii, V.A. Galaktionov, S.P. Kurdyumov, A.P. Mikhailov, *Blow-Up in Quasilinear Parabolic Equations*. de Gruyter Expositions in Mathematics, vol. 19 (Translated from the 1987 Russian original by Michael Grinfeld and revised by the authors) (Walter de Gruyter & Co., Berlin, 1995)
20. K. Scherer, On optimal global error bounds obtained by scaled local error estimates. Numer. Math. **36**, 257–277 (1981)
21. D. Schötzau, C. Schwab, Exponential convergence for *hp*-version and spectral finite element methods for elliptic problems in polyhedra. Math. Models Methods Appl. Sci. **25**(9), 1617–1661 (2015)
22. D. Schötzau, C. Schwab, T.P. Wihler, *hp*-dGFEM for elliptic problems in polyhedra. II: exponential convergence. SIAM J. Numer. Anal. **51/4**, 2005–2035 (2013). (Extended version in Technical Report 2009–29, Seminar for Applied Mathematics, ETH Zürich)
23. C. Schwab, *p and hp-FEM* (Oxford University Press, Oxford, 1998)
24. C. Schwab, Exponential convergence of simplicial hp-FEM for H^1-functions with isotropic singularities. Technical Report 2014–15, Seminar for Applied Mathematics, ETH Zürich, 2014
25. T.P. Wihler, P. Frauenfelder, C. Schwab, Exponential convergence of the *hp*-DGFEM for diffusion problems. Comput. Math. Appl. **46**, 183–205 (2003)

Higher Order Quasi Monte-Carlo Integration in Uncertainty Quantification

Josef Dick, Quoc Thong Le Gia, and Christoph Schwab

Abstract We review recent results on dimension-robust higher order convergence rates of Quasi-Monte Carlo Petrov-Galerkin approximations for response functionals of infinite-dimensional, parametric operator equations which arise in computational uncertainty quantification.

1 Introduction

Computational uncertainty quantification (UQ) for partial differential equations (PDEs) with uncertain distributed input data gives rise, upon uncertainty parametrization, to the task of numerically approximating the solution of parametric, deterministic operator equations. Due to the distributed nature of uncertain inputs, the number of parameters (and, hence, the dimension of the parameter spaces) in such UQ problems is infinite. The computation of response statistics corresponding to distributed uncertain inputs of PDEs involves, in addition, *numerical quadrature* of all possible 'uncertain scenarios', i.e., over the entire, infinite-dimensional parameter space.

This has lead to the widespread use of sampling, in particular Monte-Carlo (MC) and Markov-Chain Monte-Carlo (MCMC) methods, in the numerical treatment of these problems: MC methods afford convergence *rates* which are independent of the parameter dimension if the variance of the integrand can be bounded independently of this dimension (the computational *work* of MC methods, of course, increases linearly with the space dimension). This *dimension robustness* of MC methods is purchased at the cost of low order: the convergence rate of simple MC methods is, generically, limited to $1/2$: variance reduction and other devices can only reduce the constant, but not the rate in the convergence bounds. At the same time, however,

J. Dick • Q.T. Le Gia
School of Mathematics and Statistics, UNSW Australia, Sydney, NSW, Australia
e-mail: josef.dick@unsw.edu.au; qlegia@unsw.edu.au

C. Schwab (✉)
Seminar for Applied Mathematics, ETH, 8092 Zürich, Switzerland
e-mail: schwab@math.ethz.ch

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_41

the *parametric regularity* required of integrand functions by MC methods is very moderate: mere square integrability with respect to a probability measure on the parameter space of the integrand functions is required for the convergence rate $1/2$ (subject to evaluations of the integrand functions being defined everywhere). In UQ for problems whose solutions exhibit propagation of singularities (as, e.g., nonlinear hyperbolic conservation laws with random inputs, see e.g. [15] and the references therein), such low regularity is the best that can be expected in general. In other applications, the parametric dependence of the response maps is considerably more regular: the solutions' dependence on the parameters is, in fact, *analytic*. This observation has been the basis for the widespread use of spectral- and polynomial chaos based numerical methods in such problems (see e.g. [1–3, 10] and the references therein).

Straightforward application of standard spectral techniques entails, however, the *curse of dimensionality*: the spectral- or even exponential convergence rate afforded by analytic parameter dependence is not realized in computational practice as soon as the number of parameters is just moderately large. High order numerical methods for infinite-dimensional problems require, therefore, a more refined analysis of analytic parameter dependence where, for dimension-independent convergence rates, the size of the domains of analyticity must increase with the problem dimension.

The purpose of the paper is to present recent advances in the analysis of higher order *Quasi Monte-Carlo (QMC)* methods, which were proposed initially in [4] (see also [5]), from [6, 7]. The presented results imply, for a large class of operator equations with random coefficients, dimensionally robust high order convergence rates. The convergence rates are, in fact, only limited by a certain sparsity measure of the uncertain input.

2 Affine Parametric Operator Equations

We present a model setting of affine parametric operator equations, and their Petrov-Galerkin (PG) discretizations, following the setting in [7]. We denote by $\mathscr{X}$ and $\mathscr{Y}$ two separable and reflexive Banach spaces over $\mathbb{R}$ (all results will hold with the obvious modifications also for spaces over $\mathbb{C}$) with (topological) duals $\mathscr{X}'$ and $\mathscr{Y}'$, respectively. By $\mathscr{L}(\mathscr{X}, \mathscr{Y}')$, we denote the set of bounded linear operators $A : \mathscr{X} \to \mathscr{Y}'$. We consider *affine-parametric operator equations*: given $f \in \mathscr{Y}'$, for every parameter sequence $\boldsymbol{y}$ in the parameter domain U find $u(\boldsymbol{y}) \in \mathscr{X}$ such that

$$A(\boldsymbol{y})\, u(\boldsymbol{y}) = f\ . \tag{1}$$

For such parametrizations, the parametric operator $A(\boldsymbol{y})$ depends on $\boldsymbol{y}$ in an affine manner, i.e., there exists a sequence $\{A_j\}_{j\geq 0} \subset \mathscr{L}(\mathscr{X}, \mathscr{Y}')$ such that

$$\forall \boldsymbol{y} \in U: \quad A(\boldsymbol{y}) = A_0 + \sum_{j\geq 1} y_j A_j\ . \tag{2}$$

After possibly rescaling, we restrict ourselves to the bounded (infinite-dimensional) parameter domain $U = [-\frac{1}{2}, \frac{1}{2}]^{\mathbb{N}}$. For every $f \in \mathscr{Y}'$ and for every $\boldsymbol{y} \in U$, we solve the parametric operator equation (1), where the operator $A(\boldsymbol{y}) \in \mathscr{L}(\mathscr{X}, \mathscr{Y}')$ is of affine parameter dependence, see (2). We associate to the operators A_j the bilinear forms $\mathfrak{a}_j(\cdot,\cdot) : \mathscr{X} \times \mathscr{Y} \to \mathbb{R}$ defined by

$$\forall v \in \mathscr{X},\ w \in \mathscr{Y}: \quad \mathfrak{a}_j(v,w) \ = \ {}_{\mathscr{Y}'}\langle A_j v, w\rangle_{\mathscr{Y}} \ , \quad j = 0, 1, 2, \dots \ .$$

Similarly, we associate with the affine-parametric operator family $A(\boldsymbol{y})$, $\boldsymbol{y} \in U$, the parametric family of bilinear forms $\mathfrak{a}(\boldsymbol{y};\cdot,\cdot) : \mathscr{X} \times \mathscr{Y} \to \mathbb{R}$, $\boldsymbol{y} \in U$, via

$$\forall v \in \mathscr{X},\ w \in \mathscr{Y}: \quad \mathfrak{a}(\boldsymbol{y}; v,w) \ = \ {}_{\mathscr{Y}'}\langle A(\boldsymbol{y}) v, w\rangle_{\mathscr{Y}} \ .$$

In order for the sum in (2) to converge, we impose

Assumption 1 *The sequence $\{A_j\}_{j\geq 0} \subset \mathscr{L}(\mathscr{X}, \mathscr{Y}')$ in (2) satisfies:*

1. *$A_0 \in \mathscr{L}(\mathscr{X}, \mathscr{Y}')$ is boundedly invertible, i.e., there exists $\mu_0 > 0$ such that*

$$\inf_{0\neq v\in\mathscr{X}} \sup_{0\neq w\in\mathscr{Y}} \frac{\mathfrak{a}_0(v,w)}{\|v\|_{\mathscr{X}}\|w\|_{\mathscr{Y}}} \geq \mu_0 \ , \quad \inf_{0\neq w\in\mathscr{Y}} \sup_{0\neq v\in\mathscr{X}} \frac{\mathfrak{a}_0(v,w)}{\|v\|_{\mathscr{X}}\|w\|_{\mathscr{Y}}} \geq \mu_0 \ .$$

2. *The* fluctuation operators *$\{A_j\}_{j\geq 1}$ are small with respect to A_0 in the following sense: there exists a constant $0 < \kappa < 2$ such that*

$$\sum_{j\geq 1} \beta_{0,j} \leq \kappa < 2 \ , \quad \textit{where} \quad \beta_{0,j} := \|A_0^{-1} A_j\|_{\mathscr{L}(\mathscr{X},\mathscr{X})} \ , \quad j = 1, 2, \dots \ . \tag{3}$$

Theorem 1 ([17, Theorem 2]) *Under Assumption 1, for every realization $\boldsymbol{y} \in U$ of the parameter vector, the affine parametric operator $A(\boldsymbol{y})$ given by (2) is boundedly invertible, with inverse bounded uniformly with respect to $\boldsymbol{y}$. In particular, for every $f \in \mathscr{Y}'$ and for every $\boldsymbol{y} \in U$, the parametric operator equation*

$$\textit{find} \quad u(\boldsymbol{y}) \in \mathscr{X}: \quad \mathfrak{a}(\boldsymbol{y}; u(\boldsymbol{y}), w) \ = \ {}_{\mathscr{Y}'}\langle f, w\rangle_{\mathscr{Y}} \quad \forall w \in \mathscr{Y} \tag{4}$$

admits a unique solution $u(\boldsymbol{y})$ which satisfies the a-priori estimate

$$\|u(\boldsymbol{y})\|_{\mathscr{X}} \ \leq \ \frac{1}{\mu} \|f\|_{\mathscr{Y}'} \ , \quad \textit{with} \quad \mu = (1 - \kappa/2)\, \mu_0 \ .$$

2.1 Single-Level and Multi-Level Algorithms

The Quantity of Interest (QoI) in our study is the expected value of a linear functional $G : \mathscr{X} \to \mathbb{R}$ of the solution u,

$$I(G(u)) = \int_U G(u(\boldsymbol{y}))\,\mathrm{d}\boldsymbol{y}.$$

In the following we discuss the approximation of the QoI by the algorithm $Q_{N,s}(G(u_s^h))$, where $Q_{N,s}$ is a quadrature rule (QMC rule) and u_s^h is the Petrov-Galerkin (PG) approximation of the dimension truncated problem, which means that the set of parameters $\boldsymbol{y} \in U$ is restricted to $\boldsymbol{y}$ of the form $(y_1, y_2, \ldots, y_s, 0, 0, \ldots)$. The combined error of this *single-level algorithm* can be expressed as

$$\begin{aligned} &I(G(u)) - Q_{N,s}(G(u_s^h)) \\ &\quad = \underbrace{I(G(u)) - I(G(u_s))}_{\text{truncation error}} + \underbrace{I(G(u_s)) - Q_{N,s}(G(u_s))}_{\text{integration error}} + \underbrace{Q_{N,s}(G(u_s - u_s^h))}_{\text{PG error}}, \end{aligned} \tag{5}$$

where 'PG error' stands for the Petrov-Galerkin discretization error. We discuss the three errors and the necessary background in the subsequent sections.

To reduce the computational cost required to achieve the same error, a novel *multi-level* algorithm was introduced and analyzed in [14]. It takes the form

$$Q_*^L(G(u)) := \sum_{\ell=0}^{L} Q_{s_\ell,N_\ell}(G(u_{s_\ell}^{h_\ell} - u_{s_{\ell-1}}^{h_{\ell-1}})) \,. \tag{6}$$

In [14] the authors considered the case where each Q_{s_ℓ,N_ℓ} is a randomly shifted lattice rule with N_ℓ points in s_ℓ dimensions, and where $u_{s_{-1}}^{h_{-1}} := 0$, whereas in the higher order version of (6) in [7] the authors used an interlaced polynomial lattice rule.

It is well-known [6] that under suitable smoothness assumptions the PG error of functionals $G(\cdot) \in \mathscr{X}_{t'}'$ admits the asymptotic error bound (as $h \to 0$ for some smoothness orders $0 < t, t'$)

$$\left|G(u(\boldsymbol{y})) - G(u^h(\boldsymbol{y}))\right| \le C\, h^{t+t'}\, \|f\|_{\mathscr{Y}_t'}\, \|G\|_{\mathscr{X}_{t'}'} \,. \tag{7}$$

2.2 Parametric and Spatial Regularity of Solutions

First we establish the regularity of the solution $u(\boldsymbol{y})$ of the parametric, variational problem (4) with respect to the parameter vector $\boldsymbol{y}$. This is important in order for the integration error of a QMC rule to admit a dimension-independent error bound.

In the following, let $\mathbb{N}_0^{\mathbb{N}}$ denote the set of sequences $\boldsymbol{\nu} = (\nu_j)_{j\geq 1}$ of non-negative integers ν_j, and let $|\boldsymbol{\nu}| := \sum_{j\geq 1} \nu_j$. For $|\boldsymbol{\nu}| < \infty$, we denote the partial derivative of order $\boldsymbol{\nu}$ of u with respect to $\boldsymbol{y}$ by

$$\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u(\boldsymbol{y}) := \frac{\partial^{|\boldsymbol{\nu}|}}{\partial_{y_1}^{\nu_1} \partial_{y_2}^{\nu_2} \cdots} u(\boldsymbol{y}), \quad \boldsymbol{y} \in U .$$

Theorem 2 ([2, 11]) Parametric Regularity. *Under Assumption 1, there exists a constant $C_0 > 0$ such that for every $f \in \mathscr{Y}'$ and for every $\boldsymbol{y} \in U$, the partial derivatives of the parametric solution $u(\boldsymbol{y})$ of the parametric operator equation (1) with affine parametric, linear operator (2) satisfy the bounds*

$$\|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u(\boldsymbol{y})\|_{\mathscr{X}} \leq C_0 \, |\boldsymbol{\nu}|! \, \boldsymbol{\beta}_0^{\boldsymbol{\nu}} \, \|f\|_{\mathscr{Y}'} \quad \text{for all } \boldsymbol{\nu} \in \mathbb{N}_0^{\mathbb{N}} \text{ with } |\boldsymbol{\nu}| < \infty ,$$

where $0! := 1$, $\boldsymbol{\beta}_0^{\boldsymbol{\nu}} := \prod_{j\geq 1} \beta_{0,j}^{\nu_j}$, with $\beta_{0,j}$ as in (3).

Spatial regularity is expressed in *scales of smoothness spaces* $\{\mathscr{X}_t\}_{t\geq 0}$, $\{\mathscr{Y}_t\}_{t\geq 0}$, i.e.

$$\mathscr{X} = \mathscr{X}_0 \supset \mathscr{X}_1 \supset \mathscr{X}_2 \supset \cdots , \quad \mathscr{Y} = \mathscr{Y}_0 \supset \mathscr{Y}_1 \supset \mathscr{Y}_2 \supset \cdots , \quad \text{and}$$
$$\mathscr{X}' = \mathscr{X}_0' \supset \mathscr{X}_1' \supset \mathscr{X}_2' \supset \cdots , \quad \mathscr{Y}' = \mathscr{Y}_0' \supset \mathscr{Y}_1' \supset \mathscr{Y}_2' \supset \cdots .$$

For self-adjoint operators, usually $\mathscr{X}_t = \mathscr{Y}_t$. For Multi-Level QMC, we require

Assumption 2 (See [7, Assumption 2]) *There exists $\bar{t} \geq 0$ such that:*

1. *For every t, t' satisfying $0 \leq t, t' \leq \bar{t}$, we have*

$$\sup_{\boldsymbol{y}\in U} \|A(\boldsymbol{y})^{-1}\|_{\mathscr{L}(\mathscr{Y}_{t'}, \mathscr{X}_t)} < \infty \quad \text{and} \quad \sup_{\boldsymbol{y}\in U} \|(A^*(\boldsymbol{y}))^{-1}\|_{\mathscr{L}(\mathscr{X}_{t'}', \mathscr{Y}_{t'})} < \infty . \tag{8}$$

   Moreover, there exist summability exponents $0 \leq p_0 \leq p_t \leq p_{\bar{t}} < 1$ such that

$$\sum_{j\geq 1} \|A_j\|_{\mathscr{L}(\mathscr{X}_t, \mathscr{Y}_{t'})}^{p_t} < \infty . \tag{9}$$

2. *Let $\boldsymbol{u}(\boldsymbol{y}) = (A(\boldsymbol{y}))^{-1} f$ and $w(\boldsymbol{y}) = (A^*(\boldsymbol{y}))^{-1} G$. For $0 \leq t, t' \leq \bar{t}$, there exist constants $C_t, C_{t'} > 0$ such that for every $f \in \mathscr{Y}_t'$ and $G \in \mathscr{X}_{t'}'$ holds*

$$\sup_{\boldsymbol{y}\in U} \|u(\boldsymbol{y})\|_{\mathscr{X}_t} \leq C_t \|f\|_{\mathscr{Y}_t'} \quad \text{and} \quad \sup_{\boldsymbol{y}\in U} \|w(\boldsymbol{y})\|_{\mathscr{Y}_{t'}} \leq C_{t'} \|G\|_{\mathscr{X}_{t'}'} .$$

   Moreover, for every $0 \leq t \leq \bar{t}$ there exists a sequence $\boldsymbol{\beta}_t = (\beta_{t,j})_{j\geq 1}$ satisfying

$$\sum_{j\geq 1} \beta_{t,j}^{p_t} < \infty ,$$

such that for every $0 \le t, t' \le \bar{t}$ *and for every* $\boldsymbol{\nu} \in \mathbb{N}_0^{\mathbb{N}}$ *with* $|\boldsymbol{\nu}| < \infty$ *we have*

$$\sup_{\boldsymbol{y}\in U} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} u(\boldsymbol{y})\|_{\mathscr{X}_t} \le C_t\, |\boldsymbol{\nu}|!\, \boldsymbol{\beta}_t^{\boldsymbol{\nu}}\, \|f\|_{\mathscr{Y}_t'},$$

$$\sup_{\boldsymbol{y}\in U} \|\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} w(\boldsymbol{y})\|_{\mathscr{Y}_{t'}} \le C_{t'}\, |\boldsymbol{\nu}|!\, \boldsymbol{\beta}_{t'}^{\boldsymbol{\nu}}\, \|G\|_{\mathscr{X}_{t'}'}\,.$$

3. *The operators* A_j *are enumerated so that the sequence* $\boldsymbol{\beta}_0$ *in* (3) *satisfies*

$$\beta_{0,1} \ge \beta_{0,2} \ge \cdots \ge \beta_{0,j} \ge \cdots\,. \tag{10}$$

2.3 Dimension Truncation

We truncate the infinite sum in (2) to s terms and solve the corresponding operator equation (1) approximately using Galerkin discretization from two dense, one-parameter families $\{\mathscr{X}^h\} \subset \mathscr{X}$, $\{\mathscr{Y}^h\} \subset \mathscr{Y}$ of subspaces of $\mathscr{X}$ and $\mathscr{Y}$: for $s \in \mathbb{N}$ and $\boldsymbol{y} \in U$, we define

$$\mathfrak{a}_s(\boldsymbol{y}; v, w) := {}_{\mathscr{Y}'}\langle A^{(s)}(\boldsymbol{y})v, w\rangle_{\mathscr{Y}}, \quad \text{with} \quad A^{(s)}(\boldsymbol{y}) := A_0 + \sum_{j=1}^{s} y_j A_j.$$

For $0 < h \le h_0$ and $\boldsymbol{y} \in U$, the dimension truncated PG-solution is defined by

$$\text{find } u_s^h(\boldsymbol{y}) \in \mathscr{X}^h: \quad \mathfrak{a}_s(\boldsymbol{y}; u_s^h(\boldsymbol{y}), w^h) = {}_{\mathscr{Y}'}\langle f, w^h\rangle_{\mathscr{Y}} \quad \forall w^h \in \mathscr{Y}^h\,. \tag{11}$$

By choosing $\boldsymbol{y} = (y_1, \ldots, y_s, 0, 0, \ldots)$, the PG discretization error bound (7) remains valid for the dimensionally truncated problem (11).

Theorem 3 ([6, Theorem 2.6]) *Under Assumption 1, for every* $f \in \mathscr{Y}'$, *for every* $G \in \mathscr{X}'$, *for every* $\boldsymbol{y} \in U$, *for every* $s \in \mathbb{N}$ *and for every* $h > 0$, *the variational problem* (11) *admits a unique solution* $u_s^h(\boldsymbol{y})$ *which satisfies*

$$|I(G(u^h)) - I(G(u_s^h))| \le C\, \|f\|_{\mathscr{Y}'}\, \|G\|_{\mathscr{X}'} \left(\sum_{j\ge s+1} \beta_{0,j}\right)^2$$

for some constant $C > 0$ *independent of* f, G *and of* s *where* $\beta_{0,j}$ *is defined in* (3). *In addition, if* (9) *and* (10) *hold with* $p_0 < 1$, *then*

$$\sum_{j\ge s+1} \beta_{0,j} \le \min\left(\frac{1}{1/p_0 - 1}, 1\right) \left(\sum_{j\ge 1} \beta_{0,j}^{p_0}\right)^{1/p_0} s^{-(1/p_0 - 1)}\,.$$

3 Quasi Monte-Carlo Quadrature

In [13], Quasi-Monte Carlo (QMC for short) rules of the form $Q_{N,s}(G(u_s^h)) = \frac{1}{N}\sum_{n=0}^{N-1} G(u_s^h(\boldsymbol{y}_n - \frac{\mathbf{1}}{2}))$, where $\boldsymbol{y}_n \in [0,1]^s$, have been used to approximate the dimension truncated integral $I(G(u_s^h))$ (see also [12]). The QMC rules considered therein are so-called randomly shifted lattice rules. Using the so-called "*product and order-dependent (POD) weights*" a convergence rate of order $\mathcal{O}(N^{-\min(1/p_0-1,1-\delta)})$, for any $\delta > 0$, with $\mathcal{O}(\cdot)$ independent of s and N was shown.

Noting that the integrand is actually analytic, the authors of [6] used *interlaced polynomial lattice rules*, as introduced in [9] (which are a special type of higher order digital net [4]), to obtain improved rates of convergence. The rules can be constructed using the fast component-by-component approach of [16]. A new function space setting was introduced in [6] based on Banach spaces with *smoothness driven product and order dependent (SPOD) weights*, to show the following result.

Theorem 4 ([6, Theorem 3.1]) *Let $s \geq 1$ and $N = b^m$ for $m \geq 1$ and prime b. Let $\boldsymbol{\gamma} = (\gamma_j)_{j\geq 1}$ be a sequence of positive numbers, let $\overline{\boldsymbol{\gamma}}_s = (\gamma_j)_{1\leq j\leq s}$, and assume*

$$\exists\, 0 < p \leq 1: \quad \sum_{j=1}^{\infty} \gamma_j^p < \infty\,.$$

Suppose we have an integrand F whose partial derivatives satisfy

$$\forall\, \boldsymbol{\nu} \in \{0,1,\ldots,\alpha\}^s: \quad |(\partial_{\boldsymbol{y}}^{\boldsymbol{\nu}} F)(\boldsymbol{y})| \leq c\,|\boldsymbol{\nu}|!\,\boldsymbol{\gamma}_s^{\boldsymbol{\nu}}$$

for some constant $c > 0$. Then, an interlaced polynomial lattice rule of order $\alpha := \lfloor 1/p \rfloor + 1$ with N points can be constructed using a fast component-by-component algorithm, with cost $\mathcal{O}(\alpha\, s\, N \log N + \alpha^2\, s^2 N)$ operations, such that, as $N \to \infty$,

$$|I_s(F) - Q_{N,s}(F)| \leq C_{\alpha,\boldsymbol{\gamma},b,p}\, N^{-1/p}\,,$$

where $C_{\alpha,\boldsymbol{\gamma},b,p} < \infty$ is a constant independent of s and of N.

4 Combined Error Bound

In the case of the single level algorithm, the combined error (5) satisfies the following error bound (see [8] for numerical results).

Theorem 5 ([6, Theorem 4.1]) *Under Assumption 1 and conditions (8), $G \in \mathcal{X}_{t'}'$ and (10), the integration error using an interlaced polynomial lattice rule of order $\alpha = \lfloor 1/p_0 \rfloor + 1$ with $N = b^m$ points (with b prime) in s dimensions, combined with a Petrov-Galerkin method in the domain D with one common subspace $\mathcal{X}^h$ with*

$M_h = \dim(\mathcal{X}^h)$ *degrees of freedom and with linear cost* $\mathcal{O}(M_h)$*, satisfies*

$$|I(G(u)) - Q_{N,s}(G(u_s^h))| \le C\left(s^{-2(1/p_0-1)} + N^{-1/p_0} + h^{t+t'}\right),$$

where the constant implied in $\mathcal{O}$ *is independent of s, h and N.*

The multi-level algorithm Q_*^L in (6) additionally requires the stronger Assumption 2. The corresponding combined error bound using interlaced polynomial lattice rules is of the form (see [7, Theorem 3.4])

$$|I(G(u)) - Q_*^L(G(u_s^h))| \le C\left(s_L^{-2(1/p_0-1)} + h_L^{t+t'} + \sum_{\ell=0}^{L} N_\ell^{-1/p_t}\left(s_{\ell-1}^{-(1/p_0-1/p_t)} + h_{\ell-1}^{t+t'}\right)\right).$$

The s_ℓ and N_ℓ in (6) can be optimized using a Lagrange multiplier argument [7, 14], which, in most cases, yields an improvement compared to the single-level algorithm.

