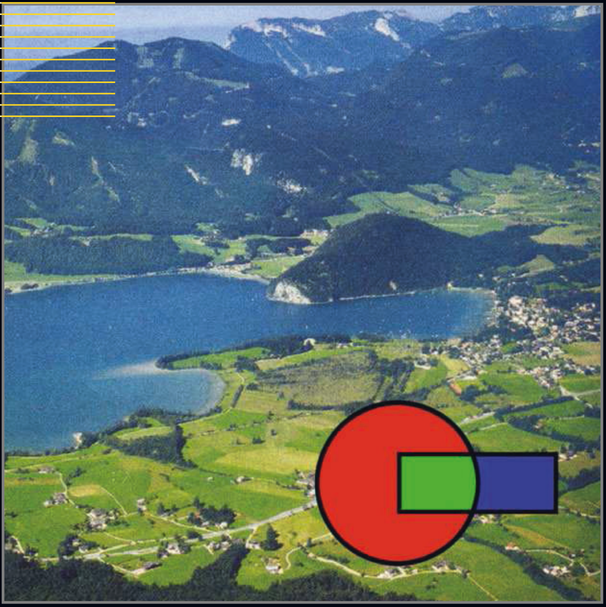
Lecture Notes in Computational Science and Engineering

60



Ulrich Langer
Marco Discacciati
David Keyes
Olof Widlund
Walter Zulehner
Editors

Domain Decomposition Methods in Science and Engineering XVII

Springer

Lecture Notes in Computational Science and Engineering 60

Ulrich Langer · Marco Discacciati
David E. Keyes · Olof B. Widlund
Walter Zulehner (Eds.)

Domain Decomposition Methods in Science and Engineering XVII

With 159 Figures and 110 Tables

Springer

Ulrich Langer
Walter Zulehner

Institute of Computational Mathematics
Johannes Kepler University Linz
Altenbergerstr. 69
4040 Linz, Austria
ulanger@numa.uni-linz.ac.at
zulehne@numa.uni-linz.ac.at

Marco Discacciati

Institute of Analysis and Scientific
Computing - CMCS
Ecole Polytechnique Fédérale de Lausanne
EPFL SB IACS CMCS
Bâtiment MA, Station 8
1015 Lausanne, Switzerland
marco.discacciati@epfl.ch

David E. Keyes

Department of Applied Physics
and Applied Mathematics
Columbia University
500 W. 120th Street, MC 4701
New York, NY 10027, USA
david.keyes@columbia.edu

Olof B. Widlund

Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012-1185, USA
widlund@cims.nyu.edu

ISBN 978-3-540-75198-4 e-ISBN 978-3-540-75199-1

Lecture Notes in Computational Science and Engineering ISSN 1439-7358

Library of Congress Control Number: 2007936914

Mathematics Subject Classification (2000): 65F10, 65N22, 65N22, 65N30, 65N55, 65Y05, 15-06, 35-06, 74-06, 76-06, 78-06

Cover design: WMX Design GmbH, Heidelberg

Printed on acid-free paper

9 8 7 6 5 4 3 2 1

spinger.com

Preface

This volume contains a selection of 71 refereed papers presented at the 17th International Conference on Domain Decomposition Methods held at St. Wolfgang/Strobl, Austria, July 3 - 7, 2006.

1 Background of Conference Series

Domain Decomposition (DD) is an active, interdisciplinary research area concerned with the development, analysis, and implementation of coupling and decoupling strategies in mathematical and computational models arising in Computational Science and Engineering. Historically, it has emerged from the analysis of partial differential equations, beginning with the work of H. A. Schwarz in 1869, in which he established the existence of harmonic functions in domains with complicated boundaries (see logo on the cover), continuing with the variational setting of the alternating Schwarz method by S.L. Sobolev in 1934, and leading to the powerful "Schwarz machinery" developed during the last two decades. Another historical origin of modern domain decomposition methods (DDM) is the classical substructuring techniques which were first developed by mechanical engineers for the finite element analysis of complex structures in the 1960s. We note that the DD technologies are also well suited for treating coupled field problems by hybrid discretization techniques.

The appearance of parallel computers, in particular, of massively parallel computers with distributed memory in the mid 1980s, led to an extensive development of parallel algorithms for solving partial differential equations — problems which play a fundamental role in computational sciences. Time was therefore then right to organize the first international conference, which was held in Paris in 1987. There are now conferences in this series with roughly 18-month intervals:

- Paris, France, 1987

- Los Angeles, CA, USA, 1988
- Houston, TX, USA, 1989
- Moscow, USSR, 1990
- Norfolk, VA, USA, 1991
- Como, Italy, 1992
- University Park, PA, USA, 1993
- Beijing, China, 1995
- Ullensvang, Norway, 1996
- Boulder, CO, USA, 1997
- Greenwich, UK, 1998
- Chiba, Japan, 1999
- Lyon, France, 2000
- Cocoyoc, Mexico, 2002
- Berlin, Germany, 2003
- New York, NY, USA, 2005
- St. Wolfgang, Austria, 2006

The DD conferences are now not only attended by numerical analysts and people interested in parallel computing, but also by scientists from all computational sciences.

The activities of the domain decomposition community are coordinated by the International Scientific Committee on Domain Decomposition Methods:

- Petter Bjørstad, Bergen
- Roland Glowinski, Houston, TX
- Ronald Hoppe, Augsburg and Houston, TX
- Hideo Kawarada, Chiba, Japan
- David Keyes, New York, NY
- Ralf Kornhuber, Berlin
- Yuri Kuznetsov, Houston, TX
- Ulrich Langer, Austria
- Jacques Periaux, Paris
- Alfio Quarteroni, Lausanne, Switzerland
- Zhong-Ci Shi, Beijing
- Olof Widlund, New York, NY
- Jinchao Xu, University Park, PA

Information on and proceedings of the domain decomposition conferences and the ongoing activities of the domain decomposition community can be found on the DDM home page

`http://www.ddm.org` .

2 The Seventeenth Conference

The 17th International Conference on Domain Decomposition Methods (DD17) was held at the Institute for Adult Education in St. Wolfgang/Strobl, Austria, July 3 - 7, 2006. The DD17 was hosted by the Johann Radon Institute for Computational and Applied Mathematics (RICAM), in cooperation with the Special Research Program F013 (SFB F013) on *"Numerical and Symbolic Scientific Computing"* and the Institute for Computational Mathematics (NuMa) of the Johannes Kepler University Linz (JKU). The conference was chaired by Ulrich Langer (NuMa, RICAM and SFB F013). 162 scientists from 29 countries participated. Among the highlights were the talks of the 15 invited speakers:

- Mark Adams (Columbia University, USA): Algebraic Multigrid Methods for Mechanical Engineering Applications,
- Mark Ainsworth (Strathclyde University, UK): Robustness of Some Simple Smoothers for Finite Element and Boundary Elements on Nonquasiuniform Meshes,
- Zoran Andjelić (ABB Schweiz AG, SWITZERLAND): BEM: Opening the New Frontiers in the Industrial Products Design,
- Martin Gander (University of Geneva, SWITZERLAND): Time Domain Decomposition Methods,
- Laurence Halpern (University of Paris 13, FRANCE): Schwarz Waveform Relaxation Algorithms: Theory and Applications,
- Matthias Heinkenschloss (Rice University, USA): Domain Decomposition Methods for PDE Constrained Optimization,
- Hyea Hyun Kim (Courant Institute of Mathematical Sciences, New York University, USA): Domain Decomposition Algorithms for Mortar Discretizations,
- Rolf Krause (University of Bonn, GERMANY): On the Multiscale Solution of Constrained Problems in Linear Elasticity,
- Yuri Kuznetsov (University of Houston, USA): Domain Decomposition Preconditioners for Anisotropic Diffusion,
- Raytcho Lazarov (Texas A&M University, USA): Preconditioning of Discontinuous Galerkin FEM of Second Order Elliptic Problems,
- Young-Ju Lee (University of California, Los Angeles, USA): Convergence Theories of the Subspace Correction Methods for Singular and Nearly Singular System of Equations,
- Günter Leugering (Friedrich-Alexander-University of Erlangen-Nürnberg, GERMANY): Domain Decomposition in Optimal Control of Partial Differential Equations on Networked Domains,
- Jacques Périaux (CIMNE/UPC Barcelona, SPAIN): A Domain Decomposition/Nash Equilibrium Methodology for the Solution of Direct and Inverse Problems in Fluid Dynamics,
- Olaf Steinbach (Graz University of Technology, AUSTRIA): Boundary Element Domain Decomposition Methods: Challenges and Applications,

- Mary Wheeler (University of Texas at Austin, USA): A Domain Decomposition Multiscale Mortar Mixed Method for Flow in Porous Media.

Ten minisymposia were organized on different topics. In addition, the many contributed talks and posters contributed to the success of the DD17.

Sponsoring Organizations:

- Institute for Computational Mathematics (NuMa) of the Johannes Kepler University, Linz (JKU)
- Johann Radon Institute for Computational and Applied Mathematics, Linz (RICAM)
- Linzer Hochschulfond
- Special Research Program SFB F013 *"Numerical and Symbolic Scientific Computing"*
- Springer Verlag
- Township St. Wolfgang
- Township Strobl

Local Organizing Committee Members:

- Sven Beuchler, JKU (Linz)
- Alfio Borzi, University of Graz (Graz)
- Martin Burger, JKU, SFB013 and JKU (Linz)
- Heinz Engl, JKU, SFB013 and JKU (Linz)
- Martin Gander, University of Geneva (Geneva)
- Gundolf Haase, University of Graz (Graz)
- Karl Kunisch, University of Graz (Graz) and RICAM (Linz)
- Ulrich Langer, JKU, SFB013 and JKU (Linz)
- Ewald Lindner, SFB013 and JKU (Linz)
- Joachim Schöberl, RICAM and SFB013 (Linz)
- Olaf Steinbach, Graz University of Technology (Graz)
- Christoph Überhuber, Vienna University of Technology (Vienna)
- Walter Zulehner, JKU (Linz)

The International Scientific Committee would like to thank the members of the Local Organizing Committee for organizing and managing the conference. Special thanks go to the conference secretaries, Magdalena Fuchs and Marion Schimpl, the technical assistants, Wolfgang Forsthuber, Oliver Koch and Markus Winkler, the program coordinators, Dr. Sven Beuchler, Dipl.-Ing. David Pusch, and Dr. Satyendra Tomar, the manager of the social program, Dr. Ewald Lindner, and, last but not least, to Dipl.-Ing. Peter Gruber and Dr. Jan Valdman for producing the book of abstracts.

More information about the conference can be found on the DD17 home page

`http://www.ricam.oeaw.ac.at/dd17` .

3 Conference Proceedings, Selected Books, and Survey Articles

1. P.E. Bjørstad, M.S. Espedal, and David E. Keyes, eds., *Proc. Ninth Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Ullensvang, 1997), Wiley, New York, 1999.
2. T.F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., *Proc. Second Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Los Angeles, 1988), SIAM, Philadelphia, 1989.
3. T.F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, eds., *Proc. Third Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Houston, 1989), SIAM, Philadelphia, 1990.
4. T.F. Chan, T. Kako, H. Kawarada, and O. Pironneau, eds., *Proc. Twelfth Int. Conf. on Domain Decomposition Methods in Sciences and Engineering* (Chiba, 1999), DDM.org, Bergen, 2001.
5. T.F. Chan and T.P. Mathew, *Domain Decomposition Algorithms*, Acta Numerica, 1994, pp. 61–143.
6. N. Débit, M. Garbey, R. Hoppe, D. Keyes, Yu. A. Kuznetsov and J. Périaux, eds., *Proc. Thirteenth Int. Conf. on Domain Decomposition Methods in Sciences and Engineering* (Lyon, 2000), CINME, Barcelona, 2002.
7. C. Farhat and F.-X. Roux, *Implicit Parallel Processing in Structural Mechanics*, Computational Mechanics Advances, Vol. 2, 1994, pp. 1–124.
8. R. Glowinski, G. H. Golub, G. A. Meurant and J. Périaux, eds., *Proc. First Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Paris, 1987), SIAM, Philadelphia, 1988.
9. R. Glowinski, Y.A. Kuznetsov, G. Meurant, J. Périaux and O. B. Widlund, eds., *Proc. Fourth Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Moscow, 1990), SIAM, Philadelphia, 1991.
10. R. Glowinski, J. Périaux and Z. Shi, eds., *Proc. Eighth Int. Conf. on Domain Decomposition Methods in Sciences and Engineering* (Beijing, 1995), Wiley, Strasbourg, 1997.
11. I. Herrera, D. Keyes, O. Widlund and R. Yates, eds., *Proc. Fourteenth Int. Conf. on Domain Decomposition Methods in Sciences and Engineering* (Cocoyoc, Mexico, 2002), National Autonomous University of Mexico (UNAM), Mexico City, 2003.
12. D. Keyes, T.F. Chan, G. Meurant, J.S. Scroggs and R.G. Voigt, eds., *Proc. Fifth Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Norfolk, 1991), SIAM, Philadelphia, 1992.
13. D. Keyes, Y. Saad and D. G. Truhlar, eds., *Domain-based Parallelism and Problem Decomposition Methods in Computational Science and Engineering*, SIAM, Philadelphia, 1995.
14. D. Keyes and J. Xu, eds., *Proc. Seventh Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (University Park, 1993), AMS, Providence, 1995.

15. V.G. Korneev and U. Langer, *Domain Decomposition and Preconditioning*, Chapter 22 in Volume 1 (Fundamentals) of the “Encyclopedia of Computational Mechanics”, ed. by E. Stein, R. de Borst and Th.J.R. Hughes, John Wiley & Sons, 2004.
16. R. Kornhuber, R. Hoppe, J. Periaux, O. Pironneau, O. Widlund and J. Xu, eds., *Proc. Fifteenth Int. Conf. on Domain Decomposition Methods in Sciences and Engineering* (Berlin, 2003), Springer, Heidelberg, 2004.
17. J. Kruis, *Domain Decomposition Methods for Distributed Computing*, Saxe-Coburg Publication, Dun Eaglais, 2005.
18. C.-H. Lai, P.E. Bjørstad, M. Cross, and O. Widlund, eds., *Proc. Eleventh Int. Conf. on Domain Decomposition Methods* (Greenwich, 1999), DDM.org, Bergen, 2000.
19. U. Langer and O. Steinbach, *Coupled Finite and Boundary Element Domain Decomposition Methods*, In “Boundary Element Analysis: Mathematical Aspects and Application”, ed. by M. Schanz and O. Steinbach, Lecture Notes in Applied and Computational Mechanic, Volume 29, Springer, Berlin, pp. 29-59, 2007.
20. V.I. Lebedev and V.I. Agoshkov, *Poincaré-Steklov operators and their applications in analysis*, Academy of Sciences USSR, Dept. of Numerical Mathematics, Moskow, 1983, (In Russian).
21. P. Le Tallec, *Domain Decomposition Methods in Computational Mechanics*, Computational Mechanics Advances, Vol. 1, No. 2, 1994, pp. 121–220.
22. J. Mandel, C. Farhat and X.-C. Cai, eds., *Proc. Tenth Int. Conf. on Domain Decomposition Methods in Science and Engineering* (Boulder, 1997) AMS, Providence, 1998.
23. S. Nepomnyaschikh, *Domain Decomposition Methods*, In “Lectures on Advanced Computational Methods in Mechanics”, ed. by J. Kraus and U. Langer, Radon Series on Computational and Applied Mathematics, de Gruyter, Berlin, 2007.
24. P. Oswald, *Multilevel Finite Element Approximation: Theory and Applications*, Teubner Skripten zur Numerik, Teubner-Verlag, Stuttgart, 1994.
25. L. Pavarino and A. Toselli, *Recent Developments in Domain Decomposition Methods. Proc. Workshop held in Zürich in 2001*, Lecture Notes in Computational Sciences and Engineering, Vol. 23, Springer, Heidelberg, 2002.
26. A. Quarteroni, J. Periaux, Y.A. Kuznetsov, and O. B. Widlund, eds., *Proc. Sixth International Conference on Domain Decomposition Methods in Science and Engineering* (Como, 1992) AMS, Providence, 1994.
27. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Sciences Publications, Oxford, 1999.
28. B. F. Smith, P. E. Bjørstad and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, Cambridge, 1996.

29. O. Steinbach, *Stability Estimates for Hybrid Coupled Domain Decomposition Methods*, Lecture Notes in Mathematics, Vol. 1809, Springer, Berlin, 2003.
30. A. Toselli and O. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, Springer, New York, 2005.
31. O. Widlund and D.E. Keyes, eds., *Proc. Sixteenth Int. Conf. on Domain Decomposition Methods in Sciences and Engineering* (New York City, 2005), Springer, Heidelberg, 2007.
32. B. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, Volume 17 of *Lecture Notes in Computational Science and Engineering*, Springer, Berlin, Heidelberg, 2001.
33. J. Xu, *Iterative Methods by Space Decomposition and Subspace Correction: A unifying approach*, SIAM Review, Vol. 34, No. 4, 1992, pp. 581–613.
34. J. Xu and J. Zou, *Some Nonoverlapping Domain Decomposition Methods*, SIAM Review, Vol. 40, No. 4, 1998, pp. 857–914.

4 Organization

Parts I and III of the proceedings collect the plenary and contributed presentations, respectively; the papers appear in alphabetical order by the first-listed author. In part II "Minisymposia", the organizers of the minisymposia provide short introductions to the minisymposia. Within each minisymposium section, the papers again appear in alphabetical order.

5 Acknowledgments

The editors would like to thank all authors for their contributions, the anonymous referees for their valuable work, and Dr. Martin Peters and Ms. Thanh-Ha Le Thi from Springer for continuing support and friendly collaboration in preparing these proceedings.

Linz,	Ulrich Langer, Walter Zulehner
Lausanne,	Marco Discacciati
New York,	David E. Keyes, Olof B. Widlund
August 2007	

Contents

Part I Plenary Presentations

Part II Minisymposia

Part III Contributed Presentations

Part I

Plenary Presentations

BEM: Opening the New Frontiers in the Industrial Products Design

Zoran Andjelić

ABB Corporate Research
zoran.andjelic@ch.abb.com

Summary. Thanks to the advances in numerical analysis achieved in the last several years, BEM became a powerful numerical technique for the industrial products design. Until recent time this technique has been recognized in a praxis as a technique offering from one side some excellent features (2D instead of 3D discretization, treatment of the open-boundary problems, etc.), but from the other side having some serious practical limitations, mostly related to the full-populated, often ill-conditioned matrices. The new, emerging numerical techniques like MBIT (Multipole-Based Integral Technique), ACA (Adaptive Cross-Approximations), DDT (Domain-Decomposition Technique) seems to bridge some of these known bottlenecks, promoting those the BEM in a high-level tool for even daily-design process of 3D real-world problems.

The aim of this contribution is to illustrate the application of BEM in the design process of the complex industrial products like power transformers or switchgears. We shall discuss some numerical aspects of both single-physics problems appearing in the Dielectric Design (Electrostatics) and multi-physics problems characteristic for Thermal Design (coupling of Electromagnetic - Heat transfer) and Electro-Mechanical Design (coupling of Electromagnetic - Structural mechanics).

Nomenclature

- x - source point
- y - integration point
- $\Gamma := \partial\Omega$ - surface around the body
- σ^e - electric surface charge density $[As/m^2]$
- ρ^e - electric volume charge density $[As/m^3]$
- σ^m - magnetic surface charge density $[Vs/m^2]$
- ρ^m - magnetic volume charge density $[Vs/m^3]$
- q - charge $[As]$
- ε - dielectric constant (permittivity, absolute) $[F/m = As/Vm]$
- ε_0 - dielectric constant of free space (permittivity)$=1/\mu_0 c_0^2 \approx 0.885419e^{-11}$
- c_0 - speed of electromagnetic waves (light) in vacuum$=2.2997925e^8$ $[m/s]$
- ε_r - relative dielectric constant

- μ - magnetic permeability (absolute) $[H/m]$
- μ_0 - magnetic permeability of the free space $[H/m] = 4\pi/10^7$
- μ_r - relative magnetic permeability
- σ - electrical conductivity $[Sm/mm^2]$
- $\boldsymbol{E}$ - electrical field strength $[V/m]$
- $\boldsymbol{D}$ - electrical flux (displacement) density $[As/m^2]$
- φ - electrical potential $[V]$
- I - electrical current $[A]$
- U - electrical voltage $[V]$
- $\varphi^{ext}(I)$ - potential of the external electrostatic field [V]
- $\mathbf{H}$ - magnetic field strength $[A/m]$
- $\mathbf{B}$ - magnetic flux density $[T]$
- $\boldsymbol{F}$ - force $[N]$
- $\mathbf{f}_v$ - volume force density $[N/m^3]$
- $\mathbf{f}_m$ - magnetic force density $[N/m^3]$
- $\mathbf{f}_m^s$ - "strain" magnetic force density $[N/m^3]$
- $\mathbf{J}$ - current density $[A/m^2]$
- $\mathbf{J}_0$ - exciting current density $[A/m^2]$
- $\mathbf{S}$ - Poynting vector
- $\bar{f}$ - time-average force density $[N/m^3]$ (volume) or $[N/m^2]$ (surface)
- Θ - solid angle
- $\mathbf{j}$ - current density (complex vector) $[A/m^2]$
- ω - angular velocity $[rad/s]$
- f - frequency $[Hz]$
- T - temperature $[^oC]$ or $[K]$
- α - heat transfer coefficient $[W/m^2K]$

1 Introduction

One of key challenges in a booming industrial market is to achieve a better *time2market* performance. This marketing syntagma one could translate as: "To be better (best) in a competition race, bring the product to the market in the fastest way (read cheapest way), simultaneously preserving / improving its functionality and reliability". One of the nowadays unavoidable ways to achieve this target is to replace partially (or completely) the traditional *Experimentally-Based Design* (EBD) with the *Simulation-Based Design* (SBD) of industrial products. Usage of SBD contributes in:

- Acceleration of the design process (avoiding prototyping),
- Better design through better understanding of the physical phenomena,
- Recognizability of the product's weak points already at the design stage.

Introduction of the SBD in the design process requires accurate, robust and fast numerical technologies for:

- **3D real-world problems** analysis, preserving the necessary structural and physical complexity,
- ... but using the numerical technologies enough **user-friendly** to be accepted by the designers,
- ... and using the numerical technologies suitable for the **daily design** process.

All these three items present quite tough requirements when speaking about the industrial products that are usually featured by huge dimensions, huge aspect ratio in model dimensions, complex physics, complex materials. For the class of the problems we are discussing here, there are basically two candidates among many numerical methods that could potentially be used: FEM (Finite Element Method) and BEM (Boundary Element Method). Our experience shows that for the electromagnetic and electromagnetically-coupled problems BEM has certain advantages when dealing with complex engineering design.

Without going into details, let us list some of the main BEM characteristics:

- Probably the most important feature of BEM is that for *linear* classes of problems the discretization needs to be performed only over the interfaces between different media. This excellent characteristic of BEM makes the discretisation/meshing of complex 3D problems more straightforward and usable for simulations in a *daily design* process.
- Also, this feature is of utmost importance when dealing with the simulation of *moving boundary problems.* Thanks to the fact that the space between the moving objects does not need to be meshed, BEM offers an excellent platform for the simulation of *dynamics*, especially in 3D geometry.
- Furthermore, the *open boundary problem* is treated easily with BEM, without need to take into account any additionally boundary condition. When using tools based on the differential approach (FEM, FDM), the *open boundary problem* requires an additional *bounding box* around the object of interest, which has a negative impact on both mesh size and computation error.
- Another important feature of BEM is its *accuracy.* Contrary to differential methods, where *adaptive mesh refinement* is almost imperative to achieve the required accuracy, with BEM it is frequently possible to obtain good results even with a relatively rough mesh. But, at this point we also do not want to say that "adaptivity" could not make life easier even when using BEM.

In spite of the above mentioned excellent features of BEM, this method had until recent time some serious limitations with respect to the practical design, mostly related to the:

- full populated matrix,
- huge memory requirements,

- bad matrix conditioning.

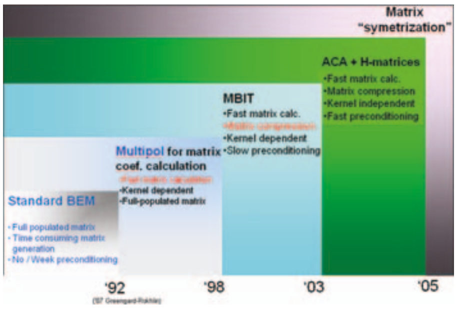


Fig. 1. Paradigm change in BEM development

Thanks to the real breakthroughs happening in the last decade in BEM-related applied mathematics, most of these bottlenecks have been removed. To author's opinion the work done by Greengard and Rochlin, Greengard [10], is probably one of the crucial ignitions contributing to this paradigm change, Figure 1. Today we can say that this work, together with a number of capital contributions of other groups working with BEM, has lunched really a new dimension in the simulation of the complex real-world problems. In the following we shall try to illustrate it on some practical examples like:

- Dielectric design of circuit breakers,
- Electro-mechanical design of circuit breakers,
- Thermal design for power transformers.

2 Dielectric Design of Circuit Breakers

Under **Dielectric Design** we usually understand the *Simulation-Based Design* (SBD) of configurations consisting of one or more electrodes loaded with either *fixed* or *floating* potential and being in contact with one or more dielectric media. From the physics point of view, here we deal with a *single-physics* problem, which can be described either by a Laplace or Poisson equation.