Acknowledgements Josef Dick is the recipient of an Australian Research Council Queen Elizabeth II Fellowship (project number DP1097023). Quoc T. Le Gia was supported partially by the ARC Discovery Grant DP120101816. The work of Christoph Schwab was supported in part by European Research Council AdG grant STAHDPDE 247277, and the Swiss National Science Foundation under Grant No. 200021-149819.

References

1. A. Chkifa, A. Cohen, C. Schwab, Breaking the curse of dimensionality in sparse polynomial approximation of parametric PDEs. J. Math. Pures Appl. **103**, 400–428 (2015)
2. A. Cohen, R. DeVore, C. Schwab, Convergence rates of best N-term Galerkin approximation for a class of elliptic sPDEs. Found. Comput. Math. **10**, 615–646 (2010)
3. A. Cohen, R. DeVore, C. Schwab, Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDEs. Anal. Appl. **9**, 1–37 (2011)
4. J. Dick, Walsh spaces containing smooth functions and Quasi-Monte Carlo rules of arbitrary high order. SIAM J. Numer. Anal. **46**, 1519–1553 (2008)
5. J. Dick, F. Pillichshammer, *Digital Nets and Sequences* (Cambridge University Press, Cambridge, 2010)
6. J. Dick, F.Y. Kuo, Q.T. Le Gia, D. Nuyens, C. Schwab, Higher order QMC Galerkin discretization for parametric operator equations. SIAM J. Numer. Anal. **52**(6), 2676–2702 (2014)
7. J. Dick, F.Y. Kuo, Q.T. Le Gia, C. Schwab, Multi-level higher order QMC Galerkin discretization for affine parametric operator equations. Research Report 2014–14, SAM, ETH Zürich, 2014. Available at arXiv:1406.4432
8. R. Gantner, C. Schwab, Computational higher order Quasi-Monte Carlo integration. Report 2014–25, Seminar for Applied Mathematics, ETH Zürich, 2014 (to appear in Proc. MCQMC2014, Springer Publ., 2015)
9. T. Goda, J. Dick, Construction of interlaced scrambled polynomial lattice rules of arbitrary high order. Found. Comput. Math. (2015). doi:10.1007/s10208-014-9226-8
10. M. Hansen, C. Schwab, Analytic regularity and best N-term approximation of high dimensional, parametric initial value problems. Vietnam J. Math. **41**(2), 181–215 (2013)

11. A. Kunoth, C. Schwab, Analytic regularity and GPC approximation for stochastic control problems constrained by linear parametric elliptic and parabolic PDEs. SIAM J. Control Optim. **51**, 2442–2471 (2013)
12. F.Y. Kuo, C. Schwab, I.H. Sloan, Quasi-Monte Carlo methods for very high dimensional integration: the standard weighted-space setting and beyond. ANZIAM J. **53**, 1–37 (2011)
13. F.Y. Kuo, C. Schwab, I.H. Sloan, Quasi-Monte Carlo finite element methods for a class of elliptic partial differential equations with random coefficient. SIAM J. Numer. Anal. **50**, 3351–3374 (2012)
14. F.Y. Kuo, C. Schwab, I.H. Sloan, Multi-level quasi-Monte Carlo finite element methods for a class of elliptic PDEs with random coefficients. Found. Comput. Math. **15**(2), 411–449 (2015)
15. S. Mishra, C. Schwab, J. Sukys, *Multi-Level Monte Carlo Finite Volume Methods for Uncertainty Quantification in Nonlinear Systems of Balance Laws*. Lecture Notes in Computational Science and Engineering, vol. 92. SAM Report 2012-08 (2013), pp. 225–294
16. D. Nuyens, R. Cools, Fast algorithms for component-by-component construction of rank-1 lattice rules in shift-invariant reproducing kernel Hilbert spaces. Math. Comput. **75**, 903–920 (2006)
17. C. Schwab, QMC Galerkin discretizations of parametric operator equations, in *Monte Carlo and Quasi-Monte Carlo Methods 2012*, ed. by J. Dick, F.Y. Kuo, G. W. Peters, I.H. Sloan (Springer, Berlin, 2013), pp. 613–630

Summation by Parts Finite Difference Approximations for Seismic and Seismo-Acoustic Computations

Björn Sjögreen and N. Anders Petersson

Abstract We develop stable finite difference approximations for a multi-physics problem that couples elastic wave propagation in one domain to acoustic wave propagation in another domain. The approximation consists of one finite difference scheme in each domain together with discrete interface conditions that couple the two schemes. The finite difference approximations use summation-by-parts (SBP) operators, which lead to stability of the coupled problem. Furthermore, we develop a new way to enforce boundary conditions for SBP discretizations of first order problems. The new method, which uses ghost points to enforce the boundary conditions, is a flexible alternative to the more established projection and SAT methods.

1 Introduction

Near surface seismic events emit both elastic waves traveling in the earth and acoustic waves propagating in the atmosphere. Acoustic waves can also occur because of other events, such as bolides or volcanic eruptions. Elastic and acoustic waves are recorded by seismographs and by infrasound instruments at various locations around the world. A coupled seismo-acoustic modeling capability is of relevance to many applications in order to analyze and understand seismograms and infrasound recordings.

We here model seismic wave propagation by the elastic wave equation. Acoustic infrasound is described by the linearized Euler equations of compressible gas dynamics. The elastic and acoustic domains are coupled by interface conditions that enforce continuity of normal stresses and of normal velocities. We here develop finite difference discretizations based on the summation-by-parts (SBP) principle [4], which make the coupled seismo-acoustic problem stable.

B. Sjögreen (✉) • N.A.Petersson
Center for Applied Scientific Computing, LLNL, P.O.Box 808, L-422, Livermore, CA 94551, USA
e-mail: sjogreen2@llnl.gov; petersson1@llnl.gov

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_42

In [3], we made use of ghost points to enforce physical boundary conditions on SBP discretizations of the elastic wave equation in second order formulation. For first order hyperbolic PDEs, boundary conditions in the SBP context have traditionally been imposed by either projection or penalty term (simultaneous approximation term (SAT) [1, 2]). In this paper, we develop ghost point enforced boundary conditions also for SBP discretizations of problems in first order formulation.

2 SBP Operators

Let D be a standard summation by parts finite difference operator for approximating a first derivative. D can be represented as a real N by N matrix acting on grid functions $u = (u_1,\ u_2,\ u_3, \ldots,\ u_N)$. The grid functions are defined on a domain $0 \le x \le 1$, with uniformly distributed grid points $x_j = (j-1)h, j = 1, 2, \ldots, N$, where $h = 1/(N-1)$ is the grid spacing. When ghost points are present they are located at the points $j = 0$ and $j = N+1$. The standard SBP identity,

$$(u, Dv)_h = -(Du, v)_h - u_1 v_1 + u_N v_N, \tag{1}$$

is assumed to hold in a scalar product

$$(u, v)_h = h \sum_{j=1}^{N} \omega_j u_j v_j, \tag{2}$$

where ω_j are positive weights. We extend the difference operator D to handle ghost points by adding an operator to the first and last row of D. The resulting operator, $\tilde{D}$, can be represented as a rectangular matrix with N rows and $N+2$ columns. We denote the extended grid function by

$$\tilde{u} = (u_0,\ u_1,\ u_2,\ \ldots, u_N,\ u_{N+1})^T,$$

and define

$$\tilde{D}\tilde{u} = Du + \frac{1}{h}\mathbf{e}_1(B\tilde{u}_1) - \frac{1}{h}\mathbf{e}_N(C\tilde{u}_N), \tag{3}$$

where $\mathbf{e}_1 = (1, 0, \ldots, 0)^T$ and $\mathbf{e}_N = (0, \ldots, 0, 1)^T$. At the first grid point, Du_1 is replaced by $Du_1 + \frac{1}{h}B\tilde{u}_1$, where $B\tilde{u}_1 = \beta_0 u_0 + \beta_1 u_1 + \ldots + \beta_r u_r$ Similarly, at the last grid point, the modified difference approximation becomes $Du_N - \frac{1}{h}C\tilde{u}_N$, where $C\tilde{u}_N = \beta_0 u_{N+1} + \beta_1 u_N + \ldots + \beta_r u_{N-r+1}$.

Lemma 1 *The difference operator $\tilde{D}$ satisfies the SBP-like identity*

$$(u, \tilde{D}\tilde{v})_h = -(Du, v)_h - u_1(v_1 - \omega_1 B\tilde{v}_1) + u_N(v_N - \omega_N C\tilde{v}_N). \tag{4}$$

Proof The definition of $\tilde{D}$ in (3) and (1) give

$$(u, \tilde{D}\tilde{v})_h = -(Du, v)_h - u_1 v_1 + u_N v_N + u_1 \omega_1 B \tilde{v}_1 - u_N \omega_N C \tilde{v}_N,$$

which leads to (4).

To illustrate the usage of (4), we consider the initial boundary value problem

$$u_t + a(x)u_x = 0, \quad 0 \le x \le 1, \quad t > 0, \tag{5}$$

$$u(0, t) = g(t), \quad t > 0, \tag{6}$$

where $a(x)$ is a real-valued function. We assume $a(0) > 0$ and $a(1) > 0$. We can write (5) as

$$u_t = -\frac{1}{2}a(x)u_x - \frac{1}{2}(au)_x + \frac{1}{2}a_x u.$$

Multiplying this equation by u and integrating over $0 \le x \le 1$ gives the estimate

$$\frac{1}{2}\frac{d}{dt}\|u\|^2 = \frac{1}{2}(u, a_x u) + \frac{1}{2}\left[a(0)u(0,t)^2 - a(1)u(1,t)^2\right]$$

$$\le \frac{1}{2}\alpha\|u\|^2 + \frac{1}{2}a(0)g(t)^2, \tag{7}$$

where $\alpha = |a_x|_\infty$. Here, (u, v) and $||u||$ denote the L^2 scalar product and norm.

Let $v_j(t)$ be the semi-discrete approximation of $u(x_j, t)$. We discretize (5) in space by mixing the standard and extended SBP operators,

$$\frac{dv}{dt} = -\frac{1}{2}a\tilde{D}\tilde{v} - \frac{1}{2}D(av) + \frac{1}{2}D(a)v. \tag{8}$$

To derive an energy estimate, we form the scalar product between v and (8),

$$(v, v_t)_h = -\frac{1}{2}(v, a\tilde{D}\tilde{v})_h - \frac{1}{2}(v, D(av))_h + \frac{1}{2}(v, D(a)v)_h.$$

We set $w = av$ in the first term on the right hand side. The SBP property (4) gives

$$(w, \tilde{D}\tilde{v})_h = -(Dw, v)_h - w_1(v_1 - \omega_1 B\tilde{v}_1) + w_N(v_N - \omega_N C\tilde{v}_N).$$

Therefore,

$$(v, v_t)_h = \frac{1}{2}(v, D(a)v)_h + \frac{1}{2}\left[a_1 v_1(v_1 - \omega_1 B\tilde{v}_1) - a_N v_N(v_N - \omega_N C\tilde{v}_N)\right].$$

We can write

$$v_1(v_1 - \omega_1 B\tilde{v}_1) = \left(v_1 - \frac{\omega_1}{2}B\tilde{v}_1\right)^2 - \frac{\omega_1^2}{4}(B\tilde{v}_1)^2,$$

and the estimate for the semi-discrete problem becomes

$$\frac{1}{2}\frac{d||v||_h^2}{dt} = \frac{1}{2}(v, D(a)v)_h + \frac{a_1}{2}\left[\left(v_1 - \frac{\omega_1}{2}B\tilde{v}_1\right)^2 - \frac{\omega_1^2}{4}(B\tilde{v}_1)^2\right] - \frac{a_N}{2}\left[\left(v_N - \frac{\omega_N}{2}C\tilde{v}_N\right)^2 - \frac{\omega_N^2}{4}(C\tilde{v}_N)^2\right]. \quad (9)$$

The boundary data at $x = 0$ (inflow) can be enforced by choosing the ghost point value v_0 such that

$$v_1 - \frac{\omega_1}{2}B\tilde{v}_1 = g(t). \quad (10)$$

At $x = 1$ (outflow), we choose the ghost point value v_{N+1} such that

$$C\tilde{v}_N = 0, \quad (11)$$

which is an extrapolation formula. With the boundary conditions (10) and (11), we arrive at the estimate

$$\frac{1}{2}\frac{d\|v\|_h^2}{dt} \le \frac{1}{2}\alpha_h\|v\|_h^2 + \frac{a_1}{2}g(t)^2,$$

where $\alpha_h = \max_j |D(a)_j|$. This corresponds to the estimate (7) for (5)–(6).

If, for example, we use a diagonal norm SBP operator that is sixth order accurate in the interior of the domain, and third order near the boundary, the solution can not be expected to be more than fourth order accurate. It is then reasonable to choose

$$B\tilde{v}_1 = \kappa(v_0 - 4v_1 + 6v_2 - 4v_3 + v_4), \quad (12)$$

$$C\tilde{v}_N = \kappa(v_{N+1} - 4v_N + 6v_{N-1} - 4v_{N-2} + v_{N-3}), \quad (13)$$

where κ is a tunable parameter. With this choice $\frac{1}{h}B\tilde{v} = \mathcal{O}(h^3)$, i.e., $\tilde{D}\tilde{v}$ has a third order truncation error on the boundary. Furthermore, (10) imposes the Dirichlet boundary condition to fourth order accuracy. Inserting (12) and (13) into (10) and (11), respectively, lead to the boundary conditions

$$v_0 = \frac{2(v_1 - g(t))}{\kappa\omega_1} + 4v_1 - 6v_2 + 4v_3 - v_4,$$

$$v_{N+1} = 4v_N - 6v_{N-1} + 4v_{N-2} - v_{N-3}.$$

Remark 1 In this simple example the ghost point value v_{N+1} is only used to set $C\tilde{v}_N = 0$. We could therefore have defined $\tilde{D}$ without the term $\mathbf{e}_N(C\tilde{v}_N)$, leading to the standard SBP procedure where no boundary condition is explicitly needed at outflow boundaries. Furthermore, if the ghost point v_0 is eliminated from (8) for $j = 1$, it turns out that the term $-a_1 \frac{v_1 - g}{h\omega_1}$ appears. Hence, for this simple semi-discrete problem the proposed technique is equivalent with an SAT method.

3 Elastic-Acoustic Coupled Problem

We consider a one dimensional domain of length $2L$, $-L \le x \le L$, with an elastic-acoustic interface at $x = 0$. The domain to the left, $-L \le x \le 0$, is a solid described by the wave equation

$$\rho_e w_{tt} = (\mu w_x)_x + g, \quad t > 0, \ -L \le x \le 0, \tag{14}$$

where w is the displacement, $\rho_e(x)$ is the density of the solid, $\mu(x)$ its shear modulus, and $g = g(x,t)$ is a given forcing function. The domain to the right, $0 \le x \le L$, is acoustic and described by the linearized and symmetrized Euler equations,

$$\mathbf{q}_t + A(x)\mathbf{q}_x = E(x)\mathbf{q} + \mathbf{f}, \tag{15}$$

where $\mathbf{q} = (s,\ u,\ r)$, with

$$s = \frac{1}{\hat{\rho}\hat{c}}p - \frac{\hat{c}}{\hat{\rho}}\rho, \qquad r = \frac{1}{\hat{\rho}\hat{c}}p,$$

and where ρ, u, and p are the density, velocity, and pressure perturbations in the air. The hat variables denote a given, steady, background field $(\hat{\rho}(x),\ \hat{u}(x),\ \hat{c}(x))$, where the background sound speed is given by $\hat{c} = \sqrt{\gamma \hat{p}/\hat{\rho}}$. Here γ is a constant, usually taken to be 1.4 in air. The matrices are given by

$$A = \begin{pmatrix} \hat{u} & 0 & 0 \\ 0 & \hat{u} & \hat{c} \\ 0 & \hat{c} & \hat{u} \end{pmatrix} \qquad E = \begin{pmatrix} \hat{u}_x - 3\frac{\hat{u}}{\hat{c}}\hat{c}_x + \frac{\hat{u}}{\hat{p}}\hat{p}_x & \frac{\gamma-1}{\hat{\rho}\hat{c}}\hat{p}_x + 2\hat{c}_x & (\gamma-1)\hat{u}_x + 2\frac{\hat{u}}{\hat{c}}\hat{c}_x \\ \frac{1}{\hat{\rho}\hat{c}}\hat{p}_x & \hat{u}_x & \frac{\gamma-1}{\hat{\rho}\hat{c}}\hat{p}_x - \hat{c}_x \\ 0 & \frac{1}{\hat{\rho}\hat{c}}\hat{p}_x & \gamma\hat{u}_x - \frac{\hat{u}}{\hat{c}}\hat{c}_x + \frac{\hat{u}}{\hat{p}}\hat{p}_x \end{pmatrix}$$

where we note that $A(x)$ is symmetric. The function $\mathbf{f} = \mathbf{f}(x,t)$ is a given forcing function. At the interface, the background velocity is assumed to vanish, $\hat{u}(0) = 0$. At the interface we impose continuity of stress, $\mu w_x(0,t) = -p(0,t)$, and velocity, $w_t(0,t) = u(0,t)$.

A grid with grid spacing h, $x_j = jh$ discretizes the domain. The interface is located at $x_0 = 0$. Here, x_{-1} is a ghost point for the acoustic domain, and x_1 is a

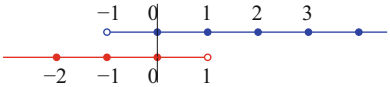


Fig. 1 Grid $x_j = jh$ near the interface at $j = 0$. The elastic domain (*red*) uses a ghost point at $j = 1$. The acoustic domain (*blue*) has a ghost point at $j = -1$

ghost point for the elastic domain. Figure 1 shows the grid points of the acoustic (blue) and elastic (red) domains near the interface.

Similarly to the scalar problem in Sect. 2, the acoustic equations are discretized in space by

$$\frac{d}{dt}\mathbf{q}_j = -\frac{1}{2}A_j\tilde{D}\tilde{\mathbf{q}}_j - \frac{1}{2\hat{\rho}_j}D(\hat{\rho}A\mathbf{q})_j + F_j\mathbf{q}_j + \mathbf{f}_j \tag{16}$$

for $j = 0, 1, \ldots, N$ with ghost points at $j = -1$ and $j = N + 1$. The matrix $F = E + \frac{1}{\hat{\rho}}(\hat{\rho}A)_x$. Here A_j denotes the matrix $A(x_j)$, and similarly for F_j, $\mathbf{f}_j$, and $\hat{\rho}_j$. The density weighting in the splitting is introduced to ensure that the scaling of the boundary terms at the interface matches the scaling of the boundary term from the wave equation in the elastic domain. Denote the SBP scalar product on the acoustic domain by $(\mathbf{u}, \mathbf{v})_{h+} = h\sum_{j=0}^{N}\omega_j^+\mathbf{u}_j^T\mathbf{v}_j$, where ω_j^+ are the SBP norm weights. The spatial discretization satisfies the estimate (if we set $\mathbf{f} = \mathbf{0}$),

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}(\hat{\rho}\mathbf{q}, \mathbf{q})_{h+} = (\mathbf{q}, \hat{\rho}\mathbf{q}_t)_{h+} &= -\frac{1}{2}(\mathbf{q}, \hat{\rho}A\tilde{D}\tilde{\mathbf{q}})_{h+} - \frac{1}{2}(\mathbf{q}, D(\hat{\rho}A\mathbf{q}))_{h+} + (\mathbf{q}, \hat{\rho}F\mathbf{q})_{h+} \\ &= \frac{1}{2}\mathbf{q}_0^T\hat{\rho}_0A_0(\mathbf{q}_0 - B\tilde{\mathbf{q}}_0) - \frac{1}{2}\mathbf{q}_N^T\hat{\rho}_NA_N(\mathbf{q}_N - C\tilde{\mathbf{q}}_N) + (\mathbf{q}, \hat{\rho}F\mathbf{q})_{h+}.\end{aligned} \tag{17}$$

This equality follows by straightforward generalization of the scalar identity (4) and by using the symmetry of A. Here, the boundary operator $B\tilde{\mathbf{q}}_0$ is defined component wise, $B\tilde{\mathbf{q}} = (B\tilde{s},\ B\tilde{u},\ B\tilde{r})^T$, and similarly for $C\tilde{\mathbf{q}}_N$. Due to the numbering of the ghost point, note that $B\tilde{u}_0 = \sum_{k=-1}^{r-1}\beta_{k+1}u_k$. The assumption $\hat{u}(0) = 0$ implies that the boundary term at $x = x_0$ can be written

$$\frac{1}{2}\mathbf{q}_0^T\hat{\rho}_0A_0(\mathbf{q}_0 - B\tilde{\mathbf{q}}_0) = (u_0 - B\tilde{u}_0)(p_0 - \hat{\rho}_0\hat{c}_0B\tilde{r}_0) - (B\tilde{u}_0)(\hat{\rho}_0\hat{c}_0B\tilde{r}_0).$$

In order to advance in time with the same method in the acoustic and elastic domains, we rewrite (14) as a system of two equations with first derivatives in time. After discretizing in space we obtain,

$$\rho_e\frac{dv_j}{dt} = G(\mu, w)_j + g_j \qquad \frac{dw_j}{dt} = v_j, \tag{18}$$

for $j = -N, \ldots, 0$. The spatial discretization $G(\mu, w)$ is the SBP operator approximating $(\mu u_x)_x$, developed in [3]. It satisfies, in the SBP scalar product $(v, w)_{h-}$,

$$(v, G(\mu, w))_{h-} = -(Dv, \mu Dw)_{h-} - (v, Pw)_{h-} - v_{-N}\mu_{-N}Sw_{-N} + v_0\mu_0 Sw_0, \tag{19}$$

where P is a positive semi-definite operator that is small and Sw_0 is a high order approximation of $w_x(x_0)$ using the stencil $w_{-m}, \ldots, w_1$, for some stencil width $m+2$.

The energy norm, N_E, of the solution over both domains satisfies

$$\begin{aligned}\frac{1}{2}\frac{dN_E}{dt} &= \frac{1}{2}\frac{d}{dt}\left((w_t, \rho_e w_t)_{h-} + (Dw, \mu Dw)_{h-} + (w, Pw)_{h-} + (\mathbf{q}, \hat{\rho}\mathbf{q})_{h+}\right) \\ &= (v, \rho_e v_t)_{h-} + (Dv, \mu Dw)_{h-} + (v, Pw)_{h-} + (\mathbf{q}, \hat{\rho}\mathbf{q}_t)_{h+} \\ &= v_0\mu_0 Sw_0 + \frac{1}{2}\mathbf{q}_0^T\hat{\rho}_0 A_0(\mathbf{q}_0 - B\tilde{\mathbf{q}}_0) + (\mathbf{q}, \hat{\rho}F\mathbf{q})_{h+} + T_2,\end{aligned} \tag{20}$$

which can be seen by combining (17) and (19). Here, T_2 denotes boundary terms from the boundaries at $x = \pm L$, and we have set $g = 0$ in (18). The interface conditions are stable if the boundary terms at the interface do not contribute to any norm increase, i.e., if

$$v_0\mu_0 Sw_0 + (u_0 - B\tilde{u}_0)(p_0 - \hat{\rho}_0\hat{c}_0 B\tilde{r}_0) - (B\tilde{u}_0)(\hat{\rho}_0\hat{c}_0 B\tilde{r}_0) = 0. \tag{21}$$

We enforce the discrete interface condition (21), by setting

$$B\tilde{r}_0 = 0 \tag{22}$$

$$\mu_0 Sw_0 = -(p_0 - \hat{\rho}_0\hat{c}_0 B\tilde{r}_0) \tag{23}$$

$$v_0 = u_0 - B\tilde{u}_0. \tag{24}$$

Here (22) determines r_{-1}, (23) determines w_1, and (24) determines u_{-1}. This means that stress and velocity are required to be continuous across the interface.

Alternatively, the approximation for the acoustic equations can be done without use of ghost points. In that case, the operator $\tilde{D}$ in (16) is replaced by D, and the extra boundary operator $B = 0$, which gives

$$\mu_0 Sw_0 = -p_0 \qquad \text{and} \qquad v_0 = u_0.$$

These two conditions are used to determine w_1 and u_0, respectively. Hence, the acoustic velocity u_0 is set by direct injection, which is equivalent with the projection method, and therefore also leads to a stable method. The projection method is straightforward and easy to use for simple Dirichlet conditions. We prefer using

the ghost point method for the elastic equation because it is very easy to implement and conceptually simpler than the SAT method.

4 Numerical Experiments

The semi-discrete acoustic-elastic problem (16), (18) with interface conditions (22)–(24) is integrated in time by the fourth order accurate Runge-Kutta method. The SBP first derivative operator D of interior order six and third order on the boundary is used in (16). The SBP operator $G(\mu, w)$, developed in [3], which has fourth order interior and second order boundary accuracy, is used in (18).