2.1 Briefly About Formulation

For 3D BEM analysis of electrostatic problems, the equations satisfying the field due to stationary charge distribution can be derived directly form the Maxwell equations, assuming that all time derivatives are equal to zero. The formulation can be reduced to the usage of I and II Fredholm integral equations[1]:

[1] The complete formulation derivation can be found in Tozoni [19], Koleciskij [15]

$$\varphi(x) = \varphi^{ext}(x) + \frac{1}{4\pi\varepsilon_0}\sum_{m=1}^{M}\int_{\Gamma_m}\sigma^e(y)K_1 d\Gamma_m(y) \tag{1}$$

$$\sigma^e(x) = \frac{\lambda}{2\pi}\sum_{m=1}^{M}\int_{\Gamma_m}\sigma^e(y)\frac{\mathbf{r}\cdot\mathbf{n}}{r^3}d\Gamma_m(y) \tag{2}$$

where $\varepsilon = \varepsilon_0 \cdot \varepsilon_r$ is absolute permittivity with $\varepsilon_0 = 8.85 \cdot 10^{-12}F/m$ the permittivity of the free space and ε_r the relative permittivity or dielectric constant, $K_1 = \frac{1}{r} = \frac{1}{|\mathbf{x}-\mathbf{y}|}$ is a *weakly singular* kernel, r is a distance between the calculation point x and integration point y, $\mathbf{n}$ is a unit normal vector in point x directed into the surrounding medium, and $\lambda = \frac{\varepsilon_i - \varepsilon_e}{\varepsilon_i + \varepsilon_e}$.
The equation (1) is usually applied for the points laying on the electrodes, and the equation (2) is applied for the points positioned on the interface between different dielectrics. Then the electrostatic field strength at any point in the space can be determined as:

$$\mathbf{E}(x) = -\nabla\varphi(x) = -\frac{1}{4\pi\varepsilon_0}\sum_{m=1}^{M}\int_{\Gamma_m}\sigma^e(y)\cdot\nabla K_1 d\Gamma_m(y) \tag{3}$$

whereby the position vector $\mathbf{r} = \mathbf{x} - \mathbf{y}$ in K_1 is pointed towards the collocation point x. The discretization of equations (1) and (2) yields a densely populated matrix, which is well known as the major bottleneck in BEM computations. The amount of storage is of order $O(N^2)$, with N being the number of unknowns. Furthermore, the essential step at the heart of the iterative solution of this system is a matrix-vector multiplication and the cost of such a multiplication is also of order $O(N^2)$. Thus a reduction of the complexity to $O(N \log N)$ or $O(N)$ would naturally be very desirable. Developments started with a seminal paper by Greengard [10] that proposed a *Fast Multipole Method*, which became highly popular in several numerical communities. Another fundamental development was brought about by Hackbusch [11, 12]. In the following we present a brief description of MBIT[2] algorithm that is used in our computations. The central idea is to split the discretized boundary integral operator into a *far-field* and a *near-field* zone. The singularity of the kernel of the integral operator is then located in the *near-field*, whereas the kernel is continuous and smooth in the *far-field*. Compression can then be achieved by a separation of variables in the *far-field*. In order to reach this goal, the boundary in the first stage is subdivided into clusters of adjacent panels that are stored in a hierarchical structure called the panel-cluster tree. The first cluster is constructed from all elements/panels (the largest set of elements/panes) and is denoted as x_{00}, Figure 2. We continue to subdivide each existing cluster level successively into smaller clusters with cluster centers x_{ij} through bifurcation, Figure 2a. After several bifurcations we obtain

[2] Multipole-Based Integral Technique

a cluster tree structure for the elements/panel set, Figure 2b. Then, in the

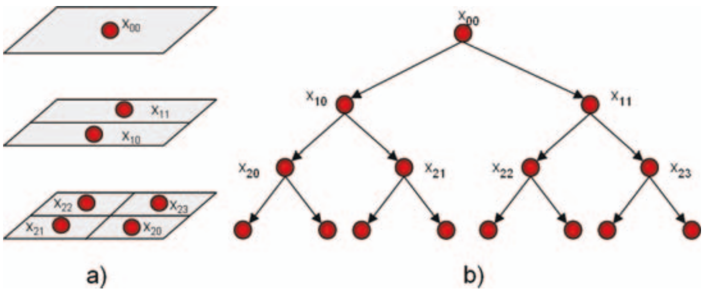


Fig. 2. a) Panel bifurcation, b) Panel cluster tree

second stage we collect all admissible pairs of clusters, i.e. pairs that fulfill the admissibility condition $|\mathbf{x}-\mathbf{x}_0|+|\mathbf{x}^c-\mathbf{x}_0^c| \leq \eta\,|\mathbf{x}_0-\mathbf{x}_0^c|$ where $0 \leq \eta < 1$ into the far-field block. The centers of gravity of the panel clusters and node/vertex clusters are here denoted by $\mathbf{x}_0$, $\mathbf{x}_0^c$, respectively. All other pairs of clusters (the non-admissible ones) belong to the *near-field*. Then the matrix entries corresponding to the *near-field* zone are computed as usual, whereas the matrix blocks of the *far-field* are only approximated. This is achieved by an expansion of the kernel function $k(\mathbf{x},\mathbf{x}^c)$ that occurs in the matrix entries

$$a_{ij} = \int\limits_{\Gamma}\int\limits_{\Gamma} \hat{\varphi}_i(\mathbf{x})k(\mathbf{x},\mathbf{x}^c)\varphi_j(\mathbf{x}^c)d\Gamma(x)d\Gamma(x^c)\,. \tag{4}$$

The expansion:

$$k(\mathbf{x},\mathbf{x}^c) \approx k_m(\mathbf{x},\mathbf{x}^c;\mathbf{x}_0,\mathbf{x}_0^c) = \sum_{(\mu,\nu)\in \mathrm{I}_m} k_{(\mu,\nu)}(\mathbf{x},\mathbf{x}_0^c)X_\mu(\mathbf{x},\mathbf{x}_0)Y_\nu(\mathbf{x}^c,\mathbf{x}_0^c) \tag{5}$$

decouples the variables $\mathbf{x}$ and $\mathbf{x}^c$ and must be done only in the *far-field*. Then, the matrix-vector products can be evaluated as:

$$\nu = \tilde{A}\cdot u = N\cdot u + \sum_{(\sigma,\tau)\in F} X_\sigma^T(F_{\sigma,\tau}(Y_\tau\cdot u))\,. \tag{6}$$

Several expansions can be used for this purpose: Multipole-, Taylor- and Chebyshev-expansion. The procedures lead to a low rank approximation of the *far-field* part and it is shown in Schmidlin [17] that one obtains exponential convergence for a proper choice of parameters. A more detailed elaboration and comparison of all three type of expansions can also be found in the same reference.

Example 1: ***SBD for a Generator Circuit-Breaker Design***

In this example it is briefly shown how Simulation-Based Design of the Generator Circuit-Breaker (GCB) is performed using a BEM[3] module for electrostatic field computation.

Generator circuit-breakers, Figure 3, are important components of electricity transmission systems. Figure 4 (left) shows the complete assembly of a GCB containing, beside the interrupting chamber as a key component, all other parts such as current and voltage transformers, earthing switches, surge capacitors, etc. The simulation details for the above shown generator circuit-breakers case were:

Fig. 3. ABB generator circuit-breaker

- The discretization of the model has been performed using *second order* triangle elements.
- The stiffness matrix has been assembled using an *Indirect* Ansatz with *collocation* in the main triangle vertices, formulas (1) and (2). It has to be mentioned here that in both the real design and consequently then in the simulation model, geometrical singularities like edges and corners have been removed through *rounding.* In the real design this is a common practice in all high-voltage devices in order to prevent the occurrence of *dielectric breakdown.* On the numerics side, this fact enables usage of the *nodal collocation* method - which is also the fastest one - without violating the mathematical correctness of the problem.
- The coefficients of the stiffness matrix have been calculated using the **multipole approach**, Greengard [10], with *monopole*, *dipole* and *quadropole* approximations for the far-field treatment, Andjelic [3]. *Diagonal matrix preconditioning* has been used, which enables fast and reliable matrix solution using GMRES. This run has been accomplished without any matrix compression, but using a parallelized version of the code, Blaszczyk [7]. For a parallel run we used a PC cluster with 22 nodes. The data about memory and CPU time are given in Table 1.
- The calculated electrostatic field distribution is shown in Figure 4 (right). It can be seen that the highest field strength appears on the small feature details, such as screws.

Validation:

Replacement of the EBD with the SBD requires a number of field tests to confirm the simulation results by the experiments. Validation is one of important

[3] This BEM module is a sub-module for electrostatic analysis in POLOPT (http://www.poloptsoftware.com), a 3D BEM-based simulation package for single and multi-physics computation.

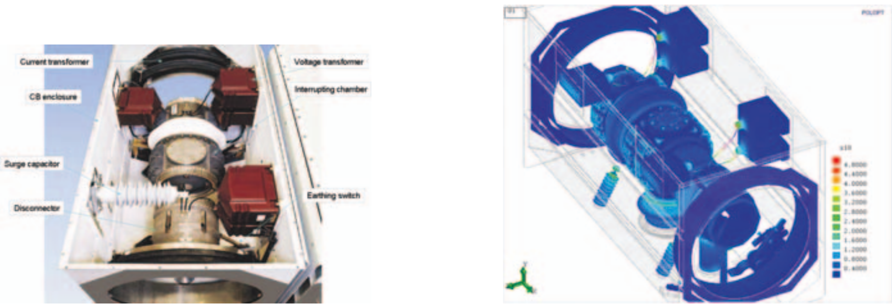


Fig. 4. GCB assembly (left). ABB Generator Circuit-Breaker: Electrostatic field distribution, E[V/m] (right)

Table 1. The analysis data for GCB example.

Elements	*Nodes*	*Main vertices*	*Memory*	*CPU*
145782	291584	80230	42GByte	2h20'

steps to gain the confidence in the simulation tools. Figure 5 (right) shows the experimental verification of the results obtained by the simulation of the GCB.

Note 1:
Calculated field distribution is just a “primary” information for the designers. For complete judgment about the products behavior, it is usually necessary to go one step forward, i.e. to evaluate the *design criteria*. Very often such criteria are based on the analysis of the field lines, Figure 5 (left), that enables further the conclusion about the breakdown probability in the inspected devices.

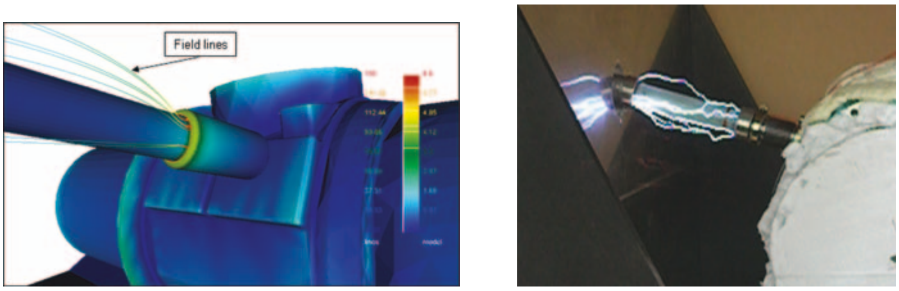


Fig. 5. Electric field strength distribution - detailed view including field lines traced from the position of the maximal field values (left). Experimental verification of the simulation results (right)

3 Electro-Mechanical Design of Circuit Breakers

In Electro-Mechanical class of problems we are dealing with coupled electromagnetic / structural-dynamic phenomena. Better to say, we are seeking to find out what is a mechanical response of the structure subjected to the action of the electromagnetic forces. Coupling of these phenomena can be either *weak* or *strong*. Under *weak* coupling we understand the sequential analysis of each phenomena separately, coupled together via an iterative scheme. In *strong* coupling we usually deal with the simultaneous solution of both problems, whereby the coupling is preserved on the equations level. In the present material we deal with the *weak* coupling, that usually assumes two main steps:

- Calculation of electromagnetic forces
- Calculation of mechanical response

Forces evaluation is a first step in this coupled simulation chain. Electromagnetic forces appear in any device conducted by either DC or AC current, or subjected to the action of an external electromagnetic field[4,5]. Force analysis itself is a bright field and will not be treated in details within this material. More info can be found in Andjelic [4]. Here we shall give only a brief overview on the Workflow for coupled EM-ME simulation tasks, Figure 6. A very first step in the simulation chain is the calculation of the excitation current / field distribution. The calculation of the stationary current distribution in the

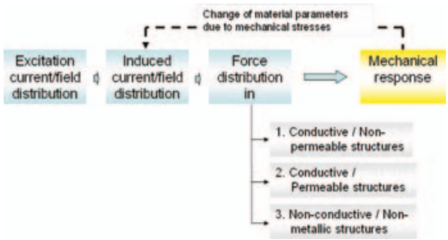

Change of material parameters
due to mechanical stresses
Excitation current/field distribution
Induced current/field distribution
Force distribution in
Mechanical response
1. Conductive / Non-permeable structures
2. Conductive / Permeable structures
3. Non-conductive / Non-metallic structures

Fig. 6. Weak coupling scheme for EM-SM problems

[4] In this material we shall not treat the electro-mechanical problems whereby the force are of electrostatic origin.

[5] In certain applications (force sensors, pressure sensors, accelerometers) we are not looking for *mechanical response* caused by the electromagnetic forces, but rather for *electrical response* caused by the mechanical forces (*piezoelectric problem*). This case will not be covered in the scope of this material. More information about BEM treatment of these classes of problems can be found in Gaul [9], Hill [14]. Here we shall also not cover the topic of coupled Electro-Magnetic / Mechanics problems related to magnetostriction phenomena (change of the shape of magnetostrictive material under the influence of a magnetic field). More information for example in Whiteman [20].

conductors assumes the solution of the Laplace problem, analogous to the previously described electrostatic case. A detailed description of the formulations for stationary current calculation can be found in Andjelic [4]. When performing a coupled electromagnetic-structural mechanics analysis, we are not interested in the *total force*, but rather in the *local force* density distribution. For stationary case the local force density (**forces per unit volume**, $[N/m^3]$) can be calculated as:

$$\mathbf{f}_m = \mathbf{J} \times \mathbf{B} - \frac{1}{2} H^2 \nabla \mu + \mathbf{f}_m^s \, . \tag{7}$$

Usually in praxis we are interested in the time-averaged force density $\bar{f}$ $[N/m^3]$:

$$\bar{f} = \frac{1}{2} \mathrm{Re} \left\{ \rho^e \mathbf{E}^* + \mathbf{J} \times \mathbf{B}^* + \rho^m \mathbf{H}^* + \mathbf{M} \times \mathbf{D}^* \right\} \tag{8}$$

where $\mathbf{M} = i\omega \mathbf{P}^m = i\omega\mu_0(\mu_r - 1)\mathbf{H}$ are the bounded magnetic currents, and ρ^m are the bounded magnetic charges.
Basically, we can distinguish between the forces acting on:

- conductive/non-permeable structures,
- conductive/permeable structures[6],
- non-conductive/non-permeable structures[7].

If we stay with the typical design cases appearing in the transformers and circuit-breakers design, that the mostly encountered problems are related to the forces in conductive/non-permeable structures (bus-bars, windings). For time-average Lorentz force density in a non-permeable current-carrying conductor (μ=1), the equation (7) reduces to:

$$\bar{f} = \frac{1}{2} \mathrm{Re} \left\{ \mathbf{J} \times \mathbf{B}^* \right\} \, . \tag{9}$$

These local forces are then further passed as an *external load* for the analysis of the mechanical quantities, last module in Figure 6. BEM formulations used in our module for linear elasticity problems is described in more details in Andjelic [4].

Example 2: ***Electro-mechanical Design of Generator Circuit-Breaker***

Let us consider now the coupled electromechanical loading of a switch found in the generator circuit breaker (GCB) seen already in the previous example. Following a current-distribution and eddy-current analysis, it is possible by Biot-Savart calculation to find the body-forces arising out of Lorentz interactions. In fact, these forces are often of interest only in a limited region of the

[6] More on the force analysis on conductive/permeable structures can be found in Henrotte [13].

[7] This class of problems is rather seldom and appears mostly in sensor design, Andjelic [1].

entire engineering system, typically in moving parts. In the GCB case presented here, a point of particular interest is the “knife” switch, where there is a tendency for the generated Lorentz forces to act so as to open the switch. Taking the example from earlier, for the mechanical part of the analysis only a limited portion of the mesh needs to be evaluated. Results were calculated using a mesh comprising 4130 triangular planar surface elements and 2063 nodes. The volume discretization (necessary for the body-force coupling) comprises 14000 tetrahedra. This model has been analyzed taking advantage of the ACA approximation for the single and double layer potentials described earlier in the outline of the formulation. Results from this analysis are shown in Figure 7. Clearly visible is the effect of the coupling forces on the switch, which has a tendency to move out of its closed position under the action of the electromagnetic loading. This quantitative and qualitative information is a valuable input into the design process leading to the development of complex electromechanical systems.

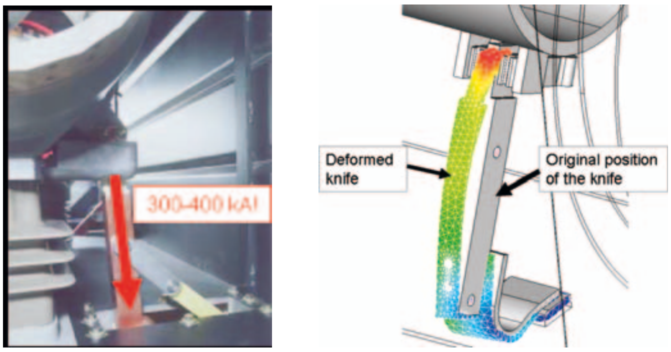


Fig. 7. A detail of the earthing-switch in GCB carrying the current of 300-400 kA! (left). Deformation of the earthing knife (overscaled), caused by the action of the short-circuit forces (right).

4 Thermal Design for Power Transformers

When speaking about the Thermal Design we are usually looking for thermal response of the structures caused by the electromagnetic losses. In reality, the physics describing this problem is rather complex. There are three major physical phenomena that should be taken into account simultaneously: the electromagnetic part responsible for the losses generation, a fluid part responsible for the cooling effects and thermal part responsible for the heat transfer. Simulation of such problems, taking into account both complex physics and complex 3D structures found in the real-world apparatus is still a challenge, especially with respect to the requirements mentioned at the beginning:

accuracy-robustness-speed. A common practice to avoid a complex analysis of the cooling effects by a fluid-dynamics simulation is to introduce the *Heat-Transfer Coefficients* (HTC) obtained either by simple analytical formulae, (see for example Boehme [8]) or based on experimental observations. For this type of analysis the link between the electromagnetic solver and heat-transfer solver is throughout the *losses* calculated on the electromagnetic side and passed further as *external loads* to the heat-transfer module.

4.1 Workflow

The Workflow used for the coupled simulation of electro-magnetic / thermal problems is shown in Figure 8. Usually the very first step in thermal simulation the industrial products like power transformer is import of the geometry from CAD tool, followed by meshing and setting appropriate boundary conditions (BC) and material data. It has to be stressed again that thanks to the excellent features of BEM, we can solve such complex diffusion problem by meshing only the interfaces between different media, i.e. avoiding completely any volume mesh[8]! The solution phase consist of three major steps: calculation of the excitation current distribution, calculation of the eddy-currents / losses distribution and finally calculation of the temperature distribution. Let us give a brief outline on the eddy-current formulation, as one of the probably most complicated problems in the computational electromagnetics. More info on the formulations of excitation current as well as thermal calculation can be found in Andjelic [4].

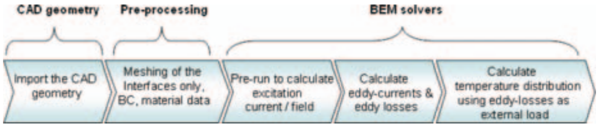

CAD geometry
Pre-processing
BEM solvers
Import the CAD geometry
Meshing of the Interfaces only, BC, material data
Pre-run to calculate excitation current / field
Calculate eddy-currents & eddy losses
Calculate temperature distribution using eddy-losses as external load

Fig. 8. EM-TH Workflow

4.2 Eddy-current Analysis

There are a number of possible formulation that can be used for BEM-based analysis of eddy-current problems. A useful overview of the available eddy-current formulations can be found in Kost [16]. Here we follow the $H - \varphi$ formulation, whereby for the treatment of the skin-effect problems an modified version of this formulation is used, Andjelic [2]. The $H-\varphi$ formulation is based

[8] This is valid so long we are working with linear problems. In the case when non-linear problem has to be treated, than when using BEM it is necessary to apply the volume mesh, but only for the parts having non-linear material behavior!

on the *indirect* Ansatz, leading thus to the minimal number of 4 degrees of freedom (DoF) per node[9]. This nice feature makes this formulation suitable for the eddy-current analysis of complex, real-world problems. The $H - \varphi$ formulation need to be used with a care in cases where the problem is *multi-valued*, i.e. when the model belongs to the class *multi-connected problems*, Tozoni [19]. The following integral representation is used[10]:

$$\begin{aligned}&\tfrac{1}{2}\mathbf{j}(x) + \tfrac{1}{4\pi}\oint\limits_{\Gamma} \mathbf{n}(x) \times \left(\mathbf{j}(y) \times \nabla \tfrac{e^{-(1+i)k\cdot r}}{r}\right) d\Gamma(y) - \\ &\tfrac{1}{4\pi}\oint\limits_{\Gamma} \sigma^m(y)(n(x) \times \nabla \tfrac{1}{r} d\Gamma(y) \\ &= -\mathbf{n}(x) \times \mathbf{H}_0(x)\end{aligned} \tag{10}$$

$$\begin{aligned}&\tfrac{1}{2}\sigma^m(x) + \tfrac{1}{4\pi}\oint\limits_{\Gamma} \sigma^m(y) \cdot \mathbf{n}(x) \cdot \nabla(\tfrac{1}{r}) d\Gamma(y) + \\ &\tfrac{\mu}{4\pi\mu_0}\oint\limits_{\Gamma} \mathbf{n}(x) \left(\mathbf{j}(y) \times \nabla \tfrac{e^{-(1+i)k\cdot r}}{r}\right) d\Gamma(y) \\ &= -\mathbf{n}(x) \cdot \mathbf{H}_0(x)\,.\end{aligned} \tag{11}$$

This boundary integral equation system can be written in operator form:

$$\begin{bmatrix} A_1 & B_1 \\ B_2 & A_2 \end{bmatrix} \begin{pmatrix} \mathbf{j} \\ \sigma^m \end{pmatrix} = \begin{pmatrix} -2\mathbf{n} \times \mathbf{H}_0 \\ -2\mathbf{n} \cdot \mathbf{H}_0 \end{pmatrix}\,. \tag{12}$$

For more details on a numerical side of this approach the reader is referred to Schmidlin [18]. Solution of the equation system (12) gives the virtual magnetic charges σ^m and virtual current density $\mathbf{j}$. Then, the magnetic field in conductive materials can be expressed as:

$$\mathbf{H}^+(x) = \frac{1}{4\pi}\oint_{\Gamma} \nabla \times [\mathbf{j}(y)K(x,y)]\, d\Gamma(y); \qquad x \in \Omega^+; y \in \Omega^+ \tag{13}$$

and

$$\mathbf{H}^-(x) = \mathbf{H}o(x) - \frac{1}{4\pi}\oint_{\Gamma} \sigma_m(y)\nabla_x G(x,y) d\Gamma(y) \qquad x \in \Omega^-; y \in \Omega^- \tag{14}$$

in the non-conductive materials. $\mathbf{H}_0$ is the primary magnetic field produced by the exciting current $\mathbf{J}_0$ and $K = e^{-(1+i)k\cdot\mathbf{r}}/r$, $G = 1/r$.

Fast BEM for Eddy-current Analysis

Although the above formulation is the minimal-order formulation for 3D eddy-current analysis, it still reaches very fast the limits (both in memory and CPU)

[9] With $H-\varphi$ formulation it is possible to work even with only 3 DoF/node, whereby the eddy-currents on the surfaces are described in a *surface* coordinate system instead of Cartesian, Yuan [21].

[10] For complete derivation of the above formulations, please look in Kost [16], Tozoni [19], Andjelic [2]

when trying to apply it to the simulation of the complex real-world problems. As said at the very beginning, the new emerging techniques like MBIT or ACA have enabled the efficient usage of this (and other BEM-based formulations) by removing most of the known bottlenecks (huge memory, big CPU, bad matrix conditioning). MBIT has enabled the efficient matrix generation, together with low-memory matrix compression. For a pity, when using MBIT an extra preconditioner is necessary in the case of bad conditioned matrices (for example Schur-complement). ACA from other side covers all three major critical points. Beside fast matrix generation, excellent compression, ACA provides also inherently the matrix preconditioning, Bebendorf [6], Bebendorf [5]. As illustration, Figure 9 shows a comparison of MBIT and ACA versus dense matrix solution.

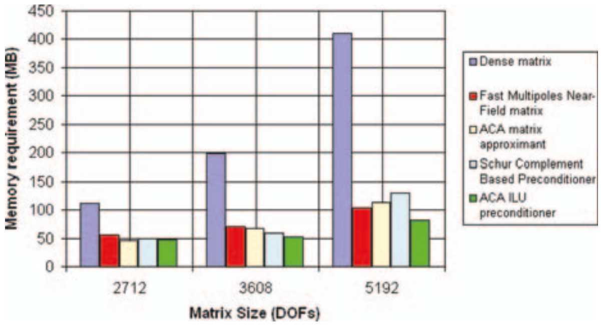


Fig. 9. Memory requirements for various matrix compression and preconditioning methods

Example 2: ***Thermal Design of Power Transformers***

The procedure described above has been used for the analysis of a number of power transformer problems, both single- and three-phase units, Figure 10 (left). Figure 10 (right) shows the distribution of the calculated excitation field over the transformer tank wallfootnote, together with the three-phase bus-bars structure. It has to be noted that typical transformers structures (for example tank or turrets) usually consist of one or more components made of different materials like magnetic or non-magnetic steel, copper or aluminum. The numerical procedure that are used have to be careful selected in order to properly resolve the penetration of electro-magnetic field into each of these materials, depending on their magnetic permeability, electrical conductivity and applied frequency. Calculation of eddy currents and losses is performed using the above described numerical procedure. Figure 11 shows the distribution of the calculated eddy-currents.

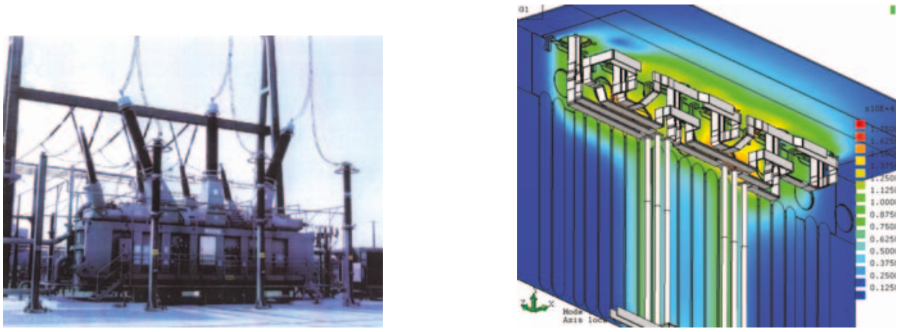


Fig. 10. 985 MVA Power Transformers, ABB (left). Excitation field distribution in the three-phase transformer bus-bars (right)

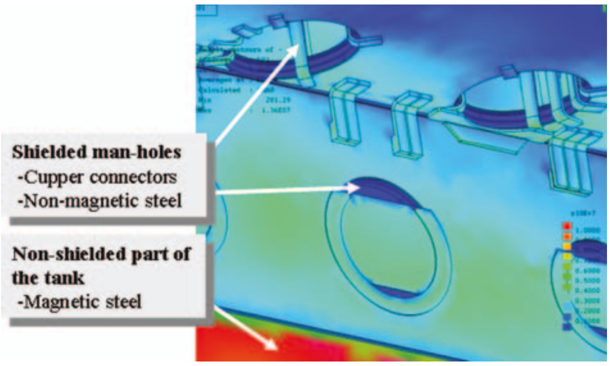


Fig. 11. Eddy current distribution (complex magnitude)- detailed view to the inner shielding details

Validation

As mentioned before, an important aspect of the practical usage of the simulation tools is its validation, i.e. its comparison with the measured data. In the following example we present as illustration the comparison between simulated and measured temperature for an 400 MVA single-phase transformer unit.[11] More on BEM-based approach for temperature analysis can be found in Andjelic [4]. The temperature calculation is obtained using previously calculated eddy losses as the external load for thermal run. The impact of the cooling effects is taken into account by the appropriate choice of the heat transfer coefficients. The simulation output has been validated by comparison with thermography recording done during the transformer operation. Figure 12 shows the comparison between the simulation results and the measured re-

[11] The parts of the tank exposed to the thermal overheating are often made of the non-magnetic steel. This allows usage of the linear Ansatz for eddy-current class of problems.

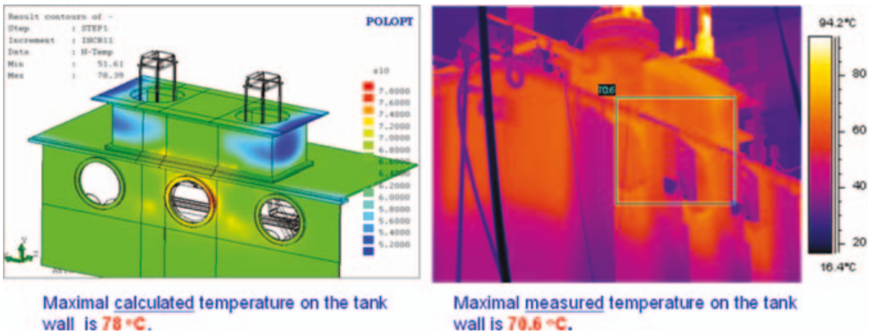


Fig. 12. Validation

sults obtained by the thermography. It can be seen that the simulation results have good agreement with the measured results. The difference between the measured and calculated results (10% in this case) could be probably explained by the inaccurate estimation of the heat transfer coefficients used in the simulation.

5 Some Concluding Remarks

In this paper we have tried to illustrate some BEM-based approaches for the simulation of different problems appearing in engineering design praxis. The excellent features of BEM for both single and multi-physics tasks are highlighted, together with some emerging numerical techniques like MBIT and ACA, recognized as the major drivers leading to the real breakthrough in BEM usage for practical design tasks.

But, beside these and many other good features of BEM, and staying at the level of *static* or *quasi-static* simulation tasks, there are still a number of potential improvements that could be made to achieve the “best in the class” tool desired for the advanced simulations in the industrial design (*strong* coupling formulations, non-linearity treatment, contact problems, preconditioning etc.).

In spite of these and other open issues, the authors general opinion is that the BEM already now offers an excellent platform for successful simulation of 3D real-world industrial problems. Especially when speaking about some of the major requirements appearing in the Simulation-Based Design nowadays, like:

- *assembly* instead of *component simulation*,
- simulation for the *daily* design process,
- *user-friendly* simulation, but still preserving the *full geometrical and physical complexity*,

BEM-based numerical technologies seems to fulfill the majority of the requirements needed today for efficient design of the industrial products.

References

[1] Z. Andjelić, P. Kripner, and A. Vogel. Simulation based design of O2 MEMS sensor. *Fifth Inter. Conf. on Modeling and Simulation of Microsystems*, 2002.

[2] Z. Andjelić, B. Krstajić, S. Milojković, A. Blaszczyk, H. Steinbigler, and M. Wohlmuth. *Integral Methods for the Calculation of Electric Fields*, volume 10. Scientific Series of the Internaltional Bureu, Forschungszentrum Jülich GmbH, 1992.

[3] Z. Andjelić and P. Marchukov. Acceleration of the electrostatic computation using multipole technique. Technical report, ABB Corporate Research, Heidelberg, 1992.