The domain is $-L \le x \le L$, with $L = 1000$. The grid near the interface is as outlined in Fig. 1. The acoustic background state is

$$(\hat{\rho},\ \hat{u},\ \hat{c}) = (1 + \cos(k_m x + \phi_1)/5,\ 10\sin(k_m x),\ 340 - 30\sin(k_m x + \phi_2)),$$

and the elastic material is

$$\rho_e = 2600 + 150\cos(kx + \phi_2), \qquad \mu = \rho_e c^2, \quad c = 1000 + 400\sin(kx + \phi_1).$$

These material properties have sizes that are realistic for a seismo-acoustic computation. The manufactured solution for the elastic domain is

$$w(x,t) = \sin(210kt - \phi_1)\cos(-2k(x - 200t) - \phi_1)$$

and the acoustic manufactured solution is

$$\begin{pmatrix} \rho \\ u \\ p \end{pmatrix} = \begin{pmatrix} \cos(kx)\sin(k(x - 150t))/20 \\ \sin(kx + \phi_1)\cos(420kt) \\ 200\sin(kx)\sin(170kt + \phi_2) \end{pmatrix}$$

The parameters have the values $k = 0.023$, $k_m = 0.021$, $\phi_1 = 0.17$, and $\phi_2 = 0.08$. The forcing functions $\mathbf{f}(x,t)$ and $g(x,t)$ are determined such that (15) and (14) are solved by the manufactured solution. Forcing functions are also inserted into the interface conditions. These are needed to enforce the jump in the manufactured solutions across the interface. The convergence under grid refinement is shown in Fig. 2. Fourth order convergence is observed in all variables except in the acoustic density, which converges somewhere between third and fourth order. This is probably due to the interface being a characteristic boundary for the acoustic equations, since recovery of fourth order convergence from the third order truncation error on the boundary is not guaranteed at such boundaries.

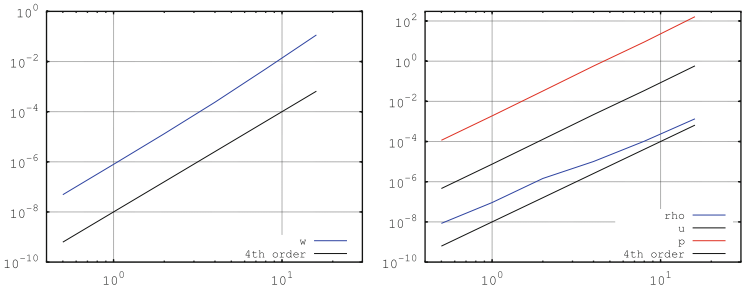


Fig. 2 Maximum norm errors of the manufactured solution at $t = 1$ vs. grid spacing. *Left subplot* shows the error in the elastic variable w, the *right subplot* shows errors in the acoustic density (*blue*), velocity (*black*), and pressure (*red*). *Thin black lines* show fourth order convergence rate

Acknowledgements Work performed under the auspices of the U.S. Department of Energy by LLNL under contract DE-AC52-07NA27344. This is contribution LLNL-PROC-659087.

References

1. M.H. Carpenter, D. Gottlieb, S. Abarbanel, Time-stable boundary conditions for finite-difference schemes solving hyperbolic systems: methodology and application to high-order compact schemes. J. Comput. Phys. **111**, 220–236 (1994)
2. K. Mattsson, J. Nordström, Summation by parts operators for finite difference approximations of second derivatives. J. Comput. Phys. **199**, 503–540 (2004)
3. B. Sjögreen, N.A.Petersson, A fourth order accurate finite difference scheme for the elastic wave equation in second order formulation. J. Sci. Comput. **52**, 17–48 (2012)
4. B. Strand, Summation by parts for finite difference approximations for d/dx. J. Comput. Phys. **110**, 47–67 (1994)

Transparent Boundary Conditions for the Wave Equation: High-Order Approximation and Coupling with Characteristic NRBCs

I. Sofronov and L. Dovgilovich

Abstract We propose and numerically investigate two approaches for extending the application area of transparent boundary conditions (TBCs) for the wave equation: a method for generating finite-difference approximations of TBCs with the fourth and sixth order in space, and a coupling procedure of TBCs on the top boundary of a cubical computational domain with characteristic BCs at the neighbor side boundaries.

1 Introduction

An important application of the wave equation in cubical computational domains comes from geophysics when considering marine and land surface seismic problems. The corresponding full waveform modeling requires using *high-order accurate* finite-difference schemes (FDS) and non-reflecting boundary conditions (NRBCs) at open boundaries; very often, the top boundary must have high-quality NRBCs to suppress so-called multiples. A known approach of providing NRBCs for the wave equation consists of using analytical *transparent boundary conditions* (TBCs) derived for both spherical/circular boundaries and waveguide cross sections (including the half-plane limit case) [1–3]. Corresponding ways of approximating TBCs developed in the cited papers and [4] permit closing the explicit time-integration schemes and obtaining stable and efficient *second-order accurate* methods.

In this paper, we address the questions of wider use of TBCs for cubical computational domains. First, we describe an approach of increasing the approximation

I. Sofronov (✉)
Schlumberger, Pudovkina 13, Moscow, Russia

MIPT, 9 Institutskiy per. Dolgoprudny, Russia
e-mail: isofronov@slb.com

L. Dovgilovich
Schlumberger, Pudovkina 13, Moscow, Russia
e-mail: ldovgilovich@slb.com

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_43

order of TBCs to match accuracy with *high-order accurate* spatial schemes inside domains. In particular, we consider the wave propagation problem in a rectangular waveguide and approximate TBCs with 6th spatial order of accuracy [5]. Second, we propose a way of coupling TBCs at the top boundary with characteristic NRBCs at neighboring side boundaries.

We also mention that there are other approaches that can provide high-order accurate closures at the waveguide open cross section: PML [6] and extended domain absorbing layers (see [7] as example of recent results on this classic idea).

2 High-Order Approximation of TBCs at the Waveguide Cross Section

For some $X, Y, Z > 0$, we consider an initial boundary value problem

$$\begin{cases} u_{tt} - c^2\left(u_{xx} + u_{yy} + u_{zz}\right) = S\left(t, \mathbf{x}\right), \quad \mathbf{x} \equiv (x, y, z) \in \Omega, \quad t > 0 \\ u|_{z=-Z} = 0, \quad \frac{\partial u}{\partial n}\big|_{\Gamma} = 0, \; u\,|_{t=0} = W_0\left(\mathbf{x}\right), \; u_t\,|_{t=0} = W_1\left(\mathbf{x}\right). \end{cases} \tag{1}$$

in a semi-infinite waveguide $\Omega = \{-Z \le z < \infty,\ 0 \le x \le X,\ 0 \le y \le Y\}$ for a function $u \equiv u\left(t, \mathbf{x}\right)$. Here, $\Gamma = \partial\Omega \setminus \{z = -Z\}$ is the side boundary; n is the outer normal; $c(\mathbf{x})$, S, and W_0, W_1 are sufficiently smooth functions of the sound speed, source, and initial data matching the boundary conditions of (1), respectively. We suppose that $S = W_0 = W_1 = 0$, and $c = \text{const}$ outside the domain $\Omega_1 = \Omega \cap \{z \le 0\}$. Our aim is to provide highly accurate NRBCs at the boundary $\Gamma_2 = \{0 \le x \le X,\ 0 \le y \le Y,\ z = 0\}$ so that solution of (1) is approximated by the solution of the same governing equations in the reduced domain Ω_1.

2.1 Solution Continuation

To generate such NRBCs, we use formulas of solution continuation into the truncated external domain $\Omega_2 = \Omega \setminus \Omega_1$, according to the TBC idea [1]. Denoting $\varphi_{\alpha,\beta}\left(x, y\right) = \cos\left(\pi\alpha x/X\right)\cos\left(\pi\beta y/Y\right)$, $\alpha, \beta = 0, \ldots, \infty$, we calculate the Fourier coefficients

$$u_{\alpha,\beta}\left(t, z\right) = \left(Qu\right)|_{\alpha,\beta} \equiv \frac{4\gamma_\alpha\gamma_\beta}{XY}\int_0^Y\int_0^X u\left(t, \mathbf{x}\right)\varphi_{\alpha,\beta}\left(x, y\right)dxdy; \;\; \gamma_{\alpha\neq 0} = 1, \;\; \gamma_0 = 0.5, \tag{2}$$

to continuing them from $z = 0$ to any $z > 0$, $t > z/c$, by

$$\widehat{u}\left(t, z\right) = \widehat{u}\left(t - z/c, 0\right) - \frac{z}{c}\int_0^{t - z/c}\widehat{u}\left(t', 0\right)\widehat{K}\left(t - t', z\right)dt' \tag{3}$$

where

$$\widehat{K}(t,z) = c^2\widehat{\lambda}^2 \frac{J_1\left(c\widehat{\lambda}\sqrt{t^2 - z^2/c^2}\right)}{c\widehat{\lambda}\sqrt{t^2 - z^2/c^2}},\ \widehat{\lambda} = \mathrm{sqrt}\left(\left(\frac{\pi\alpha}{X}\right)^2 + \left(\frac{\pi\beta}{Y}\right)^2\right), \tag{4}$$

and $J_1(z)$ is the Bessel function; the hat is used instead of «α, β»; $\widehat{K} = 0$ if $\widehat{\lambda} = 0$. A straightforward application of (2), (3), and the inverse Fourier transformation Q^{-1} gives the formulas of solution continuation to (1) from Γ_2 in Ω_2. Note that TBCs at Γ_2 are derived while taking the limit of (3) as $z \to 0$ [2, 5].

Let us derive an approximation to (3) for fast numerical computations. This is possible because we are using solution continuation for $z = ph$, $p = 1,2,3$, where $h > 0$ is a (small) grid spacing. First, we factorize the convolution kernel (4) by functions depending separately on z and t; this allows using the same convolution kernels of t for any h. We approximate the kernel (4) by the Taylor series. After some transformations involving well-known relations for the Bessel functions, we obtain

$$\widehat{K}_{\mathrm{appr}}(t,z) = c^2\widehat{\lambda}^2 \sum_{m=1}^{p} K_m\left(c\widehat{\lambda}t\right) \frac{\left(\widehat{\lambda}z\right)^{2m-2}}{2^{m-1}(m-1)!},\quad K_m(t) \equiv \frac{J_m(t)}{t^m}.$$

that provides accuracy $\widehat{K}(t,z) = \widehat{K}_{\mathrm{appr}}(t,z) + O\left(\left(\widehat{\lambda}z\right)^{2p}\right)$. Second, we approximate the convolution kernels by sums of exponentials:

$$\widehat{K}^{\mathrm{exp}}_{\mathrm{appr}}(t,z) = c^2\widehat{\lambda}^2 \sum_{m=1}^{p} K^{\mathrm{exp}}_m\left(c\widehat{\lambda}t\right) \frac{\left(\widehat{\lambda}z\right)^{2m-2}}{2^{m-1}(m-1)!},\quad K^{\mathrm{exp}}_m(t) = \sum_{l=1}^{L_m} a_{m,l}\exp\left(b_{m,l}t\right). \tag{5}$$

We have generated sets $\{a_{m,l}, b_{m,l}\}$ such that numerically proven accuracy of exponential kernels $\varepsilon_m = \max\limits_{t\geq 0}\left|K_m(t) - K^{\mathrm{exp}}_m(t)\right|$ is estimated by $\varepsilon_1 < 4.0e-6$, $\varepsilon_2 < 2.0e-7$, and $\varepsilon_3 < 6.0e-8$ for $L_1 = 64$, $L_2 = 32$, and $L_3 = 16$, respectively.

Thus, we use the following approximate formula instead of (3):

$$\widehat{u}(t,z) = \widehat{u}(t - z/c, 0) - \frac{z}{c}\int_0^{t-z/c} \widehat{u}\left(t',0\right)\widehat{K}^{\mathrm{exp}}_{\mathrm{appr}}\left(t-t',z\right)dt'\,. \tag{6}$$

2.2 Discretization Aspects

Generation of approximate TBCs on the basis of the described solution continuation is illustrated on the example of the conventional $O\left(\tau^2 + h^{2p}\right)$ order explicit central FDS for (1) with a $(6p+1)-$ point spatial stencil on a uniform grid; the boundary grid points belong to the physical boundaries of Ω_1. To implement boundary conditions, we use p ghost grid layers for each boundary. Consider the case of the open boundary Γ_2. Suppose that we know a difference solution $u^h \equiv u^h\left(t^n, \mathbf{x}_h\right)$ at Γ_2^h, the grid counterpart of Γ_2; $t^n = n\tau$. It is required to continue this solution into the grid points with $z = h, \ldots, ph$ at the time level t^{n+1}. Denote $n_p = \max\left(2p, \left[ph/(c\tau)\right]\right)$. The following sequential steps describe the continuation procedure of u^h with accuracy $O\left(\tau^{2p} + h^{2p}\right)$:

1) Apply the discrete counterpart of Fourier expansion (2) to u^h at Γ_2^h and obtain the coefficients sets $\widehat{u}^h(t^\nu, 0)$ at time levels $\nu = n, \ldots, n - n_p$.
2) For $z = h, \ldots, ph$ compute each coefficient $\widehat{u}^h\left(t^{n+1}, z\right)$ by (6) as follows:
   (2a) Compute $\widehat{u}^h\left(t^{n+1} - z/c, 0\right)$ by using the Lagrange interpolation of $2p$ degree with respect to time of the grid function $\left\{\widehat{u}^h(t^\nu, 0),\ \nu = n, \ldots, n - n_p\right\}$.
   (2b) Compute the convolution integral by recurrence formulas, using the fact that the kernel consists of a sum of exponentials; the remaining integration of $\widehat{u}^h(t, 0)$ over the interval $[t - z/c - \tau,\ t - z/c]$ is made by the explicit formulas using the same interpolation of $\left\{\widehat{u}^h\left(t^\nu, 0\right),\ \nu = n, \ldots, n - n_p\right\}$ as in (2a).
3) Apply the summation of the discrete Fourier harmonics with the computed coefficients $\widehat{u}^h\left(t^{n+1}, z\right)$ to obtain $u^h\left(t^{n+1}, \mathbf{x}_h\right)$ at $z = h, \ldots, ph$.

Estimates of both memory and operations volumes needed for the described *ghost layers TBCs* operator show that they are similar to application of explicit FDS in some additional grid layer having $C_0\left(n_p + 0.25\sum_1^p L_m\right)$ grid points in the z direction; the coefficient C_0 depends on the source code implementation quality.

2.3 Numerical Experiments

We consider a 2D test problem governed by (1) with omitted dependence on y. Let x and z be the horizontal and vertical axes, respectively. We take $X = Z = 1$, $c = 1$, $W_0 = W_1 = 0$, and define a point source with the time wavelet $S(t) = \left(1 - 2\pi^2\nu^2(t - 2/\nu)^2\right)e^{-\pi^2\nu^2(t-2/\nu)^2}$, $\nu = 5$, located at $x = 0.5$, $z = -0.08$, i.e., very close to the TBCs boundary. A square uniform grid is used with I cells in each direction. The solution absolute value snapshots for the parameters $I = 100$, $p = 2$ at time points $t = 0.75$ and $t = 1.0$ are shown in Fig. 1.

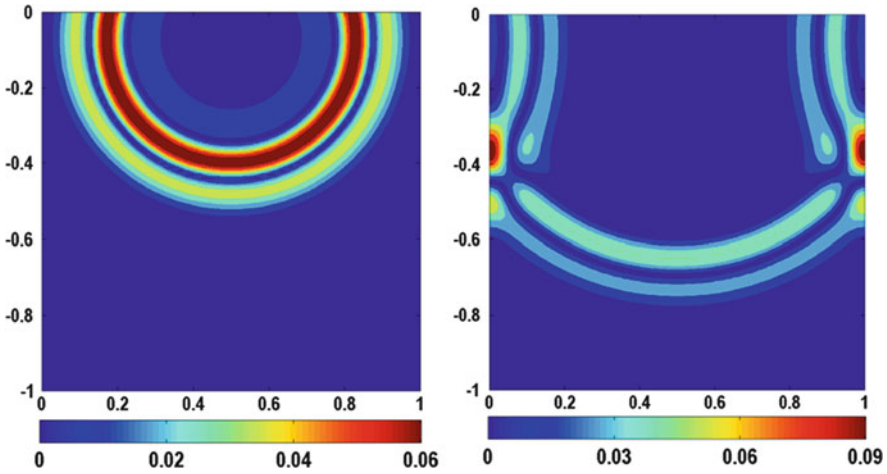


Fig. 1 Solution norm snapshots at $t = 0.75$ (*left*) and $t = 1.0$ (*right*) for a point source. TBCs are at the top boundary; Neumann BCs are at the side boundaries

To check the accuracy of the proposed TBCs, we introduce the relative error norm $\varepsilon\,[u, v, \omega]\,(t) = \max\limits_{\mathbf{x}\in\omega} |u - v| \,/\, \max\limits_{\mathbf{x}\in\omega,\, t\le T} |u|$ and consider reference solutions u_{ref} calculated on the extended domain $-Z \le z \le Z$ with TBCs operator at $z = Z$. Denote $\Delta_I = \max\limits_{t\le T} \varepsilon \left[u^h_{ref}, u^h, \Omega^h_1\right](t)$, $T = 1$. The numerically estimated order of the solution accuracy $\log_2(\Delta_I/\Delta_{2I})$ is 2.2, 4.3, and 6.0 for $p = 1, 2,$ and 3, respectively; $I = 200$, the number of discrete Fourier harmonics is $I/4$.

To verify the stability for large simulation time, we calculated the problem with TBCs up to $T = 100$. Figure 2 shows $\varepsilon[u^h, 0, \Omega^h_1](t)$ (blue), $\varepsilon[u^h_{ref}, 0, \Omega^h_1](t)$ (green), and $\varepsilon[u^h_{ref}, u^h, \Omega^h_1](t)$ (red).

3 Coupling TBCs at the Top Boundary with Characteristic NRBCs

Consider now the case of characteristic NRBCs [8]

$$\frac{\partial u}{\partial t} + c\frac{\partial u}{\partial n} = 0 \tag{7}$$

on the side boundary Γ in (1). Evidently, the immediate generating of TBCs with help of Fourier transformation over the open boundary Γ_2 cannot be done because of the time derivative in (7).

To apply conventional TBCs for this case, we propose using an auxiliary wider waveguide in the external domain $z > 0$ with the cross section $\Gamma_D =$

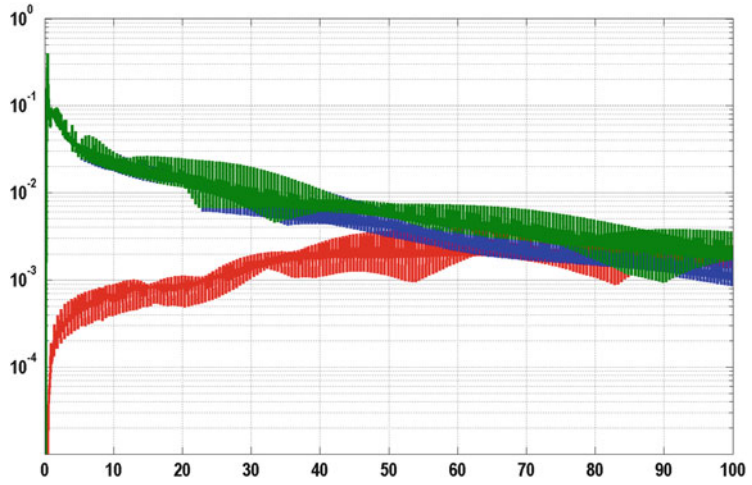


Fig. 2 Solution norm vs time for the original (*blue*) and extended (*green*) domains. Also norm of the solution relative difference is shown (*red*). I = 100, p = 2

$\{-D \le x \le X + D,\ -D \le y \le Y + D,\ z = 0\}$, $D > 0$, so that $\Gamma_2 \subset \Gamma_D$. According to this approach, the solution continuation in the ghost layers points $z = h, \ldots, ph$ is computed by applying TBCs formulas at Γ_D; the extra efforts consist in developing a special technique to prolong waveforms from Γ_2 onto Γ_D. The proposed prolongation technique is as follows. We consider an initial boundary value problem on $\Gamma_D \backslash \Gamma_2$ for the reduced 2D wave equation $u_{tt} - c^2 \left(u_{xx} + u_{yy}\right) = 0$ with a Dirichlet condition at boundaries of Γ_2. Solution of this auxiliary problem supplies an approximation of the required prolongation of waveforms. The corresponding numerical implementation is made straightforwardly by explicit FDS. As an example, we introduce a uniform rectangular grid in Γ_D (grid in Γ_2 is its subgrid). The solution at grid points of $\Gamma_D \backslash \Gamma_2$ is updated by auxiliary explicit FDS with a five-point spatial operator for $u_{tt} - c^2 \left(u_{xx} + u_{yy}\right) = 0$ using Dirichlet data at the boundary of Γ_2; the solution at Γ_2 grid points (including these Dirichlet data) is updated by FDS for (1).

3.1 Numerical Experiments

Accuracy and stability of approximate TBCs generated by the proposed approach are analyzed on 2D tests similar to those in Sect. 2. We take $D = 1, X = 1, Z = 2$, $I \times 2I$ grid cells, $I = 400$, and 300 Fourier harmonics for $\Gamma_D = \{-1 \le x \le 2\}$. Second-order approximation of all equations including TBCs is used ($p = 1$).

The solution absolute value snapshots at time points $t = 1.0$ and $t = 1.1$ are drawn in Fig. 3. One can see that no reflections from the top boundary with TBCs are

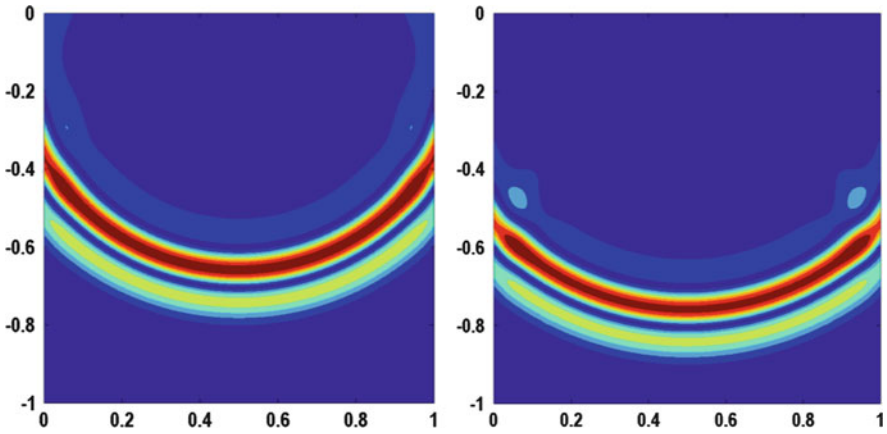


Fig. 3 Solution norm snapshot at $t = 1.0$ (*left*) and $t = 1.1$ (*right*). TBCs are at the top boundary and characteristic NRBCs are at the side boundaries

observed. The main distortions originate near the side boundaries, which is expected as simple characteristic boundary conditions (7) are used here; the distortions are stronger for $t = 1.1$ because of bigger deviation of the incidence angle from the normal direction ($z = -0.6$). However, despite approximate prolongation of waveforms in Γ_D, the introduced error only slightly influences the accuracy of TBCs; this is due to the practically normal incidence angle of waveforms to the side boundaries in the near top surface region ($z = 0$).

For the quantitative accuracy analysis, we compare our solution u^h of the above problem versus the reference solution u^h_{ref} calculated on the extended domain $-Z \leq z \leq 1$ with TBCs operator at $z = 1$, and versus u^h_{open}, the solution calculated in a very large computational domain in all directions, i.e., without any reflections from boundaries.

To exclude strong influence of side boundaries on the accuracy, we consider a little narrower domain $\Omega_0 = \{0.15 \leq x \leq 0.85, -1.0 \leq z \leq 0.0\}$ for estimating error norms. Figure 4 shows $\varepsilon[u^h_{open}, u^h, \Omega^h_0]$ (blue) and $\varepsilon[u^h_{open}, u^h_{ref}, \Omega^h_0]$ (green). The error $\varepsilon[u^h_{open}, u^h_{ref}, \Omega^h_0](t)$ is indicative of the level of reflections from the side boundaries, visible after time point $t = 0.95$. Let us analyze $\varepsilon[u^h_{open}, u^h, \Omega^h_0](t)$. We distinguish two intervals of interest. First, from $t = 0.3$ to $t = 0.95$, the residuals are determined by reflections from the top of the computational domain, where we have TBCs. Then, from $t = 0.95$, the residuals include side reflections (from characteristic boundary conditions) as well as top reflections. Since the errors $\varepsilon[u^h_{open}, u^h, \Omega^h_0]$ and $\varepsilon[u^h_{open}, u^h_{ref}, \Omega^h_0]$ are practically equal to each other after $t = 0.95$, we conclude that the reflections from the top boundary are smaller than those from the side boundaries. Comparing the level of errors before $t = 0.95$ and after $t = 1.2$, we determine that the TBC errors are smaller than the errors from characteristic boundary conditions by a factor of about 100.