[4] Z. Andjelić, J. Smajić, and M. Conry. *Boundary Integral Analysis: Mathematical Aspects and Applications*, chapter BEM-based Simulations in Engineering Design. Springer-Verlag, 2007.

[5] M. Bebendorf and S. Rjasanow. Adaptive low rank approximation of collocation matrices. *Computing*, 70:1–24, 2003.

[6] M. Bebendorf, S. Rjasanow, and E.E. Tytyshnikov. Approaximations using diagonal-plus skeleton matrices. *Chapman & Hall/CRC Research Notes in Mathematics*, 414:45–53, 1999.

[7] A. Blaszczyk, Z. Andjelić, P. Levin, and A. Ustundag. *Lecture Notes on Computer Science*, chapter Parallel Computation of Electric Fields in a Heterogeneous Workstation Cluster, pages 606–611. Springer Verlag Berlin Heidelberg, hpcn europe 95 edition, 1995.

[8] H. Boehme. *Mittelspannungstechnik.* Verlag Technik GmbH Berlin-München, 1972.

[9] L. Gaul, M. Kögl, and M. Wagner. *Boundary Element Methods for Engineers and Scientists.* Springer-Verlag Berlin, 2003.

[10] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[11] W. Hackbusch. The panel clustering technique for the boundary element method. *9th Int. Conf. on BEM*, pages 463–473, 1987.

[12] W. Hackbusch and Z.P. Nowak. On the fast matrix multiplication in the boundary element method by panel clustering. *Numer. Math.*, 54:463–491, 1989.

[13] F. Henrotte and K. Hameyer. Computation of electromagnetic force densities: Maxwell stress tensor vs. virtual work principle. *J. Comput. Appl. Math.*, 168(1-2):235–243, 2004.

[14] L.R. Hill and T.N. Farris. Three-dimensional piezoelectric boundary element method. *AIAA Journal*, 36(1), January 1998.

[15] E.C. Koleciskij. Rascet eletriceskih poljei ustroistv visokog naprezenija. *Energoatomizdat*, 1983.

[16] A. Kost. *Numerische Methoden in der Berechnung Elektromagnetischer Felder*. Springer Verlag, 1994.

[17] G. Schmidlin. Fast Solution Algorithms for Integral Equations in $\mathbb{R}^3$. Master's thesis, ETH Zurich, 2003.

[18] G. Schmidlin, U. Fischer, Z. Andjelić, and C. Schwab. Preconditioning of the second-kind boundary integral equations for 3D eddy current problems. *Internat. J. Numer. Methods Engrg.*, 51:1009–1031, 2001.

[19] O.B. Tozoni and I.D. Maergoiz. *Rascet Trehmernih Elektromagnetnih Polei.* Tehnika, Kiev, 1974.

[20] J.R. Whiteman and L. Demkowicz (eds.). Proceedings of the Eleventh Conference on The Mathematics of Finite Elements and Applications. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5), 2005.

[21] J. Yuan and A. Kost. A three-component boundary element algorithm for three-dimensional eddy current calculation. *IEEE Tran. on Mag.*, 30(5), September 1994.

A Domain Decomposition/Nash Equilibrium Methodology for the Solution of Direct and Inverse Problems in Fluid Dynamics with Evolutionary Algorithms

Hong Quan Chen[1], Roland Glowinski[2] and Jacques Périaux[3]

[1] Nanjing University of Aeronautics and Astronautics, Department of Aerodynamics, Nanjing 210016, P.R. China, hqchenam@nuaa.edu.cn

[2] University of Houston, Dept. of Math., Houston, Texas 77004-3308, USA, Roland@math.uh.edu

[3] University of Jyväskylä, Dept. of Mathematical Information Technologies, Jyväskylä, Finland, jperiaux@gmail.com

Summary. The main goal of this paper is to present the application of a decentralization optimization principle from Game Theory to the solution of direct and inverse problems in Fluid Dynamics. It is shown in particular that multicriteria optimization methods "à la Nash" combine ideally with domain decomposition methods, with or without overlapping in order to solve complex problems. The resulting methodology is flexible and in the case of design problems has shown to perform well when using adjoint based techniques or evolutionary algorithms for the optimization.

The above methodology is applied to the simulation and shape design optimization for flows in nozzles and around aerodynamical shapes.The results of various numerical experiments show the efficiency of the method presented here.

1 Introduction

In this paper we introduce a new methodology to solve inverse problems in Fluid Dynamics using Genetic Algorithms and Game Theory. This methodology amounts to finding (suitable) Nash points for "local inverse problems". These Nash points are approximated by Genetic Algorithms (GAs) suitably constructed. This is an example of a completely general method, presented in [7] and [4]. GAs are different from traditional optimization tools and based on digital imitation of biological evolution. Game Theory replaces here a global optimization problem by a non-cooperative game based on Nash equilibrium with several players solving local constrained sub-optimization tasks. The main idea developed here is to consider two Nash applications of Game Theory under conflict introduced in a flow analysis solver (1) and a GAs optimizer (2) as follows:

(1) a flow analysis solver modeled by the potential equations uses overlapping domain decomposition methods (DDM). A variant of the classical DDM Schwarz method is considered with optimal control/GAs techniques. It uses the distance of local solutions on the overlapping regions as global fitness function described in a previous paper with GAs [11]. Then a Nash/GAs game whose decentralized players are in charge of the matching of local solutions as multi fitness functions is associated to the global problem. During the evolution process the search space of each genetic point at the interfaces of overlapping domain is implemented on adapted interval. This new approach is shown to request less information for convergence than the global one.

(2) the above DDM flow solver is then used to feed a Nash/GAs optimizer for the surface pressure reconstruction of nozzle shapes parameterized with local Bézier's splines. During this Nash iteration, the information exchange between DDM flow solver is nested to the shape-GAs optimizer.

Numerical experiments presented on inverse problems of a nozzle with Laplace's solver illustrate both the efficiency and robustness of decentralized optimization strategies. The promising inherent parallel properties of Nash games implemented with GAs on distributed computers and their possible further extensions to non-linear flows are also discussed.

2 Nash and GAs

2.1 Generalities

Many multi objective optimization problems are still not solved perfectly and some are found to be difficult to solve using traditional weighted objective techniques [17, 6]. GAs have been shown to be both global and robust over a broad spectrum of problems. Shaffer was the first to propose a genetic algorithm approach for multi objectives through his Vector Evaluated Genetic Algorithms (VEGA [15]), but it was biased towards the extreme of each objective. Goldberg proposed a solution to this particular problem with both non dominance Pareto-ranking and sharing, in order to distribute the solutions over the entire Pareto front [5]. This cooperative approach was further developed in [16], and lead to many applications [14]. All of these approaches are based on Pareto ranking and use either sharing or mating restrictions to ensure diversity; a good overview can be found in [3]. Another non cooperative approach with the notion of player has been introduced by J. Nash [10] in the early 50' for multi objective optimization problems originating from Game Theory and Economics. The following section is devoted to an original non cooperative multi objective algorithm, which is based on Nash equilibria.

2.2 Definition of a Nash Equilibrium

For an optimization problem with G objectives, a Nash strategy consists in having G players, each optimizing his own criterion. However, each player has to optimize his criterion given that all the other criteria are fixed by the rest of the players. When no player can further improve his criterion, it means that the system has reached a state of equilibrium called *Nash Equilibrium.* Let E be the search space for the first criterion and F the search space for the second criterion. A strategy pair $(\overline{x}, \overline{y}) \in E \times F$ is said to be a Nash equilibrium iff:

$$\begin{aligned} f_E(\overline{x}, \overline{y}) &= \inf_{x \in E} f_E(x, \overline{y}) \\ f_F(\overline{x}, \overline{y}) &= \inf_{y \in F} f_F(\overline{x}, y) \end{aligned}$$

It may also be defined by:
$u = (u_1, \ldots, u_G)$ is a Nash equilibrium iff: $\forall i, \forall v_i$

$$J_i(u_1, .., u_{i-1}, u_i, u_{i+1}, ..u_G) \leq J_i(u_1, .., u_{i-1}, v_i, u_{i+1}, ..u_G)$$

It may be difficult to exhibit such an equilibrium in particular for non differentiable problems.

2.3 Description of a Nash/GAs

The following stage consists in merging GAs and Nash strategy in order to make the genetic algorithm *build* the Nash Equilibrium for a complete description (see [13, 18]).

Let $s = XY$ be the string representing the potential solution for a dual objective optimization, where X corresponds to the first criterion and Y to the second one. The first idea is to assign the optimization task of X to a player called *Player 1* and the optimization task of Y to *Player 2.* Thus, as advocated by Nash theory, *Player 1* optimizes s with respect to the first criterion by modifying X, while Y is fixed by *Player 2.* Symmetrically, *Player 2* optimizes s with respect to the second criterion by modifying Y while X is fixed by *Player 1* (see [13] for details).

The next step consists in creating two different populations, one for each player. *Player 1*'s optimization task is performed by population 1 whereas *Player 2*'s optimization task is performed by population 2.

Let X_{k-1} be the best value found by *Player 1* at generation $k-1$, and Y_{k-1} the best value found by *Player 2* at generation $k-1$. At generation k, *Player 1* optimizes X_k while using Y_{k-1} in order to evaluate s (in this case, $s = X_k Y_{k-1}$). At the same time, *Player 2* optimizes Y_k while using X_{k-1} ($s = X_{k-1} Y_k$). After the optimization process, *Player 1* sends the best value X_k to *Player 2* who will use it at generation $k+1$. Similarly, *Player 2* sends the best value Y_k to *Player 1* who will use it at generation $k+1$. Nash equilibrium is reached when neither *Player 1* nor *Player 2* can further improve their criteria.

This setting may seem to be similar to that of Island Models in Parallel Genetic Algorithms (PGA [9]). However, there is a fundamental difference

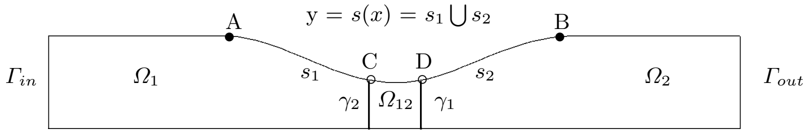


Fig. 1. Description of a nozzle with two subdomains

which lays in the notion of equilibrium for Nash approach. Nash equilibria do not correspond only to robust convergence, but have also very good stability properties compared to cooperative strategies. The mechanisms of the Nash-GAs described here are directly used in the following sections.

3 An Implementation of Nash/GAs Game for the DDM Flow Problem

3.1 Description of the DDM Flow Problem

The DDM optimization problem considered here concerns an incompressible potential flow in a nozzle modeled by the Laplace equation with Dirichlet boundary conditions at the entrance and exit and homogeneous Neumann conditions on the walls. As shown in Fig. 1, the computational domain Ω is decomposed into two subdomains Ω_1 and Ω_2 with overlapping Ω_{12} whose interfaces are denoted by γ_1 and γ_2. We shall prescribe potential values, g_1 on γ_1 and g_2 on γ_2, as extra Dirichlet boundary conditions in order to obtain potential solutions Φ in each subdomain. Using domain decomposition techniques, the problem of the flow can be reduced to minimize the following functional [2]:

$$JF(g_1, g_2) = \frac{1}{2} \parallel \Phi_1(g_1) - \Phi_2(g_2) \parallel^2 \tag{1}$$

where Φ_1 and Φ_2 are the solutions in the overlapping subdomain Ω_{12}, $\parallel \cdot \parallel$ denotes an appropriate norm, whose choice will be made precise in the examples which follow.

For the minimization problem (1), we have presented a variant of the classical DDM Schwarz method with optimal control/GAs techniques [11] and have made a further extension with genetic treatment at the interface of the subdomains (for details, see [12]). In the following sections, an implementation of Nash/GAs with decentralized players will be addressed.

3.2 Decentralized Multi-Fitness Functions

As mentioned above, in the previous work of references [11, 12], the global fitness function used in GAs is the distance of local solutions on the overlapping domains (see (1)), which could be

$$JF(g_1, g_2) = \frac{1}{2}\int_{\Omega_{12}} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\Omega\,. \tag{2}$$

In this paper, we use boundary integrals instead of the domain integral and we choose for (2) the criteria introduced in (3). The minimization problem (1) can be reduced to minimize the following function based on boundary integral:

$$JFB(g_1, g_2) = \frac{1}{2}\int_{\gamma_1} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\gamma_1 + \frac{1}{2}\int_{\gamma_2} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\gamma_2. \tag{3}$$

Being associated to the global fitness function $JFB(g_1, g_2)$, the decentralized multi fitness functions $JFB_1(g_1, \underline{g_2})$ and $JFB_2(\underline{g_1}, g_2)$ are defined with the following two minimizations:

$$\begin{aligned} &\inf_{g_1} JFB_1(g_1, \underline{g_2}) \quad \text{with} \quad JFB_1(g_1, \underline{g_2}) = \frac{1}{2}\int_{\gamma_1} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\gamma_1, \\ &\inf_{g_2} JFB_2(\underline{g_1}, g_2) \quad \text{with} \quad JFB_2(\underline{g_1}, g_2) = \frac{1}{2}\int_{\gamma_2} |\Phi_1(g_1) - \Phi_2(g_2)|^2 d\gamma_2. \end{aligned} \tag{4}$$

The inf of the functionals (2) or (3) is zero. Therefore if in searching for a Nash equilibrium (4) we find one such that $\inf_{g_1} = 0$ and $\inf_{g_2} = 0$ then it is the solution of inf (3). There could be other Nash points which would not solve the problem if $\inf_{g_1} > 0$ for instance. The global DDM solution can be found through searching a Nash equilibrium between the above two minimizations based on the treatments described in the next sections.

3.3 An Implementation of Nash/GAs Game

Following the description of section 2.3, we can simulate the DDM flow optimization problem as a Nash game with two decentralized players, *Flow-GA1* and *Flow-GA2* in charge of objective functions $JFB_1(g_1, \underline{g_2})$ and $JFB_2(\underline{g_1}, g_2)$, respectively. Note that each player optimizes the corresponding objective function with respect to non-underlined variables. After discretization of the problem, we have $g_1 = g_{1i}$ and $g_2 = g_{2i}$, $i = 1, ny$ (ny is mesh size in y direction). Following the genetic treatment at the interface of reference [12], for each interface, one point is binary encoded (for instance, g_{11} for γ_1 and g_{21} for γ_2). Other values of g_{1i} and g_{2i} ($i \geq 2$) are corrected by numerical values (for details, see [12]). The whole structure of the implementation based on the information exchange between players is described as follows:

Step 0: (Initialization) Given initial interval (g_{min}, g_{max}) as search space for two genetic points, g_{11} and g_{21}, and then start with two set of randomly created genetic points to form two initial populations for each players, *Flow-GA1* and *Flow-GA2*.

Step 1: *Flow-GA1* and *Flow-GA2* run separately until the iteration number equals the exchange frequency number.

Step 2: Exchange current the fittest flow information between *Flow-GA1* and *Flow-GA2*.

Step 3: Repeat the Step 1 to Step 2 until no player can further improve his fitness.

It should be noted that *Flow-GA1* operates for the left part and *Flow-GA2* for the right part of the nozzle. In fact, we have prescribed $\delta^0 = \frac{1}{2}(g_{max} - g_{min})$ in the initialization step. In this paper, *Flow-GA1* updates $\underline{g_2}$ from the fittest individual of *Flow-GA2*. Besides, the search space of g_{11} is adapted with:

$$(\underline{g_{21}} - \delta^n, \underline{g_{21}} + \delta^n)$$

where $\underline{g_{21}}$ is the first component of vector $\underline{g_2}$ updated by other player, *Flow-GA2*, through Nash-exchange and $\delta^n = fa\delta^{n-1}$, where $fa < 1$. In other words, δ^n is adapted and gradually approached to a small value with the Nash generation, which can ensure accuracy similar to a real value encoding. Numerical experiments have shown that this treatment is helpful for the present method to have the Nash equilibrium. In the meantime *Flow-GA2* player is doing the same as *Flow-GA1* player.

The significant extent of parallelism properties gained from the above method has further improvement compared with previous work of reference [11] or other flow solvers using Domain Decomposition techniques. This DDM flow solver will be used to feed a Nash/GAs shape optimizer described in the following section.

4 Shape Optimization Problem Using Nash/GAs with DDM

The DDM shape optimization problem considered here involves the inverse problem of a nozzle using a reconstruction technique and domain decomposition method using Nash/GAs. For the inverse problem, the global shape optimization is to find a shape (denoted, $y = s(x)$, $x \in [A, B]$, see Fig. 1) of a nozzle which realizes a prescribed pressure distribution on its boundary for a given flow condition. This problem has the following formulation:

$$\inf_s JS(s) \quad \text{with} \quad JS(s) = \frac{1}{2}\int_{[AB]} |p_s - p_t|^2 ds \tag{5}$$

where p_t is a given target pressure and p_s the actual flow pressure on the shape s. Let $s_1(x), x \in [A, D]$ and $s_2(x), x \in [C, B]$ be the split shapes, then if $s(x) = s_1(x) \bigcup s_2(x)$, we consider the two following local optimization problems:

$$\begin{aligned}
&\inf_{s_1} JS_1(s_1, \underline{s_2}) \quad \text{with} \quad JS_1(s_1, \underline{s_2}) = \frac{1}{2}\int_{[AD]} |p_{s_1} - p_t|^2 ds_1 \\
&\inf_{s_2} JS_2(\underline{s_1}, s_2) \quad \text{with} \quad JS_2(\underline{s_1}, s_2) = \frac{1}{2}\int_{[CB]} |p_{s_2} - p_t|^2 ds_2
\end{aligned} \tag{6}$$

with the constraint that $s_1 = s_2$ on interval C, D. Then $\inf JS_1 = 0$ on s_1 and $\inf JS_2 = 0$ on s_2 is the solution of (6) considered in the sequel. The global shape optimization solution can be found through searching a Nash equilibrium between the above two minimizations. The DDM flow problem described in the section 3 will provide information to the shape optimization problem using Nash strategy.

4.1 Parameterization of the Shape of the Nozzle

Using GAs, the candidate shapes of the inverse problem mentioned above are represented by a Bézier curve of order n, which reads [1]:

$$x(t) = \sum_{i=0}^{n} c_n^i t^i (1-t)^{n-i} x_i, \quad y(t) = \sum_{i=0}^{n} c_n^i t^i (1-t)^{n-i} y_i$$

where $c_n^i = \frac{n!}{i!(n-i)!}$ and (x_i, y_i) are control points of the curve, t is the parameter whose values vary between [0,1]. To limit the size of the search space, we vary the control points only in the y direction with fixed x_i values. $JS(s)$ is used as fitness function and real coding is used for y_i, which forms a string denoted $\{y_0 y_1 y_2 ... y_{n-1} y_n\}$. One site uniform crossover and non-uniform mutation are used in the present work (for details, see the work of Michalewicz [8]). The treatment of continuity between two split shapes mentioned above will be described in the next section.

4.2 Solution Method and Its Implementation

Following the description of section 2.3, we can now play a practical game of this DDM-shape optimization problem with two players, *Shape-GA1* and *Shape-GA2* in charge of objective functions $JS_1(s_1, \underline{s_2})$ and $JS_2(\underline{s_1}, s_2)$, respectively. With DDM, *Shape-GA1* has a follower *Flow-GA1* with objective function $JF_1(g_1, \underline{g_2})$ and *Shape-GA2* has another follower *Flow-GA2* with objective function $JF_2(\underline{g_1}, g_2)$. Note that each player or follower optimizes the corresponding objective function with respect to non-underlined variables. The whole structure of the implementation based on the information exchange between players is described as follows:

Step 0: (Initialization) Start with a randomly created shape $s(x), x \in [A, B]$ and split it into two curves s_1 *and* s_2 as starting curves for *Shape-GA1* and *Shape-GA2*

Step 1: *Shape-GA1* and *Shape-GA2* run separately until the iteration number equals the exchange frequency number.

Step 2: Exchange current the fittest shape information between *Shape-GA1* and *Shape-GA2*.

Step 3: Repeat the Step 1 to Step 2 until no player can further improve his fitness.

It is noted that *Shape-GA1* operates for the left part and *Shape-GA2* for the right part of the nozzle. In this paper, *Shape-GA1* receives the y coordinate value and slope of the point D from the fittest curve $\underline{s_2}$ of *Shape-GA2*. This value will be used for the end control point of the Bézier curve of s_1 in *Shape-GA1* for the next step. This treatment ensures continuity and is expected to have smoothness at the overlapping segment $\widehat{CD}$. *Shape-GA2* does the same as *Shape-GA1* meanwhile.

The calculation of each shape fitness requires to solve the flow equations by CFD solvers over the whole domain. Combining DDM with the local geometrical optimization, the flow field can be solved separately by two followers *Flow-GA1* and *Flow-GA2* in each subdomain. The *flow-GAs* returns the current fittest flow solution to *Shape-GAs* for computing fitness of each shape and the information exchange between two followers happens during the exchange between the shape players. We are satisfied when each local problem gives "zero" (very small) for the local criteria.

The problem (6) with *Shape-GA1* and *Shape-GA2* is a Nash problem solved with a floating point coded GA, whereas the problem (4) with *Flow-GA1* and *Flow-GA2* is solved with a binary coded Nash GA. Problems (4) and (6) are coupled since a precise solution of the DDM flow solver via (4) is necessary to evaluate candidate solutions of optimization problem via (6).

5 Results and Analysis

With the method presented above, we have tested both the DDM flow problem and the nozzle reconstruction problem, respectively. Exchange frequency for Nash/GAs is 1. The potential values are predicted by a finite element Laplace's solver based on a direct Choleski method. The probability of crossover $Pc = 0.85$ and the probability of mutation $Pm = 0.09$ are not carefully selected but are fixed for *Flow-GAs*. The parameters used in *Shape-GAs* are 0.6 for crossover rate and 0.108 for mutation rate.

We first present the preliminary results of the DDM flow problem with the Nash/GAs game described in the the section 3. The convergence histories of the fittest individual are shown in Fig. 2. Following the trace of the domain integral of the current fittest values of $JF(\underline{g_1}, \underline{g_2})$, we find that the value of the domain integral JF has been reduced from 1.2E-2 to 1.6E-7, which confirms that the present Nash/GAs method works well for the test case.

The numerical results of the method described in the section 4 tested for a nozzle reconstruction problem are presented in Figs. 3-5. As the pressure distribution Cp matches the target, the corresponding nozzle shape is reconstructed successfully (see Fig. 5).

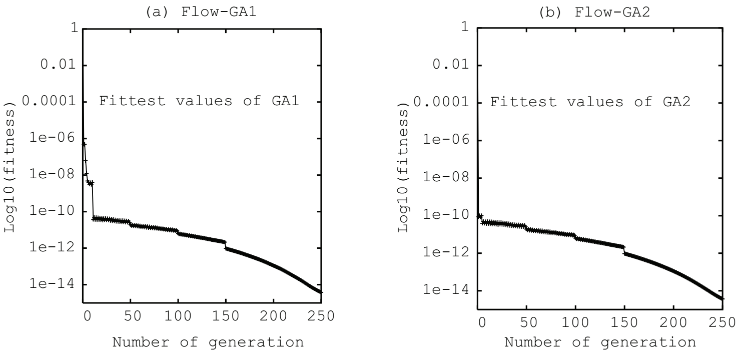


Fig. 2. Convergence histories: (a) Flow-GA1 and (b) Flow-GA2

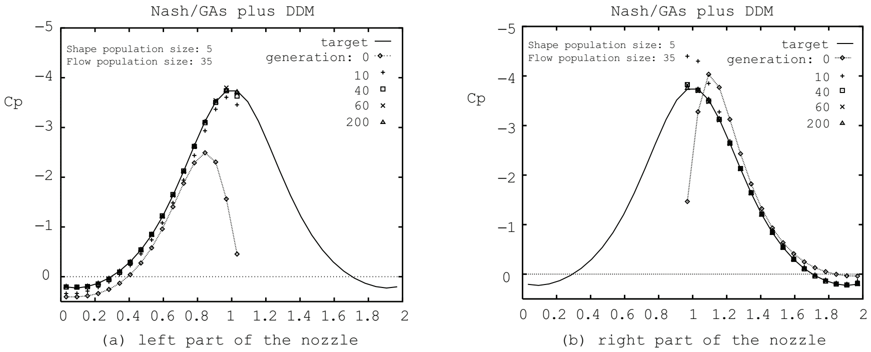


Fig. 3. Successive Cp distributions: (a) left part and (b) right part

6 Conclusion and Possible Extensions

From the experiments described in this paper, it is clear that GAs and DDM may provide robust tools to solve complex distributed optimization problems. It is shown that one can decompose a "global" cost function into a sum of "local" cost functions and under circumstances it is sufficient to look for Nash equilibrium points (or special Nash points). The multi objective techniques with decentralized players discussed here demonstrate convincingly that combining ideas from Economics or Game Theory with GAs may lead to powerful distributed optimization methods for Engineering problems. A significant saving in the above process in terms of elapsed time in a distributed parallel networked environment is anticipated by replacing expensive global commu-

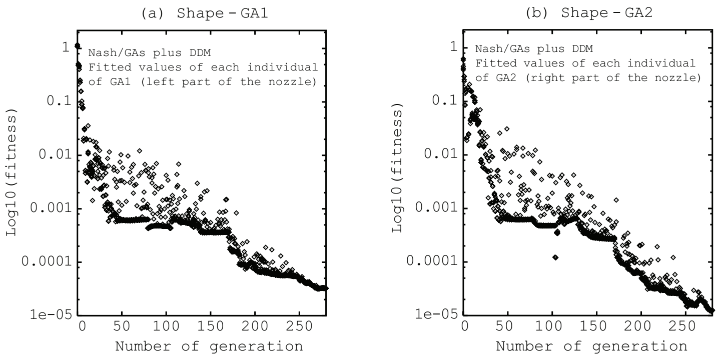


Fig. 4. Convergence histories: (a) Shape-GA1 and (b) Shape-GA2

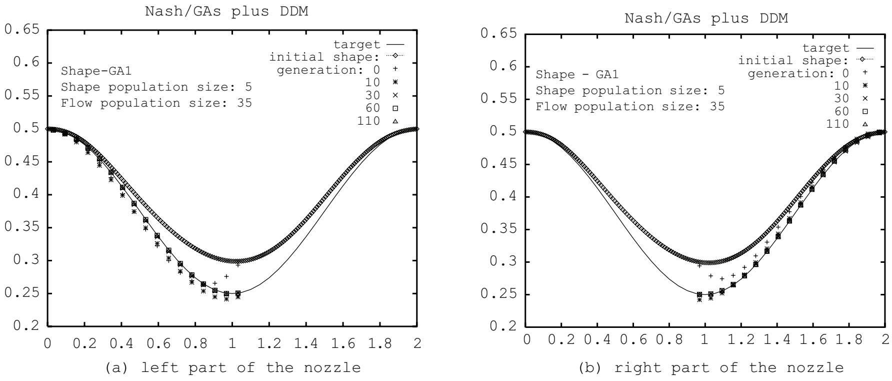


Fig. 5. Successive shapes: (a) left part and (b) right part

nication (standard strong collective optimality) by local communication (non standard weaker individual optimality).

The preliminary results presented above should be checked on many subdomains in dimension 3 and extended to non linear flow situations. Very many other problems can be considered by related methods. Some of them are indicated in the CRAS note by the authors [7] and several papers are in preparation.

Acknowledgement. The authors would like to thank M. Sefrioui, B. Mantel and H. T. Sui for fruitful discussions and providing suggestions and software about Game Theory and GAs for non cooperative optimization strategies, finite element and Bézier spline-parameterization of shapes.

References

[1] J.-A. Desideri. Hierarchical optimum-shape algorithms using embedded Bezier parametrization. In E. Heikkola, Y. Kuznetsov, P. Neittaanmaki, and O. Pironneau, editors, *Numerical Methods for Scientific Computing, Variational Problems and Applications*, pages 45–56. CIMNE Barcelona Publisher, 2003.

[2] Q.V. Dinh, R. Glowinski, J. Périaux, and G. Terrasson. On the coupling of viscous and inviscid models for incompressible fluid flows via domain decomposition. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987)*, pages 350–369. SIAM, Philadelphia, PA, 1988.

[3] C.M. Fonseca and P.J. Fleming. An overview of evolutionary algorithms in multiobjective optimization. *Evolutionary Computation*, 3:1–16, 1995.

[4] R. Glowinski, J.-L. Lions, and O. Pironneau. Decomposition of energy spaces and applications. *C. R. Acad. Sci. Paris Sér. I Math.*, 329(5):445–452, 1999.

[5] D. Goldberg. *Genetic Algorithms in Search, Optimization, and Machine Learning.* Addison-Wesley, Reading, Mass, 1989.

[6] T. E. Labrujere. Single and two-point airfoil design using Euler and Navier-Stokes equations. Test case TE2 and TE4. In J. Périaux, G. Bugeda, P.K. Chaviaropoulos, K. Giannakoglou, S. Lanteri, and B. Mantel, editors, *Optimum Aerodynamic Design & Parallel Navier-Stokes Computations, ECARP–European Computational Aerodynamics Research Project*, volume 61 of *Notes on Numerical Fluid Mechanics*, pages 214–230. Vieweg, 1998.

[7] J.L. Lions. Private communication, 1999.

[8] Z. Michalewicz. *Genetic Algorithms + Data Structures = Evolution Programs.* Artificial Intelligence. Springer-Verlag, Berlin, 1992.

[9] H. Muhlenbein, M. Schomisch, and J. Born. The parallel genetic algorithm as function optimizer. *Parallel Computing*, 17:619–632, 1991.

[10] J. Nash. Non-cooperative games. *Ann. of Math. (2)*, 54:286–295, 1951.

[11] J. Périaux and H.Q. Chen. Domain decomposition method using gas for solving transonic aerodynamic problems. In R. Glowinski, J. Périaux, Shi Z.-C., and O.B. Widlund, editors, *Domain Decomposition Methods in Sciences and Engineering. Proceedings of DDM-8, Beijing*, pages 427–431. John Wiley, 1995.

[12] J. Périaux, B. Mantel, and H.Q. Chen. Intelligent interfaces of a Schwarz domain decomposition method via genetic algorithms for solving nonlinear PDEs. Application to transonic flows simulations. In P. Bjørstad et al., editor, *Proceedings of DD9.* John Wiley & Sons Ltd, 1996.

[13] J. Périaux and M. Sefrioui. Genetic algorithms, game theory and hierarchical models: application to CFD and CEM problems. In J. Périaux, G. Degrez, and H. Deconinck, editors, *Genetic Algorithms for*

Optimization in Aeronautics and Turbomachinery, Lecture Series 2000-07, pages 1–44. Von Karman Institute for Fluid Dynamics, 2000.

[14] C. Poloni. Hybrid QA for multi objective aerodynamic shape optimization. In G. Winter, J. Périaux, and P. Cuesta, editors, *Genetic Algorithms in Engineering and Computer Science*, pages 397–415. John Wiley, 1995.

[15] J.D. Schaffer. Multiple objective optimization with vector evaluated genetic algorithms. In J. J. Grefenstette, editor, *Proceedings of the 1st International Conference on Genetic Algorithms*, pages 93–100, Carnegie-Mellon, Pittsburgh, 1985. Lawrence Erlbaum Associates.

[16] N. Srinivas and K. Deb. Multiobjective optimization using nondominated sorting in genetic algorithms. *Evolutionary Computation*, 2:221–248, 1995.

[17] W. Stadler. Fundamentals of multicriteria optimization. In *Multicriteria optimization in engineering and in the sciences*, volume 37 of *Math. Concepts Methods Sci. Engrg.*, pages 1–25. Plenum, New York, 1988.

[18] J.F. Wang and J. Périaux. Search space decomposition of Nash/Stackelberg games using genetic algorithms for multi-point design optimization in aerodynamics. In N. Debit, M. Garbey, R. Hoppe, D. Keyes, Y. Kuznetsov, and J. Périaux, editors, *Proceedings of the 13th Domain Decomposition Methods in Science and Engineering*, pages 139–150. CIMNE Barcelona Publisher, 2002.

Preconditioning of Symmetric Interior Penalty Discontinuous Galerkin FEM for Elliptic Problems

Veselin A. Dobrev[1], Raytcho D. Lazarov[1], and Ludmil T. Zikatanov[2]

[1] Department of Mathematics, Texas A&M University, {dobrev,lazarov}@math.tamu.edu
[2] Department of Mathematics, Penn State University, ludmil@psu.edu

Summary. This is a further development of [9] regarding multilevel preconditioning for symmetric interior penalty discontinuous Galerkin finite element approximations of second order elliptic problems. We assume that the mesh on the finest level is a results of a geometrically refined fixed coarse mesh. The preconditioner is a multilevel method that uses a sequence of finite element spaces of either continuous or piece-wise constant functions. The spaces are nested, but due to the penalty term in the DG method the corresponding forms are not inherited. For the continuous finite element spaces we show that the variable V-cycle provides an optimal preconditioner for the DG system. The piece-wise constant functions do not have approximation property so in order to control the energy growth of the inter-level transfer operator we apply W–cycle MG. Finally, we present a number of numerical experiments that support the theoretical findings.

1 Introduction

Consider the following model second order elliptic problem on a bounded domain with a polygonal boundary $\Omega \subset R^d$, $d = 2, 3$:

$$-\nabla \cdot (a(x)\nabla u) = f(x) \quad \text{in } \Omega, \quad u(x) = g \quad \text{on } \partial\Omega. \tag{1}$$

Here a is a uniformly positive in Ω and piece-wise $W^1_\infty(\Omega)$-function that may have jumps along some interfaces. The theoretical results can be easily extended to a coefficient matrix a and more general boundary conditions.

Our goal is to study iterative methods for a symmetric interior penalty discontinuous Galerkin finite element approximations of (1) over a partition $\mathcal{T}$ of Ω into finite elements denoted by $\mathcal{K}$. We assume that the partition is quasi uniform and regular. For a finite element $\mathcal{K}$ we denote by $h_\mathcal{K}$ its size and $h = \max_{\mathcal{K}\in\mathcal{T}} h_\mathcal{K}$. Further, we use the following notations concerning $\mathcal{T}$: $\mathcal{E}^0$ is the set of all interior edges/faces, $\mathcal{E}^b$ is the set of the edges/faces on the boundary $\partial\Omega$ and $\mathcal{E} = \mathcal{E}^0 \cup \mathcal{E}^b$. In fact, $\mathcal{T}$, $\mathcal{E}$, etc are sets depending on the mesh-size h. However, in order to avoid proliferation of indices and since we are dealing exclusively with algebraic problems we shall not explicitly denote this dependence on the mesh-size. We also use a hierarchy of meshes $\mathcal{T}_1 \subset \cdots \subset \mathcal{T}_J$ which are obtained by geometric refinement of a coarse mesh $\mathcal{T}_1$. Thus, $\mathcal{T} = \mathcal{T}_J$ and $\mathcal{T}_k$ is the mesh generated after $k-1$ levels of refinement of $\mathcal{T}_1$. When the index k, showing the dependence on the refinement level, is suppressed this means that the quantities are defined on the finest level.

We introduce the spaces

$$H^s(\mathcal{T}) = \left\{ v \in L^2(\Omega) : v|_{\mathcal{K}} \in H^s(\mathcal{K}),\ \forall \mathcal{K} \in \mathcal{T} \right\},\ \text{for } s \geq 0 \tag{2}$$

and for $r \geq 0$ integer we define the finite element space

$$\mathcal{V} := \mathcal{V}(\mathcal{T}) := \{ v \in L^2(\Omega) : v|_{\mathcal{K}} \in P_r(\mathcal{K}),\ \mathcal{K} \in \mathcal{T} \}, \tag{3}$$

where P_r is the set of polynomials of total degree at most r restricted to $\mathcal{K}$. On $\mathcal{V}$ we define the bilinear forms

$$(a\nabla u, \nabla v)_{\mathcal{T}} := \sum_{\mathcal{K} \in \mathcal{T}} \int_{\mathcal{K}} a\nabla u, \nabla v \; dx, \quad \langle p, q \rangle_{\mathcal{E}} := \sum_{\mathrm{e} \in \mathcal{E}} \int_{\mathrm{e}} pq \; ds.$$

On $\mathrm{e} = \bar{\mathcal{K}}_1 \cap \bar{\mathcal{K}}_2 \in \mathcal{E}$ we define the jump of a scalar function $v \in \mathcal{V}$ by

$$[\![v]\!]_{\mathrm{e}} := \begin{cases} v|_{\mathcal{K}_1} \mathbf{n}_{\mathcal{K}_1} + v|_{\mathcal{K}_2} \mathbf{n}_{\mathcal{K}_2}, & \mathrm{e} = \bar{\mathcal{K}}_1 \cap \bar{\mathcal{K}}_2, \text{ i.e. } \mathrm{e} \in \mathcal{E}^0, \\ v|_{\mathcal{K}} \mathbf{n}_{\mathcal{K}}, & \mathrm{e} = \bar{\mathcal{K}} \cap \partial\Omega, \text{ i.e. } \mathrm{e} \in \mathcal{E}^b \end{cases}$$

and the average value of the traces of $a\nabla v$ for $v \in \mathcal{V}$:

$$\{\{ a\nabla v \}\}\,|_{\mathrm{e}} := \begin{cases} \frac{1}{2}\{ a\nabla v|_{\mathcal{K}_1} + a\nabla v|_{\mathcal{K}_2} \}, & \mathrm{e} = \bar{\mathcal{K}}_1 \cap \bar{\mathcal{K}}_2, \text{ i.e. } \mathrm{e} \in \mathcal{E}^0, \\ a\nabla v|_{\mathcal{K}}, & \mathrm{e} = \bar{\mathcal{K}} \cap \partial\Omega, \text{ i.e. } \mathrm{e} \in \mathcal{E}^b. \end{cases}$$

Here $\mathbf{n}_{\mathcal{K}}$ is the external unit vector normal to the boundary $\partial\mathcal{K}$ of $\mathcal{K} \in \mathcal{T}$.

Next, we define the piecewise constant function $h_{\mathcal{E}}$ on $\mathcal{E}$ as

$$h_{\mathcal{E}} = h_{\mathcal{E}}(x) = |\mathrm{e}|^{\frac{1}{d-1}}, \quad \text{for } x \in \mathrm{e}, \quad \mathrm{e} \in \mathcal{E}, d = 2, 3. \tag{4}$$

And finally, we introduce the following mesh-dependent norm on $\mathcal{V}$:

$$|\!|\!| v |\!|\!|^2 = (a\nabla v, \nabla v)_{\mathcal{T}} + \left\langle h_{\mathcal{E}}^{-1} \kappa_{\mathcal{E}} [\![v]\!] , [\![v]\!] \right\rangle_{\mathcal{E}}. \tag{5}$$

The stabilization factor $\kappa_{\mathcal{E}}$ is weighted by the coefficient a, namely, $\kappa_{\mathcal{E}} = \kappa \{\{a\}\}$, where $\{\{a\}\}$ is the average value of a from both sides of $\mathrm{e} \in \mathcal{E}$. This choice of the penalty gives rise to a DG bilinear form (7) that is equivalent to the norm (5) with constants independent of the jumps of a.

We consider the following symmetric interior penalty discontinuous Galerkin (SIPG) finite element approximation of (1) (see, e.g. [1, 2]):

$$\text{find } \ u_h \in \mathcal{V} \ \text{ such that } \ \mathcal{A}(u_h, v) = \mathcal{L}(v) \ \ \forall v \in \mathcal{V}, \tag{6}$$

where $\mathcal{V}$ is the finite element space and $\mathcal{A}(\cdot,\cdot)$, $\mathcal{L}(\cdot)$ are bilinear and linear forms on $\mathcal{V}$ defined by

$$\begin{aligned}\mathcal{A}(u_h, v) \equiv (a\nabla u_h, \nabla v)_{\mathcal{T}} &- \langle \{\{a\nabla u_h\}\}\,,\, [\![v]\!]\rangle_{\mathcal{E}} - \langle [\![u_h]\!]\,,\, \{\{a\nabla v\}\}\rangle_{\mathcal{E}} \\ &+ \left\langle h_{\mathcal{E}}^{-1}\kappa_{\mathcal{E}}\, [\![u_h]\!]\,,\, [\![v]\!]\right\rangle_{\mathcal{E}}\end{aligned} \tag{7}$$

and

$$\mathcal{L}(v) = \int_{\Omega} f v dx + \int_{\partial\Omega} (h_{\mathcal{E}}^{-1}\kappa_{\mathcal{E}} v - a\nabla v \cdot \mathbf{n})\, g\, ds. \tag{8}$$

It is known (see, e.g. [2]) that SIPG (6) – (8) is stable for sufficiently large $\kappa > 0$ and has optimal convergence in H^1-like norm (5). This is just one example of a large number of DG FEM approximations of second order elliptic problems that have been introduced and studied in the last several years (see, e.g. [2, 9]).

The aim of this paper is to introduce and study multilevel iterative methods for the corresponding algebraic problems. Note that the condition number of the DG FE system grows like $O(h^{-2})$ on a quasi uniform mesh with meshsize h. Therefore construction of optimal solution methods, i.e. with arithmetic work proportional to the numbers of unknowns, that is robust with respect to large variations of the coefficient a is an important problem from both theoretical and practical points of view.

The work of Gopalakrishnan and Kanschat [10], the first one we are aware of, studied the variable V-cycle multigrid operator as a preconditioner of the symmetric DG system. Under certain weak regularity assumptions on geometrically nested meshes it was shown in [10] that the condition number of the preconditioned system is $O(1)$, i.e. bounded independently of h. The analysis of the preconditioner is based on the abstract multigrid theory [7] for non-inherited bilinear forms and the estimates for interior penalty finite element method. Further, Brenner and Zhao [8] studied V-cycle, W-cycle, and F-cycle algorithms for the symmetric DG FE schemes on rectangular meshes and showed that they produce uniform preconditioners for sufficiently many pre- and post smoothing steps. Their analysis is based on certain mesh dependent norms and a relationship of the discontinuous FE spaces to some higher order continuous finite element spaces. Our approach is slightly different, it could be seen as the classical two-level method applied to the DG linear systems. We explore two different possibilities for a choice of the second level, namely, continuous piece-wise polynomial functions and piece-wise constant functions.

2 MG Preconditioner Using Spaces of Continuous Functions

We assume that we have a sequence of nested globally quasi-uniform triangulations $\mathcal{T}_k$, $k = 1, \ldots, J$, of the domain Ω with $\mathcal{T}_1$ being the coarsest triangulation. According to the convention from the introduction the set of all edges/faces of elements in $\mathcal{T}_k$ is denoted by $\mathcal{E}_k$, the sets of the interior and boundary edges/faces are denoted by $\mathcal{E}_k^0$ and $\mathcal{E}_k^b$, respectively, and h_k is the diameter of a typical element in $\mathcal{T}_k$ and $h_{\mathcal{E}_k}$ is defined by (4) on $\mathcal{E}_k$. Then $H^s(\mathcal{T}_k)$ and $\mathcal{V}_k$ are the spaces (2) and (3), respectively, defined on $\mathcal{T}_k$. The corresponding continuous discrete spaces are defined as $\mathcal{V}_k^c = \mathcal{V}_k \cap C(\overline{\Omega})$.

For functions u and v in $H^s(\mathcal{T}_k)$, $s > \frac{3}{2}$, we define the interior penalty (SIPG) bilinear and linear forms according to (7) for the mesh $\mathcal{T}_k$:

$$\begin{aligned}\mathcal{A}_k(u,v) = (a\nabla u, \nabla v)_{\mathcal{T}_k} + &\left\langle h_{\mathcal{E}_k}^{-1}\kappa_{\mathcal{E}} [\![u]\!], [\![v]\!]\right\rangle_{\mathcal{E}_k} \\ &- \langle \{\{a\nabla u\}\}, [\![v]\!]\rangle_{\mathcal{E}_k} - \langle \{\{a\nabla v\}\}, [\![u]\!]\rangle_{\mathcal{E}_k},\end{aligned}$$

$$\mathcal{L}_k(v) = \int_{\Omega} fv + \left\langle h_{\mathcal{E}_k}^{-1}\kappa_{\mathcal{E}} g, v\right\rangle_{\mathcal{E}_k^b} - \langle a\nabla v\cdot \mathbf{n}, g\rangle_{\mathcal{E}_k^b}.$$

With these definitions, the interior penalty discontinuous Galerkin method for the elliptic problem (1) reads: find $u_h \in \mathcal{V}_J$ such that

$$\mathcal{A}_J(u_h, v) = \mathcal{L}_J(v), \quad \forall v \in \mathcal{V}_J. \tag{9}$$

Let $|\!|\!|\cdot|\!|\!|_k$ be the norm (5) defined on the mesh $\mathcal{T}_k$. It is well known that there exists κ_0 such that for $\kappa > \kappa_0$ the following norm equivalence on $\mathcal{V}_k$ holds $\mathcal{A}_k(v,v) \simeq |\!|\!|v|\!|\!|_k^2$, $\forall v \in \mathcal{V}_k$, with constants in the norm equivalence independent of h_k, i.e. $\mathcal{A}_k(v,v)^{\frac{1}{2}}$ is a norm on $\mathcal{V}_k$.

Lemma 1. *Consider the case of homogeneous boundary condition, $g = 0$, and assume that the solution u of (1) belongs to $H^{1+\alpha}(\Omega)$ for some $\frac{1}{2} < \alpha \leq 1$. Let $u_k \in \mathcal{V}_k$ (or $\mathcal{V}_k^c$) be the solution of $\mathcal{A}_k(u_k, v) = \mathcal{L}_k(v)$, $\forall v \in \mathcal{V}_k$ ($\mathcal{V}_k^c$). Then the following error estimate holds*

$$|\!|\!|u - u_k|\!|\!|_k \leq C h_k^{\alpha} \|u\|_{1+\alpha}$$

with a constant C independent of h_k.

Sketch of the proof. To prove this estimate one can use the Galerkin orthogonality, the boundedness of $\mathcal{A}_k(\cdot,\cdot)$ in the norm $|\!|\!|u|\!|\!|_{\alpha,k} = |\!|\!|u|\!|\!|_k^2 + \sum_{\mathcal{K}\in\mathcal{T}_k} h_k^{2\alpha}|u|_{1+\alpha,\mathcal{K}}^2$ for $u \in H^{1+\alpha}(\mathcal{T}_k)$ and the approximation properties of the space $\mathcal{V}_k$. Note that in contrast to the work [10] instead of using the quantity $\mathcal{A}_k(\cdot,\cdot)^{\frac{1}{2}}$, which in general is not a norm on $H^{1+\alpha}(\mathcal{T}_k)$, we work directly in the norm $|\!|\!|u|\!|\!|_{\alpha,k}$.

Now we define the variable V-cycle MG preconditioner.

3 Variable *V*-Cycle Multigrid Preconditioner

In this Section we shall follow the general theory of multigrid methods as presented by Bramble and Zhang in [7, Chapter II, Section 7]. We will use the following sequence of nested spaces: $M_{J+1} = \mathcal{V}$, i.e. this is the space where the SIPG method is defined; for $k = 1, \dots, J$ we take $M_k = \mathcal{V}_k^c$ the continuous finite element space. The corresponding bilinear forms $\mathcal{A}_k(\cdot,\cdot)$ are defined above for $k = 1, \dots, J$; for $k = J+1$ we let $\mathcal{A}_{J+1}(u,v) = \mathcal{A}(u,v)$. Define the operators $A_k : M_k \to M_k$, $Q_k : L^2(\Omega) \to M_k$, and $P_k : M_{k+1} \to M_k$ by

$$\begin{aligned} (A_k u, v) &= \mathcal{A}_k(u,v), && \forall v \in M_k, \quad k = 1, \dots, J+1, \\ (Q_k u, v) &= (u,v), && \forall v \in M_k, \quad k = 1, \dots, J+1, \\ \mathcal{A}_k(P_k u, v) &= \mathcal{A}_{k+1}(u,v), && \forall v \in M_k, \quad k = 1, \dots, J, \end{aligned}$$

where $(\cdot,\cdot)$ denotes the inner product in $L^2(\Omega)$. Note that because of the penalty term the forms $\mathcal{A}_k(u,v)$ defined on the spaces $\mathcal{V}_k$ vary. Assume we are given the smoothing operators $R_k : M_k \to M_k$ that satisfy appropriate smoothing property (see, [7, Chapter II, Section 7, p. 260]). One can show that scaled Jacobi and Gauss-Seidel iterations satisfy this requirement.

Let B_k be the operator of the MG method based on the sequence of spaces $M_1 \subset \cdots \subset M_J \subset M_{J+1}$, with m_k pre- and post-smoothing steps with the smoother R_k. Note that to retain the symmetry of certain operators on odd steps we apply R_k, while on even steps we apply R_k^t, where the transposition is with respect to the $(\cdot,\cdot)$-inner product.

The following assumption will be used in the study of the MG method.

Assumption A.1: For any $f \in H^{-1+\rho}(\Omega)$ with $\frac{1}{2} < \rho \le 1$ and $g = 0$ the problem (1) has a unique solution $u \in H^{1+\rho}(\Omega)$ and $\|u\|_{H^{1+\rho}} \le C_\Omega \|f\|_{H^{-1+\rho}}$ with a constant C_Ω.

For this setting, we prove the following main result (see, e.g. [7]):

Theorem 1. *Let the Assumption A.1 hold. Assume also that for some* $1 < \beta_0 \le \beta_1$ *we have* $\beta_0 m_k \le m_{k-1} \le \beta_1 m_k$. *Then there is a constant* M *independent of* k *such that*

$$\eta_k^{-1}\, \mathcal{A}_k(v,v) \le \mathcal{A}_k(B_k A_k v, v) \le \eta_k\, \mathcal{A}_k(v,v), \quad \forall v \in M_k$$

with $\eta_k = \frac{M + m_k^\alpha}{m_k^\alpha}$ *and* α *as in Lemma 1.*

Sketch of the proof. The proof essentially checks the conditions (of "smoothing and approximation") from [7] under which this theorem is proved. The first condition essentially requires that R_k is a smoother. It is well known that Gauss-Seidel or scaled Jacobi satisfy this condition.

Now we outline the main steps in the proof of the second condition which is: for some $\alpha \in (0,1]$ there is a constant C_P independent of k such that

$$|\,\mathcal{A}_k((I - P_{k-1})v, v)| \le C_P \left(\frac{\|A_k v\|^2}{\lambda_k} \right)^{\alpha} [\mathcal{A}_k(v,v)]^{1-\alpha}, \tag{10}$$

where λ_k is the largest eigenvalue of the operator A_k. This is established in several steps.

First, we show that under the Assumption A.1 for all $u \in M_k$, $k = 2, \dots, J+1$ we have $|||u - P_{k-1}u|||_k \leq Ch_k^\rho \|A_k u\|_{-1+\rho}$, where $|||\cdot|||_{J+1} = |||\cdot|||_J$. Next, we show that

$$\|A_k u\|_{-1} \leq C\,|||u|||_k\,, \quad \forall u \in M_k \tag{11}$$

and

$$|||u - P_{k-1}u|||_k \leq Ch_k^\rho \|A_k u\|_{-1+\rho} \leq Ch_k^\rho \|A_k u\|_{-1}^{1-\rho} \|A_k u\|^\rho. \tag{12}$$

Finally, using the estimates (12) and (11) and the fact that $H^{-1+\rho}(\Omega)$ is an intermediate space between $H^{-1}(\Omega)$ and $L^2(\Omega)$ we obtain

$$\begin{aligned} |\,\mathcal{A}_k(u - P_{k-1}u, u)| &\leq Ch_k^\rho \|A_k u\|_{-1}^{1-\rho} \|A_k u\|^\rho\,|||u|||_k \\ &\leq C\frac{\|A_k u\|^\rho}{\lambda_k^{\rho/2}}\,|||u|||_k^{2-\rho} = C\left(\frac{\|A_k u\|^2}{\lambda_k}\right)^{\frac{\rho}{2}} \mathcal{A}_k(u,u)^{1-\frac{\rho}{2}} \end{aligned}$$

which is exactly the required result with $\alpha = \rho/2$.

Remark 1. This results is quite similar to the results of [10] and [8] in the sense that it proves the convergence of the variable V-cycle MG and ensures better convergence for smoother solutions. The difference is the choice of the hierarchy of finite element spaces used on the consecutive levels and the proof of the fundamental estimate (10). After closer inspection of the proof one can see easily that one can take $M_k = \mathcal{V}_k$, for all $k \geq k_0 \geq 1$. In fact, making this choice with $k_0 = 1$ will lead to the result of [10] (with a slightly different proof).

4 Multigrid *W*-Cycle for Piecewise-Constant Spaces

In this section we consider a method for the solution of the coarse problem, when a two level method with coarse space, denoted here with M_J, of piecewise constant functions. We will also take a standard multilevel hierarchy of this space, given by the subspaces M_k, of piecewise constant functions on grids with size h_k. Let us note that such two level algorithm is attractive, because of its simplicity and low number of degrees of freedom. However, it is well known that using the hierarchy given by M_k and applying standard V-cycle on M_J does not lead to an optimal algorithm.

In this section we briefly describe how a general ν-fold cycle can be applied to solve the coarse grid problem when piece-wise constant functions are used to define this problem. Note that on general meshes the piecewise constant functions do not provide approximation and one cannot apply the theory of MG methods in a manner used in [5] for cell-centered schemes on regular rectangular meshes. To introduce the ν-fold MG cycle algorithm, we consider the recursive definition of a general multilevel method as in [7]. Assuming that we know the action of B_{k-1} on M_{k-1}, for a given $f \in M_k$ we define the action $B_k f$ as follows.

Recursive definition of a multilevel algorithm:

1. $x = R_k g$.
2. $y = x + Z_k B_{k-1} Q_{k-1}(f - A_k x)$.
3. $B_k f = y + R_k^t (g - A_k y)$.

Now, for a fixed $e \in M_k$, we consider $E_k e = (I - B_k A_k)e$. It is easy to derive the following error equation:

$$E_k e = (I - R_k^t A_k)(I - Z_k B_{k-1} A_{k-1} P_{k-1})(I - R_k A_k)e.$$

In the case, when $\{M_k\}_{k=1}^J$ are the spaces of discontinuous piece-wise constant functions we shall define Z_k, using the techniques from [12, 3, 4], namely we shall choose Z_k to be a polynomial in $(B_{k-1}A_{k-1})$. Indeed, in such case the second term in the product form of the error equation is as follows.

$$X_k = I - Z_k B_{k-1} A_{k-1} P_{k-1} = I - (I - p_\nu(B_{k-1}A_{k-1}))P_{k-1}.$$

Usually, $p_\nu(t)$ is of degree less than or equal to ν, $p_\nu(t)$ is non-negative for $t \in [0, 1]$, and $p_\nu(0) = 1$. Taking $p_\nu(t) = (1 - t)^\nu$ gives the ν-fold MG cycle. For $\nu = 1$ this is the V-cycle and for $\nu = 2$, this is the W-cycle. Note also that, for $p_\nu(t) = (1-t)^\nu$, we have $X_k = I - P_{k-1} + E_{k-1}^\nu P_{k-1}$. Hence, if the degree of the polynomial is sufficiently large and E_{k-1} is a contraction on M_{k-1}, then the corresponding ν-fold cycle can be made as close as we please to a two-level iteration. As it is well known, the two level iteration, is uniformly convergent [9].

We would like to point out that an adaptive choice of the polynomials p_ν is possible, and we refer to [12, 3, 4] for strategies how to make such choices and also for many theoretical results for these methods.

A crucial property of the coarser spaces, that determines the convergence of such multilevel process, in general, is the stability of projections on coarser spaces. A basic assumption in the analysis is the existence of constants $q \geq 1$ and C (both independent of k and l) and such that

$$\|Q_l v\|^2 \leq C q^{k-l} \|v\|_{A_k}, \quad \forall v \in M_k, \quad k > l. \tag{13}$$

Clearly, if $q = 1$, then the resulting V-cycle algorithm has convergence rate depending only logarithmically on the mesh size, without any regularity assumptions on the underlying elliptic equation (see [6]). The ν-fold cycle, however, works even in cases, when $q > 1$, by increasing the polynomial degree ν when needed. Since the goal is to construct an optimal algorithm, the overall computational complexity gives a restriction on ν. Practical values are $\nu = 2$ or $\nu = 3$. In case $\nu = 2$ (W-cycle), which we have used in most of our numerical experiments in the next section, a uniform convergence result can be proved in a fashion similar to the case of variable V-cycle. In such analysis, an essential ingredient are bounds on q from (13) and such estimates for piecewise constant spaces on uniformly refined hexahedral, quadrilateral as well as simplicial grids are given in [11, 9].

5 Numerical Experiments

We present three test problems of elliptic equation with homogeneous Dirichlet boundary conditions:
Test Problem 1: The equation $-\Delta u = 1$ in the cube $\Omega = (0,1)^3$;
Test Problem 2: The equation $-\nabla \cdot (a\nabla u) = 1$ in $\Omega = (0,1)^3 \setminus [0.5,1)^3$ where the coefficient a has jumps (a 3-D chess-board pattern) as follows: $a = 1$, in $(I_1 \times I_1 \times I_1) \cup (I_2 \times I_2 \times I_1) \cup (I_1 \times I_2 \times I_2) \cup (I_2 \times I_1 \times I_2)$ and $a = \epsilon$, in the other parts of Ω, where $I_1 = (0, 0.5]$ and $I_2 = (0.5, 1]$, and we vary the value of ϵ according to the data in the Tables;
Test Problem 3: The equation $-\Delta u = 1$ in the domain shown on Figure 1.