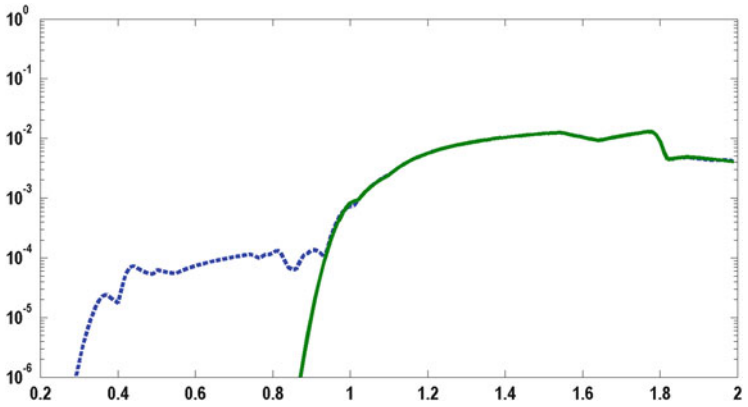


Fig. 4 Norms of solution relative errors vs time: $\varepsilon[u^h_{open}, u^h_{ref}, \Omega^h_0](t)$ (*red*) and $\varepsilon[u^h_{open}, u^h, \Omega^h_0](t)$ (*green*)

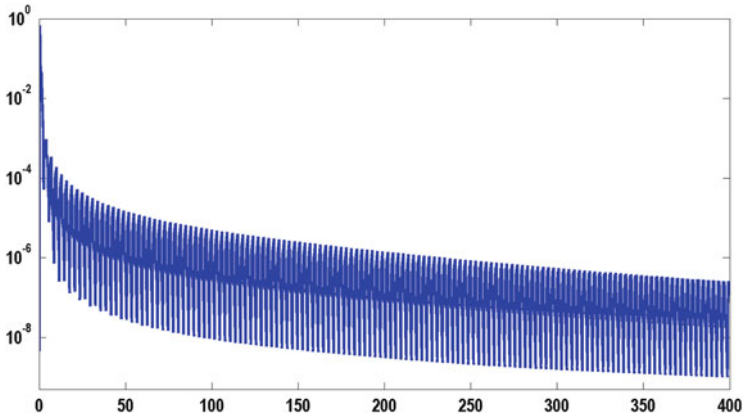


Fig. 5 Solution norm u^h in Ω^h_0 vs time

To check the stability of the algorithm with TBCs, we run computations up to $t = 400$, see norm of $\varepsilon[u^h, 0, \Omega^h_0](t)$ in Fig. 5.

4 Conclusions

We developed and numerically investigated two approaches of expanding the scope of the application of TBCs for modeling wave propagation problems in computational domains with open boundaries. The first approach is a high-order accuracy approximation of TBCs aimed to match high-order FD schemes in the interior. Test numerical calculations confirm the expected 2-nd, 4-th, and 6-th order

of convergence. The second approach is coupling of TBCs with characteristic NRBCs at neighbor boundaries of a cubical domain. Again, test problems confirm the expected accuracy of the "TBCs boundary", at least until spurious reflections of "NRBCs boundaries" reach the domain of interest. In all cases, the computations are stable for large simulation time.

Due to use of convolution kernels approximated by Taylor series and sum of exponentials with respect to spatial and time variables, respectively, the computational resources required for the TBCs operator are similar to the application of FDS in some additional grid layer with a fixed number of grid intervals in the normal direction.

Acknowledgments The authors are grateful to Schlumberger and MIPT for permission to publish the work, and RFBR project 13-01-00338

References

1. I.L. Sofronov, Conditions for complete transparency on the sphere for the three-dimensional wave equation. Russian Acad. Sci. Dokl. Math. **46**, 397–401 (1993)
2. I.L. Sofronov, Non-reflecting inflow and outflow in wind tunnel for transonic time-accurate simulation. J. Math. Anal. Appl. **221**, 92–115 (1998)
3. T. Hagstrom, Radiation boundary conditions for the numerical simulation of waves. Acta Numer. **8**, 47–106 (1999)
4. J. Ballmann, G. Britten, I. Sofronov, Time-accurate inlet and outlet conditions for unsteady transonic channel flow. AIAA J. **40**, 1745–1754 (2002)
5. I.L. Sofronov, L. Dovgilovich, N. Krasnov, Application of transparent boundary conditions to high-order finite-difference schemes for the wave equation in waveguides. Appl. Numer. Math. **93**, 195–205 (2015)
6. K. Duru, The role of numerical boundary procedures in the stability of perfectly matched layers. arXiv:1405.0536 [math.NA]
7. D. Appelö, T. Colonius, A high order super-grid-scale absorbing layer and its application to linear hyperbolic systems. J. Comput. Phys. **228**, 4200–4217 (2009)
8. B. Engquist, A. Majda, Absorbing boundary conditions for the numerical evaluation of waves. Math. Comput. **31**(139), 629–651 (1977)

Comparison of Clenshaw–Curtis and Leja Quasi-Optimal Sparse Grids for the Approximation of Random PDEs

Fabio Nobile, Lorenzo Tamellini, and Raul Tempone

Abstract In this work we compare different families of nested quadrature points, i.e. the classic Clenshaw–Curtis and various kinds of Leja points, in the context of the quasi-optimal sparse grid approximation of random elliptic PDEs. Numerical evidence suggests that both families perform comparably within such framework.

1 Introduction

While it is nowadays widely acknowledged that Uncertainty Quantification problems can be conveniently tackled with polynomial approximation schemes whenever the output quantities of interest depend smoothly on a moderate number of random parameters, the search for algorithms whose performance is resilient with respect to the number of such random parameters is a very active research area.

In the context of sparse grid approximation [1, 4, 13], this has lead on the one hand to the development of more efficient sparse grid algorithms, which exploit the anisotropic structure of the problem (either via an "a-priori" analysis, see e.g. [2, 3, 11], or with an "a-posteriori" adaptation, see [6, 8, 12]), and on the other hand to the study of appropriate univariate collocation points to be used as a basis for the sparse grid construction.

To maximize the efficiency of the sparse grids, such collocation points are typically chosen to be nested. Clenshaw–Curtis points are a classical choice in this sense; more recently, an increasing attention has been devoted to the study of the performance of the so-called Leja points (see [5, 6, 10, 12]), which are promising since the cardinality of Leja quadrature rules grows slower than that of Clenshaw–Curtis rules when increasing the approximation level. In the literature,

F. Nobile • L. Tamellini (✉)
CSQI - MATHICSE, École Polytechnique Fédérale de Lausanne, Lausanne, Switzerland
e-mail: fabio.nobile@epfl.ch; lorenzo.tamellini@epfl.ch

R. Tempone
Applied Mathematics and Computational Science, King Abdullah University of Science and Technology, Thuwal, Saudi Arabia
e-mail: raul.tempone@kaust.edu.sa

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_44

Leja points have only been applied to "a-posteriori" adaptive sparse grids [6, 10, 12]: the aim of this work is to test their performance in the context of the quasi-optimal "a-priori/a-posteriori" sparse grids that we have proposed in a series of papers [2, 3, 11], focusing on the case of elliptic PDEs with diffusion coefficients parametrized by uniform random variables.

The rest of this work is organized as follows. The general problem setting will be introduced in Sect. 2, and quasi-optimal sparse grids in Sect. 3. Clenshaw–Curtis and Leja points will be discussed in Sect. 4, while numerical tests and some conclusions will be presented in Sect. 5.

2 Problem Setting

Let $N \in \mathbb{N}$ and $\Gamma \subset \mathbb{R}^N$ be an N-dimensional hyper-rectangle $\Gamma = \Gamma_1 \times \ldots \times \Gamma_N$, and assume that each Γ_n is endowed with a uniform probability measure $\varrho_n(y_n)dy_n = \frac{1}{|\Gamma_n|}dy_n$, so that $\varrho(\mathbf{y})d\mathbf{y} = \prod_{n=1}^N \varrho_n(y_n)dy_n$ is a uniform probability measure on Γ and $(\Gamma, B(\Gamma), \varrho(\mathbf{y})d\mathbf{y})$ is a probability space, $B(\Gamma)$ being the Borel σ-algebra on Γ. Given a convex polygonal domain D in $\mathbb{R}^d$, $d = 1, 2, 3$, we consider the following problem:

Problem 1 Find a real-valued function $u : \overline{D} \times \Gamma \to \mathbb{R}$, such that $\varrho(\mathbf{y})d\mathbf{y}$-almost everywhere there holds:

$$\begin{cases} -\operatorname{div}(a(\mathbf{x}, \mathbf{y})\nabla u(\mathbf{x}, \mathbf{y})) = f(\mathbf{x}) & \mathbf{x} \in D, \\ u(\mathbf{x}, \mathbf{y}) = 0 & \mathbf{x} \in \partial D, \end{cases}$$

where the operators div and ∇ imply differentiation with respect to the physical coordinates only, and $a : \overline{D} \times \Gamma \to \mathbb{R}$ is such that

$$0 < a_{min} \leq a(\mathbf{x}, \mathbf{y}) \leq a_{max} < \infty \tag{1}$$

for some positive and bounded constants a_{min}, a_{max}.

By introducing the Hilbert space $V = H_0^1(D)$, the above problem is well-posed in the Bochner space $L^2_\varrho(\Gamma; V) = \left\{u : \Gamma \to V \text{ s.t. } \int_\Gamma \|u(\mathbf{y})\|_V^2 \varrho(\mathbf{y})d\mathbf{y} < \infty\right\}$, due to the boundedness assumption (1). Moreover, under additional assumptions on a (e.g. $\mathbf{y}$-linearity or mild assumptions on the growth of its $\mathbf{y}$-derivatives), it can be shown that the map $\mathbf{y} \to u(\cdot, \mathbf{y})$ is analytic, see [2, 7].

3 Quasi-Optimal Sparse Grid Approximation

Let $\mathbb{P}_r(\Gamma_n)$ be the set of polynomials of degree at most r over Γ_n, $C^0(\Gamma_n)$ the set of continuous functions over Γ_n, and for a given interpolation level i_n let $\mathcal{U}_n^{m(i_n)} : C^0(\Gamma_n) \to \mathbb{P}_{m(i_n)-1}(\Gamma_n)$ be the Lagrangian interpolant operator over $m(i_n)$ points, with $m : \mathbb{N} \to \mathbb{N}$ a non-decreasing function, the so-called "level-to-nodes" function. Next, for any multi-index with non-zero components $\mathbf{i} \in \mathbb{N}_+^N$ let us define the "hierarchical surplus" operator $\boldsymbol{\Delta}^{m(\mathbf{i})} = \bigotimes_{n=1}^N (\mathcal{U}_n^{m(i_n)} - \mathcal{U}_n^{m(i_n-1)})$, and let $\{\mathcal{I}(\mathrm{w})\}_{\mathrm{w}\in\mathbb{N}}$ denote a sequence of index sets with non-zero components with $\mathcal{I}(0) = [1, 1, \ldots, 1]$, $\mathcal{I}(\mathrm{w}) \subset \mathcal{I}(\mathrm{w}+1)$ and $\bigcup_{\mathrm{w}\in\mathbb{N}} \mathcal{I}(\mathrm{w}) = \mathbb{N}_+^N$. The sparse grid approximation of u is then written as

$$\mathcal{S}^m_{\mathcal{I}(\mathrm{w})}[u](\mathbf{y}) = \sum_{\mathbf{i}\in\mathcal{I}(\mathrm{w})} \boldsymbol{\Delta}^{m(\mathbf{i})}[u](\mathbf{y}), \tag{2}$$

where one usually requires the sets $\mathcal{I}(\mathrm{w})$ to be *downward closed sets*,[1] see e.g. [8]. In practice, to build a sparse grid one has to specify (a) the family of interpolation nodes, that should be chosen according to the probability measure over Γ (as previously mentioned, in this work we will use Leja and Clenshaw–Curtis points, which are suitable for uniform measures), (b) the function $m(\cdot)$, and (c) the sequence of index sets $\mathcal{I}(\mathrm{w})$.

To detail the choice of the sequence $\mathcal{I}(\mathrm{w})$, let us now denote by $\Delta E(\mathbf{i})$ the error reduction obtained by adding a given hierarchical surplus $\boldsymbol{\Delta}^{m(\mathbf{i})}$ to the sparse grid approximation of u and by $\Delta W(\mathbf{i})$ the associated cost, i.e. the number of interpolation points added to the sparse grid by $\boldsymbol{\Delta}^{m(\mathbf{i})}$, and let us define the *profit* $P(\mathbf{i})$ of each $\boldsymbol{\Delta}^{m(\mathbf{i})}$ as the ratio $P(\mathbf{i}) = \frac{\Delta E(\mathbf{i})}{\Delta W(\mathbf{i})}$. The optimal sequence $\mathcal{I}(\mathrm{w})$ should then progressively add to the sparse grid approximation of u the hierarchical surpluses $\boldsymbol{\Delta}^{m(\mathbf{i})}$ ordered by decreasing profits, see [2, 8, 9, 11],

$$\mathcal{I}(\mathrm{w}) = \left\{\mathbf{i} \in \mathbb{N}_+^N : P(\mathbf{i}) \geq \epsilon_{\mathrm{w}}\right\}, \tag{3}$$

with $\{\epsilon_{\mathrm{w}}\}_{\mathrm{w}\in\mathbb{N}}$ a positive sequence decreasing to 0. Note that $\mathcal{I}(\mathrm{w})$ in (3) may not be a downward closed set, and this condition will have to be explicitly enforced.

The above criterion (3) can be implemented either by an "a-posteriori" adaptive procedure (see e.g. [6, 8, 12]) that explores the space of hierarchical surpluses and adds to $\mathcal{I}(\mathrm{w})$ the most profitable one, or, as we have previously detailed in [2, 3, 11], with a procedure based on a-priori estimates of $\Delta E(\mathbf{i})$ and $\Delta W(\mathbf{i})$, tuned to the problem at hand by some cheap preliminary computations ("a-priori/a-posteriori" approach); in this work, we consider the latter approach. Of course, if on the one hand the "a-priori/a-posteriori" approach saves the computational cost of the

[1] Also known as *admissible sets* or *lower sets*, i.e. such that $\forall\, \mathbf{i} \in \mathcal{I}(\mathrm{w})$ and $\forall\, \mathbf{j} \in \mathbb{N}_+^N$ s.t. $\mathbf{j} \preceq \mathbf{i}$, there holds $\mathbf{j} \in \mathcal{I}(\mathrm{w})$, where the inequality is to be understood component-wise.

exploration of the space of hierarchical surpluses, on the other hand it will be effective only if the estimates of $\Delta E(\mathbf{i})$ and $\Delta W(\mathbf{i})$ are sufficiently sharp.

The work contribution $\Delta W(\mathbf{i})$ can actually be computed exactly if the points used in the sparse grid construction are nested (as it is the case in this work) and $\mathcal{I}(\mathrm{w})$ is downward closed:

$$\Delta W(\mathbf{i}) = \prod_{n=1}^{N} (m(i_n) - m(i_n - 1)). \tag{4}$$

As for the error contribution $\Delta E(\mathbf{i})$, we propose to use certain problem-dependent estimates that we will specify later on.

4 Leja and Clenshaw–Curtis Quadrature Rules

A Leja sequence on a generic compact set X is defined recursively, by first choosing $x_1 \in X$ and then letting $x_n = \mathrm{argmin}_{x \in X} \prod_{k=1}^{n-1} (x - x_k)$, see e.g. [5, 6, 10, 12], while the corresponding quadrature weights are computed by enforcing the maximal degree of polynomial exactness. More specifically, we will consider the following families of Leja points:

Line Leja: Let $X = [-1, 1]$ and $x_1 = -1$. Then $x_2 = 1$, $x_3 = 0$, and $x_n = \mathrm{argmin}_{(-1,1)} \prod_{k=1}^{n-1} (x - x_k)$.

Sym-Line Leja: Let $x_1 = 0$, $x_2 = 1$, $x_3 = -1$, $x_n = \mathrm{argmin}_{(-1,1)} \prod_{k=1}^{n-1} (x - x_k)$ for n even, and x_{n+1} be the symmetric point of x_n with respect to 0. Observe that this is *not* a Leja sequence according to the definition above.

P-Disk Leja: Let $x_k = \cos \phi_k$, with $\phi_1 = 0$, $\phi_2 = \pi$, $\phi_3 = \pi/2$, $\phi_{2k+2} = \frac{\phi_{k+2}}{2}$, and $\phi_{2k+3} = \phi_{2k+2} + \pi$. These points correspond to the projection on the real axis (with no repetitions) of the Leja sequence obtained with $x_1 = 1$ and X the complex unit ball (see [5]), and are *not* a Leja sequence.

We will test the Leja families above with two different level-to-nodes functions, i.e. $m_s(i_n) = i_n$ and $m_t(i_n) = 2i_n - 1$. The latter "two-stepping" rule has been introduced in the adaptive context (see e.g. [12]), where the error contributions $\Delta E(\mathbf{i})$ are estimated via successive differences of the integral of u (or of its approximation by e.g. finite elements) over the parameter space: indeed, observe that whenever one point is added to a symmetric quadrature rule, the corresponding quadrature weight will be zero, by symmetry; hence if one were using the "single-stepping" rule $m_s(i_n)$, two consecutive integrals may be equal (up to numerical roundoff) and the algorithm might prematurely stop. Finally, Clenshaw–Curtis points (cf. e.g. [11]) are defined as

$$x_j = \cos\left(\frac{(j-1)\pi}{m(i_n) - 1}\right), 1 \le j \le m(i_n),$$

together with the following level-to-nodes relation $m_d(i_n)$, that ensures their nestedness[2]: $m_d(0) = 0$, $m_d(1) = 1$, $m_d(i_n) = 2^{i_n-1} + 1$. Observe $m_d(i_n)$ grows exponentially in i_n, while $m_s(i_n)$ and $m_t(i_n)$ grow linearly; quoting [10], we say that Leja points have a much finer "granularity".

5 Numerical Tests

In this section we consider two different examples of Problem 1; in both cases, we will introduce a bounded linear functional $\Theta : V \to \mathbb{R}$, and monitor the convergence of the quantity

$$\varepsilon = \sqrt{\mathbb{E}\left[\left(\Theta(\mathcal{S}^m_{\mathcal{I}(\mathrm{w})}[u]) - \Theta(u)\right)^2\right]}, \tag{5}$$

with respect to the number of sparse grid points, that will converge with the same rate as the full error $\mathbb{E}\left[\left(\mathcal{S}^m_{\mathcal{I}(\mathrm{w})}[u] - u\right)^2\right]^{1/2}$, given the linearity of Θ. In practice, we have estimated (5) with a Monte Carlo sampling (see Fig. 1 for the sample size for each test); we underline that the sample sizes have been verified to be sufficient for our purposes.

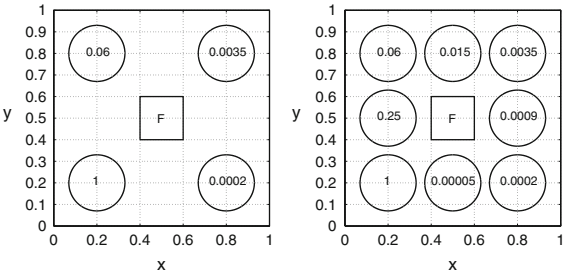
In the first test, we consider $\Gamma_n = [-1, 1]$, $D = (0, 1)$ and two different expressions of $a(\mathbf{x}, \mathbf{y})$, both complying with condition (1), that is $a_1(x, \mathbf{y}) = 4 + y_1 + 0.2\sin(\pi x)y_2 + 0.04\sin(2\pi x)y_3 + 0.008\sin(3\pi x)y_4$, and $\log a_2(x, \mathbf{y}) = y_1 + 0.2\sin(\pi x)y_2 + 0.04\sin(2\pi x)y_3 + 0.008\sin(3\pi x)y_4$. We also set $f(\mathbf{x}) = 1$ and $\Theta(u) = u(0.7)$. For this case, the estimate for the error contribution $\Delta E(\mathbf{i})$ in (3) is (cf. [2])

$$\Delta E(\mathbf{i}) \leq Ce^{-\sum_{n=1}^N g_n m(i_n-1)} \left(\prod_{n=1}^N \mathbb{L}_n^{m(i_n)}\right) \frac{|m(\mathbf{i})|!}{m(\mathbf{i})!},$$

where C is a positive constant, $\mathbb{L}_n^{m(i_n)}$ is the Lebesgue constant associated to the interpolation scheme $\mathcal{U}_n^{m(i_n)}$ that can either be computed numerically or estimated a-priori (cf. [11]), $m(\mathbf{i})! = \prod_n m(i_n)!$, $|m(\mathbf{i})| = \left(\sum_n m(i_n)\right)!$, and g_n can be tuned with cheap preliminary computations, see e.g. [2].

In the second test, we consider instead $\Gamma_n = [-0.99, 0.99]$, $D = [0, 1]^2$ and $a(\mathbf{x}, \mathbf{y}) = 1 + \sum_{n=1}^N \gamma_n \chi_n(\mathbf{x}) y_n$, for $N = 4, 8$. Here $\chi_n(\mathbf{x})$ are the indicator functions of the disjoint circular sub-domains $D_n \subset D$ as in Fig. 1, and γ_n are real coefficients such that (1) holds true; more specifically, we consider both an isotropic setting, $\gamma_n = 1$ for each subdomain, and an anisotropic setting, see Fig. 1 for the values of

[2]When $2^m + 1$ p-Disk Leja points are computed, they coincide with the Clenshaw–Curtis points.



Test	Sample size
Test 1, case a_1	1000
Test 1, case a_2	1000
Test 2, $N = 4$, iso.	5000
Test 2, $N = 4$, aniso.	5000
Test 2, $N = 8$, iso.	50000
Test 2, $N = 8$, aniso.	25000

Fig. 1 *Left*: domains for test 2 with $N = 4$ and $N = 8$, with values of the coefficients γ_n for the anisotropic settings. *Right*: sample size for the Monte Carlo estimate of (5) for each test

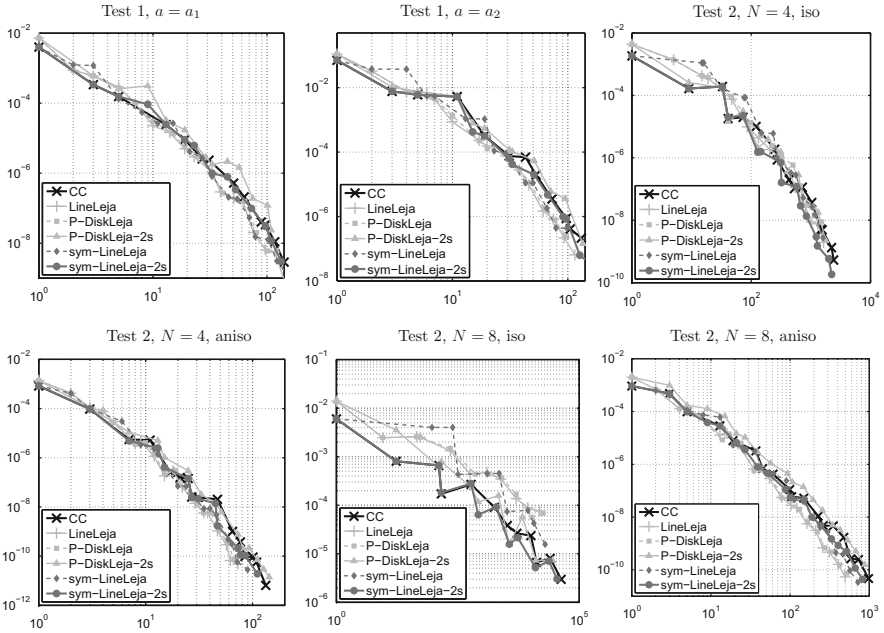


Fig. 2 Convergence of error (5) vs. sparse grids cardinality. The suffix "2s" refers to the "two-stepping" function for Leja points

γ_n in this latter setting. Finally, we set $f(\mathbf{x}) = 100\chi_F(\mathbf{x})$ and $\Theta(u) = \int_F u(\mathbf{x})d\mathbf{x}$. In this case, the estimate for the error contribution $\Delta E(\mathbf{i})$ in (3) is

$$\Delta E(\mathbf{i}) = Ce^{-\sum_{n=1}^{N} g_n m(i_n-1)} \left(\prod_{n=1}^{N} \mathbb{L}_n^{m(i_n)} \right),$$

see [11], where we also provide a convergence estimate for the resulting sparse grid.