The second test problem is designed to check the robustness of the methods with respect to jumps of the coefficient a. The mesh of test problem 3 has a number of finite elements of high aspect ration and the aim was to see how the iteration methods perform on such grids.

For all test examples we have used a coarse tetrahedral mesh which is uniformly refined to form a sequence of nested meshes. In SIPG we use linear and quadratic finite elements. The value of the penalty term was experimentally chosen to be $\kappa = 15$ for linear, and $\kappa = 30$ for quadratic finite elements (cf. (5), (7)).

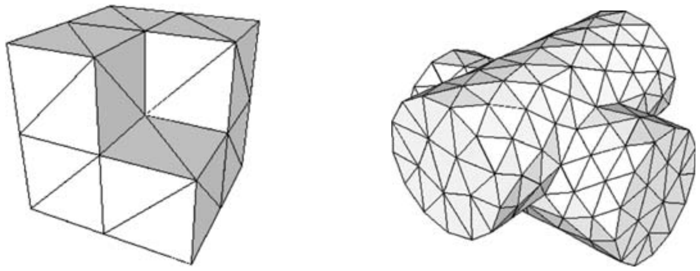


Fig. 1. Coarse meshes for the second (left) and third (right) test problems.

We test the following multilevel preconditioners for the SIPG method:

1. the V-cycle preconditioner based on continuous elements with one pre- and one post-smoothing Gauss-Seidel iteration.
2. W-cycle preconditioner based on piecewise constant coarse spaces using one pre- and post-smoothing steps of symmetric Gauss-Seidel smoother.
3. variable V-cycle preconditioner based on continuous elements described in Section 3 with one pre- and post-smoothing Gauss-Seidel iteration on the finest level and double the pre- and post-smoothing iteration on each consecutive coarser level.

The numerical results are summarised below. In each table we give the number of iterations in the PCG algorithm and the corresponding average reduction factor for each test run. In addition we include the number of degrees of freedom (DOF) in the DG space, $\mathcal{V}$, and the DOF for the first coarse space (defined on the finest mesh) of either continuous piecewise polynomial functions or piecewise constants.

Table 1. Numerical results for SIPG with **linear** FE: V-cycle based on continuous linear FE and W-cycle based on piece-wise constant functions with one pre- and one post-smoothing Gauss-Seidel iteration.

Test Problem 1	Level 2	Level 3	Level 4	Level 5	Level 6
DOF SIPG	3 072	24 576	196 608	1 572 864	12 582 912
preconditioner DOF	189	1 241	9 009	68 705	536 769
continuous FE	14/0.2556	14/0.2614	14/0.2572	14/0.2487	13/0.2344
preconditioner DOF	768	6 144	49 152	393 216	3 145 728
piecewise constant	24/0.4493	29/0.5238	30/0.5374	30/0.5342	29/0.5276

Table 2. Numerical results for SIPG with **quadratic** FE: V-cycle preconditioner based on continuous FE and W-cycle preconditioner based on piecewise constant functions each with one pre- and one post-smoothing Gauss-Seidel iteration.

Test Problem 1	Level 1	Level 2	Level 3	Level 4	Level 5
DOF SIPG	960	7 680	61 440	491 520	3 932 160
preconditioner DOF	189	1 241	9 009	68 705	536 769
continuous FE	10/0.1414	11/0.1717	11/0.1747	11/0.1657	10/0.1514
preconditioner DOF	96	768	6 144	49 152	393 216
piecewise constant	22/0.4315	35/0.5810	42/0.6442	43/0.6509	43/0.6496

In Tables 1 and 2 we present the computational results for test problem 1. These results show that both preconditioners, the V-cycle, that uses continuous finite elements, and the W-cycle, that uses piece-wise constant function on all coarser levels are optimal with respect to the number of iterations. The W-cycle preconditioner, based on piecewise constant functions, performs according to the W-cycle theory. However, it needs two times more iterations compared with the V-cycle, based on continuous functions. While the former has a matrix of size about 6 times larger than size of the matrix of the latter (for linear FE), one should have in mind that in the case of piece-wise constant functions the corresponding matrix has only five nonzero entries per row, i.e. it is about five times sparser than the matrix produced by continuous linear elements. Unfortunately, we do not have a theory for the V-cycle.

It is known that the choice of the stabilization factor $\kappa_{\mathcal{E}}$ could affect the properties of the method. To test sensitivity of the preconditioners with respect to the jumps of the coefficient a we considered two different choices, $\kappa_{\mathcal{E}} = \kappa \{\{a\}\}$, as defined in the SIPG method, and $\kappa_{\mathcal{E}} = \kappa \|a\|_{L^\infty} = 15$,

Table 3. V-cycle and variable V-cycle based on continuous coarse spaces for the SIPG with **linear** elements and stabilization factor $\kappa_{\mathcal{E}}$ that does not depend on the jumps of a.

Test Problem 2	Level 1	Level 2	Level 3	Level 4
DOF of SIPG	1 344	10 752	86 016	688 128
precond. DOF - continuous linear	117	665	4 401	31 841
$\epsilon = 1$, V-cycle	15/0.2750	16/0.2946	15/0.2908	15/0.2838
$\epsilon = 0.1$, V-cycle	17/0.3322	19/0.3645	19/0.3717	19/0.3675
$\epsilon = 0.01$, V-cycle	17/0.3219	19/0.3632	19/0.3746	19/0.3713
$\epsilon = 0.001$, V-cycle	15/0.2929	17/0.3377	18/0.3527	18/0.3488
$\epsilon = 1$, variable V-cycle	15/0.2738	15/0.2900	15/0.2850	15/0.2759
$\epsilon = 0.1$, variable V-cycle	17/0.3310	18/0.3593	19/0.3658	18/0.3566
$\epsilon = 0.01$, variable V-cycle	17/0.3211	18/0.3568	19/0.3684	18/0.3582
$\epsilon = 0.001$, variable V-cycle	15/0.2919	17/0.3333	18/0.3457	17/0.3337

which obviously is independent of the jumps. As shown in Table 3 the variable V-cycle preconditioner, covered by our theory, gives the same number of iterations as the V-cycle. Both preconditioners are not sensitive to the choice of $\kappa_{\mathcal{E}}$. From Table 3 one can see that the preconditioners based on continuous coarse spaces are robust in this case with respect to the jumps in a. However, this is not the case for the preconditioners based on piece-wise constant coarse spaces. We observe this in Table 4 where the performance of the W-cycle is given. From these experiments we see that a proper weighting of the jumps is essential for the performance of the W-cycle iteration based on piece-wise constant functions. In Table 5 we present results for test problem 2 with properly scaled stabilization parameter: $\kappa_{\mathcal{E}} = \kappa \{\{a\}\}$. We tested the following preconditioners: V-cycle and variable V-cycle based on continuous coarse spaces and W-cycle based on piece-wise constant coarse spaces. Once again one can see that V-cycle and variable V-cycle based on continuous coarse spaces perform almost identically. Note that the iteration counts are slightly larger than those of the case $\kappa_{\mathcal{E}} = \kappa \|a\|_{L^\infty}$ (cf. Table 3) but they are insensitive to large jumps. In the case of piece-wise constant coarse spaces (W-cycle) the advantage of

Table 4. W-cycle based on piece-wise constant coarse spaces for the SIPG with **linear** elements and stabilization factor $\kappa_{\mathcal{E}}$ that does not depend on the jumps of a

Test Problem 2	Level 1	Level 2	Level 3	Level 4
DOF of SIPG	1 344	10 752	86 016	688 128
precond. DOF - piecewise constant	336	2 688	21 504	172 032
$\epsilon = 1$, W-cycle	22/0.4151	27/0.4940	29/0.5224	29/0.5297
$\epsilon = 0.1$, W-cycle	38/0.6106	72/0.7706	85/0.8027	91/0.8160
$\epsilon = 0.01$, W-cycle	48/0.6804	157/0.8869	210/0.9156	238/0.9255

the weighted stabilization is evident – the numerical experiments show that the number of PCG iterations is essentially independent of the jumps.

Table 5. Numerical results for Test Problem 2: SIPG with **linear** elements and stabilization parameter $\kappa_{\mathcal{E}} = \kappa \{\{a\}\}$.

Test Problem 2	Level 1	Level 2	Level 3	Level 4	Level 5
DOF of SIPG	1 344	10 752	86 016	688 128	5 505 024
precond. DOF - continuous	117	665	4 401	31 841	241 857
$\epsilon = 1$, V-cycle	15/0.2750	16/0.2946	15/0.2908	15/0.2838	15/0.2766
$\epsilon = 0.1$, V-cycle	16/0.3161	20/0.3812	21/0.4105	22/0.4187	22/0.4196
$\epsilon = 0.01$, V-cycle	20/0.3800	24/0.4539	29/0.5228	31/0.5518	33/0.5687
$\epsilon = 0.001$, V-cycle	19/0.3782	24/0.4603	30/0.5377	33/0.5674	36/0.5957
$\epsilon = 10^{-4}$, V-cycle	18/0.3546	24/0.4535	30/0.5312	32/0.5622	34/0.5753
$\epsilon = 10^{-5}$, V-cycle	18/0.3411	23/0.4488	28/0.5100	30/0.5405	32/0.5622
$\epsilon = 10^{-6}$, V-cycle	17/0.3279	23/0.4416	26/0.4911	29/0.5298	30/0.5375
$\epsilon = 1$, var. V-cycle	15/0.2738	15/0.2900	15/0.2850	15/0.2759	14/0.2628
$\epsilon = 0.1$, var. V-cycle	16/0.3157	20/0.3782	21/0.4038	21/0.4107	21/0.4056
$\epsilon = 0.01$, var. V-cycle	20/0.3796	24/0.4508	29/0.5170	31/0.5448	32/0.5559
$\epsilon = 0.001$, var. V-cycle	19/0.3779	24/0.4574	30/0.5329	33/0.5612	35/0.5886
precond. DOF - p.w. constant	336	2 688	21 504	172 032	1 376 256
$\epsilon = 1$, W-cycle	22/0.4151	27/0.4940	29/0.5224	29/0.5297	29/0.5251
$\epsilon = 0.1$, W-cycle	23/0.4400	28/0.5057	29/0.5284	30/0.5357	30/0.5343
$\epsilon = 0.01$, W-cycle	22/0.4300	28/0.5012	30/0.5321	30/0.5385	31/0.5420
$\epsilon = 0.001$, W-cycle	23/0.4410	28/0.5001	30/0.5332	30/0.5403	31/0.5438
$\epsilon = 10^{-4}$, W-cycle	22/0.4302	27/0.4980	30/0.5333	30/0.5405	31/0.5442
$\epsilon = 10^{-5}$, W-cycle	22/0.4209	26/0.4880	30/0.5333	30/0.5405	31/0.5442
$\epsilon = 10^{-6}$, W-cycle	21/0.4112	25/0.4730	30/0.5333	30/0.5405	31/0.5442

Table 6. Numerical results for Test Problem 3 for V-cycle and W-cycle for the SIPG with **linear** elements.

Test Problem 3	Level 1	Level 2	Level 3	Level 4
precond. DOF of SIPG	24 032	192 256	1 538 048	12 304 384
precond. DOF of cont. FE	1 445	9 693	70 633	538 513
V-cycle	18/0.3530	18/0.3559	18/0.3529	19/0.3785
precond. DOF p.w. constants	6 008	48 064	384 512	3 076 096
W-cycle	35/0.5907	40/0.6307	45/0.6578	48/0.6788

Finally, in Table 6 we present the results iteration for V-cycle and W-cycle preconditioners for test Problem 3. The mesh of this example has a number of finite elements with high aspect ratio. The computations show that the preconditioner based on piecewise constant functions is slightly more sensitive with respect to the aspect ratio.

Acknowledgement. The second author has been supported in parts by NSF DEIA-0540136 and by CASC, Lawrence Livermore National Laboratory; the third author was supported in part by NSF DMS-058110, DMS-0619587, and by Lawrence Livermore National Laboratory under subcontract number B551021.
The authors thanks the referees for their helpful remarks and useful comments.

References

[1] D.N. Arnold. An interior penalty finite element method with discontinuous elements. *SIAM J. Numer. Anal.*, 19(4):742–760, 1982.

[2] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779 (electronic), 2001/02.

[3] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods. I. *Numer. Math.*, 56(2-3):157–177, 1989.

[4] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods. II. *SIAM J. Numer. Anal.*, 27(6):1569–1590, 1990.

[5] J.H. Bramble, R.E. Ewing, J.E. Pasciak, and J. Shen. The analysis of multigrid algorithms for cell centered finite difference methods. *Adv. Comput. Math.*, 5(1):15–29, 1996.

[6] J.H. Bramble, J.E. Pasciak, J.P. Wang, and J. Xu. Convergence estimates for multigrid algorithms without regularity assumptions. *Math. Comp.*, 57(195):23–45, 1991.

[7] J.H. Bramble and X. Zhang. The analysis of multigrid methods. In *Handbook of Numerical Analysis, Vol. VII*, pages 173–415. North-Holland, Amsterdam, 2000.

[8] S.C. Brenner and J. Zhao. Convergence of multigrid algorithms for interior penalty methods. *Appl. Numer. Anal. Comput. Math.*, 2(1):3–18, 2005.

[9] V.A. Dobrev, R.D. Lazarov, P.S. Vassilevski, and L.T. Zikatanov. Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations. *Numer. Linear Algebra Appl.*, 13(9):753–770, 2006.

[10] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numer. Math.*, 95(3):527–550, 2003.

[11] R.D. Lazarov, P.S. Vassilevski, and L.T. Zikatanov. Algebraic multilevel iteration for preconditioning DG problems. Technical report, 2005.

[12] P.S. Vassilevski. Hybrid V-cycle algebraic multilevel preconditioners. *Math. Comp.*, 58(198):489–512, 1992.

Nonlinear Convergence Analysis for the Parareal Algorithm

Martin J. Gander and Ernst Hairer

Section de Mathématiques, Université de Genève, CP 64, 1211 Genève,
Switzerland, {martin.gander,ernst.hairer}@math.unige.ch

1 Introduction

Time domain decomposition methods have a long history: already [10] made the following visionary statement:

> "For the last 20 years, one has tried to speed up numerical computation mainly by providing ever faster computers. Today, as it appears that one is getting closer to the maximal speed of electronic components, emphasis is put on allowing operations to be performed in parallel. In the near future, much of numerical analysis will have to be recast in a more "parallel" form."

Nievergelt proposed a parallel algorithm based on a decomposition of the time direction for the solution of ordinary differential equations. While his idea targeted large scale parallelism, [9] proposed a little later a family of naturally parallel Runge Kutta methods for small scale parallelism:

> "It appears at first sight that the sequential nature of the numerical methods do not permit a parallel computation on all of the processors to be performed. We say that the front of computation is too narrow to take advantage of more than one processor... Let us consider how we might widen the computation front."

Waveform relaxation methods, introduced in [6] for the large scale simulation of VLSI design, are another fundamental way to introduce time parallelism into the solution of evolution problems. For an up to date historical review and further references, see [4].

The present research was motivated by the introduction of the parareal algorithm in [7]. We show in this paper a general superlinear convergence result for the parareal algorithm applied to a nonlinear system of ordinary differential equations.

2 Derivation of the Parareal Algorithm

The parareal algorithm is a time parallel algorithm for the solution of the general nonlinear system of ordinary differential equations

$$\mathbf{u}'(t) = \mathbf{f}(\mathbf{u}(t)), \quad t \in (0,T), \quad \mathbf{u}(0) = \mathbf{u}^0, \tag{1}$$

where $\mathbf{f} : \mathbb{R}^M \longrightarrow \mathbb{R}^M$ and $\mathbf{u} : \mathbb{R} \longrightarrow \mathbb{R}^M$.

To obtain a time parallel algorithm for (1), we decompose the time domain $\Omega = (0,T)$ into N time subdomains $\Omega_n = (T_n, T_{n+1})$, $n = 0, 1, \dots N-1$, with $0 = T_0 < T_1 < \dots < T_{N-1} < T_N = T$, and $\Delta T_n := T_{n+1} - T_n$, and consider on each time subdomain the evolution problem

$$\mathbf{u}_n'(t) = \mathbf{f}(\mathbf{u}_n(t)),\ t \in (T_n, T_{n+1}),\ \mathbf{u}_n(T_n) = \mathbf{U}_n,\ n = 0, 1, \dots, N-1, \tag{2}$$

where the initial values $\mathbf{U}_n$ need to be determined such that the solutions on the time subdomains Ω_n coincide with the restriction of the solution of (1) to Ω_n, i.e. the $\mathbf{U}_n$ need to satisfy the system of equations

$$\mathbf{U}_0 = \mathbf{u}^0, \quad \mathbf{U}_n = \boldsymbol{\varphi}_{\Delta T_{n-1}}(\mathbf{U}_{n-1}), \quad n = 1, \dots, N-1, \tag{3}$$

where $\boldsymbol{\varphi}_{\Delta T_n}(\mathbf{U})$ denotes the solution of (1) with initial condition $\mathbf{U}$ after time ΔT_n. This time decomposition method is nothing else than a multiple shooting method for (1), see [3]. Letting $\mathbf{U} = (\mathbf{U}_0^T, \dots \mathbf{U}_{N-1}^T)^T$, the system (3) can be written in the form

$$\mathbf{F}(\mathbf{U}) = \begin{pmatrix} \mathbf{U}_0 - \mathbf{u}^0 \\ \mathbf{U}_1 - \boldsymbol{\varphi}_{\Delta T_0}(\mathbf{U}_0) \\ \vdots \\ \mathbf{U}_{N-1} - \boldsymbol{\varphi}_{\Delta T_{N-2}}(\mathbf{U}_{N-2}) \end{pmatrix} = \mathbf{0}, \tag{4}$$

where $\mathbf{F} : \mathbb{R}^{M \cdot N} \longrightarrow \mathbb{R}^{M \cdot N}$. System (4) defines the unknown initial values $\mathbf{U}_n$ for each time subdomain, and needs to be solved, in general, by an iterative method. For a direct method in the case where (1) is linear and the system (4) can be formed explicitly, see [1].

Applying Newtons method to (4) leads after a short calculation to

$$\begin{aligned} \mathbf{U}_0^{k+1} &= \mathbf{u}^0, \\ \mathbf{U}_n^{k+1} &= \boldsymbol{\varphi}_{\Delta T_{n-1}}(\mathbf{U}_{n-1}^k) + \boldsymbol{\varphi}'_{\Delta T_{n-1}}(\mathbf{U}_{n-1}^k)(\mathbf{U}_{n-1}^{k+1} - \mathbf{U}_{n-1}^k), \end{aligned} \tag{5}$$

where $n = 1, \dots, N-1$. Chartier and Philippe [3] showed that the method (5) converges quadratically, once the approximations are close enough to the solution. However in general, it is too expensive to compute the Jacobian terms in (5) exactly. An interesting recent approximation is the parareal algorithm, which uses two approximations with different accuracy: let $\mathbf{F}(T_n, T_{n-1}, \mathbf{U}_{n-1})$ be an accurate approximation to the solution $\boldsymbol{\varphi}_{\Delta T_{n-1}}(\mathbf{U}_{n-1})$ on time subdomain Ω_{n-1}, and let $\mathbf{G}(T_n, T_{n-1}, \mathbf{U}_{n-1})$ be a less accurate approximation, for example on a coarser grid, or a lower order method, or even an approximation using a simpler model than (1). Then, approximating the time subdomain solves in (5) by $\boldsymbol{\varphi}_{\Delta T_{n-1}}(\mathbf{U}_{n-1}^k) \approx \mathbf{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k)$, and the Jacobian term by

$$\boldsymbol{\varphi}'_{\Delta T_{n-1}}(\mathbf{U}_{n-1}^k)(\mathbf{U}_{n-1}^{k+1} - \mathbf{U}_{n-1}^k) \approx \mathbf{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k+1}) - \mathbf{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k),$$

we obtain as approximation to (5)

$$\begin{aligned} \mathbf{U}_0^{k+1} &= \mathbf{u}^0, \\ \mathbf{U}_n^{k+1} &= \mathbf{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k) + \mathbf{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k+1}) - \mathbf{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k), \end{aligned} \tag{6}$$

which is the parareal algorithm, see [7] for a linear model problem, and [2] for the formulation (6). A natural initial guess is the coarse solution, i.e. $\mathbf{U}_n^0 = G(T_n, T_{n-1}, \mathbf{U}_{n-1}^0)$.

3 Convergence Analysis

To simplify the exposition, we assume in this section that all the time subdomains are of the same size, $\Delta T_n = \Delta T := \frac{T}{N}$, $n = 0, 1, \ldots, N-1$, and that $\mathbf{F}$ is the exact solution, i.e. $\mathbf{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k) = \boldsymbol{\varphi}_{\Delta T_{n-1}}(\mathbf{U}_{n-1}^k)$. We also assume that the difference between the approximate solution given by $\mathbf{G}$ and the exact solution can be expanded for ΔT small,

$$\mathbf{F}(T_n, T_{n-1}, x) - \mathbf{G}(T_n, T_{n-1}, x) = c_{p+1}(x)\Delta T^{p+1} + c_{p+2}(x)\Delta T^{p+2} + \ldots, \tag{7}$$

which is possible if the right hand side function $\mathbf{f}$ in (1) is smooth enough, and $\mathbf{G}$ is a Runge Kutta method for example. We finally assume that $\mathbf{G}$ satisfies the Lipschitz condition

$$\|\mathbf{G}(t + \Delta T, t, \mathbf{x}) - \mathbf{G}(t + \Delta T, t, \mathbf{y})\| \leq (1 + C_2 \Delta T)\|\mathbf{x} - \mathbf{y}\|. \tag{8}$$

Theorem 1. *Let $\mathbf{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k) = \boldsymbol{\varphi}_{\Delta T_{n-1}}(\mathbf{U}_{n-1}^k)$ be the exact solution on time subdomain Ω_{n-1}, and let $\mathbf{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k)$ be an approximate solution with local truncation error bounded by $C_3 \Delta T^{p+1}$, and satisfying (7), where the c_j, $j = p+1, p+2, \ldots$ are continuously differentiable, and assume that $\mathbf{G}$ satisfies the Lipschitz condition (8). Then, at iteration k of the parareal algorithm (6), we have the bound*

$$\begin{aligned} \|\mathbf{u}(T_n) - \mathbf{U}_n^k\| &\leq \frac{C_3}{C_1} \frac{(C_1 \Delta T^{p+1})^{k+1}}{(k+1)!} (1 + C_2 \Delta T)^{n-k-1} \prod_{j=0}^{k} (n-j) \\ &\leq \frac{C_3}{C_1} \frac{(C_1 T_n)^{k+1}}{(k+1)!} e^{C_2 (T_n - T_{k+1})} \Delta T^{p(k+1)}. \end{aligned}$$

Proof. From the definition of the parareal algorithm (6), we obtain, using that $\mathbf{F}$ is the exact solution and adding and subtracting $\mathbf{G}(T_n, T_{n-1}, \mathbf{u}(T_{n-1}))$

$$\begin{aligned}\mathbf{u}(T_n) - \mathbf{U}_n^{k+1} &= \mathbf{F}(T_n, T_{n-1}, \mathbf{u}(T_{n-1})) - \mathbf{G}(T_n, T_{n-1}, \mathbf{u}(T_{n-1})) \\ &\quad - \big(\mathbf{F}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k) - \mathbf{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^k)\big) \\ &\quad + \mathbf{G}(T_n, T_{n-1}, \mathbf{u}(T_{n-1})) - \mathbf{G}(T_n, T_{n-1}, \mathbf{U}_{n-1}^{k+1}).\end{aligned}$$

Now using expansion (7) for the first two terms on the right hand side, and (8) on the last one, we obtain on taking norms

$$\|\mathbf{u}(T_n) - \mathbf{U}_n^{k+1}\| \le C_1 \Delta T^{p+1} \|\mathbf{u}(T_{n-1}) - \mathbf{U}_{n-1}^k\| + (1 + C_2 \Delta T)\|\mathbf{u}(T_{n-1}) - \mathbf{U}_{n-1}^{k+1}\|.$$

This motivates to study the recurrence relation

$$e_n^{k+1} = \alpha e_{n-1}^k + \beta e_{n-1}^{k+1}, \quad e_n^0 = \gamma + \beta e_{n-1}^0, \tag{9}$$

where $\alpha = C_1 \Delta T^{p+1}$, $\beta = 1 + C_2 \Delta T$ and $\gamma = C_3 \Delta T^{p+1}$, since e_n^k is then an upper bound on $\|\mathbf{u}(T_n) - \mathbf{U}_n^k\|$. Multiplying (9) by ζ^n and summing over n, we find that the generating function $\rho_k(\zeta) := \sum_{n \ge 1} e_n^k \zeta^n$ satisfies the recurrence relation

$$\rho_{k+1}(\zeta) = \alpha\zeta\rho_k(\zeta) + \beta\zeta\rho_{k+1}(\zeta), \quad \rho_0(\zeta) = \gamma\frac{\zeta}{1-\zeta} + \beta\zeta\rho_0(\zeta).$$

Solving for $\rho_k(\zeta)$, we obtain after induction

$$\rho_k(\zeta) = \gamma\alpha^k \frac{\zeta^{k+1}}{(1-\zeta)} \frac{1}{(1-\beta\zeta)^{k+1}}.$$

Replacing the factor $1 - \zeta$ in the denominator by $1 - \beta\zeta$ only increases the coefficients in the power series of $\rho_k(\zeta)$. Using now the binomial series expansion

$$\frac{1}{(1-\beta\zeta)^{k+2}} = \sum_{j \ge 0} \binom{k+1+j}{j} \beta^j \zeta^j,$$

we obtain for the n-th coefficient e_n^k the bound

$$e_n^k \le \gamma\alpha^k\beta^{n-k-1}\binom{n}{k+1},$$

which concludes the proof.

4 Numerical Experiments

We show now several numerical experiments, first for small systems of ordinary differential equations, where the only potential for parallelization lies in the time direction, and then also for a partial differential equation, namely the viscous Burgers equation.

4.1 Brusselator

The brusselator system of ordinary differential equations models a chain of chemical reactions and is given by

$$\dot{x} = A + x^2 y - (B+1)x, \qquad \dot{y} = Bx - x^2 y.$$

We chose for the parameters $A = 1$ and $B = 3$, and since $B > A^2 + 1$, the system will form a limit cycle, see [5]. We start the simulation with the initial conditions $x(0) = 0$, $y(0) = 1$, and compute an approximate solution over the time interval $t \in [0, T = 12]$ using the classical fourth order Runge Kutta method with coarse time step $\varDelta T = \frac{T}{32}$, and fine time step $\varDelta t = \frac{T}{640}$, which gives a solution with an accuracy of $5.62e-6$. In Figure 1, we show the initial guess from the coarse solver, and the first five iterates of the parareal algorithm in the phase plane, and also the difference between the parareal approximation and the complete fine approximation as a function of time. The larger dot in the phase plane, and the vertical line in the error plots indicate how far one could have computed the fine solution sequentially in the same computation time, neglecting the cost of the coarse solve. The fine dashed line indicates the accuracy of the fine grid solution. Clearly there is

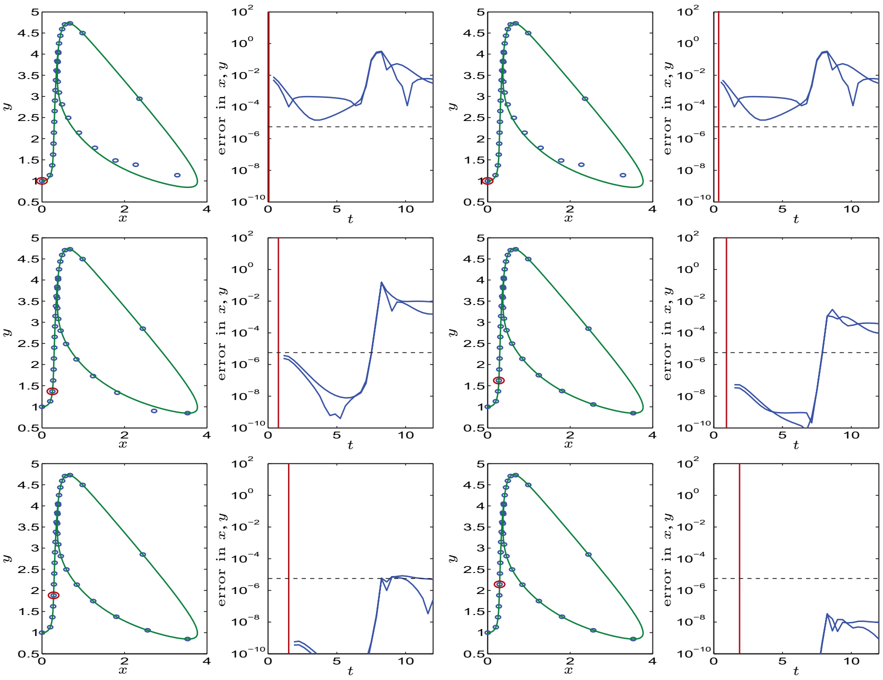


Fig. 1. Parareal approximation of the solution of the Brusselator problem.

a parallel speedup with this type of time parallelization: with 32 processors, one could have computed the numerical approximation to the same accuracy of $5.62e-6$ about eight times faster than with one processor.