Numerical results for the different cases are shown in Fig. 2. It can be seen that Sym-Line Leja points with "two-stepping" seem to have the same (or slightly

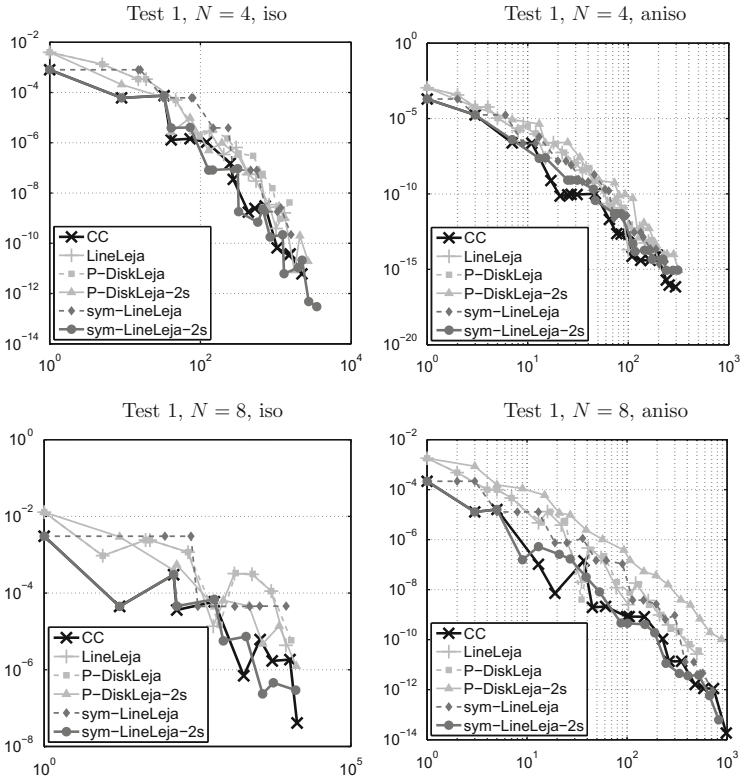


Fig. 3 Convergence of quadrature error for $\Theta(u)$ vs. sparse grids cardinality. The suffix "2s" refers to the "two-stepping" function for Leja points

better) performance than Clenshaw–Curtis points, while the other families of Leja points present slight improvements in some cases but underperform in other tests. Additional tests carried out to monitor the quadrature error for $\Theta(u)$ rather than the interpolation error (5) (see Fig. 3), show again that the performance of Sym-Leja points with two-stepping is comparable to that of Clenshaw–Curtis, while this time the other Leja families always show a slight performance deterioration. This is likely due to the fact that Leja points are designed to minimize the Lebesgue constant, hence more suited for interpolation than for quadrature. Similar results have been found in [10].

In conclusion, these tests seem to suggest that Leja points do not exhibit significative advantages over Clenshaw–Curtis points in the framework of the quasi-optimal sparse grids; moreover, despite the little granularity of the univariate Clenshaw–Curtis points, the number of points in the resulting sparse grids grows similarly to that of grids built with Leja points, due to the fact the quasi-optimal construction adds only one or few hierarchical surpluses per level.

Acknowledgements F. Nobile and L. Tamellini have been partially supported by the Swiss National Science Foundation under the Project No. 140574 "Efficient numerical methods for flow and transport phenomena in heterogeneous random porous media" and by the Center for ADvanced MOdeling Science (CADMOS). R. Tempone is a member of the KAUST SRI Center for Uncertainty Quantification in Computational Science and Engineering.

References

1. I. Babuška, F. Nobile, R. Tempone, A stochastic collocation method for elliptic partial differential equations with random input data. SIAM Rev. **52**(2), 317–355 (2010)
2. J. Beck, F. Nobile, L. Tamellini, R. Tempone, On the optimal polynomial approximation of stochastic PDEs by Galerkin and collocation methods. Math. Models Methods Appl. Sci. **22**(09) (2012)
3. J. Beck, F. Nobile, L. Tamellini, R. Tempone, A quasi-optimal sparse grids procedure for groundwater flows, in *Spectral and High Order Methods for Partial Differential Equations - ICOSAHOM 2012*. Lecture Notes in Computational Science and Engineering, vol. 95 (Springer International Publishing, Switzerland, 2014), pp. 1–16
4. H. Bungartz, M. Griebel, Sparse grids. Acta Numer. **13**, 147–269 (2004)
5. A. Chkifa, On the Lebesgue constant of Leja sequences for the complex unit disk and of their real projection. J. Approx. Theory **166**(0), 176–200 (2013)
6. A. Chkifa, A. Cohen, C. Schwab, High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDEs. Found. Comput. Math. **14**(4), 601–633 (2014)
7. A. Cohen, R. Devore, C. Schwab, Analytic regularity and polynomial approximation of parametric and stochastic elliptic PDE'S. Anal. Appl. **9**(1), 11–47 (2011)
8. T. Gerstner, M. Griebel, Dimension-adaptive tensor-product quadrature. Computing **71**(1), 65–87 (2003)
9. M. Griebel, S. Knapek, Optimized general sparse grid approximation spaces for operator equations. Math. Comput. **78**(268), 2223–2257 (2009)
10. A. Narayan, J.D. Jakeman, Adaptive Leja sparse grid constructions for stochastic collocation and high-dimensional approximation. SIAM J. Sci. Comput. **36**(6), A2952–A2983 (2014)
11. F. Nobile, L. Tamellini, R. Tempone, Convergence of quasi-optimal sparse-grids approximation of Hilbert-space-valued functions: application to random elliptic PDEs. Accepted for publication on Numerische Mathematik. Also available as Mathicse report 12/2014, EPFL
12. C. Schillings, C. Schwab, Sparse, adaptive Smolyak quadratures for Bayesian inverse problems. Inverse Probl. **29**(6) (2013)
13. D. Xiu, J. Hesthaven, High-order collocation methods for differential equations with random inputs. SIAM J. Sci. Comput. **27**(3), 1118–1139 (2005)

From Rankine-Hugoniot Condition to a Constructive Derivation of HDG Methods

Tan Bui-Thanh

Abstract This chapter presents a constructive derivation of HDG methods for convection-diffusion-reaction equation using the Rankine-Hugoniot condition. This is possible due to the fact that, in the first order form, convection-diffusion-reaction equation is a hyperbolic system. As such it can be discretized using the standard upwind DG method. The key is to realize that the Rankine-Hugoniot condition naturally provides an upwind HDG framework. The chief idea is to first break the uniqueness of the upwind flux across element boundaries by introducing single-valued new trace unknowns on the mesh skeleton, and then re-enforce the uniqueness via algebraic conservation constraints. Essentially, the HDG framework is a redesign of the standard DG approach to reduce the number of coupled unknowns. In this work, an upwind HDG method with one trace unknown is systematically constructed, and then extended to a family of penalty HDG schemes. Various existing HDG methods are rediscovered using the proposed framework.

1 Introduction

The high-order discontinuous Galerkin (DG) method was originally developed by Reed and Hill [12] for the neutron transport equation, first analyzed in [8, 9], and then has been extended to other problems governed by partial differential equations (PDEs) [2]. Roughly speaking, DG combines advantages of classical finite volume and finite element methods. However, for steady state problems or time-dependent ones that require implicit time-integrators, DG methods typically have many more (coupled) unknowns compared to the other existing numerical approaches, and hence more expensive in general.

Recently, Cockburn and his coworkers have introduced a hybridizable (also known as hybridized) discontinuous Galerkin (HDG) methods for various type of PDEs including Poisson equation [4, 5], and convection-diffusion equation [3, 11].

T. Bui-Thanh (✉)
Department of Aerospace Engineering and Engineering Mechanics, Institute for Computational Engineering and Sciences, Austin, TX 78712, USA
e-mail: tanbui@ices.utexas.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_45

The beauty of the HDG method is that it reduces the number of coupled unknowns substantially while retaining all other attractive properties of the DG counterpart. The coupled unknowns are in fact unknown traces introduced on the mesh skeleton, i.e. the faces, to hybridize the numerical flux. Once they are solved for, the usual DG unknowns can be recovered in an element-by-element fashion, completely independent of each other. Thus, the HDG methods are well suited for current and future supercomputing systems. Existing HDG constructions however vary from one type of PDE to another, though they do share some similarities. Moreover, they are parameter-dependent method. Consequently, practitioners may be wary of deriving/applying the HDG approach to a new PDE.

In this chapter we seek to develop a systematic and constructive hybridized discontinuous Galerkin (HDG) methods for partial differential equations. For concreteness and clarity of the exposition we choose to present our development for convection-diffusion-reaction equation, though it can be extended to other PDEs. This paper is a continuation of our recent effort [1] on unifying the construction and theory HDG method. Unlike [1], in which we construct HDG schemes from the Godunov approach with upwind flux, in this work we discover a new way to unify HDG methods using the Rankine-Hugoniot jump condition. In fact, we shall show that Rankine-Hugoniot jump condition is, perhaps, the most natural way to construct HDG schemes. In the following, we provide step-by-step the construction of our new unified HDG framework and we refer the readers to [1, 3–5, 11] for a complete description of HDG methodology, its novelties, and its efficiency.

2 Upwind HDG Method and Its Variants for Convection-Diffusion-Reaction Equation

In this section we will systematically devise an upwind HDG scheme for convection-diffusion-reaction in the following first order form

$$\varepsilon^{-1}\boldsymbol{\sigma} + \nabla u = 0 \quad \text{in } \Omega, \quad \text{and} \quad \nabla \cdot \boldsymbol{\sigma} + \nabla \cdot (\boldsymbol{\beta} \cdot u) + \nu u = f \quad \text{in } \Omega \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, and we take $d = 3$ for concreteness; the velocity field $\boldsymbol{\beta}$ is assumed to be continuous; ε is the diffusion coefficient; ν is the reaction parameter; and f is the forcing term. Since the boundary condition plays no role in the basic construction and understanding of our upwind HDG framework, it will be ignored.

If we define $\mathbf{u} := [\boldsymbol{\sigma}, u]$ we can rewrite (1) in a more compact form as

$$\nabla \cdot \mathscr{F}(\mathbf{u}) + \mathbf{C}\mathbf{u} = \mathbf{f}, \quad \text{in } \Omega, \tag{2}$$

where $\mathbf{f} := [\mathbf{0}, f]$, and $\mathbf{C}$ is a 4×4 matrix with $\mathbf{C}(1,1) = \mathbf{C}(2,2) = \mathbf{C}(3,3) = \varepsilon^{-1}$, $\mathbf{C}(4,4) = \nu$ and $C(i,j) = 0$ otherwise. Here, the flux tensor $\mathscr{F}$ is given by

$\mathscr{F}(\mathbf{u}) := \mathscr{A}\mathbf{u}$ and $\mathscr{A}$ is a tensor with three components defined as

$$\mathscr{A}^1 := \left[\begin{array}{ccc|c} 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \hline 1 & 0 & 0 & \boldsymbol{\beta}^1 \end{array}\right], \quad \mathscr{A}^2 := \left[\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \\ \hline 0 & 1 & 0 & \boldsymbol{\beta}^2 \end{array}\right], \text{ and } \quad \mathscr{A}^3 := \left[\begin{array}{ccc|c} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ \hline 0 & 0 & 1 & \boldsymbol{\beta}^3 \end{array}\right].$$

Now, let $\mathbf{n} := \left[\mathbf{n}^1, \mathbf{n}^2, \mathbf{n}^3\right]$ be an arbitrary unit vector, we observe that

$$\mathbf{A} := \mathscr{A} \cdot \mathbf{n} = \left[\begin{array}{ccc|c} 0 & 0 & 0 & \mathbf{n}^1 \\ 0 & 0 & 0 & \mathbf{n}^2 \\ 0 & 0 & 0 & \mathbf{n}^3 \\ \hline \mathbf{n}^1 & \mathbf{n}^2 & \mathbf{n}^3 & \boldsymbol{\beta} \cdot \mathbf{n} \end{array}\right] \tag{3}$$

has four eigenvalues $[c_1, c_2, c_2, c_3]$:

$$[c_1, c_2, c_2, c_3] := \left[\frac{\boldsymbol{\beta} \cdot \mathbf{n}}{2} - \frac{\sqrt{|\boldsymbol{\beta} \cdot \mathbf{n}|^2 + 4}}{2}, 0, 0, \frac{\boldsymbol{\beta} \cdot \mathbf{n}}{2} + \frac{\sqrt{|\boldsymbol{\beta} \cdot \mathbf{n}|^2 + 4}}{2}\right].$$

It can be inspected that the eigen-values are real and eigen-vectors are independent. Consequently, (1) is a steady state hyperbolic system (see, e.g., [13] for definition of hyperbolicity), though the original convection-diffusion-reaction is not purely hyperbolic (in fact elliptic if $\boldsymbol{\beta} = \mathbf{0}$). As such, it can be discretized and solved using upwind numerical methods such as DG.

The *goal* of this section is to provide a systematic construction of an upwind HDG framework for convection-diffusion-reaction equation (1). Let us begin by introducing some notations and conventions. The domain Ω is partitioned into N_{el} non-overlapping elements $K_j, j = 1, \ldots, N_{\text{el}}$ with Lipschitz boundaries such that $\Omega_h := \cup_{j=1}^{N_{\text{el}}} K_j$ and $\overline{\Omega} = \overline{\Omega}_h$. We denote the skeleton of the mesh by $\mathscr{E}_h := \cup_{j=1}^{N_{\text{el}}} \partial K_j$; it is the set of all (uniquely defined) faces e. We conventionally identify the normal vector $\mathbf{n}^-$ on the boundary ∂K of the element K under consideration (also denoted as K^-) and $\mathbf{n}^+ = -\mathbf{n}^-$ as the normal of the boundary of a neighboring element (also denoted as K^+). On the other hand, we use $\mathbf{n}$ to denote either $\mathbf{n}^-$ or $\mathbf{n}^+$ in an expression that is valid for both cases, and this convention is also used for other quantities (restricted) on $e \in \mathscr{E}_h$. For the sake of convenience, we denote by $\mathscr{E}_h^\partial$ the sets of all boundary faces and define $\mathscr{E}_h^o := \mathscr{E}_h \setminus \mathscr{E}_h^\partial$ the set of all interior faces.

For simplicity in writing we define $(\cdot, \cdot)_K$ as the L^2-inner product on a domain $K \in \mathbb{R}^d$ and $\langle \cdot, \cdot \rangle_K$ as the L^2-inner product on a domain K if $K \in \mathbb{R}^{d-1}$. We shall use bold-face lowercase/uppercase letters for vector-valued functions and in that case the inner product is defined as $(\mathbf{u}, \mathbf{v})_K := \sum_{i=1}^m \left(\mathbf{u}^i, \mathbf{v}^i\right)_K$, and similarly as $\langle \mathbf{u}, \mathbf{v} \rangle_K := \sum_{i=1}^m \left\langle \mathbf{u}^i, \mathbf{v}^i \right\rangle_K$, where m is the number of components $(\mathbf{u}^i, i = 1, \ldots, m)$ of $\mathbf{u}$. We also employ upper case calligraphic letter, e.g. $\mathscr{F}$, to denote tensors. It is

our convention that superscripts are used to denote the components of vector, matrix, and tensor. We shall not distinguish row and column vectors in what follows.

We define $\mathscr{P}^p(K)$ as the space of polynomials of degree at most p on the domain K. Next, we introduce two discontinuous piecewise polynomial spaces

$$\mathbf{V}_h(\Omega_h) := \left\{\mathbf{v} \in \left[L^2(\Omega)\right]^m : \mathbf{v}|_K \in [\mathscr{P}^p(K)]^m, \forall K \in \Omega_h\right\},$$
$$\Lambda_h(\mathscr{E}_h) := \left\{\lambda \in L^2(\mathscr{E}_h) : \lambda|_e \in \mathscr{P}^p(e), \forall e \in \mathscr{E}_h\right\},$$

and similarly for $\mathbf{V}_h(K)$, and $\boldsymbol{\Lambda}_h(e)$ by replacing Ω_h with K and $\mathscr{E}_h$ with e. If $m = 1$, i.e. scalar-valued functions, we define

$$V_h(\Omega_h) := \left\{v \in L^2(\Omega) : v|_K \in \mathscr{P}^p(K), \forall K \in \Omega_h\right\}.$$

From now on we conventionally use $\mathbf{u}$ for DG solution. We would like to find local finite element solution $\mathbf{u} \in \mathbf{V}_h(K)$ on each element $K \in \Omega_h$. To that end, multiplying (2) by $\mathbf{v}$ and integrating by parts we have

$$-(\mathscr{F}(\mathbf{u}), \nabla\mathbf{v})_K + \langle \mathscr{F}(\mathbf{u}) \cdot \mathbf{n}, \mathbf{v}\rangle_{\partial K} + (\mathbf{C}\mathbf{u}, \mathbf{v})_K = (\mathbf{f}, \mathbf{v})_K, \quad \forall \mathbf{v} \in \mathbf{V}_h(K). \tag{4}$$

At this point, the flux $\mathscr{F}(\mathbf{u})\cdot\mathbf{n}$ on $e \in \partial K$ is not well-defined since the traces of both $\mathbf{u}^-$ of element K^- and $\mathbf{u}^+$ of element K^+ co-exist on e. Godunov's type methods [6] resolves this by introducing some (typically upwind, see e.g. [10, 13]) numerical flux $\mathscr{F}^*\left(\mathbf{u}^-, \mathbf{u}^+\right)$ to replace $\mathscr{F}(\mathbf{u})$ on the boundary term in (4) so that (4) becomes

$$-(\mathscr{F}(\mathbf{u}), \nabla\mathbf{v})_K + \left\langle \mathscr{F}^*\left(\mathbf{u}^-, \mathbf{u}^+\right) \cdot \mathbf{n}, \mathbf{v}\right\rangle_{\partial K} + (\mathbf{C}\mathbf{u}, \mathbf{v})_K = (\mathbf{f}, \mathbf{v})_K. \tag{5}$$

It should be pointed out that for simplicity in writing we have ignored the fact (5) must hold for all test functions $\mathbf{v} \in \mathbf{V}_h(K)$; to the end of the chapter, this should be implicitly understood.

It is the upwind numerical flux $\mathscr{F}^*$ that couples local unknowns on elements K^+ and K^- that share a face $e \in \partial K$. Consequently, local unknowns on all elements are coupled (for steady state problems or time-dependent problem with implicit time-integrators), and they must be solved together. This leads to the "usual complaint" that DG has so many coupled unknowns, and hence is expensive, though it has many attractive properties.

What we are going to do next is to remove this coupling by introducing new trace unknowns that live on the mesh skeleton. The beauty of this approach is that the actual globally coupled unknowns are those newly introduced trace unknowns, and hence the resulting system is substantially smaller and sparser. Once the trace unknowns are computed, the local DG unknown $\mathbf{u}$ is computed locally element-by-element independent of each other. *We shall show that the Rankine-Hugoniot condition (see, e.g. [13]) provides all the necessary ingredients for accomplishing this decoupling task*. To that end, let us sketch in Fig. 1 the wave

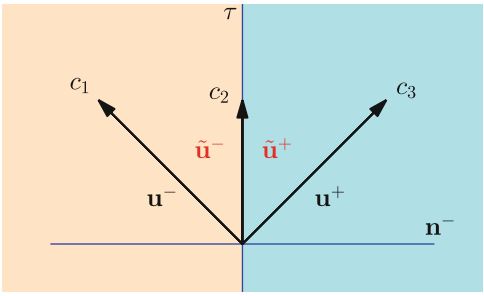


Fig. 1 The wave structure in the Riemann problem for first order form of convection-diffusion-reaction equation (1) with pseudo-time τ

structure of the Riemann problem for the first order PDE system (1) along the normal direction of the interface between K^- and K^+. Here, τ *is the pseudo-time*.

Applying the Rankine-Hugoniot condition across each wave we obtain

$$(\mathscr{A}^- \cdot \mathbf{n}^-)\,\tilde{\mathbf{u}}^- - (\mathscr{A}^- \cdot \mathbf{n}^-)\,\mathbf{u}^- = c_1\,(\tilde{\mathbf{u}}^- - \mathbf{u}^-)\,, \tag{6a}$$

$$\left(\mathscr{A}^+ \cdot \mathbf{n}^-\right)\tilde{\mathbf{u}}^+ - (\mathscr{A}^- \cdot \mathbf{n}^-)\,\tilde{\mathbf{u}}^- = 0, \tag{6b}$$

$$\left(\mathscr{A}^+ \cdot \mathbf{n}^-\right)\mathbf{u}^+ - \left(\mathscr{A}^+ \cdot \mathbf{n}^-\right)\tilde{\mathbf{u}}^+ = c_3\left(\mathbf{u}^+ - \tilde{\mathbf{u}}^+\right). \tag{6c}$$

On the other hand, from definition of $\mathscr{A}$ and the continuity of $\boldsymbol{\beta}$ we have

$$\mathscr{A}^- = \mathscr{A}^+ = \mathscr{A}, \quad \text{and } (\mathscr{A} \cdot \mathbf{n})\,\mathbf{u} = [u\mathbf{n}, \boldsymbol{\sigma} \cdot \mathbf{n} + \boldsymbol{\beta} \cdot \mathbf{n}u]$$

which, together with (6b), imply

$$u^* := \tilde{u}^- = \tilde{u}^+, \quad \text{and} \quad \boldsymbol{\sigma}^* \cdot \mathbf{n} := \tilde{\boldsymbol{\sigma}}^- \cdot \mathbf{n} = \tilde{\boldsymbol{\sigma}}^+ \cdot \mathbf{n}$$

where $\mathbf{u}^* := [\boldsymbol{\sigma}^*, u^*]$ is defined as the upwind state, which is also the Riemann solution in this case [see (10a) and (10b)]. The upwind flux is then defined as

$$\mathscr{F}^* \cdot \mathbf{n} := (\mathscr{A} \cdot \mathbf{n})\,\mathbf{u}^*.$$

Using the definition of c_1, c_3, and $\mathscr{A}$, we can rewrite both (6a) and (6c) in a general form, referring to either K^- or K^+, as

$$\mathscr{F}^* \cdot \mathbf{n} = \begin{bmatrix} u\mathbf{n} + \frac{1}{2}\,(\alpha - \boldsymbol{\beta} \cdot \mathbf{n})\,(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*) \\ \boldsymbol{\beta} \cdot \mathbf{n}u + \boldsymbol{\sigma} \cdot \mathbf{n} + \frac{1}{2}\,(\alpha - \boldsymbol{\beta} \cdot \mathbf{n})\,(u - u^*) \end{bmatrix}, \tag{7}$$

with α given by $\alpha := \sqrt{|\boldsymbol{\beta} \cdot \mathbf{n}|^2 + 4}$. Since the first three components of left hand sides of (7) is a vector parallel to $\mathbf{n}$, the tangent component of the corresponding

vector consisting of the first three components of right hand side must vanish. This observation allows us to rewrite the Rankine-Hugoniot conditions (7) as

$$\mathscr{F}^* \cdot \mathbf{n} = \begin{bmatrix} u\mathbf{n} + \frac{1}{2}(\alpha - \boldsymbol{\beta}\cdot\mathbf{n})(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*)\cdot\mathbf{n}\,\mathbf{n} \\ \boldsymbol{\beta}\cdot\mathbf{n}u + \boldsymbol{\sigma}\cdot\mathbf{n} + \frac{1}{2}(\alpha - \boldsymbol{\beta}\cdot\mathbf{n})(u - u^*) \end{bmatrix}. \tag{8}$$

Since $\mathscr{F}^* \cdot \mathbf{n}$ is the upwind flux, it obviously satisfies

$$[\![\mathscr{F}^* \cdot \mathbf{n}]\!] = \mathbf{0}, \tag{9}$$

where we have defined the "jump" operator as $[\![(\cdot)]\!] := (\cdot)^- + (\cdot)^+$. We also define "average" operator as $\{\!\{(\cdot)\}\!\} := \frac{1}{2}[\![(\cdot)]\!]$.

Lemma 1 *The following hold true:*

i) The upwind state $\mathbf{u}^*$ *satisfies*

$$u^* = \{\!\{u\}\!\} + \frac{\boldsymbol{\beta}\cdot\mathbf{n}}{2\alpha}[\![u\mathbf{n}]\!]\cdot\mathbf{n} + \frac{1}{\alpha}[\![\boldsymbol{\sigma}\cdot\mathbf{n}]\!], \tag{10a}$$

$$\boldsymbol{\sigma}^*\cdot\mathbf{n} = \{\!\{\boldsymbol{\sigma}\}\!\}\cdot\mathbf{n} + \frac{1}{\alpha}[\![u\mathbf{n}]\!]\cdot\mathbf{n} - \frac{\boldsymbol{\beta}\cdot\mathbf{n}}{2\alpha}[\![\boldsymbol{\sigma}\cdot\mathbf{n}]\!], \tag{10b}$$

ii) The upwind flux is given by

$$\mathscr{F}^*\cdot\mathbf{n} = \left[u^*\mathbf{n}_1,\ u^*\mathbf{n}_2,\ u^*\mathbf{n}_3,\ \boldsymbol{\beta}\cdot\mathbf{n}u + \boldsymbol{\sigma}\cdot\mathbf{n} + \frac{1}{2}(\alpha - \boldsymbol{\beta}\cdot\mathbf{n})\left(u - u^*\right)\right], \tag{11}$$

where

$$u^* = u + \frac{2}{\alpha}\left(\boldsymbol{\sigma} - \boldsymbol{\sigma}^*\right)\cdot\mathbf{n} + \frac{\boldsymbol{\beta}\cdot\mathbf{n}}{\alpha}\left(u - u^*\right). \tag{12}$$

Proof We know that the conservation (9) gives us four equations for the upwind state $\mathbf{u}^*$. Solving for u^* and $\boldsymbol{\sigma}^*\cdot\mathbf{n}$ in terms of u and $\boldsymbol{\sigma}$ we obtain the desired result. The second assertion immediately follows by substituting (10) into (8) and inspecting that (12) is true.

Up to this point, we have used the exact upwind state $\mathbf{u}^*$ and the upwind flux $\mathscr{F}^*\cdot\mathbf{n}$ to derive identities in Lemma 1. In particular, we have shown that the upwind flux of the form (11) naturally arises from the Rankine-Hugoniot condition. *The appealing feature of this form is that the upwind flux depends on the DG unknowns of only one side of a face* $e \in \partial K$ *and the single-valued upwind state* u^*. As such, it is completely determined using only information from either side (K^- or K^+) of the face $e \in \partial K$, as long as u^* is (either exactly or approximately) provided. More importantly, this in turn shows that we can solve Eq. (5) for $\mathbf{u}$ element-by-element independent of each other. This observation suggests that we should treat u^* as the

extra unknown and solve for it on the skeleton of the mesh instead of using the upwind value which couples the local unknown $\mathbf{u}$ on elements. To signify this step, let us rename u^* to $\hat{u}$ and $\mathscr{F}^*$ to $\hat{\mathscr{F}}$, i.e.,

$$\hat{\mathscr{F}} \cdot \mathbf{n} = \left[\hat{u}\mathbf{n}_1, \quad \hat{u}\mathbf{n}_2, \quad \hat{u}\mathbf{n}_3, \quad \boldsymbol{\beta} \cdot \mathbf{n} u + \boldsymbol{\sigma} \cdot \mathbf{n} + \frac{1}{2} (\alpha - \boldsymbol{\beta} \cdot \mathbf{n}) (u - \hat{u}) \right], \tag{13}$$

where $\hat{u}$ is the single-valued trace unknown on the mesh skeleton that needs to be solve for.