4.2 Arenstorf Orbit

Arenstorf orbits are closed orbits of a light object (e.g. a satellite) moving under the influence of gravity of two heavy objects (e.g. planets, moons). The equations of motion for the example of two heavy objects are

$$\ddot{x} = x + 2\dot{y} - b\frac{x+a}{D_1} - a\frac{x-b}{D_2}, \quad \ddot{y} = y - 2\dot{x} - b\frac{y}{D_1} - a\frac{y}{D_2},$$

where D_j, $j = 1, 2$ are function of x and y,

$$D_1 = ((x+a)^2 + y^2)^{\frac{3}{2}}, \; D_2 = ((x-b)^2 + y^2)^{\frac{3}{2}}.$$

If the parameters are $a = 0.012277471$ and $b = 1-a$, and the initial conditions are chosen to be $x(0) = 0.994$, $\dot{x} = 0$, $y(0) = 0$, $\dot{y}(0) = -2.00158510637908$, then the solution is a nice closed orbit with period $T = 17.06521656015796$, see [5]. There have been earlier attempts to compute planetary orbits in parallel, see [11], where a multiple shooting method was developed. We use again the parareal algorithm to compute the Arenstorf orbit in parallel, with the classical fourth order Runge Kutta method and coarse time step $\Delta T = \frac{T}{250}$, and fine time step $\Delta t = \frac{T}{80000}$, such that the fine trajectory has an accuracy of $9.98e-6$. We show in Figure 2 the initial guess and the first five iterations of the parareal algorithm, as in the case of the brusselator problem. While the initial guess is completely off, and simply spirals outward, the first iteration already reveals the shape of the Arenstorf orbit, and the algorithm has converged to the precision of the fine time step approximation after four iterations. Neglecting the cost of the coarse grid solve, one could have computed this trajectory with 250 processors about 62 times faster in parallel, than with one processor sequentially. The fact that the initial guess is so off is due to the tremendous sensitivity of the solution to the initial conditions, so it would be better to use an adaptive method here. We are currently studying the use of adaptivity in the context of the parareal algorithm.

4.3 Lorenz Equations

Weather prediction could be an important application of the parareal algorithm, since predictions have to be made in real time. If a large scale parallel computer is available, and the parallelization in space of the partial differential equation modeling the evolution of the weather is already saturated, the only way to speed up the computation is to try to parallelize the time direction. A very simple model for weather prediction is the model given by the Lorenz equations,

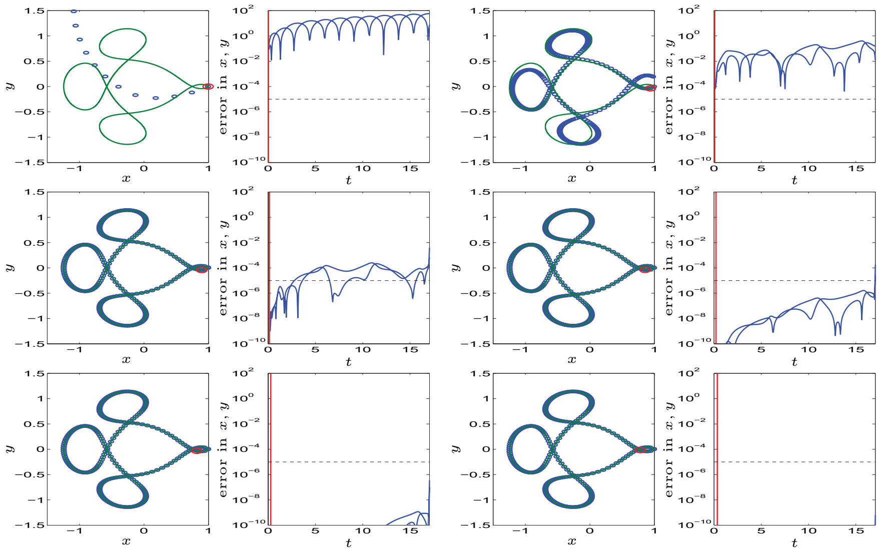


Fig. 2. Parareal approximation of the Arenstorf orbit.

$$\dot{x} = -\sigma x + \sigma y, \quad \dot{y} = -xz + rx - y, \quad \dot{z} = xy - bz.$$

These equations were first studied by Lorenz [8], who discovered that in certain cases approximations to their solution are very sensitive to small changes in the initial data (he noticed this when he interrupted a computation and wrote the current position of the solution down by hand to continue the next day, but his notes included only the first four digits, and not the full precision). A legend then says that looking at the solution of his equations, which looks on the attractor like a butterfly, Lorenz concluded that the wings of a butterfly in Europe could create a thunderstorm in the US.

We chose for the parameters in the Lorenz equations $\sigma = 10$, $r = 28$ and $b = \frac{8}{3}$, such that the system is in the chaotic regime, and trajectories converge to the butterfly attractor. We start with the initial conditions $(x, y, z)(0) = (20, 5, -5)$, and compute with the parareal algorithm an approximate solution on the time interval $[0, T = 10]$, using again the classical fourth order Runge Kutta method with coarse time step $\Delta T = \frac{T}{180}$, and fine time step $\Delta t = \frac{T}{14400}$, which leads to an accuracy in the fine trajectory of $2.4e - 6$. We show in Figure 3 the initial guess and the first five iterations of the parareal algorithm, together with error curves for the coordinates, as a function of time. One can see that for the first two iterations, the approximate parareal trajectory is not in the same wing of the butterfly attractor as the converged trajectory. At iteration three, the situation changes and the parareal approximation follows now the converged trajectory. From this iteration on, the algorithm converges on the entire time interval, as one can see in Figure 4, where we show iteration six to eleven. At iteration ten, an overall accuracy of $1e-6$, which corresponds

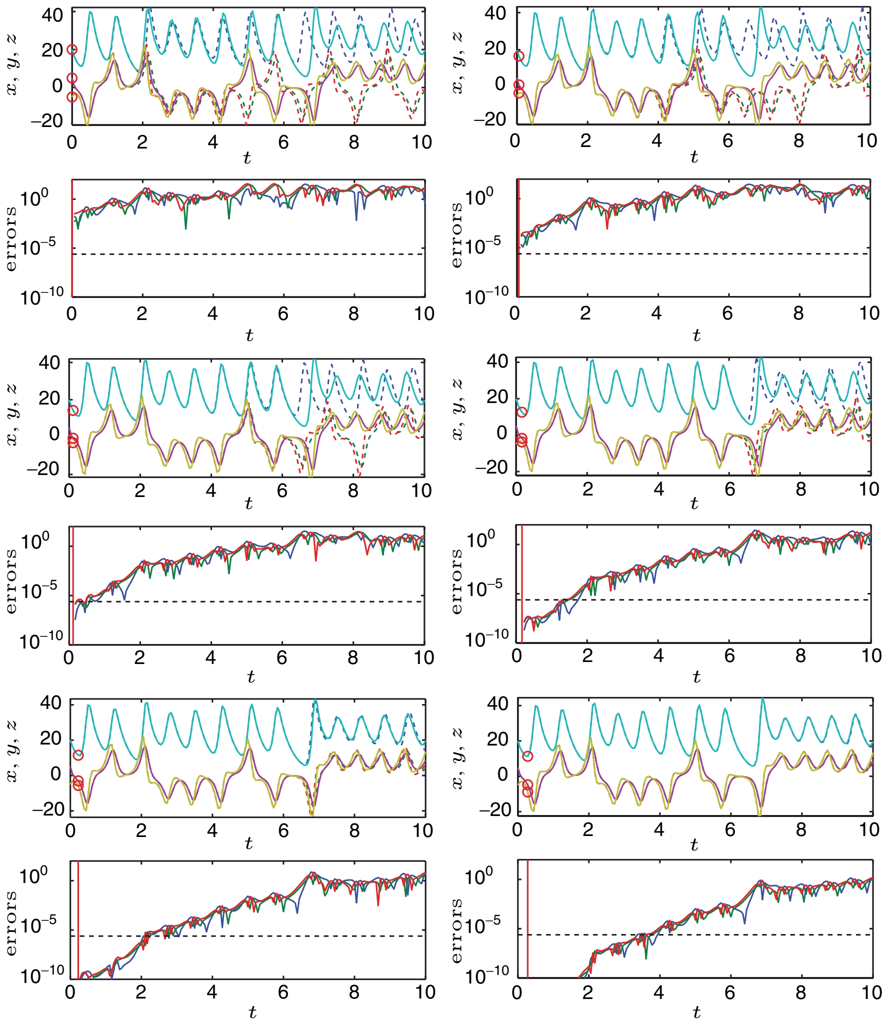


Fig. 3. Initial guess and first five parareal approximations of the solution of the Lorenz equations.

to the fine grid solution accuracy, is achieved. Neglecting the cost of the coarse solver, one could therefore have computed a fine grid accurate solution with 180 processors about 18 times faster than sequentially, as indicated by the dots and the vertical line on the graphs.

In Figure 5 on the left, we show how the difference of the parareal approximation and the converged solution, measured in the L^2-norm in space, and in the L^∞-norm in time, diminishes as a function of the iterations of the parareal algorithm. One can clearly see that the convergence is superlinear.

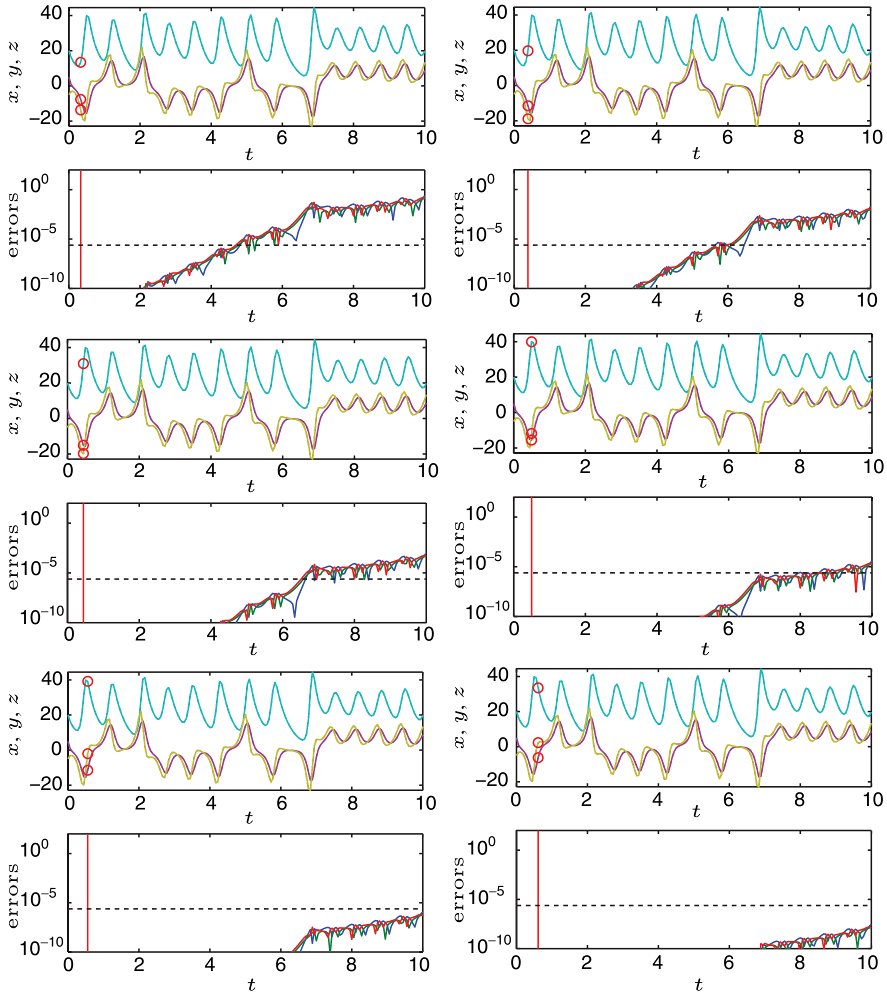


Fig. 4. Sixth to eleventh parareal approximation of the solution of the Lorenz equations.

In the context of the Lorenz equations, it is interesting to investigate the behavior of the parareal algorithm with respect to the chaotic nature of the system. In Figure 5, we show on the right the convergence behavior of the parareal algorithm for an implementation with variable precision arithmetic, using 16, 32 and 48 digits of accuracy. One can see that the theoretical result of superlinear convergence stops at a certain level before the numerical precision has been reached, and the algorithm stagnates, or in other words, the trajectory has converged to a different solution from the one computed sequentially, due to roundoff errors.

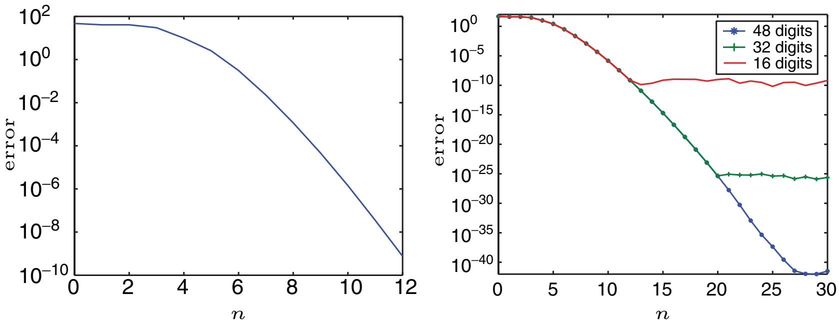


Fig. 5. Convergence behavior of the parareal algorithm applied to the Lorenz equations.

4.4 Viscous Burgers Equation

We finally show numerical experiments for a non-linear partial differential equation, the viscous Burgers equation,

$$u_t + uu_x = \nu u_{xx} \quad \text{in } \Omega = [0,1], \quad u(x,0) = \sin(2\pi x),$$

with homogeneous boundary data, such that the solution forms Friedrich's N-wave. We chose for the viscosity parameter $\nu = \frac{1}{50}$, used a centered finite difference discretization with spatial step $\Delta x = \frac{1}{50}$, and a backward Euler discretization in time. We only parallelized the solution in time using the parareal algorithm, with coarse time step $\Delta T = \frac{1}{10}$, and fine time step $\Delta t = \frac{1}{100}$, which gives a numerical accuracy of $4e-2$. We show in Figure 6 on the left the converged solution over a short time interval, $[0, T = 0.1]$, where one can see how the N-wave is forming, and on the right the same solution over a longer time interval, $[0, T = 1]$. In Figure 7, we show on the left the convergence

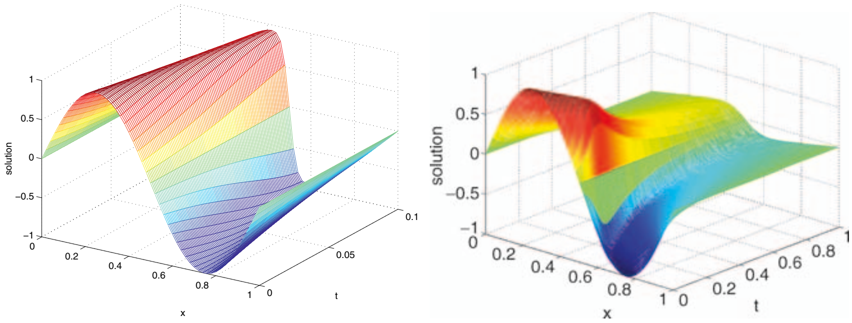


Fig. 6. Converged approximate solution for the Burgers equation over a short and long time interval.

behavior of the parareal algorithm applied to the Burgers equation, when the problem is posed over time intervals of various length. Again we measure the

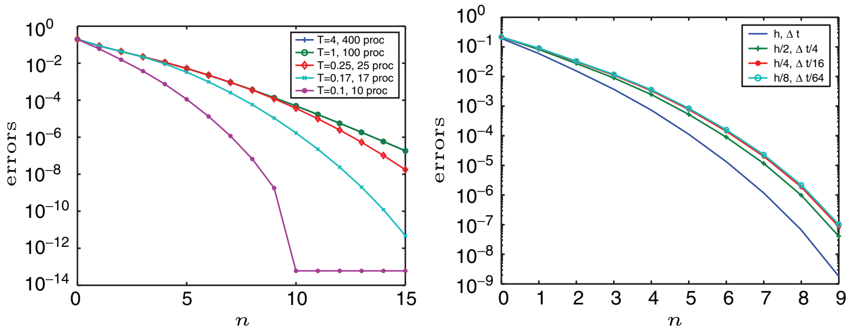


Fig. 7. Convergence behavior of the parareal algorithm applied to Burgers equation, on the left for various lengths of the time interval, and on the right when the accuracy of the discretization is increased.

error in the L^2-norm in space, and the L^∞-norm in time. Over short time intervals, the convergence of the parareal algorithm is faster than over long time intervals. In the case of $T = 0.1$, the algorithm converges at step two to the accuracy of the discretization error, and one could therefore, neglecting the coarse solve, compute this approximation with ten processors five times faster in parallel than with one processor. Note also that as one continues to iterate, the algorithm converges further toward the fine grid solution, until the roundoff error accuracy is reached at step 10, as indicated by Theorem 1. Over longer time intervals, for example $T = 1$, with the same parareal configuration, the accuracy of the discretization error is reached at iteration four. Computing with one hundred processors in parallel, this solution could have been obtained 25 times faster than sequentially on one processor.

In Figure 7 on the right, we show how the discretization error affects the parareal algorithm. For $T = 0.1$, we computed more and more refined solutions, both in space and time, with truncation error $4e-2$, $1e-2$, $2.5e-3$ and $6.2e-4$, using the parareal algorithm with 10 coarse time intervals. The convergence plot shows that the convergence rate becomes independent of the mesh parameters, as it was proved for the linear case in [4].

5 Conclusions

We showed that the parareal algorithm applied to a nonlinear system of ordinary differential equations converges superlinearly on any bounded time interval. We illustrated this result with four non-linear examples coming from chemical reactions, planetary orbits, weather forecast and fluid flow problems.

These examples show that parallel speedup in time is possible, although not at the same level as in space, where one often asks for perfect speedup, i.e. the computation with one hundred processors should be one hundred times faster. For time parallelization with the parareal algorithm, one has to be satisfied with less, but if this is the only option left to speedup the solution time, it might be worthwhile considering it.

References

[1] P. Amodio and L. Brugnano. Parallel implementation of block boundary value methods for ODEs. *J. Comp. Appl. Math.*, 78:197–211, 1997.

[2] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah. Parallel-in-time molecular-dynamics simulations. *Physical Review E*, 66:057706–1–4, 2002.

[3] P. Chartier and B. Philippe. A parallel shooting technique for solving dissipative ODEs. *Computing*, 51:209–236, 1993.

[4] M.J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 2006. In press.

[5] E. Hairer, S.P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems.* Springer-Verlag, Second Revised Edition, 1993.

[6] E. Lelarasmee, A.E. Ruehli, and A.L. Sangiovanni-Vincentelli. The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Trans. on CAD of IC and Syst.*, 1:131–145, 1982.

[7] J.-L. Lions, Y. Maday, and G. Turinici. A parareal in time discretization of PDE's. *C.R. Acad. Sci. Paris, Serie I*, 332:661–668, 2001.

[8] E.N. Lorenz. On the prevalence of aperiodicity in simple systems. In M. Grmela and J.E. Marsden, editors, *Global Analysis*, volume 755, pages 53–75, Calgary, 1978. Lecture Notes in Mathematics.

[9] W.L. Miranker and W. Liniger. Parallel methods for the numerical integration of ordinary differential equations. *Math. Comp.*, 91:303–320, 1967.

[10] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Comm. ACM*, 7:731–733, 1964.

[11] P. Saha, J. Stadel, and S. Tremaine. A parallel integration method for solar system dynamics. *Astron. J.*, 114:409–415, 1997.

Schwarz Waveform Relaxation Algorithms

Laurence Halpern

Laboratoire Analyse, Géométrie et Applications Université Paris XIII, 99 Avenue J.-B. Clément, 93430 Villetaneuse, France. halpern@math.univ-paris13.fr

Summary. Optimized Schwarz Waveform Relaxation algorithms have been developed over the last few years for the computation in parallel of evolution problems. In this paper, we present their main features.

1 Introduction

Model complexity in today's computation emphasizes the need for coupling models in different geographical zones. For evolution problems, it is desirable to design a coupling process where, the lesser subdomain boundary information is exchanged the better. This goal is different from the usual domain decomposition purpose, where either the scheme is explicit, and the exchange of information takes place every time-step, or the scheme is implicit (leading to a steady problem, usually elliptic), and domain decomposition techniques can be used, see [16] and [18].

The Schwarz Waveform Relaxation algorithms take their name from the waveform relaxation algorithms, developed in circuit simulation, see [3], and Schwarz method for solving in parallel, partial differential equations of elliptic type, see [17, 12]. The purpose is to solve the space-time partial differential equation in each subdomain in parallel, and to transmit domain boundary information to the neighbors at the end of the time interval. The basic idea comes from the world of absorbing boundary conditions: for a model problem, approximations of the Dirichlet-Neumann map are developed, which can be written in the Fourier variables. These approximations lead to transmission conditions which involve time and tangential derivatives. The coefficients in these transmission conditions are in turn computed so as to optimize the convergence factor in the algorithm. This process can be written as a complex best approximation problem of a homographic type, and solved either explicitly or asymptotically. This gives a convergent algorithm that we call Optimized Schwarz Waveform Relaxation algorithm, which outperforms the classical one, i.e. where transmission is made only by exchange of Dirichlet data. It can be performed with or without overlap, and convergence is in any case much faster. It can be used with any high performance numerical method in the subdomains, extended to variables coefficients by a freezing process, [14], and to systems of equations, [13]. Finally, it can be used to couple different discretizations in different subdomains, and acts as a preconditioner for the full interface problem in space time [4].

The purpose of this paper is to present the main features of the optimized Schwarz waveform relaxation algorithms, to give a few proofs which are unpublished and which illustrate our mathematical techniques. For an extensive historical presentation, see [1], and for examples of applications see [9].

2 Description of the Schwarz Waveform Relaxation Algorithm

Suppose we want to solve an evolution equation $\mathcal{L}u = f$ in the domain Ω on the time domain $(0,T)$, with initial data u_0, and boundary conditions which will not be considered in this paper. Ω is split into subdomains Ω_j, $1 \leq j \leq J$, overlapping or non-overlapping. For each index j, $\mathcal{V}(j)$ is the set of indices of the neighbors of Ω_j, and we write for $k \geq 1$

$$\left\{\begin{aligned} \mathcal{L}u_j^k &= f && \text{in } \Omega_j \times (0,T), \\ u_j^k(\cdot,0) &= u_0 && \text{in } \Omega_j, \\ \mathcal{B}_{jl}u_j^k &= \mathcal{B}_{jl}u_l^{k-1} && \text{on } \partial\Omega_j \cap \bar{\Omega}_l \times (0,T), l \in \mathcal{V}(j), \end{aligned}\right. \tag{1}$$

with an initial guess $\mathcal{B}_{jl}u_l^0$ on the interfaces. This algorithm can be viewed as a Jacobi type iterative method, or as a preconditioner for an interface problem. We are only interested here in discussing the transmission conditions.

3 Classical Schwarz Waveform Relaxation Algorithm

As presented in the original paper [17], and analyzed in [12], the method concerns the Laplace equation and Dirichlet transmission conditions, *i.e.* $\mathcal{B}_j \equiv I$ and subdomains overlap. The algorithm is convergent, and the larger the overlap, the faster the method. In the evolution case, this algorithm is also convergent, but the mode of convergence depends on the type of equation. The starting point of our research was the example of the advection-diffusion equation, presented in DD11, [8]. The convergence curve exhibits a linear behavior for large time intervals, and is superlinear for small time intervals. The behavior is similar in higher dimension, [14].

For the wave equation, due to the finite speed of propagation, the convergence takes place in a finite number of steps $n_0 > \frac{cT}{L}$, where c is the wave speed, and L the size of the overlap. We showed in [7] an example, with a finite differences discretization, which shows that the error decays very slowly until iteration n_0, and reaches 10^{-12} at iteration n_0.

The case of the Schrödinger equation is also very interesting, and has been addressed in [11]. See also the contribution by J. Szeftel in the minisymposium "Domain Decomposition Methods Motivated by the Physics of the Underlying Problem" in this issue. The Dirichlet transmission creates a highly oscillatory solution, and the convergence begins late, and this phenomenon gets worse as the final time increases.

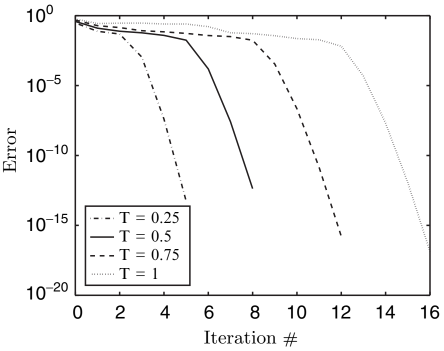


Fig. 1. Schrödinger equation: error with Dirichlet transmission conditions as a function of the iteration number for several final times.

We now describe a strategy to obtain more efficient transmission between subdomains.

4 Optimal Transmission Conditions

Let $\mathcal{L}$ denote a partial differential operator in time and space, which is elliptic in the space variables. For a domain Ω in $\mathbb{R}^n$, the Dirichlet-Neumann map $\mathcal{T}_\Omega$ maps a function h defined on $\partial\Omega \times (0,T)$ to the normal derivative $\frac{\partial u}{\partial n}$ where n is the unit exterior normal to $\partial\Omega$, and u the solution of $\mathcal{L}u = 0$ in $\Omega \times (0,T)$, with vanishing initial data. The importance of this map lies in the important result concerning a domain decomposition in layers: if it is used as transmission operator on the boundaries of the subdomains, then the convergence is achieved in J iterations, see in [15] a formal proof for an elliptic problem.

4.1 Example: the Advection-Diffusion Equation

We consider the operator

$$\mathcal{L} = \partial_t + (\mathbf{a} \cdot \boldsymbol{\nabla}) - \nu\Delta + cId,$$

with $\mathbf{a} = (a, \mathbf{b})$, $a \in \mathbb{R}^+, \mathbf{b} \in \mathbb{R}^{n-1}$. We search for the Dirichlet-Neumann map for a half-space $\mathbb{R}^\pm \times \mathbb{R}^{n-1}$, and denote by x the first coordinate, and $\mathbf{y}$ the $(n-1)-$dimensional coordinate. By Fourier transform $t \leftrightarrow \omega, \mathbf{y} \leftrightarrow \boldsymbol{\zeta}$, we see that the Fourier transform $\hat{w} = \mathcal{F}w$ of any solution of $\mathcal{L}w = 0$ is solution of the ordinary differential equation

$$-\nu \frac{\partial^2 \hat{w}}{\partial x^2} + a \frac{\partial \hat{w}}{\partial x} + \big(i(\omega + \mathbf{b} \cdot \boldsymbol{\zeta}) + \nu |\boldsymbol{\zeta}|^2 + c\big)\hat{w} = 0, \tag{2}$$

with characteristic roots

$$r_\pm(\boldsymbol{\zeta}, \omega) = \frac{a \pm \sqrt{\delta(\boldsymbol{\zeta}, \omega)}}{2\nu}, \quad \delta(\boldsymbol{\zeta}, \omega) = a^2 + 4\nu(i(\omega + \mathbf{b} \cdot \boldsymbol{\zeta}) + \nu|\boldsymbol{\zeta}|^2 + c). \tag{3}$$

The complex square root in this text is always with strictly positive real part. In order to work with at least square integrable functions in time and space, we seek for solutions which do not increase exponentially in time. Since the real parts of the roots, $\Re r_\pm$, are such that $\Re r_+ > 0$ and $\Re r_- < 0$, the Dirichlet-Neumann map $\mathcal{T}_+$ for $(L, +\infty) \times \mathbb{R}^{n-1}$ (resp. $\mathcal{T}_-$ for $(-\infty, L) \times \mathbb{R}^{n-1}$) is given by

$$\begin{gathered} \mathcal{F}\mathcal{T}_\pm(h)(\boldsymbol{\zeta}) = r_\mp(\boldsymbol{\zeta}, \omega)\mathcal{F}h(\boldsymbol{\zeta}), \\ \mathcal{T}_\pm = \frac{a \mp \sqrt{a^2 + 4\nu(\partial_t + \mathbf{b} \cdot \boldsymbol{\nabla} - \nu \Delta_{\mathbf{y}} + c)}}{2\nu}. \end{gathered} \tag{4}$$

Since the operator $\mathcal{L}$ has constant coefficients, the Dirichlet-Neumann maps do not depend on L. We consider J subdomains $\Omega_j = (a_j, b_j) \times \mathbb{R}^{n-1}$, with $a_1 = -\infty$, $b_J = +\infty$, and $a_j \leq b_{j-1} < b_j$ for $1 \leq j \leq J$, and transmission operators of the form

$$\mathcal{B}_{ij} \equiv \partial_x - S_{ij}(\partial_{\mathbf{y}}, \partial_t).$$

The Fourier transform of the error $e_j^k = u - u_j^k$ in each subdomain is given by

$$\mathcal{F}e_j^k = \alpha_j^k e^{r_+ x} + \beta_j^k e^{r_- x}, \qquad 1 \leq j \leq J, \tag{5}$$

with $\beta_1^k = 0$ and $\alpha_J^k = 0$.