An immediate question that arises is how to compute $\hat{u}$. To answer this question, we note that $\hat{u}$ is a new unknown that is introduced on ∂K so that (5) can be solved in an element-by-element fashion. To ensure the well-posedness of our formulation, we need to introduce an extra equation on ∂K. Clearly, at this point $\hat{u}$ is not the upwind state and hence identity (9) is in general no longer satisfied for $\hat{\mathscr{F}}$. It is therefore natural to use (9) as the extra equation. This additional algebraic equation ensures that what coming out from element K through its boundary ∂K must enter the neighboring elements that share (part of) the boundary ∂K. This is the statement of conservation and it is exactly conveyed by (9). Due to the single-valued nature of $\hat{u}$, the first three components of our HDG flux (13) automatically satisfy the conservation condition (9). For the fourth one, enforcing (9) weakly is sufficient for local conservation, i.e., $\forall e \in \mathscr{E}_h^o$:

$$\left\langle [\![\boldsymbol{\beta} \cdot \mathbf{n} u + \boldsymbol{\sigma} \cdot \mathbf{n} + \frac{1}{2} (\alpha - \boldsymbol{\beta} \cdot \mathbf{n}) (u - \hat{u})]\!], \mu \right\rangle_e = 0, \quad \forall \mu \in \Lambda_h(e). \tag{14}$$

In summary, *we define an upwind HDG method by hybridizing the upwind flux of the standard DG scheme*. In particular, it has the usual DG local unknown $\mathbf{u}$ and the extra "trace" unknown $\hat{u}$. These unknowns can be solved for using the global conservation constraint (14) and the local solver (5) with $\mathscr{F}^*$ replaced by $\hat{\mathscr{F}}$.

We now generalize our upwind HDG approach to a class of penalty HDG schemes, a member of which is the upwind HDG itself. To that end, we first observe that $u - \hat{u}$ is the mismatch between the volume unknown restricted on the mesh skeleton and trace unknown. This mismatch vanishes for the exact solution, but converges to zero for the HDG solution as the mesh (or solution order) is refined. This suggests that one can control the mismatch by introducing a penalty parameter λ to form a penalized family of HDG fluxes as follows

$$\hat{\mathscr{F}} \cdot \mathbf{n} = [\hat{u}\mathbf{n}_1, \quad \hat{u}\mathbf{n}_2, \quad \hat{u}\mathbf{n}_3, \quad \boldsymbol{\beta} \cdot \mathbf{n} u + \boldsymbol{\sigma} \cdot \mathbf{n} + \lambda (u - \hat{u})]. \tag{15}$$

Clearly, when $\lambda = \frac{1}{2} (\alpha - \boldsymbol{\beta} \cdot \mathbf{n})$ we recover the proposed upwind HDG scheme.

Next, we discuss the relation of our penalty HDG family, and hence upwind HDG, with other existing HDG ones. It is necessary brief since a more detailed discussion can be found in our previous work [1]. To begin, we observe that, for general convection-diffusion-reaction problem (and similarly for pure convection

problem), if we replace λ by $\lambda - \boldsymbol{\beta}\cdot\mathbf{n}$ in the HDG flux (15), we obtain

$$\hat{\mathscr{F}}\cdot\mathbf{n} = \left[\hat{u}\mathbf{n}_1, \quad \hat{u}\mathbf{n}_2, \quad \hat{u}\mathbf{n}_3, \quad \boldsymbol{\beta}\cdot\mathbf{n}\hat{u} + \boldsymbol{\sigma}\cdot\mathbf{n} + \lambda\,(u-\hat{u})\right].$$

This is exactly the HDG scheme proposed in [11].

For the Poisson equation, our penalty HDG flux (15) simplifies to

$$\hat{\mathscr{F}}\cdot\mathbf{n} = \left[\hat{u}\mathbf{n}_1, \quad \hat{u}\mathbf{n}_2, \quad \hat{u}\mathbf{n}_3, \quad \boldsymbol{\sigma}\cdot\mathbf{n} + \lambda\,(u-\hat{u})\right], \tag{16}$$

which is exactly the HDG method originally proposed in [4]. It is important to point out that since the differential part of the Helmholtz equation is the same as that of the Poisson equation, HDG flux for the Helmholtz equation using our framework is identical to (16). That is, we have also recovered the HDG scheme for Helmholtz equation proposed in [7]. Finally, we refer to [1, 3–5, 7, 11] for a rigorous analysis of all HDG methods presented in this chapter.

3 Conclusions

We have presented a constructive methodology to derive HDG methods for convection-diffusion-reaction equation. In particular, we have shown that the Rankine-Hugoniot condition, in its primitive form, is already a hybridization of the upwind flux. The chief idea is to first break the uniqueness of the upwind flux across element boundaries by introducing single-valued trace unknowns on the mesh skeleton, and then re-enforce the uniqueness via algebraic conservation constraints. We have devised in details the construction of our upwind HDG method and extended it to a family of penalty HDG schemes. The proposed framework allows one to rediscover many existing HDG methods. Ongoing work is to apply the proposed framework to constructively derive HDG methods for other PDEs.

References

1. T. Bui-Thanh, From Godunov to a unified hybridized discontinuous Galerkin framework. J. Comput. Phys. **295**, 114–146 (2015)
2. B. Cockburn, G.E. Karniadakis, C.-W. Shu, *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Lecture Notes in Computational Science and Engineering, vol. 11 (Springer, Heidelberg, 2000)
3. B. Cockburn, B. Dong, J. Guzman, M. Restelli, R. Sacco, A hybridizable discontinuous Galerkin method for steady state convection-diffusion-reaction problems. SIAM J. Sci. Comput. **31**, 3827–3846 (2009)
4. B. Cockburn J. Gopalakrishnan, R. Lazarov, Unified hybridization of discontinuous Galerkin, mixed, and continuous Galerkin methods for second order elliptic problem. SIAM J. Numer. Anal. **47**, 1319–1365 (2009)

5. B. Cockburn, J. Gopalakrishnan, F.-J. Sayas, A projection-based error analysis of HDG methods. Math. Comput. **79**, 1351–1367 (2010)
6. S.K. Godunov, A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. Mat. Sb. **47**, 357–393 (1959)
7. R. Griesmaier, P. Monk, Error analysis for a hybridizable discontinuous Galerkin method for the Helmholtz equation. J. Sci. Comput. **49**, 291–310 (2011)
8. C. Johnson, J. Pitkäranta, An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. Math. Comput. **46**, 1–26 (1986)
9. P. LeSaint, P.A. Raviart, On a finite element method for solving the neutron transport equation, in *Mathematical Aspects of Finite Element Methods in Partial Differential Equations* (Academic, New York, 1974)
10. R.J. LeVeque, *Finite Volume Methods for Hyperbolic Problems* (Cambridge University Press, Cambridge, 2002)
11. N.C. Nguyen, J. Peraire, B. Cockburn, An implicit high-order hybridizable discontinuous Galerkin method for linear convection-diffusion equations. J. Comput. Phys. **228**, 3232–3254 (2009)
12. W.H. Reed, T.R. Hill, Triangular mesh methods for the Neutron Transport Equation. Los Alamos Scientific Laboratory, 1973
13. E.F. Toro, *Riemann Solvers and Numerical Methods for Fluid Dynamics* (Springer, Heidelberg, 1999)

Numerical Simulation of Two-Phase Flows Using Fourier Pseudospectral Method

Mariana Fernandes dos Santos Villela, Felipe Pamplona Mariano, and Aristeu da Silveira-Neto

Abstract The present work proposes the extension of the IMERSPEC methodology for numerical simulations of two-phase flows. This methodology consists of the fusion between the Fourier pseudospectral method and the immersed boundary method for non-periodical problems. This method was originally developed for single-phase and incompressible flows (Mariano et al., Comput Model Eng Sci 59:181–216, 2010). In the present paper, we extend this methodology for two-phase flows using the front-tracking method to model the fluid-fluid interface. The results involving the spurious currents, mass conservation and analysis through numerical experimental bubbles rise, show that the proposed method can be considered validated and promising to computational fluid dynamics (CFD).

1 Introduction

Multiphase flows have a significant role in a vast area of geophysics and industrial processes. All of these applications stimulate the research of bubbles. One of the main issues is to understand how a bubble moves in flow and as the continuous phase is affected by the dispersed phase. Experimental studies of rising gas bubbles in a static fluid began in the decade 60 with the works of [4] and others. Some years later, Clift et al. [2] proposed to unify the treatment of solid, liquid droplets and gas bubbles.

The improvement of computers has allowed the direct numerical simulation of flows, using the Navier-Stokes equations as another way to perform experiments.

M.F.d.S. Villela (✉) • A. da Silveira-Neto
Laboratory of Mechanical of Fluids, Department of Mechanical Engineering, Federal University of Uberlândia, Campus Sta. Mônica, Av: João Naves de Ávila, 2121, Bloco 5p, CEP: 38400-902, Uberlândia, Brazil
e-mail: marianamat_ufu@yahoo.com.br; aristeus@mecanica.ufu.br

F.P. Mariano
School of Electric, Mechanical and Computational Engineering, Federal University of Goiás, Av.: Universitária, 1488, Bloco: A, Piso: 3, CEP: 74.605-010, Goiânia, Brazil
e-mail: fpmariano@ufg.br

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_46

This methodology becomes a powerful tool for validating results, and make possible the extension to complex cases of fluid dynamics. The search for more accurate and low computational cost for the simulation of two-phase flow methods is of great interest for industry, since they require information from the flows with higher level of accuracy.

The Fourier pseudospectral method is a methodology with a high rate of numerical convergence, providing high accuracy at a low computational cost, when compared with other high accuracy methodologies, owing to the use of Fast Fourier Transform (FFT). It is also observed that for the Navier-Stokes equations, considering incompressible flow in spectral space, the projection method is used to eliminate the pressure gradient term, leading to uncoupling of the pressure-velocity fields and eliminating the need to solve the Poisson equation for the pressure. The major limitation of this methodology is in the boundary conditions, which are required to be periodical, an exigence of the Fourier spectral method, [1]. However, [5] overcame this limitation by coupling the Fourier pseudospectral method with the immersed boundary method, which allowed to simulate non-periodic problems, using Fourier pseudospectral method.

Thus, the purpose of the present paper is to show a proposition of a new mathematical and computational methodology to solve two-phase flows problems which provides at the same time, high computational efficiency and high accuracy. For this, we used Fourier pseudospectral method (FSM) coupled with immersed boundary (IB) and with the Front-Tracking method (FTM). Results obtained by analysis through numerical experiment of rising bubbles show that the proposed method can be considered validated and very promising.

2 Mathematical Modeling for Two Phase-Flow

The present work is based on the merging FPSM, IBM and FTM method. The fluid as a whole is represented by $\Omega = \Omega_1 \cup \Omega_2$ domain. The interface Γ is called Lagrangian, which can move and deform. The subdomains Ω_1 and Ω_2 represent the outside and inside of the interface Γ, respectively. Variables with uppercase letters are related to Lagrangian domain (Γ) whereas a lowercase letter is related to Eulerian domain (Ω).

The mathematical model for incompressible and isothermal flows of Newtonian fluids with variable physical properties are composed by mass conservation and Navier-Stokes equations, Eqs. (1) and (2), respectively. Such equations present source terms that model the boundary conditions for momentum and the interface force.

$$\frac{\partial u_l}{\partial x_l} = 0, \tag{1}$$

$$\left(\frac{\partial u_l}{\partial t}+\frac{\partial u_l u_k}{\partial x_k}\right)=-\frac{1}{\rho(\phi)}\frac{\partial p}{\partial x_l}+\frac{1}{\rho(\phi)}\frac{\partial}{\partial x_k}\left[\mu(\phi)\left(\frac{\partial u_l}{\partial x_k}+\frac{\partial u_k}{\partial x_l}\right)\right]+g_l+\frac{1}{\rho(\phi)}f_{\sigma_l}+\frac{1}{\rho(\phi)}f_{FI_l}, \tag{2}$$

where $\rho(\phi)$ and $\mu(\phi)$ are, respectively, the density and the coefficient of dynamic viscosity of the fluid; ϕ is used as phase indicator; u_l is the component l of the velocity vector; p is the pressure field; g_l is the component l of the gravitational acceleration vector and x_l, with $l = 1, 2$, for two-dimensional problems, are the spatial coordinates and t is the time.

The source term f_{FI_l} models non-periodic boundary conditions for momentum equation through multi-direct-forcing method, as presented by Mariano et al. [5]. The source term f_{σ_l} represents the interface and its equation is given by:

$$f_l(\mathbf{x},t)=\int_\Gamma F_l(\mathbf{X},t)\delta(\mathbf{x}-\mathbf{X})d\mathbf{X}, \tag{3}$$

where $\delta(\mathbf{x}-\mathbf{X})$ is the Dirac delta function. $F_l(\mathbf{X},t)$ is the component l of the interface force vector calculated at Γ and X_l is the component l of the position vector of the interface (Γ) (see Fig. 1).

The discretization process of the δ function is replaced by a smooth function D_h, Eq. (4), [6]:

$$D_h(\mathbf{x}-\mathbf{X})=\frac{1}{h^2}W_{cos}\left(\frac{X-x}{h}\right)W_{cos}\left(\frac{Y-y}{h}\right), \tag{4}$$

$$W_{cos}(r)=\begin{cases}\frac{1}{4}\left[1+cos\left(\frac{\pi|r|}{2}\right)\right] & \text{se } 0\neq|r|<2\\ 0 & \text{se } 2\leq|r|\end{cases}, \tag{5}$$

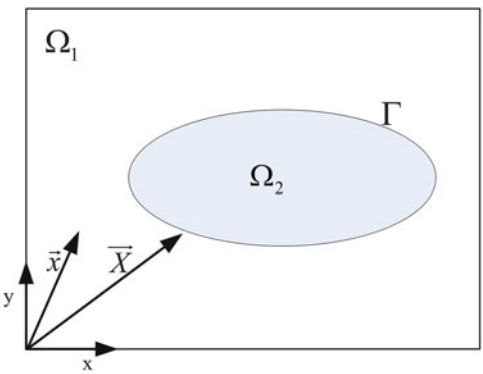


Fig. 1 Representation of Eulerian domains $\Omega = \Omega_1 \cup \Omega_2$ and Lagrangian Γ interface

where $\mathbf{r} = \frac{\mathbf{X}-\mathbf{x}}{h}$, and h spacing of discrete domain, Δx e Δy [5]. It has a behavior similar to a Gaussian and attend the property unitary integral in the range $[-\infty, \infty]$.

An indicator function $\phi(\mathbf{x}, t)$ is used to determine the distribution of the Eulerian physical properties, such as the density ρ and the dynamic viscosity μ. This function is calculated at each time step solving a Poisson equation, which does not require the solution of a linear system when using the spectral Fourier method, reducing the computational cost. The values of the Eulerian physical properties range from 0 for continuous phase to 1 for the dispersed phase. Whenever $\phi(\mathbf{x}, t)$ is obtained, $\rho(\mathbf{x}, t)$ and $\mu(\mathbf{x}, t)$ are determined.

3 Mathematical Modeling for Interfacial Force

The interface Γ is represented parametrically by $(X(s,t), Y(s,t))$, where X and Y are the Lagrangian coordinates, and s is the arc length parameter, $0 \leq s \leq L_b$, where $1 \leq q \leq N_L$, N_L is the total number of Lagrangian points and L_b is the total length of the interface (Fig. 2).

The modeling of the Lagrangian density force F_{σ_l}, is given by a balance of forces on an arbitrary point of the segment of interface and we obtain:

$$F_{\sigma_l} = \sigma \kappa n_l, \tag{6}$$

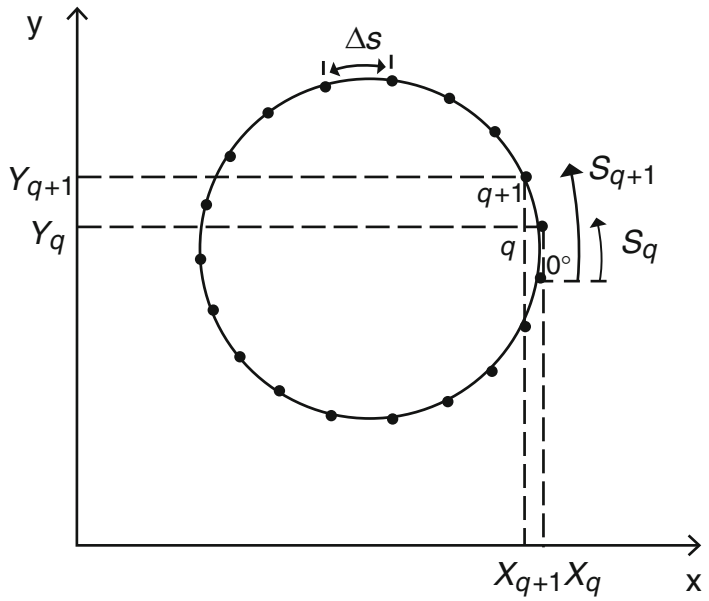


Fig. 2 Parameter s for an interface Γ represented by a *closed curve*

where κ is the mean curvature of the interface, σ is the surface tension coefficient and n_l is a component l the normal unit vector of the interface, n_l and κ are given by:

$$n_l = \frac{\partial \tau_l}{\partial s} / \parallel \frac{\partial \tau_l}{\partial s} \parallel, \tag{7}$$

$$\kappa = \parallel \frac{\partial \tau_l}{\partial s} \parallel / \parallel \frac{\partial X_l}{\partial s} \parallel, \tag{8}$$

where $\tau_l = \frac{\partial X_l}{\partial s} / \parallel \frac{\partial X_l}{\partial s} \parallel$ is a component l the tangent unit vector of the interface.

4 Mathematical Modeling for Fourier Spectral Method

The Navier-Stokes and continuity equations must be transformed to the Fourier spectral space. For incompressible flows, the projection method for decoupling the velocity from the pressure is used. Further, the properties of the Fourier transform are applied [1]. Rewriting Eqs. (1) and (2) in the Fourier spectral space, results:

$$ik_l \widehat{u_l} = 0, \tag{9}$$

$$\widehat{\frac{\partial u_l}{\partial t}} = \wp_{lm} \left[-\widehat{TNL_m} + \widehat{DIF_m} + \widehat{g_m} + \frac{1}{\widehat{\rho(\phi)}} * \widehat{f_{\sigma_m}} + \frac{1}{\widehat{\rho(\phi)}} * \widehat{f_{FI_m}} \right], \tag{10}$$

where $\widehat{TNL_l}$ is the nonlinear term of the Navier-Stokes equation $(\widehat{ik_k(u_l u_k)})$ and $\widehat{DIF_l}$ is the diffusive term of the Navier-Stokes equation $\left(\frac{1}{\hat{\rho}} * ik_k * [\hat{\mu} * (ik_k \widehat{u_l} + ik_l \widehat{u_k})]\right)$ and the symbol $*$ represent the convolution product.

The resolution of the convolution product undergoes a convolution integral that is the result of processing the product of two functions. It is not feasible to solve computationally. Therefore, use is made of Fourier pseudospectral method, that is the multiplication of two functions in the physical space [8].

In Eq. (10) pressure field term vanishes on the right hand side. However, this term can be recovered using the equation:

$$p(\mathbf{x}, t) = IFT \left\{ \frac{ik_m}{k^2} \left[\hat{\rho} * \left(\widehat{TNL_m} - \widehat{DIF_m} - \widehat{g_m} - \frac{1}{\widehat{\rho(\phi)}} * \widehat{f_{\sigma_m}} - \frac{1}{\widehat{\rho(\phi)}} * \widehat{f_{FI_m}} \right) \right] \right\}, \tag{11}$$

where *IFT* is inverse Fourier transform.

5 Results

5.1 Code Verification

Code verification is a study of the computer program which is to ensure that the equations chosen for a given model are resolved correctly and in quantifying the numerical errors of the solution. It is a purely mathematical exercise and no physical realism needs to be satisfied. Good practice of the code verification is to simulate a problem that has exact solution that mimics the physical problem of interest. This kind of accurate solution can be obtained by the method of manufactured solutions (MMS), which is based on the introduction of source terms in the governing equations, creating an unrealistic problem, but an analytical solution [3].

To verify the implementation of the Navier-Stokes equations with variable physical properties is done through the MMS and the analytical solutions for the horizontal and vertical velocity fields, u, v, the pressure field p and the density (ρ_e) and the viscosity (μ_e) variables are given by Eq. (12), [8]:

$$u^e(x,t) = sin^2(2\pi x + 2\pi y + t), \tag{12}$$

$$v^e(x,t) = cos^2(2\pi x + 2\pi y + t), \tag{13}$$

$$p^e(x,t) = cos(2\pi x + 2\pi y + t), \tag{14}$$

$$\rho_e(x,t) = 1 + c(sin^2(2\pi x + 2\pi y + t)), \tag{15}$$

$$\mu_e(x,t) = 1 + c(cos^2(2\pi x + 2\pi y + t)). \tag{16}$$

The Table 1 presents the results of the standard error L_2, given by:

$$L_2 = \sqrt{\frac{\sum_{i=1}^{n_x}\sum_{j=1}^{n_y}(\psi_{ij}^e - \psi_{ij})^2}{n_x n_y}}, \tag{17}$$

Table 1 Standard error L_2 for u, v and p with ρ and μ variables at $t = 5[s]$

Meshes	Variables	Standard error L_2
32×32	u	5.3265×10^{-15}
	v	5.3222×10^{-15}
	p	3.9150×10^{-15}
64×64	u	1.5811×10^{-15}
	v	1.5637×10^{-15}
	p	2.7136×10^{-15}
128×128	u	1.1230×10^{-15}
	v	1.1084×10^{-15}
	p	2.9079×10^{-15}

obtained for uniform meshes and periodic boundary conditions to the variables $\psi = u$, v and p.

The Table 1 shows that the error given by the L_2 norm, reaches round-off error machine when using double precision. This fact that demonstrates the high accuracy of FPSM.

5.2 Rising Bubbles

To validate the methodology proposed by numerical simulation of the rise of a single bubble in a two-dimensional domain. We assume a fluid at rest. The numbers of E*ö*tv*ö*s and Morton [2], are provided. From these numbers we obtain the density and dynamic viscosity which we use in order to compare the Reynolds number and geometrical shapes in the steady state. The geometric shape of the bubble is compared with experimental data diagram of [2].

Table 2 presents a comparison between the Reynolds number in steady state of the present work with the experiment of the [2] and numerical simulations of the [7]. The comparison shows a good approximation of the present work with the experimental result, demonstrating that the methodology proposed in the present work is able to simulate the two-phase flows problems.

The Fig. 3 shows the evolution of the vorticity during the simulation of the *Wobbling* regime, performed in the present work, using the proposed methodology.

Table 2 Reynolds number observed in the experiments of Clift et al. [2], and numerical simulations from Villar [7] and obtained in the present work

Clift et al. [2]	Villar [7]	Present work
$Eo = 0{,}50$ $M = 5{,}00 \times 10^{-3}$ $Re = 0{,}36$	$Re = 0{,}29$	$Re = 0{,}39$
$Eo = 5{,}0$ $M = 5{,}0 \times 10^{-7}$ $Re = 125{,}00$	$Re = 69{,}14$	$Re = 100{,}34$
$Eo = 5{,}09$ $M = 5{,}0 \times 10^{-10}$ $Re = 900,00$	$Re = 475{,}65$	$Re = 599{,}00$
$Eo = 50{,}00$ $M = 5{,}0 \times 10^{-4}$ $Re = 85{,}00$	$Re = 53{,}07$	$Re = 77{,}53$
$Eo = 410{,}00$ $M = 5{,}0 \times 10^{-6}$ $Re = 1200{,}00$	$Re = 641{,}15$	$Re = 1055{,}57$
$Eo = 5{,}09$ $M = 5{,}10 \times 10^{-10}$ $Re = 900{,}00$	$Re = 475{,}65$	$Re = 599{,}99$

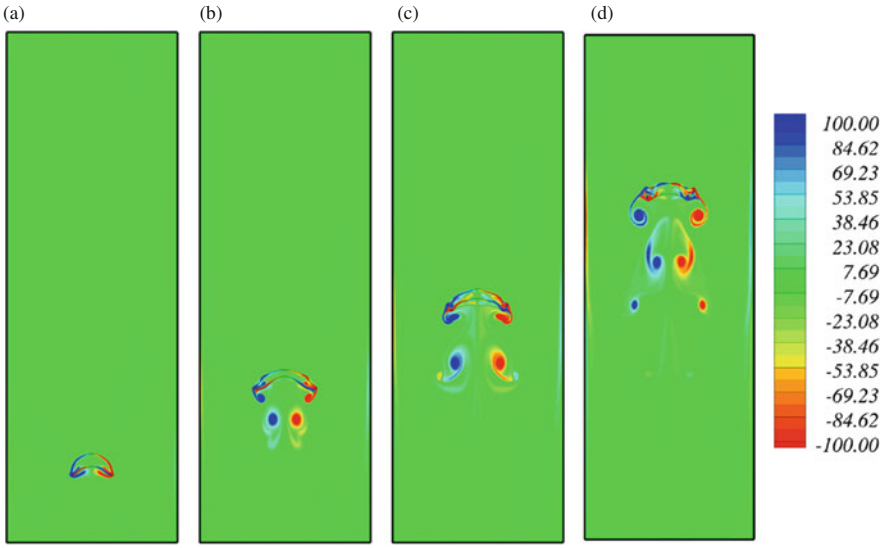


Fig. 3 Vorticity: (**a**) $t = 0$; (**b**) $t = 4,02$; (**c**) $t = 7,23$; (**d**) $t = 12,88$

These results is the last case shown in Table 2. The shape and trajectory of bubble evolutes along the time. The bubble undergoes changes in its shape and vortices are released in agreement with these phases of transient displacement. We notice in the figure 3 that physical details are shown.

6 Conclusions

As conclusion of the present work, a new methodology for two-phase flows was proposed and implemented. The results is in good agreement with experimental results and the proposed methodology is considered validated.

Acknowledgements The authors would like to thank to PETROBRAS, CAPES, FAPEMIG, FAPEG, CAPES/PROEX, CNPq, UFU and UFG for the support for the present work development.

References

1. C. Canuto, M.Y. Hussaini, A. Quarteroni, T.A. Zang Jr., *Spectral Methods: Evolution to Complex Geometries and Applications to Fluid Dynamics* (Springer, New York, 2007), 596 p.
2. R. Clift, J.R. Grace, M.E. Weber, *Bubbles, Drops, and Particles* (Academic Press, New York, 1978), 380 p.

3. H.G. da Silva, M.M. Villar, *Verificação e validação de códigos computacionais, Coleção Cadernos de Turbulência* (ABCM - Associação Brasileira de Engenharia e Ciências Mecânicas, Ilha Solteira, 2010), pp. 51–94
4. R.L. Datta, D.H. Napier, D.M. Newitt, The properties and behaviour of gas bubbles formed at circular orifices. Trans. Inst. Chem. Eng. **28**, 14–26 (1950)
5. F.P. Mariano, L.Q. Moreira, N.A. Silveira, C.F.N.B. da Silva, J.C.F. Pereira, A new incompressible Navier-Stokes solver combining Fourier pseudo-spectral and immersed boundary method. Comput. Model. Eng. Sci. **59**, 181–216 (2010). http://www.techscience.com/doi/10.3970/cmes.2010.059.181.html
6. C.S. Peskin, Numerical analysis of blood flow in the heart. J. Comput. Phys. **25**, 220–252 (1977). http://www.sciencedirect.com/science/article/pii/0021999177901000
7. M.M. Villar, *Análise Numérica Detalhada de Escoamentos Multifásicos Bidimensionais* (Universidade Federal de Uberlândia, Uberlândia, 2007), 277 p.
8. M.F.S. Villela, *Modelagem matemática de escoamentos bifásicos usando o método espectral de Fourier* (Universidade Federal de Uberlândia, Uberlândia, 2011), 83 p.

Multiwavelets and Jumps in DG Approximations

Mathea J. Vuik and Jennifer K. Ryan

Abstract In general, solutions of nonlinear hyperbolic PDEs contain shocks or develop discontinuities. One option for improving the numerical treatment of the spurious oscillations that occur near these artifacts is through the application of a limiter. The cells where such treatment is necessary are referred to as troubled cells. In this article, we discuss the multiwavelet troubled-cell indicator that was introduced by Vuik and Ryan (J Comput Phys 270:138–160, 2014). We focus on the relation between the highest-level multiwavelet coefficients and jumps in (derivatives of) the DG approximation. Based on this information, we slightly modify the original multiwavelet troubled-cell indicator. Furthermore, we show one-dimensional test cases using the modified multiwavelet troubled-cell indicator.

1 Introduction

In general, solutions of nonlinear hyperbolic PDEs contain shocks or develop discontinuities. One option for improving the numerical treatment of the spurious oscillations that occur near these artifacts is through the application of a limiter. The cells where such treatment is necessary are referred to as troubled cells.

In [11], a multiwavelet troubled-cell indicator was introduced, which is used to detect discontinuities in (the derivatives of) the DG approximation. This indicator used the global DG approximation to detect troubled cells. However, because discontinuities are local phenomena, it is useful to find the relation between jumps in (derivatives of) the DG approximation and multiwavelet coefficients.

M.J. Vuik (✉)
Delft Institute of Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628CD Delft, The Netherlands
e-mail: M.J.Vuik@tudelft.nl

J.K. Ryan
School of Mathematics, University of East Anglia, Norwich NR4 7TJ, UK
e-mail: Jennifer.Ryan@uea.ac.uk

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_47

In this paper, we investigate the relation between the multiwavelet expansion and the DG formulation, [2, 3, 11]. We show that the multiwavelet coefficients are related to the jumps in (derivatives of) the DG approximation. Furthermore, we use this information to slightly modify the multiwavelet troubled-cell indicator [11]. We demonstrate the robust performance of this indicator on one-dimensional test problems, using the moment limiter in the identified troubled cells [9].

The outline of this paper is as follows: in Sect. 2, the relation between DG approximations and multiwavelet coefficients and the definition of the modified multiwavelet troubled-cell indicator are given. The effectivity of this method is presented in Sect. 3. We conclude with a discussion and future work in Sect. 4.

2 DG Approximations and Multiwavelet Coefficients

In this section, we present the relation between a DG approximation on 2^n elements and the coefficients of the multiwavelet expansion. Here, we only investigate the one-dimensional domain $[-1, 1]$. This derivation can be easily extended to general domains in one and two dimensions [11].

For the sake of brevity, we neglect discussion of the DG scheme [4–7].

2.1 Multiwavelet Decomposition

In [11], it was shown that any global one-dimensional DG approximation of degree k can be written as

$$u_h(x) = 2^{-\frac{n}{2}} \sum_{j=0}^{2^n-1} \sum_{\ell=0}^{k} u_j^{(\ell)} \phi_{\ell j}^n(x),$$

where $\phi_{\ell j}^n$ are the scaling functions related to the scaled Legendre polynomials. The corresponding multiwavelet decomposition is

$$u_h(x) = \sum_{\ell=0}^{k} s_{\ell 0}^0 \phi_\ell(x) + \sum_{m=0}^{n-1} \sum_{j=0}^{2^m-1} \sum_{\ell=0}^{k} d_{\ell j}^m \psi_{\ell j}^m(x),$$

where $s_{\ell 0}^0$ are the scaling-function coefficients belonging to u_h, and $d_{\ell j}^m$ are the corresponding multiwavelet coefficients, [3, 11]. The multiwavelets ψ_ℓ have been developed by Alpert [1], and are also explained in [8].

It is useful to note that this expansion has n levels, where level 0 gives only coarse details and level $n-1$ gives the finest details. Level $n-1$ is the most important level for the multiwavelet decomposition, since the multiwavelet contribution at this level

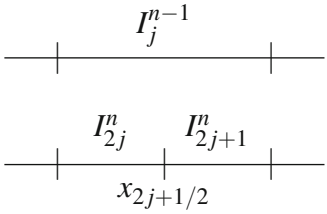


Fig. 1 Multiwavelet $\psi_{\ell j}^{n-1}$ is a piecewise polynomial on I_{2j}^n and I_{2j+1}^n

is used in the multiwavelet troubled-cell indicator, see Sect. 2.4 and [11]. Using dilation and translation, the multiwavelets at level $n-1$ are defined as

$$\psi_{\ell j}^{n-1}(x) = 2^{(n-1)/2}\psi_\ell(2^{n-1}(x+1) - 2j - 1), \quad x \in I_j^{n-1}, \tag{1}$$

where

$$I_j^{n-1} = (-1 + 2^{-n+2}j, -1 + 2^{-n+2}(j+1)]. \tag{2}$$

By construction, multiwavelets ψ_ℓ are piecewise polynomials on $[-1, 0]$ and $[0, 1]$. Extending this relation to level $n-1$, $\psi_{\ell j}^{n-1}$ is a piecewise polynomial on I_{2j}^n and I_{2j+1}^n, as visualized in Fig. 1. Note that these elements are in the DG mesh.

Note that the DG approximation in I_j^{n-1} is discontinuous at the boundary

$$x_{2j+1/2} = -1 + 2^{-n+2}(j + 1/2) := y_j. \tag{3}$$

2.2 *Multiwavelets and Vanishing Moments*

Multiwavelets have a vanishing-moment property. Using a DG approximation space of degree k, each multiwavelet ψ_ℓ, $\ell \in \{0, \ldots, k\}$, is a piecewise polynomial of degree k, and its first $\ell + k + 1$ moments vanish:

$$\int_{-1}^{1} x^m \psi_\ell(x)\, dx = 0, \quad m = 0, \ldots, \ell + k, \tag{4a}$$

[1, 8]. This means that

$$\int_{-1}^{0} x^m \psi_\ell(x)\, dx = -\int_{0}^{1} x^m \psi_\ell(x)\, dx, \tag{4b}$$

and that the value in Eq. (4b) is only nonzero if $x^m \psi_\ell(x)$ is odd.

In the following lemma, the vanishing-moment property is extended to the decomposition level $n-1$.

Lemma 1 *Let u_h be the DG approximation of degree k on $[-1,1]$, using 2^n elements, and let $\{\psi_\ell\}_{\ell=0}^{k}$ be the corresponding multiwavelet basis. Then, the following relation holds:*

$$\int_{I_j^{n-1}} (x-y_j)^m \psi_{\ell j}^{n-1}(x)\,dx = 0,\ m = 0,\ldots,k+\ell,\ j = 0,\ldots,2^{n-1}-1, \tag{5}$$

where $\psi_{\ell j}^{n-1}$, I_j^{n-1} and y_j are defined as in Eqs. (1)–(3).

Proof Using Eqs. (1)–(3), the left-hand side of Eq. (5) equals

$$2^{\frac{n-1}{2}} \int_{-1+2^{-n+2}j}^{-1+2^{-n+2}(j+1)} \left(x+1-2^{-n+2}\left(j+\frac{1}{2}\right)\right)^m \psi_\ell(2^{n-1}(x+1)-2j-1)\,dx. \tag{6}$$

Applying the transformation $z = 2^{n-1}(x+1)-2j-1$, Eqs. (5) and (6) transform into

$$\int_{I_j^{n-1}} (x-y_j)^m \psi_{\ell j}^{n-1}(x)\,dx = 2^{(m+1/2)(-n+1)} \cdot \int_{-1}^{1} z^m \psi_\ell(z)\,dz = 0, \tag{7}$$

using the relation in Eq. (4a). □

A direct consequence of Lemma 1 is the following result:

Corollary 1

$$\int_{-1+2^{-n+2}j}^{y_j} (x-y_j)^m \psi_{\ell j}^{n-1}(x)\,dx = -\int_{y_j}^{-1+2^{-n+2}(j+1)} (x-y_j)^m \psi_{\ell j}^{n-1}(x)\,dx. \tag{8}$$

This property will be used in Sect. 2.3 in order to derive the relation between multiwavelets and jumps in (the derivatives of) the DG approximation.

2.3 Jumps in DG Approximations and Multiwavelet Coefficients

In this section, it will be shown that the multiwavelet coefficients on level $n-1$ are related to jumps in (derivatives of) the DG approximation. In Walnut [12], the ideas were explained for the Haar wavelet system, and general functions f.

The multiwavelet coefficient $d_{\ell j}^{n-1}$ is computed by a projection of the DG approximation onto the space of multiwavelets [3]:

$$d_{\ell j}^{n-1} = \int_{I_j^{n-1}} u_h(x)\psi_{\ell j}^{n-1}(x)\,dx. \tag{9}$$

Below we relate the value of this coefficient to the DG approximation.

Theorem 1 *Let u_h be a DG approximation of degree k on $[-1, 1]$, using 2^n elements. Then the multiwavelet coefficients on level $n-1$ of the decomposition are equal to*

$$d_{\ell j}^{n-1} = 2^{-\frac{n-1}{2}} \sum_{m=0}^{k} c_{m\ell}^{n} \cdot \left(u_h^{(m)}(y_j^+) - u_h^{(m)}(y_j^-) \right), \tag{10a}$$

with

$$c_{m\ell}^{n} = \frac{2^{(-n+1)m}}{m!} \cdot \int_0^1 x^m \psi_\ell(x)\,dx, \tag{10b}$$

$\ell = 0, \ldots, k$, $j = 0, \ldots, 2^{n-1} - 1$.

Proof In general, the DG approximation, u_h, is a piecewise polynomial of degree k on element I_j^{n-1}, with a discontinuity at y_j (see Fig. 1). This means that we can express u_h as a Taylor polynomial about y_j^- in element I_{2j}^n and about y_j^+ in I_{2j+1}^n:

$$u_h(x) = u_h(y_j^-) + u_h'(y_j^-)(x - y_j) + \ldots + \frac{1}{k!}u_h^{(k)}(y_j^-)(x - y_j)^k, \;\; x \in I_{2j}^n, \tag{11a}$$

$$u_h(x) = u_h(y_j^+) + u_h'(y_j^+)(x - y_j) + \ldots + \frac{1}{k!}u_h^{(k)}(y_j^+)(x - y_j)^k, \;\; x \in I_{2j+1}^n. \tag{11b}$$

Using this relation in Eq. (9), multiwavelet coefficient $d_{\ell j}^{n-1}$ can be expressed as

$$\begin{aligned} d_{\ell j}^{n-1} = &\sum_{m=0}^{k} \frac{1}{m!} u_h^{(m)}(y_j^-) \int_{-1+2^{-n+2}j}^{y_j} (x - y_j)^m \psi_{\ell j}^{n-1}(x)\,dx \\ &+ \sum_{m=0}^{k} \frac{1}{m!} u_h^{(m)}(y_j^+) \int_{y_j}^{-1+2^{-n+2}(j+1)} (x - y_j)^m \psi_{\ell j}^{n-1}(x)\,dx. \end{aligned} \tag{12}$$

Using Corollary 1, we arrive at

$$d_{\ell j}^{n-1} = \sum_{m=0}^{k} \frac{1}{m!} \left(u_h^{(m)}(y_j^+) - u_h^{(m)}(y_j^-) \right) \int_{y_j}^{-1+2^{-n+2}(j+1)} (x - y_j)^m \psi_{\ell j}^{n-1}(x)\, dx. \tag{13}$$

Note that the integral in Eq. (13) is equal to the integral in the left-hand side of Eq. (7), except for the lower integration limit. Using the same transformation as in Eq. (7), the theorem is proved. □

This theorem gives a direct relation between multiwavelet coefficients $d_{\ell j}^{n-1}$ on level $n-1$ and jumps in (derivatives of) the DG approximation over the element boundary $x_{2j+1/2}$. Since the DG method adopts a discontinuous nature at element boundaries, the wavelet coefficients are in general never exactly equal to zero. However, when the underlying function is sufficiently smooth, the inter-element jumps in the approximation and its derivatives will be noticeably smaller than when a discontinuity (in one of the derivatives) is present. This information can be used to detect troubled cells. In theory, it is possible that large jumps are cancelled in the summation of Eq. (10a). In practice, however, this will not occur at more than one successive time step and therefore, the impact will be negligible.

By the vanishing-moment property, $c_{m\ell}^n$ is only nonzero when $x^m \psi_\ell(x)$ is an odd function [Eq. (4)]. Because ψ_k is an odd function [1], coefficient c_{0k}^n is always nonzero [Eqs. (4) and (10)]. Therefore, d_{kj}^{n-1} contains information about the jump $u_h(y_j^+) - u_h(y_j^-)$, and this coefficient will be used for indication.

2.4 Modified Multiwavelet Troubled-Cell Indicator

In this section we discuss a slight modification to the multiwavelet troubled-cell indicator introduced in [11].

Note that d_{kj}^{n-1} contains information about boundary $x_{2j+1/2}, j = 0, \ldots, 2^{n-1} - 1$. In order to also investigate boundaries $x_{2j-1/2}$, we virtually renumber the internal elements, $I_1, \ldots, I_{2^n-2}$, to $I_0, \ldots, I_{2^n-3}$, and apply the decomposition procedure on these elements as well.

Similar to [11], we detect an element as troubled when

$$|d_{kj}^{n-1}| > C \cdot \max\{|d_{kj}^{n-1}|, j = 0, \ldots, 2^n - 1\},\ C \in [0, 1]. \tag{14}$$

The element where $|d_{kj}^{n-1}|$ is maximal, is assumed to be the element where the strongest shock occurs. If $C = 1$, then no element will be detected. In this way, the value of C is a useful tool to prescribe the strictness of the limiter. The lower the

value of C, the more cells are limited. For each problem, the optimal value of C will be obtained by using several tests. This troubled-cell indicator is combined with the moment limiter [9].

Notice that this approach is much faster than the original approach in [11]. Here, we do not need to compute multiwavelet averages over each element. Furthermore, the new procedure is more accurate, as all boundaries are investigated by the indicator.

3 Numerical Results

In [11], many numerical test cases are used to show the effectivity of the multiwavelet troubled-cell indicator, both in one and two dimensions. In this section, we present the results using the modified multiwavelet troubled-cell indicator for Sod's shock tube [10] and the blast-wave problem [13]. The moment limiter [9] is applied in the troubled cells. In Figs. 2 and 3, time-history plots of detected troubled cells are shown, together with the approximations at the final time.

For Sod's shock tube, it is clearly visible that both the shock and the contact discontinuity are detected. Note that also one end point of the rarefaction wave (where the derivative of the approximation is discontinuous) is detected for $k = 1$. This means that our indicator is very accurate if the value of C is chosen properly.

The blast-wave problem is extremely nonlinear. However, it should be noted that only a few elements should be limited in order to get nonoscillatory results. Our

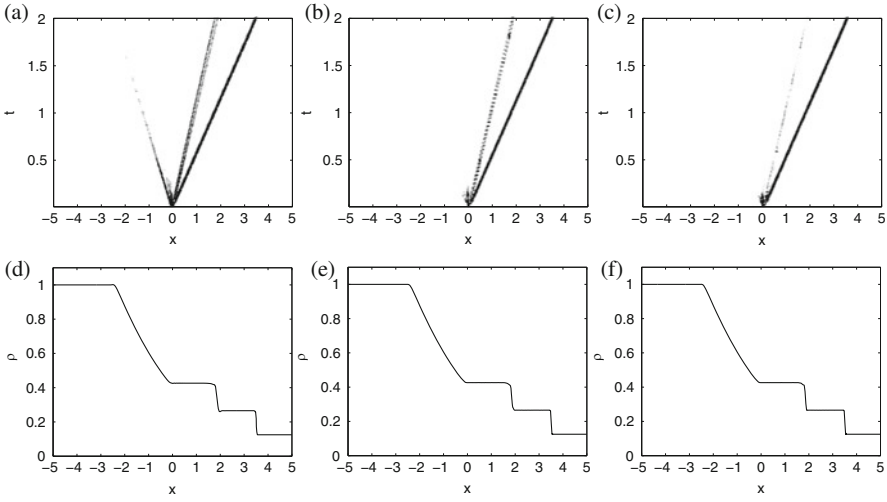


Fig. 2 Sod's shock tube, troubled cells and approximation at $T = 2$, using $C = 0.1$, 256 elements. (**a**) $k = 1$. (**b**) $k = 2$. (**c**) $k = 3$. (**d**) $k = 1$. (**e**) $k = 2$. (**f**) $k = 3$

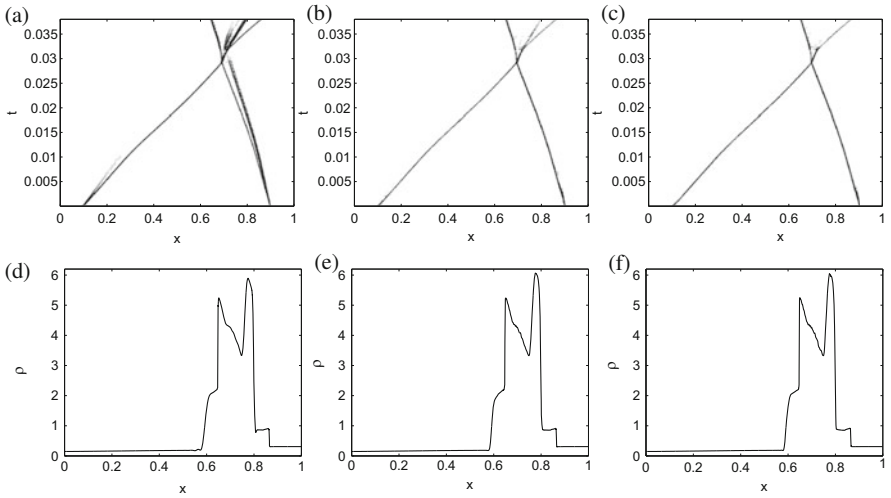


Fig. 3 Blast-wave problem, troubled cells and approximation at $T = 0.038$, using $C = 0.05$, 512 elements. (**a**) $k = 1$. (**b**) $k = 2$. (**c**) $k = 3$. (**d**) $k = 1$. (**e**) $k = 2$. (**f**) $k = 3$

parameter C is a useful tool to prevent limiting too many elements. The multiwavelet indicator detects regions that are visible in the exact shock solution, which was given by Woodward et al. [13].

The proper choice of C is ongoing work.

4 Conclusions

In this paper we have explained the relation between jumps in (derivatives of) the DG approximation and the multiwavelet expansion in order to identify troubled cells. Furthermore, a modified multiwavelet troubled-cell indicator has been constructed, which is less computationally expensive and more accurate than the original detector in [11]. In the numerical results, we demonstrated that this technique performs well, even in the vicinity of a strong shock with weaker local shocks.

Future work will be to see if we can improve upon the performance in detecting local structures, to decide in advance which value of the parameter we should use, and to extend this to unstructured meshes.

References

1. B.K. Alpert, A class of bases in L^2 for the sparse representation of integral operators. SIAM J. Math. Anal. **24**(1), 246–262 (1993)
2. B.K. Alpert, G. Beylkin, D. Gines, L. Vozovoi, Adaptive solution of partial differential equations in multiwavelet bases. J. Comput. Phys. **182**, 149–190 (2002)
3. R.K. Archibald, G.I. Fann, W.A. Shelton, Adaptive discontinuous Galerkin methods in multiwavelets bases. Appl. Numer. Math. **61**(7), 879–890 (2011)
4. B. Cockburn, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws II: general framework. Math. Comput. **52**(186), 411–435 (1989)
5. B. Cockburn, C.-W. Shu, The Runge-Kutta discontinuous Galerkin method for conservation laws V: multidimensional systems. J. Comput. Phys. **141**(2), 199–224 (1998)
6. B. Cockburn, S.-Y. Lin, C.-W. Shu, TVB Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws III: one-dimensional systems. J. Comput. Phys. **84**, 90–113 (1989)
7. B. Cockburn, S. Hou, C.-W. Shu, The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws IV: the multidimensional case. Math. Comput. **54**(190), 545–581 (1990)
8. N. Hovhannisyan, S. Müller, R. Schäfer, Adaptive multiresolution discontinuous Galerkin schemes for conservation laws. Report 311, Institut für Geometrie und Praktische Mathematik, Aachen (2010). http://www.igpm.rwth-aachen.de/forschung/preprints2010
9. L. Krivodonova, Limiters for high-order discontinuous Galerkin methods. J. Comput. Phys. **226**, 879–896 (2007)
10. G.A. Sod, A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws. J. Comput. Phys. **27**, 1–31 (1978)
11. M.J. Vuik, J.K. Ryan, Multiwavelet troubled-cell indicator for discontinuity detection of discontinuous Galerkin schemes. J. Comput. Phys. **270**, 138–160 (2014)
12. D.F. Walnut, *An Introduction to Wavelet Analysis*. Applied and Numerical Harmonic Analysis, 1st edn. (Birkhäuser, Boston, 2002)
13. P. Woodward, P. Colella, The numerical simulation of two-dimensional fluid flow with strong shocks. J. Comput. Phys. **54**, 115–173 (1984)

Efficient and High-Order Explicit Local Time Stepping on Moving DG Spectral Element Meshes

Andrew R. Winters and David A. Kopriva

Abstract We outline and extend results for an explicit local time stepping (LTS) strategy designed to operate with the discontinuous Galerkin spectral element method (DGSEM). The LTS procedure is derived from Adams-Bashforth multirate time integration methods. The new results of the LTS method focus on parallelization and reformulation of the LTS integrator to maintain conservation. Discussion is focused on a moving mesh implementation, but the procedures remain applicable to static meshes. In numerical tests, we demonstrate the strong scaling of a parallel, LTS implementation and compare the scaling properties to a parallel, global time stepping (GTS) Runge-Kutta implementation. We also present time-step refinement studies to show that the redesigned, conservative LTS approximations are spectrally accurate in space and have design temporal accuracy.