Theorem 1. *With the transmission conditions $S_{jj-1} = \mathcal{T}_-, S_{jj+1} = \mathcal{T}_+$, the algorithm is optimal: the convergence is achieved in J iterations.*

Proof. Inserting (5) in the transmission conditions, we get at each step k a system of $2J$ equations with $2J$ unknowns (α_j^k, β_j^k). In the case of the theorem, the system reduces to $\alpha_j^k = \alpha_{j+1}^{k-1}$ and $\beta_j^k = \beta_{j-1}^{k-1}$. Since $\beta_1^1 = 0$ and $\alpha_J^1 = 0$, we deduce that $\alpha_j^J = 0$ and $\beta_j^J = 0$ for all j. Thus at iteration J, the solution of the algorithm is equal to u in each subdomain.

Note that the result still holds for any partial differential equation with constant coefficients, like Schrödinger equations (see [11]), or even systems, like the shallow-water system, see [13].

4.2 The Quasi Optimal Algorithm in One Dimension

The transparent operator is global in time and space. When used in the context of absorbing boundary conditions, enormous efforts have been made for the approximation of the Dirichlet-Neumann map by local operators, in order to obtain sparse matrices in the actual computations (see [10] in the context of parabolic equations). However in one dimension, we have to transmit informations on the interface over the whole time interval. Therefore we can afford to use pseudodifferential operators in time. For the advection-diffusion equation, we find in [6] a discrete Dirichlet-Neumann map for several numerical schemes, *e.g.* Euler and Crank-Nicolson. We have used this strategy for the Schrödinger equation, and we have shown that the convergence is achieved in very few iterations, even with non constant potential, see [11]. We intend to explore this direction in higher dimension.

5 Optimized SWR Algorithm for the Advection-Diffusion Equation

We now try to improve locally the transmission between the subdomains. Therefore we restrict ourselves to the case of two subdomains, and we develop the analysis in $\mathbb{R}^n \times (0,T)$.

5.1 Partial Differential Transmission Conditions

In this part, we write differential transmission conditions as follows. We replace in $r_\pm$ the square root by a polynomial of degree lower than or equal to 1, *i.e.* we set

$$\tilde{r}_\mp = \frac{a \pm \left(p + q\left(i(\omega + \mathbf{b}\cdot\boldsymbol{\zeta}) + \nu|\boldsymbol{\zeta}|^2\right)\right)}{2\nu},$$

with real parameters p and q to be chosen. This defines the new transmission operators

$$\widetilde{\mathcal{B}}_{ii+1} = \partial_x - \frac{a - p_{ii+1}}{2\nu} + q_{ii+1}(\partial_t + \mathbf{b}\cdot\boldsymbol{\nabla} - \nu\Delta_y),$$

$$\widetilde{\mathcal{B}}_{ii-1} = \partial_x - \frac{a + p_{ii-1}}{2\nu} - q_{ii-1}(\partial_t + \mathbf{b}\cdot\boldsymbol{\nabla} - \nu\Delta_y),$$

with real parameters p_{ij} and q_{ij} to be chosen. In the case where $q_{ij} = 0$, the transmission condition reduces to Robin transmission condition, which is already used as a preconditioner for domain decomposition in the steady case. We call it transmission condition of order 0, whereas when $q_{ij} \neq 0$, we talk about first order transmission condition.

5.2 Well-Posedness and Convergence of the Algorithm

We examine in details the Robin case. It was proved in [14] that in the case of two half-spaces with only one coefficient $p > 0$, the boundary value problems are well-posed in suitable Sobolev spaces, and the algorithm is convergent. The proof relies on Lebesgue Theorem in the overlapping case. In the non overlapping case, the proof uses energy estimates as in [5], and therefore holds for space varying advection. We prove below that this result holds in any reasonable geometry, as depicted in Figure 2.

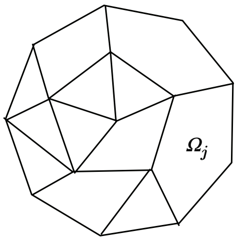


Fig. 2. Decomposition in space with nonoverlapping subdomains

The operators B_{jl} are given by

$$B_{jl} = \frac{\partial}{\partial_{n_j}} - \frac{\mathbf{a} \cdot \mathbf{n_j} - p_{jl}}{2\nu}, \tag{6}$$

where n_j is the normal to $\partial\Omega_j$, exterior to Ω_j.

Theorem 2. *In the nonoverlapping case, the domain decomposition algorithm* (1) *with transmission conditions* (6) *converges for any choice of the positive coefficients p_{jl} with $p_{jl} = p_{lj}$, provided each domain is visited infinitely many times.*

Proof. In order to shorten the proof, we work here with the heat equation. We write the equation for the error $e_j^k = u - u_j^k$ in Ω_j. We multiply it by e_j^k and integrate by parts:

$$\frac{1}{2}\frac{d}{dt}\|e_j^k\|^2_{\Omega_j} + \nu\|e_j^k\|^2_{\Omega_j} - \sum_{l \in \mathcal{V}(j)} \int_{\Gamma_{jl}} \nu \frac{\partial e_j^k}{\partial_{n_j}} e_j^k ds = 0.$$

We rewrite the boundary term as

$$-\nu \int_{\Gamma_{jl}} \frac{\partial e}{\partial_{n_j}} e\, ds = \frac{1}{2p_{jl}} \left[\int_{\Gamma_{jl}} \left(\nu \frac{\partial e}{\partial_{n_j}} - \frac{p_{jl}}{2} e \right)^2 ds - \int_{\Gamma_{jl}} \left(\nu \frac{\partial e}{\partial_{n_j}} + \frac{p_{jl}}{2} e \right)^2 ds \right],$$

and obtain

$$\frac{1}{2}\frac{d}{dt}\|e_j^k\|_{\Omega_j}^2 + \nu\|e_j^k\|_{\Omega_j}^2 + \sum_{l\in\mathcal{V}(j)}\int_{\Gamma_{jl}} \frac{1}{2p_{jl}}\left[\nu\frac{\partial e_j^k}{\partial_{n_j}} - \frac{p_{jl}}{2}\,e_j^k\right]^2 ds$$
$$= \sum_{l\in\mathcal{V}(j)}\int_{\Gamma_{jl}} \frac{1}{2p_{jl}}\left[\nu\frac{\partial e_j^k}{\partial_{n_j}} + \frac{p_{jl}}{2}\,e_j^k\right]^2 ds.$$

We use on the right hand side the transmission condition:

$$\frac{1}{2}\frac{d}{dt}\|e_j^k\|_{\Omega_j}^2 + \nu\|e_j^k\|_{\Omega_j}^2 + \sum_{l\in\mathcal{V}(j)}\int_{\Gamma_{jl}} \frac{1}{2p_{jl}}\left[\nu\frac{\partial e_j^k}{\partial_{n_j}} - \frac{p_{jl}}{2}\,e_j^k\right]^2 ds$$
$$= \sum_{l\in\mathcal{V}(j)}\int_{\Gamma_{jl}} \frac{1}{2p_{jl}}\left[\nu\frac{\partial e_l^{k-1}}{\partial_{n_j}} + \frac{p_{jl}}{2}\,e_l^{k-1}\right]^2 ds$$
$$= \sum_{l\in\mathcal{V}(j)}\int_{\Gamma_{jl}} \frac{1}{2p_{jl}}\left[-\nu\frac{\partial e_l^{k-1}}{\partial_{n_l}} + \frac{p_{jl}}{2}\,e_l^{k-1}\right]^2 ds.$$

Summing up on all domains, we have on the left a boundary term at step k and on the right the same term at step $k-1$. Summing up on all steps k, we obtain

$$\frac{d}{dt}\sum_{k=0}^{K}\sum_{j=1}^{J}\|e_j^k\|_{\Omega_j}^2 + 2\nu\sum_{k=0}^{K}\sum_{j=1}^{J}\|e_j^k\|_{\Omega_j}^2 + \sum_{j=1}^{J}\sum_{l\in\mathcal{V}(j)}\int_{\Gamma_{jl}} \frac{1}{p_{jl}}\left[\nu\frac{\partial e_j^K}{\partial_{n_j}} - \frac{p_{jl}}{2}\,e_j^K\right]^2 ds$$
$$= \sum_{j=1}^{J}\sum_{l\in\mathcal{V}(j)}\int_{\Gamma_{jl}} \frac{1}{p_{jl}}\left[\nu\frac{\partial e_l^0}{\partial_{n_l}} - \frac{p_{jl}}{2}\,e_l^0\right]^2 ds.$$

which proves that $\sum_{j=1}^{J}\|e_j^k\|_{L^\infty(0,T;\Omega_j)}^2 + 2\nu\sum_{j=1}^{J}\|e_j^k\|_{L^2(0,T;\Omega_j)}^2$ is bounded. By assumption, we have an infinite sequence e_j^k, and for every j, e_j^k tends to 0 as k tends to infinity.

There is no proof available for the overlapping domains in general geometry. Concerning the case $q \neq 0$, there is a proof of well-posedness and convergence for the layered case in [2]. The well-posedness in general geometry, even in the nonoverlapping case, has not yet been addressed.

5.3 Optimization of the Convergence Factor

In order to improve the performances of the method, it is important to optimize the convergence between two subdomains. In this case, the notations are much simpler, there are only two parameters p and q, and we write

$$\mathcal{F}e_j^{k+2}(0,\boldsymbol{\zeta},\omega) = \rho(\boldsymbol{\zeta},\omega,P,L)\mathcal{F}e_j^k(0,\boldsymbol{\zeta},\omega),$$

with the convergence factor defined for a polynomial $P \in \mathbf{P}_n$ by

$$\rho(\boldsymbol{\zeta},\omega,P,L) = \left(\frac{P\left(i(\omega+\mathbf{b}\cdot\boldsymbol{\zeta})+\nu|\boldsymbol{\zeta}|^2\right) - \sqrt{\delta(\boldsymbol{\zeta},\omega)}}{P\left(i(\omega+\mathbf{b}\cdot\boldsymbol{\zeta})+\nu|\boldsymbol{\zeta}|^2\right) + \sqrt{\delta(\boldsymbol{\zeta},\omega)}} \right)^2 e^{-2\frac{L}{\nu}\sqrt{\delta(\boldsymbol{\zeta},\omega)}}.$$

The convergence factor has two terms: the exponential one comes from the overlap, and kills the high frequencies in time or space, whereas the fractional term comes from the transmission condition and tends to 1 as frequencies in time or space become high. Since the overlap has to be small for computational reasons, and also since it is of interest to have a nonoverlapping strategy, we want to make the fractional part as small as possible. In a real computation, only frequencies supported by the grid are relevant, ω and $\boldsymbol{\zeta}$ live in the compact set

$$K = \{(\omega,\boldsymbol{\zeta}),\ \omega_m \le |\omega| \le \omega_M,\ \zeta_{j,m} \le |\zeta_j| \le \zeta_{j,M},\ j=1,\cdots,n-1\}.$$

with $\omega_m = \pi/T$, $\omega_M = \pi/\Delta t$, where Δt is the time step, and similarly $\zeta_{j,m} = \pi/Y_j$, $\zeta_{j,M} = \pi/\Delta y_j$, where Y_j is the length in the y_j direction and Δy_j the mesh size. The optimization of the convergence factor is formulated as a best approximation problem in $\mathbf{P}_n$ for $n=0$ or 1,

$$\sup_{(\boldsymbol{\zeta},\omega)\in K} |\rho(\boldsymbol{\zeta},\omega,P_n^*)| = \inf_{P\in\mathbf{P}_n} \sup_{(\boldsymbol{\zeta},\omega)\in K} |\rho(\boldsymbol{\zeta},\omega,P)|. \tag{7}$$

Problem (7) has been solved in a more general setting in [2], and exact formulas in the one-dimensional case, together with asymptotic results, have been given. A general result asserts that the infimum is strictly small than 1, the problem has a unique solution, and furthermore, the solution is a real polynomial, and is the solution of the best approximation problem set in the space of real polynomials. Here we study Problem (7) for $n=0$, which corresponds to Robin transmission conditions, and for $L=0$, which means without overlap. In particular we have $p^* > 0$. We are going now to characterize p^*.

We choose new variables $\tau = \omega + \mathbf{b}\cdot\boldsymbol{\zeta}$, $\eta = \sqrt{a^2+4\nu c+4\nu|\boldsymbol{\zeta}|^2}$, and use $\xi = \Re\sqrt{\delta}$. With the new notations we have

$$|\rho(\boldsymbol{\zeta},\omega,p)| = R(\tau,\eta,p) = \frac{(\xi-p)^2+\xi^2-\eta^2}{(\xi+p)^2+\xi^2-\eta^2},\ \xi(\tau,\eta) = \sqrt{(\eta^2+\sqrt{\tau^2+\eta^4}\,)/2}\ ,$$

and the best approximation problem for ρ has the same solution as the one for R in the subspace of $\mathbb{R}^2$, $D = [\eta_m,\eta_M]\times[\tau_m,\tau_M]$, with $\tau_m = 0$. A point M in the plane will be defined by its coordinates (η,τ) and we will call A_j the edges of D: $A_1 = (\eta_m,\tau_m)$, $A_2 = (\eta_M,\tau_m)$, $A_3 = (\eta_M,\tau_M)$, $A_4 = (\eta_m,\tau_M)$. We will interchangeably use $R(M,p)$ or $R(\tau,\eta,p)$.

The upper bounds τ_M,η_M are inversely proportional to the time and space steps. Depending on whether an explicit or implicit scheme is used, we can have Δt of the order of Δy or Δy^2, respectively. Therefore we assume here that $\tau_M = C\eta_M^\beta$, with $\beta = 1$ or 2.

Theorem 3. *For $n = 0$ and $L = 0$, problem (7) has a unique solution $p^* > 0$. If η_M is large and $\tau_M = C\eta_M^\beta$, it is given by*

$$p^* = \sqrt{\frac{\xi(A_1)\sqrt{\tau_M^2+\eta_M^4}-\xi(A_3)\sqrt{\tau_m^2+\eta_m^4}}{\xi(A_3)-\xi(A_1)}} \text{ if } \beta = 1, \text{ or } \beta = 2 \text{ and } C < C_0,$$
$$p^* = \sqrt{\eta_m^2 + 2\eta_m\xi(A_4)} \text{ if } \beta = 2 \text{ and } C > C_0.$$

Proof. We proceed in several steps.

Lemma 1. *For any positive p, the maximum of $(\tau, \eta) \mapsto R(\tau, \eta, p)$ on D is reached on one of the edges of D.*

The analytic function ρ can reach extrema only on the boundary of the domain. The partial derivatives of R with respect to τ and η delimit regions in the plane, see Figure 3, from which we infer that an extremum on any segment of the boundary, distinct from the edges, can only be a minimum.

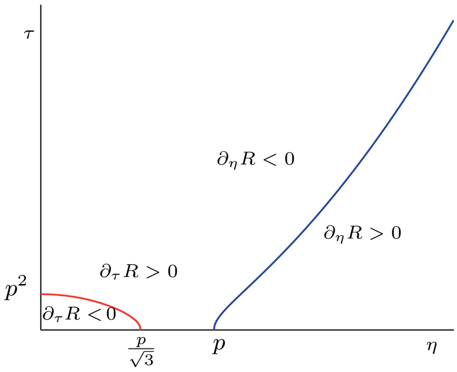


Fig. 3. Regions delimited by the zeros of the partial derivatives of R

Lemma 2. *If either η_M or τ_M is large, there exists a unique $p_* > 0$, such that $R(A_1, p_*) = R(A_3, p_*)$, and it is given by*

$$p_* = \sqrt{\frac{\xi(A_1)\sqrt{\tau_M^2 + \eta_M^4} - \xi(A_3)\sqrt{\tau_m^2 + \eta_m^4}}{\xi(A_3) - \xi(A_1)}}.$$

Further, there exists a unique $p_{**} > 0$, such that $R(A_1, p_{**}) = R(A_4, p_{**})$, and it is given by

$$p_{**} = \sqrt{\eta_m^2 + 2\eta_m\xi(A_4)}.$$

This can be seen by writing for any points M_1, M_2 in D,

$$R(M_1, p) - R(M_2, p) = \frac{4p(\xi(M_2)-\xi(M_1))}{D(M_1,M_2)}\left[Q(M_1, M_2) - p^2\right],$$
$$Q(M_1, M_2) = 2\xi(M_2)\xi(M_1) + \frac{\xi(M_2)\eta_1^2-\xi(M_1)\eta_2^2}{\xi(M_2)-\xi(M_1)},$$

with a positive denominator $D(M_1, M_2)$, and discussing the sign of Q.

Lemma 3. *For large η_M we have:*

1. *If $\beta = 1$, or $\beta = 2$ and $C < C_0$,* $\sup\limits_{M \in D} R(M, p_*) = R(A_1, p_*) = R(A_3, p_*)$,

$$p_* \sim C_* \sqrt{\eta_m \eta_M}, \quad \sup_{M \in D} R(M, p_*) \sim 1 - 4\frac{\eta_m}{p_*},$$

$$C_* = 1 \textit{ if } \beta = 1, \quad C_* = \left(\frac{2(C^2+1)}{1 + \sqrt{(C^2+1)}} \right)^{1/4} \textit{ if } \beta = 2.$$

2. *If $\beta = 2$ and $C > C_0$,* $\sup\limits_{M \in D} R(M, p_{**}) = R(A_1, p_{**}) = R(A_4, p_{**})$,

$$p_{**} \sim (2C)^{1/4} \sqrt{\eta_m \eta_M}, \quad \sup_{M \in D} R(M, p_*) \sim 1 - 4\frac{\eta_m}{p_*}.$$

$$C_0 \textit{ is the only positive root of the equation } \frac{C^2+1}{1+\sqrt{(C^2+1)}} = C.$$

These results are obtained by comparing the asymptotic values of R at the edges.

Lemma 4. *The values p_* and p_{**} in Lemma 3 are in each case a strict local minimum for the function $p \mapsto \sup\limits_{(\tau,\eta) \in D} R(\tau, \eta, p)$.*

Proof. For any positive p, we define $\bar{R}(p) = \sup_{(\tau,\eta) \in D} R(\tau, \eta, p)$, and $\mu(p) = \frac{1+\bar{R}(p)}{1-\bar{R}(p)}$. We write

$$R(\tau, \eta, p) - \bar{R}(p) = (1 - \bar{R}(p)) \frac{\bar{Q}(\tau, \eta, p, \mu)}{(\xi + p)^2 + \xi^2 - \eta^2}$$

with $\bar{Q}(\tau, \eta, p, \mu) = 2\xi^2 - 2\mu p \xi - \eta^2 + p^2$. In the sequel, we will consider $\bar{Q}$ as a polynomial in the independent variables τ, η, p and μ. Defining $\mu_* = \mu(p_*)$ we have

$$\bar{Q}(A_1, p_*, \mu_*) = \bar{Q}(A_2, p_*, \mu_*) = 0,$$

and $\bar{Q}(M, p_*, \mu_*) \leq 0$ for any M in D. Now for p_* to be a strict local minimum, it is sufficient that there exists no variation $(\delta p, \delta\mu)$ with $\delta\mu < 0$, such that $\bar{Q}(A_j, p_* + \delta p, \mu_* + \delta\mu) < 0$. By the Taylor formula, it is equivalent to proving that for $j = 1$ and $j = 3$, we can not have $\delta p(p_* - \mu_* \xi_j) - p_* \xi_j \delta\mu < 0$ for $\delta\mu < 0$. From the asymptotic behavior, we see that for $j = 1$, it gives $\delta p - \delta\mu < 0$, and for $j = 3$, it gives $-\delta p - \delta\mu < 0$, which together contradicts the fact that $\delta\mu < 0$. The arguments hold in all cases.

Another general result in [2] asserts that any strict local minimum is a global minimum. Therefore p_* is a global minimum, and equal to p^*, which concludes the proof of the theorem in the first case. The second proof is similar.

5.4 Numerical Results

To show that the optimization process is indeed important, we draw in Figure 4, for eight subdomains in one dimension, the convergence rates of the algorithm with Dirichlet transmission conditions compared to the same algorithm with the new optimized transmission conditions of zeroth or first order. The convergence curves are similar in the 2D case [14].

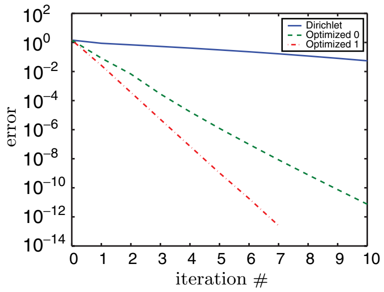


Fig. 4. Convergence rates

6 Conclusion

We have presented the main features of the Optimized Schwarz Waveform Relaxation algorithms, specifically for the advection-diffusion equation. Extension to nonlinear equations, and application to real problems in ocean modeling or combustion or waste disposal simulations will be the next step.

References

[1] D. Bennequin, M.J. Gander, and L. Halpern. Optimized Schwarz waveform relaxation methods for convection reaction diffusion problems. Technical Report 2004-24, LAGA, Université Paris 13, 2004.

[2] D. Bennequin, M.J. Gander, and L. Halpern. A homographic best approximation problem with application to optimized Schwarz waveform relaxation. Technical report, CNRS, 2006. : https://hal.archives-ouvertes.fr/hal-00111643.

[3] K. Burrage. *Parallel and Sequential Methods for Ordinary Differential Equations*. Oxford University Press Inc., New York, 1995.

[4] P. D'Anfray, L. Halpern, and J. Ryan. New trends in coupled simulations featuring domain decomposition and metacomputing. *M2AN*, 36(5):953–970, 2002.

[5] B. Després. Décomposition de domaine et problème de Helmholtz. *C.R. Acad. Sci. Paris*, 1(6):313–316, 1990.

[6] M. Ehrhardt. *Discrete Artificial Boundary Conditions.* PhD thesis, Technischen Universität Berlin, Berlin, 2001.

[7] M.J. Gander and L. Halpern. Absorbing boundary conditions for the wave equation and parallel computing. *Math. Comp.*, 74(249):153–176, 2004.

[8] M.J. Gander, L. Halpern, and F. Nataf. Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation. In C.-H. Lai, P.E. Bjørstad, M. Cross, and O.B. Widlund, editors, *11th International Conference on Domain Decomposition in Science and Engineering*, pages 253–260. Domain Decomposition Press, 1999.

[9] L. Halpern. Absorbing boundary conditions and optimized Schwarz waveform relaxation. *BIT*, 52(4):401–428, 2006.

[10] L. Halpern and J. Rauch. Absorbing boundary conditions for diffusion equations. *Numer. Math.*, 71:185–224, 1995.

[11] L. Halpern and J. Szeftel. Optimized and quasi-optimal Schwarz waveform relaxation for the one dimensional Schrödinger equation. Technical report, CNRS, 2006. http://hal.ccsd.cnrs.fr/ccsd-00067733/en/.

[12] P.-L. Lions. On the Schwarz alternating method. I. In R. Glowinski, G.H. Golub, G.A. Meurant, and J. Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 1–42. SIAM, 1988.

[13] V. Martin. A Schwarz waveform relaxation method for the viscous shallow water equations. In R. Kornhuber, R.H.W. Hoppe, J. Périaux, O. Pironneau, O.B. Widlund, and J. Xu, editors, *Proceedings of the 15th International Domain Decomposition Conference.* Springer, 2003.

[14] V. Martin. An optimized Schwarz waveform relaxation method for unsteady convection diffusion equation. *Appl. Numer. Math.*, 52(4):401–428, 2005.

[15] F. Nataf, F. Rogier, and E. de Sturler. Domain decomposition methods for fluid dynamics, Navier-Stokes equations and related nonlinear analysis. *Edited by A. Sequeira, Plenum Press Corporation*, pages 367–376, 1995.

[16] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford Science Publications, 1999.

[17] H.A. Schwarz. Über einen Grenzübergang durch alternierendes Verfahren. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich*, 15:272–286, May 1870.

[18] A. Toselli and O.B. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics.* Springer, 2005.

Integration of Sequential Quadratic Programming and Domain Decomposition Methods for Nonlinear Optimal Control Problems

Matthias Heinkenschloss[1] and Denis Ridzal[2]

[1] Department of Computational and Applied Mathematics, MS-134, Rice University, 6100 Main Street, Houston, TX 77005-1892, USA. heinken@rice.edu

[2] Computational Mathematics and Algorithms, MS-1320, Sandia National Laboratories [†], P.O. Box 5800, Albuquerque, NM 87185-1320, USA. dridzal@sandia.gov

Summary. We discuss the integration of a sequential quadratic programming (SQP) method with an optimization-level domain decomposition (DD) preconditioner for the solution of the quadratic optimization subproblems. The DD method is an extension of the well-known Neumann-Neumann method to the optimization context and is based on a decomposition of the first order system of optimality conditions. The SQP method uses a trust-region globalization and requires the solution of quadratic subproblems that are known to be convex, hence solving the first order system of optimality conditions associated with these subproblems is equivalent to solving these subproblems. In addition, our SQP method allows the inexact solution of these subproblems and adjusts the level of exactness with which these subproblems are solved based on the progress of the SQP method. The overall method is applied to a boundary control problem governed by a semilinear elliptic equation.

1 Introduction

Optimization algorithms for PDE constrained optimization problems which use second order derivative information require the solution of large-scale linear systems that involve linearizations of the governing PDE and its adjoint. Domain decomposition methods can be used to effectively solve these subproblems. In this paper we discuss the integration of a sequential quadratic programming (SQP) method with an optimization-level domain decomposition (DD) preconditioner for the quadratic optimization subproblems arising inside the SQP method.

[†] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under Contract DE-AC04-94-AL85000.

As an example problem we consider the following boundary control problem with states y and controls u.

$$\text{Minimize } \frac{1}{2}\int_{\Omega} l(y(x),x)dx + \frac{\alpha}{2}\int_{\partial\Omega_c} u^2(x)dx \tag{1a}$$

subject to

$$-\epsilon\Delta y(x) + g(y(x),x) = 0, \qquad x \in \Omega, \tag{1b}$$

$$y(x) = 0, \qquad x \in \partial\Omega \setminus \partial\Omega_c, \tag{1c}$$

$$\epsilon\frac{\partial}{\partial\mathbf{n}}y(x) = u(x), \qquad x \in \partial\Omega_c. \tag{1d}$$

Here $\alpha > 0$ is a given parameter. Because of page restrictions, we limit our presentation to semilinear elliptic optimal control problems in which the functions g, l and the problem data are such that the optimal control problem (1) has a solution $y \in H^1(\Omega)$, $u \in L^2(\partial\Omega_c)$. Furthermore, we assume that the state equation and the objective functional are twice Fréchet differentiable in $H^1(\Omega) \times L^2(\partial\Omega_c)$, that the linearized state equation has a unique solution in $H^1(\Omega)$ that depends continuously on the right hand side and boundary data, and that a second order sufficient optimality condition is satisfied at the solution. These assumptions are satisfied for the example problem considered in Section 4 as well as those discussed, e.g., in [8, 11, 10, 16]. To establish Fréchet differentiability and second order optimality conditions for other semilinear elliptic optimal control problems, however, a more involved setting and analysis is required. See, e.g., [17, 24]. Our approach can be adapted to many of those problems. We note that our approach can also be applied to the optimal control of incompressible Navier-Stokes equations. However, since these are systems of PDEs and because the compatibility conditions that are implied by the incompressibility condition require a careful treatment, the presentation of our approach for the optimal control of incompressible Navier-Stokes equations is too lengthy and will be given elsewhere.