1 Introduction

In this work we describe and evaluate a high-order local time stepping (LTS) integrator designed for use with the nodal discontinuous Galerkin spectral element methods (DGSEM). In particular, this paper serves as an update to the work in [13]. We will demonstrate the parallelizability of the LTS procedure on moving meshes as well as refactor the LTS strategy to ensure the approximation remains conservative. The LTS method [13] is similar to that presented by Gödel et al. [5] where we use an Adams-Bashforth time integrator as a base for the LTS strategy and compute intermediate coupling terms with polynomial interpolants in time. The method of Gödel et al. constructs a time interpolant of the entire time derivative (RHS) term to compute the coupling between time scales [5]. Our method differs as we construct a time interpolant of the solution along element boundaries and use the reconstructed solution value to compute the time derivative at intermediate times [13]. Both LTS

A.R. Winters (✉) • D.A. Kopriva
Department of Mathematics, The Florida State University, 208 Love Building, 1017 Academic Way, Tallahassee, FL 32306, USA
e-mail: awinters@math.fsu.edu; kopriva@math.fsu.edu

R.M. Kirby et al. (eds.), *Spectral and High Order Methods for Partial Differential Equations ICOSAHOM 2014*, Lecture Notes in Computational Science and Engineering 106, DOI 10.1007/978-3-319-19800-2_48

integrators are non-conservative, but the structure of our time interpolation strategy allows us to derive a conservative version here.

The paper is organized as follows: Sect. 2 provides a brief overview of the moving domain, arbitrary Lagrangian-Eulerian semi-discrete discontinuous Galerkin spectral element approximation (ALE-DGSEM). In Sect. 3 we discuss the LTS procedure of [13], extending the strategy with respect to parallelization and conservation. We give numerical results in Sect. 4 that show strong scaling of the parallel LTS procedure and a time-step refinement study to show the design accuracy of the new conservative LTS scheme. Section 5 presents concluding remarks.

2 Semi-discrete DG Approximation of an ALE Conservation Law

We study the approximation of problems modeled by a system of conservation laws

$$\mathbf{q}_t + \nabla \cdot \mathscr{F} = 0, \tag{1}$$

on the moving domain Ω_t.

The development of an ALE-DGSEM approximation has the following steps: The moving physical domain is decomposed into multiple elements with moving boundaries. Each moving element is mapped onto a static reference element $E = [-1, 1]^d$, d is the number of spatial dimensions, where a strong form of the equations still applies [4, 10, 11],

$$\tilde{\mathbf{q}}_t + \nabla_\xi \cdot \mathcal{F} = 0, \tag{2}$$

with

$$\begin{aligned} \tilde{\mathbf{q}} &= \mathcal{J}\mathbf{q}, \\ \mathcal{F} &= \mathcal{J}\mathbf{a}^i \cdot (\mathscr{F} - \mathbf{q}\mathbf{x}_t), \end{aligned} \tag{3}$$

and

$$\begin{aligned} \mathcal{J} a_n^i &= -\hat{x}_i \cdot \nabla_\xi \times (x_l \nabla_\xi x_m), \quad i = 1, 2, 3;\ n = 1, 2, 3;\ (n, m, l) \text{ cyclic}, \\ \mathcal{J} &= \mathbf{a}_1 \cdot (\mathbf{a}_2 \times \mathbf{a}_3). \end{aligned} \tag{4}$$

For complete details on the ALE transformation of the conservation law see Acosta and Kopriva [1]. Notice that in the transformed variables (3) the solution $\tilde{\mathbf{q}}$ incorporates the time-dependent Jacobian $\mathcal{J}$ and the flux $\mathcal{F}$ incorporates the mesh velocity $\mathbf{x}_t$.

The divergence-free form of the contravariant basis vectors (4) is particularly important to prevent spurious oscillations generated by the mesh in the solution on curved sided hexahedral elements [8]. However, for two dimensional problems and straight-sided hexahedral meshes the less computationally intensive, cross product formulation [9]

$$\mathcal{J}\mathbf{a}^i = \mathcal{J}\nabla\xi^i = \mathbf{a}_j \times \mathbf{a}_k \quad (i,j,k) \text{ cyclic}, \tag{5}$$

is sufficient to prevent the generation of spurious waves [8].

A nodal DG method approximation imposes that the solution and fluxes are approximated by polynomials of degree less than or equal to N in each element, i.e., $\tilde{\mathbf{q}} \approx \tilde{\mathbf{Q}} \in \mathbb{P}^N$ and $\mathcal{F} \approx \tilde{\mathbf{F}} \in \mathbb{P}^N$. It starts with the weak form of the Eq. (2)

$$\int_E (\tilde{\mathbf{Q}}_t + \nabla \cdot \tilde{\mathbf{F}})\varphi \, d\boldsymbol{\xi} = 0. \tag{6}$$

There is no continuity of $\varphi \in \mathbb{P}^N$ assumed between elements. We then integrate by parts and replaces boundary fluxes with the solution of a Riemann solver to obtain the DG approximation on the reference element

$$\int_E \tilde{\mathbf{Q}}_t \varphi \, d\boldsymbol{\xi} + \int_{\partial E} \tilde{\mathbf{F}}^* \cdot \hat{n}_\xi \varphi \, dS - \int_E \tilde{\mathbf{F}} \cdot \nabla\varphi \, d\boldsymbol{\xi} = 0. \tag{7}$$

To complete the spatial discretization, we choose the test function and location of the nodes in the approximation. For the test function, φ, we select the Lagrange basis that interpolates the Legendre-Gauss nodes. We approximate the integrals in (7) with Legendre-Gauss quadrature and arrive at the semi-discrete approximation of (2)

$$\frac{d\tilde{\mathbf{Q}}_{ijk}}{dt} + \sum_{n=1}^{3} D_{\xi^n} \tilde{\mathbf{F}}^n_{ijk} = 0, \tag{8}$$

where

$$D_{\xi^1}\tilde{\mathbf{F}}^1_{ijk} = \left[\tilde{\mathbf{F}}^*(1,\eta_j,\zeta_k)\frac{\ell_i(1)}{\omega_i^{(\xi)}} - \tilde{\mathbf{F}}^*(-1,\eta_j,\zeta_k)\frac{\ell_i(-1)}{\omega_i^{(\xi)}} \right] + \sum_{m=0}^{N} \tilde{\mathbf{F}}_{mjk}\hat{D}^{(\xi)}_{im}, \tag{9}$$

etc., and $\hat{D}$ is the transpose of the derivative matrix scaled by the quadrature weights [9].

The primary work in the approximation (9) is to compute the fluxes $\tilde{\mathbf{F}}^n_{ijk}$ from the solution and to evaluate the Riemann solver at element faces, e.g. $\tilde{\mathbf{F}}^*(-1,\eta_j,\zeta_k)$. Apart from the solution of the Riemann problem, the components of (9) are computed locally on each element.

3 Local Time Stepping Strategy

To integrate (8) in time we select the explicit local time stepping (LTS) method using an Adams-Bashforth linear multistep method described in [13], where the motivation and analysis of the LTS method is explored at length.

The crux of the LTS procedure lies in the locality of a DG approximation. The only coupling is through the Riemann problem which must be solved at the boundary of each element. The LTS strategy can integrate elements at different time scales. Thus, it may occur that the solution in an element and a neighbor reside at different times. To reconstruct the solution in the neighbor to be at the same time as the solution in the current element the LTS procedure uses a polynomial interpolant in time. Now it is possible to solve the Riemann problem in the current element and integrate forward one local time step.

Though this LTS strategy is computationally efficient [13], it has two important aspects which deserve consideration: parallelization and conservation. As the LTS integrator exploits the locality of the DGSEM approximation, the computation remains highly localized. So, the LTS integrator does not change the fact that the DGSEM is an embarrassingly parallel procedure [3]. We describe the parallelization and a particular load balancing of the LTS integrator in Sect. 3.1. The use of interpolants in time to recover the solution on neighboring elements renders the LTS method non-conservative. However, we redesign the method of [13] to maintain conservation in Sect. 3.2.

3.1 Parallelization

Due to the weak coupling the DGSEM approximation is inherently parallel. In fact, for a global time stepping integrator, the naïve approach of load balancing by partitioning a mesh into equally sized sub-meshes leads to a highly efficient implementation [2, 3].

The parallelization of the LTS-DGSEM is largely the same as a global time stepping integrator. First the computation is partitioned into subproblems. Next, one breaks the computation into components along partition edges and components in the partition interior. While sending necessary neighbor data to other processes, local calculations can be performed to hide the latency. The only difference arises in whether one sends the current solution on a given element (if the neighbor's solution is at the same point in time) or if one sends the solution that is reconstructed using a time interpolant.

The major change for the parallel LTS procedure is in the load balancing. At the beginning of the LTS procedure elements of similar size are placed into groups. Because small elements are integrated in time more often than large elements, we do not want to pack many small elements into a single partition. A simple strategy to balance the load of a parallel LTS computation is to weight elements in the

partitioning process by size [6]. We show strong scaling of a unstructured mesh, parallel LTS implementation using this load balancing strategy on a small cluster in Sect. 4. However, for parallel implementations on structured meshes or those with sophisticated data structures to assist with latency hiding, e.g. [2], weighting elements according to size may result in a suboptimal strategy to balance the load. For large clusters we may want to limit the number of and/or volume of communications between partitions [2, 6, 7].

3.2 Conservation

Next we address the issue of conservation loss in the LTS procedure. To simplify the discussion we make the assumption that the approximation has only two time scales, but the new, conservative LTS procedure easily generalizes to an arbitrary number.

To redesign the LTS strategy and maintain conservation of the DGSEM we abandon the use of polynomial interpolants to recover an unknown solution value at an intermediate time. Instead, we use an idea of Tirupathi et al. [12] where we redefine the Adams-Bashforth method to allow the solution to evolve in the small time scale at any time interfaces. Then, the boundary flux terms at the half time step required by the Riemann solver in the small time scale neighbor are available.

The approximation in the small time scale remains conservative, but the large time scale does not. The update to the solution in the large time scale is missing the numerical flux contribution at the half time step, $\tilde{\mathbf{F}}^*_{n+1/2}$. Because the boundary and interior terms of the DGSEM approximation are decoupled, we adjust the boundary terms to integrate at the small time scale at the temporal interface. For instance, integrating the boundary flux in the x-direction on the right edge of an element that borders the small time scale we have

$$\int_{t_n}^{t_{n+1}} \mathcal{I}_{ijk}\, ds = \int_{t_{n+1/2}}^{t_{n+1}} \mathcal{I}_{ijk}\, ds + \int_{t_n}^{t_{n+1/2}} \mathcal{I}_{ijk}\, ds, \tag{10}$$

where

$$\mathcal{I}_{ijk} = \tilde{\mathbf{F}}^*(1, \eta_j, \zeta_k) \frac{\ell_i(1)}{\omega_i^{(\xi)}}. \tag{11}$$

By separating the time integral of the surface contributions (10) to include the intermediate time $t_{n+1/2}$ we can ensure that the LTS method remains conservative because the boundary flux terms at half time steps are incorporated into the large time scale approximation. So, at each time, the interface fluxes are defined uniquely for adjacent elements.

4 Numerical Results

We provide three numerical examples that combine the ALE-DGSEM spatial discretization with the LTS time integrator. The first solves a two-dimensional moving mesh problem to demonstrate the strong scaling of the parallelized LTS method on up to 200 processors. Second, we compare the scaling properties of a parallel LTS integrator and a parallel global time stepping (GTS) Runge-Kutta method. The final test problem shows the spectral convergence and design time accuracy of the newly described, conservative LTS method from Sect. 3.2 on a static, one dimensional mesh with local refinement.

We solve the classical wave equation written as a conservation law, e.g. in two dimensions

$$\begin{bmatrix} p \\ u \\ v \end{bmatrix}_t + \begin{bmatrix} -x_t & \rho c^2 & 0 \\ 1/\rho & -x_t & 0 \\ 0 & 0 & -x_t \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_x + \begin{bmatrix} -y_t & 0 & \rho c^2 \\ 0 & -y_t & 0 \\ 1/\rho & 0 & -y_t \end{bmatrix} \begin{bmatrix} p \\ u \\ v \end{bmatrix}_y = 0, \tag{12}$$

where $\mathbf{x}_t = (x_t, y_t)$ is the mesh velocity incorporated from the ALE transformations. We choose initial and boundary conditions so that the solution is a Gaussian plane wave

$$\begin{bmatrix} p \\ u \\ v \end{bmatrix} = \begin{bmatrix} 1 \\ \frac{k_x}{\rho c} \\ \frac{k_y}{\rho c} \end{bmatrix} e^{-\frac{(k_x(x-x_0)^2 + k_y(y-y_0)^2 - ct)^2}{d^2}}, \tag{13}$$

with the wavevector $\mathbf{k}$ normalized to satisfy $k_x^2 + k_y^2 = 1$. We take $c = 1$, $\rho = 1$, and vary x_0 and y_0 to adjust the initial position. For the parallel speedup examples in Sect. 4.1, we take $d = \omega/2\sqrt{\ln(2)}$, $\omega = 0.2$, and choose $x_0 = 0.0$, $y_0 = -0.5$ in the wave form (13).

4.1 Moving Mesh, Parallel Local Time Stepping

We show that the ALE-DGSEM with LTS remains embarrassingly parallel. To balance the computational load we assign each element in the mesh a weight according to size in METIS [6]. For the LTS parallel speedup test we consider a moving domain of 5000 elements with polynomial order $N = 7$ in each direction on each element. We integrate the solution to the wave equation (12) up to $T = 1.0$. We run the calculation on up to 200 processors and show the speedup versus a serial implementation of the LTS algorithm in Fig. 1. For a large number of processors the parallel implementation of LTS has 93 % efficiency. For a small number of processors we see that speedup can be super linear due to cache effects [3].

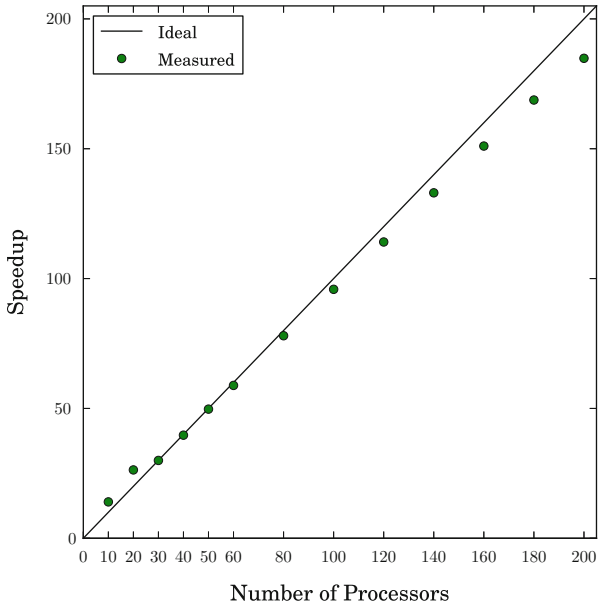


Fig. 1 Speedup of a moving mesh 2D ALE-DGSEM approximation with LTS

From previous work by Kelly and Giraldo [7] it is not surprising that we observe strong scaling for the LTS-DGSEM approximation. The test problem we consider, seventh order polynomial approximation in each spatial direction on each of 5000 elements for each governing equation, means that dividing the work among 200 processors yields 1600 grid points per processor per equation. The volume of computation, in this case, was enough to hide the latency of the communication. Kelly and Giraldo [7] developed a geometric predictor of scalability that uses a ratio of the volume of communication to the surface area of partition edges that defines the computational cost. They determined that, if the ratio of the on-processor work to communication is ≥ 100 then the communication is overwhelmed by the computation and we observe strong scalability. For two-dimensional approximations this ratio is given by

$$\mathcal{R}_P^N = \frac{\mathcal{V}_P^N}{\mathcal{S}_P^N} \approx \left(\frac{K}{P}\right)^{\frac{1}{3}} (N+1)^2, \tag{14}$$

where K is the number of elements, P is the number of processors, and N is the polynomial order of the approximation. We find for the parallel LTS test problem that $\mathcal{R}_{200}^7 \approx 187$ and therefore would expect to see strong scaling of the parallel approximation [7].

Next we compare the LTS and GTS parallel implementations. To balance the load for the GTS computation we divide the workload into equal pieces using METIS.

The load balancing for the LTS implementation is the same as the previous example. For the computation we solve the wave equation on a moving domain with 394 elements with polynomial order $N = 7$ in each direction on each element. The calculation was distributed up to a maximum of 12 processors.

We note that the parallel LTS speedup compounds with the natural speedup gained when one switches from a GTS to the LTS integrator. We see that both the global and local time integrator implementations present strong scaling, but the LTS can offer significant speedup without a large number of processors. For instance, we found that, for 12 processors, the parallel GTS integrator has a speedup of 13.3 and the LTS integrator a parallel speedup of 12.45. Previously, we found that the LTS integrator can achieve a factor of 10.5 speedup between global and local time stepping on a moving mesh [13]. Intuitively, the speedup of the LTS method depends on the level of mesh refinement and the ratio of large to small elements. A detailed theoretical and computational analysis of meshes that present the greatest speedup for the LTS integrator is provided in [13]. So, the parallel LTS integrator has a total speedup of (10.5)(12.45) = 130.7. Thus, the parallel LTS algorithm offers a competitive choice when one wants to solve large problems in parallel without the use of a large cluster. Figure 2 shows the comparison of the total speedup of a parallel, LTS implementation and a parallelized GTS integrator on up to twelve processors.

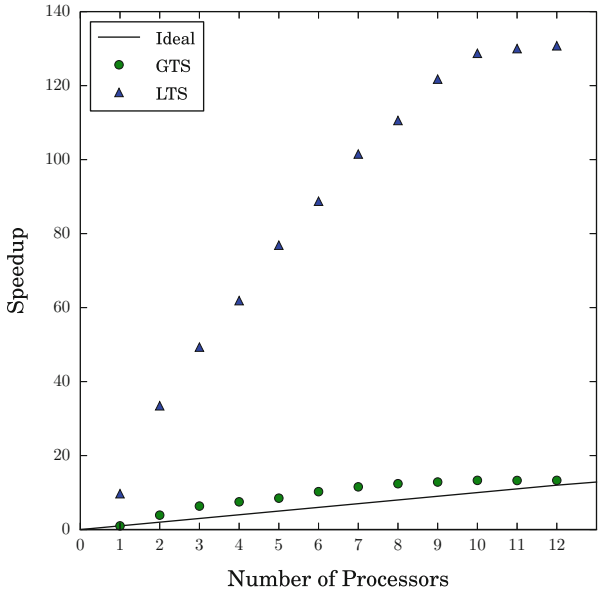


Fig. 2 Comparison between the parallel implementations of a LTS and GTS integrator. The LTS results present the total speedup, which combine the speedup of the parallel implementation and the speedup observed between the LTS and GTS integrators. Both methods are compared to the GTS serial implementation

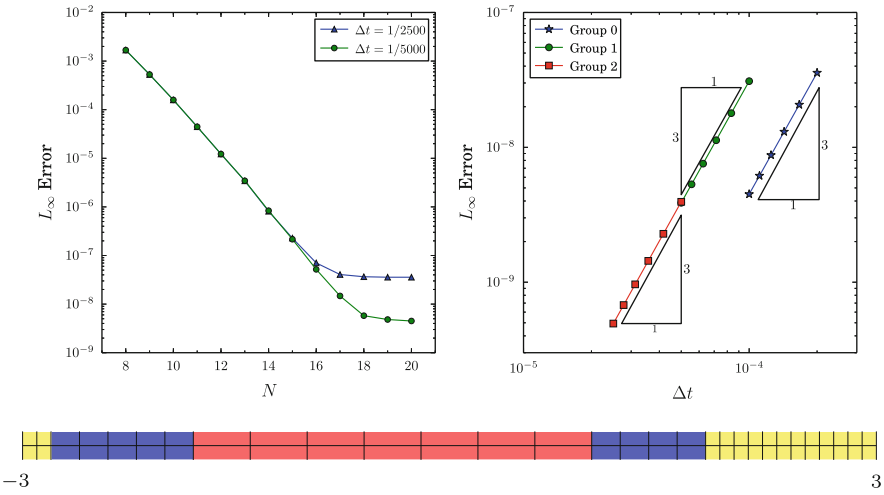


Fig. 3 Spectral convergence (*left*) and design third order time accuracy (*right*) of the conservative LTS integrator. (*Bottom*) The one-dimensional mesh with three element sizes used for convergence testing

4.2 Conservative Local Time Stepping

We demonstrate that the new, conservative LTS-DGSEM integrator retains spectral accuracy in space and design accuracy in time. We consider a one-dimensional problem on the domain $\Omega = [-3, 3]$. We divide the domain into a mesh of $K = 30$ elements with three element sizes, shown in Fig. 3. For the one-dimensional wave equation, we choose the same plane wave parameters and final time T as the parallel test cases, except $x_0 = -0.5$.

The left of Fig. 3 shows exponential convergence in space until $N = 17$, where the error is dominated by time integrator errors. Here Δt is the time step in the largest group of elements. We see that when the value of Δt is halved the error in the approximation is reduced by a factor of 8.

The right plot of Fig. 3 demonstrates design third order temporal accuracy in each group of elements. To produce the plot, we fixed $N = 20$, $T = 1.0$ and let Δt range from $1/2500$ to $1/5000$.

5 Conclusion

In this paper we demonstrated that the LTS integrator derived in [13] remains embarrassingly parallel and, with a slight restructure of the time integral approximation, can be made conservative. For the explicit LTS and GTS parallel implementations and test problems studied in this paper, we have found that the LTS method to be

highly parallelizable and competitive with a GTS method on a small cluster. We redesigned the LTS integrator to maintain conservation and showed this redesign did not affect the method's accuracy. Though promising, the issue of load balancing a parallel LTS method of this type remains an open question on very large scale computations.

References

1. C.A. Acosta Minoli, D.A. Kopriva, Discontinuous Galerkin spectral element approximations on moving meshes. J. Comput. Phys. **230**, 1876–1902 (2010)
2. C. Altmann, A.D. Beck, F. Hindenlang, M. Staudenmaier, G.J. Gassner, C.-D. Munz, An efficient high performance parallelization of a discontinuous Galerkin spectral element method, in *Facing the Multicore-Challenge III* (Springer, Heidelberg, 2013), pp. 37–47
3. A. Baggag, H. Atkins, D. Keyes, Parallel implementation of the discontinuous Galerkin method, in *Parallel Computational Fluid Dynamics: Towards Teraflops, Optimization, and Novel Formulations* (North-Holland, Amsterdam, 2000), pp. 115–122
4. S. Étienne, A. Garon, D. Pelletier, Perspective on the geometric conservation law and finite element methods for ALE simulations of incompressible flow. J. Comput. Phys. **228**, 2313–2333 (2009)
5. N. Gödel, S. Schomann, T. Warburton, M. Clemens, Local timestepping discontinuous Galerkin methods for electromagnetic RF field problems, in *3rd European Conference on Antennas and Propagation, 2009. EuCAP 2009* (IEEE, Berlin, 2009), pp. 2149–2153
6. G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM J. Sci. Comput. **20**, 359–392 (1998)
7. J.F. Kelly, F.X. Giraldo, Continuous and discontinuous Galerkin methods for a scalable three-dimensional nonhydrostatic atmospheric model: limited-area mode. J. Comput. Phys. **231**, 7988–8008 (2012)
8. D.A. Kopriva, Metric identities and the discontinuous spectral element method on curvilinear meshes. J. Sci. Comput. **26**, 301–327 (2006)
9. D.A. Kopriva, *Implementing Spectral Methods for Partial Differential Equations* (Springer, Heidelberg, 2009)
10. I. Lomtev, R.M. Kirby, G.E. Karniadakis, A discontinuous Galerkin ALE method for compressible viscous flows in moving domains. J. Comput. Phys. **155**, 128–159 (1999)
11. D.J. Mavriplis, Z. Yang, Construction of the discrete geometric conservation law for high-order time-accurate simulations on dynamic meshes. J. Comput. Phys. **213**, 557–573 (2006)
12. S. Tirupathi, J.S. Hesthaven, Y. Liang, M. Parmentier, Multilevel and local time-stepping discontinuous Galerkin methods for magma dynamics. Comput. Geosci. **19**(4), 965–978 (2015)
13. A.R. Winters, D.A. Kopriva, High-order local time stepping on moving DG spectral element meshes. J. Sci. Comput. **58**, 176–202 (2014)

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 33.3 % 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:

Timothy J. Barth
NASA Ames Research Center
NAS Division
Moffett Field, CA 94035, USA
barth@nas.nasa.gov

Michael Griebel
Institut für Numerische Simulation
der Universität Bonn
Wegelerstr. 6
53115 Bonn, Germany
griebel@ins.uni-bonn.de

David E. Keyes
Mathematical and Computer Sciences
and Engineering
King Abdullah University of Science
and Technology
P.O. Box 55455
Jeddah 21534, Saudi Arabia
david.keyes@kaust.edu.sa

and

Department of Applied Physics
and Applied Mathematics
Columbia University
500 W. 120 th Street
New York, NY 10027, USA
kd2112@columbia.edu

Risto M. Nieminen
Department of Applied Physics
Aalto University School of Science
and Technology
00076 Aalto, Finland
risto.nieminen@aalto.fi

Dirk Roose
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3001 Leuven-Heverlee, Belgium
dirk.roose@cs.kuleuven.be

Tamar Schlick
Department of Chemistry
and Courant Institute
of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012, USA
schlick@nyu.edu

Editor for Computational Science
and Engineering at Springer:
Martin Peters
Springer-Verlag
Mathematics Editorial IV
Tiergartenstrasse 17
69121 Heidelberg, Germany
martin.peters@springer.com

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 2014*.

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.* 4th 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*.

For further information on these books please have a look at our mathematics catalogue at the following URL: www.springer.com/series/5151