In this work we use the optimization-level DDM introduced in [3, 13] for the solution of convex quadratic subproblems arising in the solution of (1). These optimization-level DDMs are extensions of the well known Neumann-Neumann methods (see, e.g., [20, 22, 23]) or the Robin-Robin methods for problems with advection (see, e.g., [1, 2]) from the PDE to the optimization context. In particular, all subproblem solves that arise in our DDM correspond to the solution of subdomain optimal control problems, which are essentially smaller copies of the original one. We note that our DDM is not the only optimization-level DDM. By 'optimization-level' we mean that the DDM is applied directly to the optimization problem, not individually to the state and adjoint PDEs. For example the DDM used in [18, 19] may be viewed as the optimization-level version of the restrictive additive Schwarz method discussed, e.g., in [6]. Heinkenschloss and Nguyen [12] analyze an optimization-level additive Schwarz method. Overall, however, the theoretical properties of optimization-level DDMs are still relatively poorly understood. We also point out that many optimization-level DDMs, including ours and the ones in [18, 19] are obtained by applying DDM to the system of optimality conditions, the so-called KKT system. This is only possible if the system of optimality conditions is necessary and sufficient, i.e., if the optimization problem is convex. This restriction is not always made explicit enough and is typically important for nonlinear PDE constrained optimization problems.

SQP algorithms coupled with DDMs have been discussed in [4, 5, 18, 19].

Our SQP method builds on the works [15, 21]. There are important features that distinguish our SQP from those in [4, 5, 18, 19]. First, all quadratic subproblems that arise in our SQP method are known a-priori to be convex. This allows us to apply optimization-level DDMs to these subproblems, which are based on a decomposition of the first order optimality conditions, the so-called KKT conditions. Since our subproblems are convex, solving these optimality systems is equivalent to solving the quadratic optimization problems. Secondly, we allow the inexact solution of the large scale linear KKT systems that arise as subproblems inside the SQP algorithms, and provide a rigorous way to control the level of inexactness with which these systems have to be solved. The level of inexactness is coupled to the progress of the SQP algorithm, which enables us to apply coarse, more inexpensive solves away from the solution. Our DDM is used as a preconditioner for the large scale linear KKT systems that arise in the SQP algorithm. Other preconditioners could be used as well. In particular, it is possible to incorporate the DD Krylov-Schur preconditioner used by [4, 5] or (restricted) additive Schwarz preconditioners as used by [18, 19].

2 Optimal Control of Advection Diffusion Equations

We begin with a discussion of our DD approach for convex linear-quadratic optimal control problems governed by an advection diffusion equation. The example problem is given as follows.

$$\text{Minimize } \frac{1}{2}\int_{\Omega}(y(x)-\widehat{y}(x))^2dx+\frac{\alpha}{2}\int_{\partial\Omega_c}u^2(x)dx \tag{2a}$$

subject to

$$-\epsilon\Delta y(x)+\mathbf{a}(x)\cdot\nabla y(x)+r(x)y(x)=f(x), \qquad x\in\Omega, \tag{2b}$$

$$y(x)=0, \qquad x\in\partial\Omega\setminus\partial\Omega_c, \tag{2c}$$

$$\epsilon\frac{\partial}{\partial\mathbf{n}}y(x)=u(x), \qquad x\in\partial\Omega_c, \tag{2d}$$

where $\partial\Omega_c$ is the control boundary, $\mathbf{a}, f, g, r, \widehat{y}$ are given functions, $\epsilon, \alpha > 0$ are given scalars, and $\mathbf{n}$ denotes the outward unit normal. Our main interest is not in this particular optimal control problem. As we will see in more detail later, our SQP method applied to (1) requires the repeated solution of convex linear-quadratic optimal control subproblems governed by linear elliptic PDEs. The governing PDEs in these linear-quadratic subproblems are of the form (2b-d), with $\mathbf{a}$, r, and f determined by the SQP algorithm. The objective function in these subproblems is slightly different from (2a) and is given by a quadratic model of the Lagrangian associated with (1). However, the problem structure of the SQP subproblems and that of (2) are close enough so that a study of (2) reveals how to deal with the subproblems arising in our SQP method for (1).

The system of necessary and sufficient optimality conditions for (2) is given by the adjoint equations

$$\begin{aligned} -\epsilon \Delta p(x) - \mathbf{a}(x)\cdot\nabla p(x) & \\ +(r(x) - \nabla\cdot\mathbf{a}(x))p(x) &= -(y(x)-\hat{y}(x)), && x\in\Omega, && \text{(3a)}\\ p(x) &= 0, && x\in\partial\Omega\setminus\partial\Omega_c, && \text{(3b)}\\ \epsilon\frac{\partial}{\partial\mathbf{n}}p(x) + \mathbf{a}(x)\cdot\mathbf{n}(x)\,p(x) &= 0, && x\in\partial\Omega_c, && \text{(3c)} \end{aligned}$$

by the equation

$$p(x) = \alpha u(x) \qquad x\in\Omega_c, \tag{4}$$

and by the state equation (2b-d).

We apply DD to the system of optimality conditions (2b-d), (3), (4). For simplicity, we consider the two-subdomain case only. Everything can be extended to more than two subdomains following the discussions in [3, 13]. We decompose Ω into two subdomains Ω_1, Ω_2 with interface $\Gamma = \overline{\Omega_1}\cap\overline{\Omega_2}$. The outer unit normal for subdomain i is denoted by $\mathbf{n}_i$. By γ_Γ we denote the trace operator and we define $V_\Gamma = \{\gamma_\Gamma v \ : \ v\in H^1(\Omega),\ v=0 \text{ on } \partial\Omega\setminus\partial\Omega_c\}$ We now split (2b-d), (3), (4) as follows. Given $y_\Gamma, p_\Gamma \in V_\Gamma$ and $i\in\{1,2\}$ we consider the system

$$\begin{aligned} -\epsilon\Delta y_i(x) + \mathbf{a}(x)\cdot\nabla y_i(x) + r(x)y_i(x) &= f(x) && \text{in } \Omega_i, && \text{(5a)}\\ y_i(x) &= 0 && \text{on } \partial\Omega_i\cap\partial\Omega\setminus\partial\Omega_c, && \text{(5b)}\\ \epsilon\frac{\partial}{\partial\mathbf{n}}y_i(x) &= u_i(x), && \text{on } \partial\Omega_i\cap\partial\Omega_c, && \text{(5c)}\\ y_i(x) &= y_\Gamma(x) && \text{on } \Gamma, && \text{(5d)} \end{aligned}$$

$$\begin{aligned} -\epsilon\Delta p_i(x) - \mathbf{a}(x)\cdot\nabla p_i(x) & \\ +(r(x)-\nabla\cdot\mathbf{a}(x))p_i(x) &= -(y_i(x)-\hat{y}(x)) && \text{in } \Omega_i, && \text{(5e)}\\ p_i(x) &= 0, && \text{on } \partial\Omega_i\cap\partial\Omega\setminus\partial\Omega_c, && \text{(5f)}\\ \epsilon\frac{\partial}{\partial\mathbf{n}}p_i(x) + \mathbf{a}(x)\cdot\mathbf{n}(x)\,p_i(x) &= 0, && \text{on } \partial\Omega_i\cap\partial\Omega_c, && \text{(5g)}\\ p_i(x) &= p_\Gamma(x) && \text{on } \Gamma, && \text{(5h)} \end{aligned}$$

$$\alpha u_i(x) - p_i(x) = 0 \qquad\qquad \text{on } \partial\Omega_c \cap \partial\Omega_i. \tag{5i}$$

The system (5) together with the interface conditions

$$\begin{aligned}
\left(\epsilon\tfrac{\partial}{\partial \mathbf{n}_i} - \tfrac{1}{2}\mathbf{a}(x)\mathbf{n}_i\right) y_i(x) &= -\left(\epsilon\tfrac{\partial}{\partial \mathbf{n}_j} - \tfrac{1}{2}\mathbf{a}(x)\mathbf{n}_j\right) y_j(x) \ x \in \partial\Omega_i \cap \partial\Omega_j,\\
\left(\epsilon\tfrac{\partial}{\partial \mathbf{n}_i} + \tfrac{1}{2}\mathbf{a}(x)\mathbf{n}_i\right) p_i(x) &= -\left(\epsilon\tfrac{\partial}{\partial \mathbf{n}_j} + \tfrac{1}{2}\mathbf{a}(x)\mathbf{n}_j\right) p_j(x) \ x \in \partial\Omega_i \cap \partial\Omega_j,
\end{aligned} \tag{6}$$

on Γ are equivalent to the original optimality system (2b-d), (3), (4).

It can be shown that for given $y_\Gamma, p_\Gamma \in V_\Gamma$ the system (5) has a unique solution (y_i, p_i, u_i). If we view (y_i, p_i, u_i), $i = 1, 2$, as a function of $y_\Gamma, p_\Gamma \in V_\Gamma$ defined through (5), then (6) becomes an equation in y_Γ, p_Γ. Since $y_\Gamma, p_\Gamma \in V_\Gamma$, $i = 1, 2$, depends on y_Γ, p_Γ in an affine linear way, (5), (6) can be written as

$$(S_1 + S_2)\begin{pmatrix} y_\Gamma \\ p_\Gamma \end{pmatrix} = r_1 + r_2, \tag{7}$$

where S_i, $i = 1, 2$, is applied to y_Γ, p_Γ by first solving (5) with $f = 0$ and then evaluating $\left(\epsilon\frac{\partial}{\partial \mathbf{n}_i} - \frac{1}{2}\mathbf{a}(x)\mathbf{n}_i\right) y_i(x)$, $\left(\epsilon\frac{\partial}{\partial \mathbf{n}_i} + \frac{1}{2}\mathbf{a}(x)\mathbf{n}_i\right) p_i(x)$. The right hand side is computed by solving (5) with $y_\Gamma = p_\Gamma = \mathbf{0}$ and then evaluating $\left(\epsilon\frac{\partial}{\partial \mathbf{n}_i} - \frac{1}{2}\mathbf{a}(x)\mathbf{n}_i\right) y_i(x)$, $\left(\epsilon\frac{\partial}{\partial \mathbf{n}_i} + \frac{1}{2}\mathbf{a}(x)\mathbf{n}_i\right) p_i(x)$.

One can show that (5) is the system of optimality conditions for a subdomain optimal control problem that is essentially a restriction of (2) to subdomain Ω_i, but with the additional interface boundary condition (5d) and with an additional interface normal derivative term in the objective that leads to (5h). See [3, 13].

One can also show that the subdomain operators S_i, $i = 1, 2$, are invertible and that

$$S_i^{-1}\begin{pmatrix} r_\Gamma^u \\ r_\Gamma^\lambda \end{pmatrix} = \begin{pmatrix} \gamma_\Gamma y_i \\ \gamma_\Gamma p_i \end{pmatrix},$$

where γ_Γ denotes the trace operator and where y_i, p_i are obtained by solving

$$-\epsilon\Delta y_i(x) + \mathbf{a}(x)\cdot\nabla y_i(x) + r(x) y_i(x) = 0 \qquad \text{in } \Omega_i, \tag{8a}$$

$$y_i(x) = 0 \qquad \text{on } \partial\Omega_i \cap \partial\Omega \setminus \partial\Omega_c, \tag{8b}$$

$$\epsilon\frac{\partial}{\partial \mathbf{n}} y_i(x) = u_i(x), \qquad \text{on } \partial\Omega_i \cap \partial\Omega_c, \tag{8c}$$

$$\epsilon\frac{\partial}{\partial \mathbf{n}_i} y_i(x) - \tfrac{1}{2}\mathbf{a}(x)\cdot\mathbf{n}_i y_i(x) = r_i^y(x) \qquad \text{on } \Gamma, \tag{8d}$$

$$\begin{aligned}
-\epsilon\Delta p_i(x) - \mathbf{a}(x)\cdot\nabla p_i(x)& \\
+(r(x) - \nabla\cdot\mathbf{a}(x))p_i(x) &= -(y_i(x) - \hat{y}(x)) && \text{in } \Omega_i, && \text{(8e)}\\
p_i(x) &= 0, && \text{on } \partial\Omega_i\cap\partial\Omega\setminus\partial\Omega_c, && \text{(8f)}\\
\epsilon\frac{\partial}{\partial\mathbf{n}}p_i(x) + \mathbf{a}(x)\cdot\mathbf{n}(x)\, p_i(x) &= 0, && \text{on } \partial\Omega_i\cap\partial\Omega_c, && \text{(8g)}\\
\epsilon\frac{\partial}{\partial\mathbf{n}_i}p_i(x) + \tfrac{1}{2}\mathbf{a}(x)\cdot\mathbf{n}_i p_i(x) &= r_i^p(x) && \text{on } \Gamma, && \text{(8h)}\\
\alpha u_i(x) - p_i(x) &= 0 && \text{on } \partial\Omega\cap\partial\Omega_i. && \text{(8i)}
\end{aligned}$$

See [3, 13]. One can show that (8) is the system of optimality conditions for a subdomain optimal control problem that is essentially a restriction of (2) to subdomain Ω_i, but with the additional interface boundary condition (8d) and with an additional interface boundary term in the objective that involves $y_i r_\Gamma^\lambda$ which leads to (8h).

We solve (7) using a preconditioned Krylov subspace method such as GMRES or sQMR with preconditioner $S_1^{-1} + S_2^{-1}$. As we have mentioned earlier, everything can be extended to the case of many subdomains. See [3, 13]. One can show that the discrete versions of S_i are Schur complements. They are symmetric and highly indefinite. The number of positive and negative eigenvalues is proportional to the number of discretized states y_i on the interface. While the observed performance of these methods is comparable to that of Neumann-Neumann (Robin-Robin) methods for elliptic PDEs, there is no theoretical explanation for this observed behavior in the optimization case yet.

3 Inexact Trust-Region-SQP Method

Many nonlinear optimal control problems can abstractly be written as a nonlinear programming problem (NLP) in Hilbert space,

$$\begin{aligned}
&\min\ f(x) && \text{(9a)}\\
&\text{s.t.}\ c(x) = 0, && \text{(9b)}
\end{aligned}$$

where $f : \mathcal{X} \to \mathbb{R}$ and $c : \mathcal{X} \to \mathcal{Y}$ for some Hilbert spaces $\mathcal{X}$ and $\mathcal{Y}$. In our example problem (1) we have $x = (y, u)$, $\mathcal{X} = H^1(\Omega) \times (L^2(\partial\Omega_c))^2$ and $\mathcal{Y} = (H^1(\Omega))'$, where $'$ is used to denote the dual, and $c(x) = 0$ represents the weak formulation of the semilinear elliptic equations (1b-d). The corresponding Lagrangian functional $L : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is given by $L(x,\lambda) = f(x) + \langle\lambda, c(x)\rangle_{\mathcal{Y}}$. We use subscript x to denote Fréchet derivatives with respect to x. Given estimates x_k, λ_k for the solution of (9) and corresponding Lagrange multiplier, SQP methods approximately solve

$$\begin{aligned}
&\min\ \frac{1}{2}\langle H_k s_k, s_k\rangle_{\mathcal{X}} + \langle\nabla_x L(x_k,\lambda_k), s_k\rangle_{\mathcal{X}} && \text{(10a)}\\
&\text{s.t.}\ c_x(x_k)s_k + c(x_k) = 0 && \text{(10b)}
\end{aligned}$$

and use the solution s_k to obtain a better approximation of the solution of (9). In (9) H_k is the Hessian $\nabla_{xx}L(x_k, \lambda_k)$ of the Lagrangian or a replacement thereof, obtained, e.g., using a quasi-Newton method. If x_k is sufficiently close to the solution and if a second order sufficiency condition is satisfied at the solution, then $x_{k+1} = x_k + s_k$ can be used at the new iterate. To ensure global convergence and to deal with possible negative curvature of H_k when x_k is away from the solution, we add a trust-region constraint $\|s_k\|_{\mathcal{X}} \leq \Delta_k$ to (10), where $\Delta_k > 0$ is the trust-region radius, which is adapted by the optimization algorithm. To deal with the possible incompatibility of the trust-region constraint and (10b), we use a composite step algorithm (see [7, Ch. 15] for an overview). The trial step s_k is decomposed as $s_k = n_k + t_k$, where for a given parameter $\xi \in (0, 1)$, the so-called quasi-normal step n_k is an approximate solution of

$$\min \ \|c_x(x_k)n + c(x_k)\|_{\mathcal{Y}} \tag{11a}$$

$$\text{s.t.} \ \|n\|_{\mathcal{X}} \leq \xi\Delta_k, \tag{11b}$$

and the so-called tangential step t_k is an approximate solution of

$$\min \ \frac{1}{2}\langle H_k t, t\rangle_{\mathcal{X}} + \langle \nabla_x L(x_k, \lambda_k) + H_k n_k, t\rangle_{\mathcal{X}} \tag{12a}$$

$$\text{s.t.} \ c_x(x_k)t = 0 \tag{12b}$$

$$\|t\|_{\mathcal{X}} \leq \Delta_k - \|n_k\|_{\mathcal{X}}. \tag{12c}$$

Once the trial step $s_k = n_k + t_k$ is computed, an augmented Lagrangian merit function and a quadratic approximation of it are used to decide whether to accept the trial step, i.e, set $x_{k+1} = x_k + s_k$, or to reject it, i.e., set $x_{k+1} = x_k$, and how to update the trust-region radius. The rules are fairly easy to implement, but their precise description is lengthy. Because of page limitations, we refer to [15, 21] for the details and instead focus on the issue of linear system solves that relates to the use of DD methods.

One way to compute an approximate solution of the quasi-normal step subproblem (11) that is suitable for use within our SQP method is the so-called dog-leg approach, which requires the computation of the minimum norm solution of $\min \|c_x(x_k)n + c(x_k)\|_{\mathcal{Y}}$. The minimum norm solution can be computed by solving

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} n \\ y \end{pmatrix} = \begin{pmatrix} 0 \\ -c(x_k) \end{pmatrix} \tag{13}$$

for $y \in \mathcal{Y}$, $n \in \mathcal{X}$. The quasi-normal step is then computed as a linear combination of the minimum norm solution n and of $-c_x(x_k)^* c(x_k)$ or by a simple scaling of the minimum norm solution. For a detailed description of the quasi-normal step computation see, e.g., [7, Sec. 15.4.1.2], [21]. In our context it is important to note that the quasi-normal step computation requires the solution of (13), which in our example application (1) leads to a subproblem of the type (2). Note that (13) is the system of necessary and sufficient optimality conditions for the quadratic problem

$$\min \ \frac{1}{2}\|n\|_{\mathcal{X}}^2 \tag{14a}$$

$$\text{s.t.} \ \ c_x(x_k)n + c(x_k) = 0. \tag{14b}$$

With a bounded linear operator W_k whose range is the null space of $c_x(x_k)$, we can eliminate (12b). Various such operators exist. We use the orthogonal projection onto the null space. In this case $W_k = W_k^* = W_k^2 \in \mathcal{L}(\mathcal{X}, \mathcal{X})$ and $s = W_k w$ can be computed by solving the system

$$\begin{pmatrix} I & c_x(x_k)^* \\ c_x(x_k) & 0 \end{pmatrix} \begin{pmatrix} s \\ z \end{pmatrix} = \begin{pmatrix} w \\ 0 \end{pmatrix}. \tag{15}$$

Using this operator, (12) can be written equivalently as

$$\min \ \frac{1}{2}\langle W_k H_k W_k t, t\rangle_{\mathcal{X}} + \langle \nabla_x L(x_k, \lambda_k) + H_k n_k, W_k t\rangle_{\mathcal{X}} \tag{16a}$$

$$\text{s.t.} \ \ \|t\|_{\mathcal{X}} \le \Delta_k - \|n_k\|_{\mathcal{X}}. \tag{16b}$$

An approximate solution of (16) that is suitable for use within our SQP method can be computed using the Steihaug-Toint modification of the conjugate gradient method (see, e.g., [7]). With W_k given by (15), the Steihaug-Toint CG method can be implemented in an elegant way that in each CG iteration requires the application of W_k. See, e.g., [9]. Note that each application of W_k requires the solution of (15), which is the system of necessary and sufficient optimality conditions for

$$\min \ \frac{1}{2}\|s\|_{\mathcal{X}}^2 - \langle w, s\rangle_{\mathcal{X}} \tag{17a}$$

$$\text{s.t.} \ \ c_x(x_k)s = 0. \tag{17b}$$

We remark that it is easy to apply a preconditioned Steihaug-Toint CG method by replacing I in (15) by $\widetilde{H}_k$, where $\widetilde{H}_k$ is a selfadjoint operator that is strictly positive on the null-space of $c_x(x_k)$ and approximates H_k. (One can even set $\widetilde{H}_k = H_k$, if it is strictly positive on the null-space of $c_x(x_k)$.) In this case $\|s\|_{\mathcal{X}}^2$ in (17) has to be replaced by $\langle \widetilde{H}_k s, s\rangle_{\mathcal{X}}$. The requirements on $\widetilde{H}_k$ guarantee that the modified quadratic program (17) remains convex.

We conclude by noting that each iteration of our trust-region SQP method requires the solution of systems of the type (13) and (15) or, equivalently, the solution of convex quadratic programs of the type (14) and (17). The solves are done iteratively. Consequently, the SQP algorithm needs to provide stopping tolerances to the linear system solvers. These stopping tolerances need to be chosen to guarantee convergence of the overall algorithm, but at the same time it is desirable to choose them as large as possible to make the solution of these subproblems as inexpensive as possible. A rigorous approach that accomplishes this is detailed in [14, 15, 21]. It is used to generate the numerical results shown in the following section.

4 Optimal Control of a Semilinear Elliptic Equation

Our example problem (1) is a special case of (9) and is solved using the trust-region SQP method with inexact linear system solves outlined in the previous section. Each iteration of our trust-region SQP method requires the iterative solution of convex quadratic programs of the type (14) and (17). For the example problem (1) these quadratic programs are essentially of the form (2), with $\mathbf{a}$, r, f given by the current state and control determined by the SQP algorithm. The objective function in these subproblems is slightly different from (2a), but the domain decomposition approach outlined in Section 2 can easily be applied to these subproblems. We remark that all quadratic programs arising in our trust-region SQP method are known to be convex. Hence our optimization-level domain decomposition approach which decomposes the system of first order optimality conditions can be safely applied.

For our numerical example, we solve

$$\text{minimize } \frac{1}{2}\int_{\Omega}(y-\widehat{y})^2\,dx + \frac{\alpha}{2}\int_{\partial\Omega} u^2\,ds \tag{18a}$$

subject to

$$-\Delta y + y^3 - y = f \;\text{ in } \Omega, \qquad \frac{\partial y}{\partial n} = u \text{ on } \partial\Omega. \tag{18b}$$

See, e.g., [11, 24]. We use $\Omega = (0,1)^2$, $\alpha = 1$, $\widehat{y}(x) = \cos(\pi x_1)\cos(\pi x_2)$, and $f(x) = \cos(\pi x_1)\cos(\pi x_2)(2\pi^2 + \cos^2(\pi x_1)\cos^2(\pi x_2) - 1)$.

The problem (18) is discretized using piecewise linear finite elements for states and controls. The domain Ω is subdivided into triangles by first subdividing it into squares of size $h \times h$ and then subdividing each square into two triangles. The domain Ω is subdivided into square subdomains of size $H \times H$.

Tables 1 and 2 show the behavior of our SQP method with a one-level and two-level optimization-level Neumann-Neumann DD preconditioner for varying mesh and subdomain sizes. The number of outer SQP iterations is constant over varying mesh sizes and subdomain sizes. This is not too surprising (although not yet proven for our class of SQP methods), since we use an SQP method with exact second order derivative information and there are known mesh independence results for many Newton-like methods.

Within each iteration of the SQP method, a KKT-type system has to be solved for the computation of a Lagrange multiplier estimate, to compute the quasi-normal step (cf., (14)), and within each iteration of the Steihaug-Toint CG algorithm used to compute the tangential step (cf., (16)). Tables 1 and 2 show only a mild increase in the number of calls to GMRES as the number of subdomains is increased or the mesh size is decreased.

A significant difference is seen in the average number of GMRES iterations used to solve a KKT-type system depending on whether a one-level Neumann-Neumann DD preconditioner is used or a two-level preconditioner. This is

Table 1. One-level preconditioner: Number of SQP iterations, number of calls to GMRES, the total number of GMRES iterations, and the average number of GMRES iterations per call.

$1/h$		64×64			128×128	
$1/H$	2×2	4×4	8×8	2×2	4×4	8×8
SQP iter's	5	5	5	5	5	5
GMRES calls	36	41	45	40	50	53
GMRES total	195	1313	4733	197	1719	5895
GMRES avg	5.4	32.0	105.2	4.9	34.4	111.2

Table 2. Two-level preconditioner: Number of SQP iterations, number of calls to GMRES, the total number of GMRES iterations, and the average number of GMRES iterations per call.

$1/h$		64×64			128×128	
$1/H$	2×2	4×4	8×8	2×2	4×4	8×8
SQP iter's	5	5	5	5	5	5
GMRES calls	36	41	44	40	49	50
GMRES total	96	348	393	132	515	574
GMRES avg	2.7	8.5	8.9	3.3	10.5	11.5

expected since the performance of the one-level Neumann-Neumann DD preconditioner deteriorates as the number of subdomains increases, whereas the performance of the two-level preconditioner is insensitive to the number of subdomains. For single PDEs, this is shown theoretically as well as numerically, see, e.g., [22, 23]. For the optimization case this has been observed numerically ([13]), but not yet proven theoretically.

Figure 1 shows the relative residual stopping tolerances required for GMRES during its calls within the Steihaug-Toint CG algorithm used to compute the tangential step (cf., (16)). Each box/star indicates one call to GMRES, each box indicates a new SQP iteration. This figure shows that our SQP algorithm adjusts the stopping tolerance and has the capability to coarsen the relative residual stopping tolerance. We note that the dynamic adjustment is particularly beneficial over using a fixed stopping tolerance when the preconditioner is less effective and many GMRES iterations have to be executed to achieve a lower tolerance.

Acknowledgement. This work was supported in part by NSF grants ACI-0121360 and DMS-0511624.

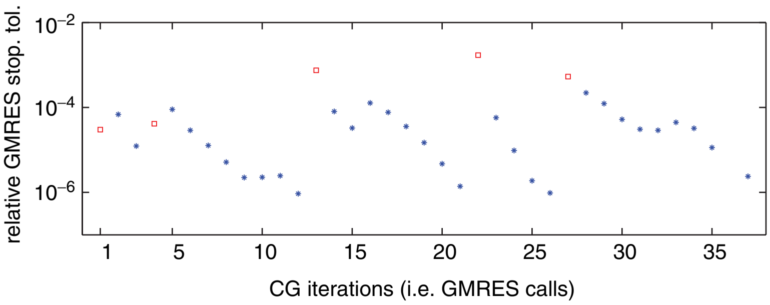


Fig. 1. Relative stopping tolerances for every call to GMRES within the Steihaug-Toint CG algorithm. One CG iteration corresponds to one GMRES call. The empty square indicates the beginning of a new SQP iteration

References

[1] Y. Achdou and F. Nataf. A Robin-Robin preconditioner for an advection-diffusion problem. *C. R. Acad. Sci. Paris Sér. I Math.*, 325(11):1211–1216, 1997.

[2] Y. Achdou, P. Le Tallec, F. Nataf, and M. Vidrascu. A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 184(2-4):145–170, 2000.

[3] R. A. Bartlett, M. Heinkenschloss, D. Ridzal, and B. van Bloemen Waanders. Domain decomposition methods for advection dominated linear–quadratic elliptic optimal control problems. *Comput. Methods Appl. Mech. Engrg.*, 195:6428–6447, 2006.

[4] G. Biros and O. Ghattas. Parallel Lagrange–Newton–Krylov–Schur methods for PDE–constrained optimization. part I: The Krylov–Schur solver. *SIAM J. Sci. Comput.*, 27(2):587–713, 2005.

[5] G. Biros and O. Ghattas. Parallel Lagrange–Newton–Krylov–Schur methods for PDE–constrained optimization. part II: The Lagrange–Newton solver and its application to optimal control of steady viscous flows. *SIAM J. Sci. Comput.*, 27(2):714–739, 2005.

[6] X.-C. Cai, M. Dryja, and M. Sarkis. Restricted additive Schwarz preconditioners with harmonic overlap for symmetric positive definite linear systems. *SIAM J. Numer. Anal.*, 41(4):1209–1231, 2003.

[7] A. R. Conn, N. I. M. Gould, and Ph. L. Toint. *Trust–Region Methods.* SIAM, Philadelphia, 2000.

[8] M. Delgado, J. A. Montero, and A. Suárez. Optimal control for the degenerate elliptic logistic equation. *Appl. Math. Optim.*, 45(3):325–345, 2002.

[9] N. I. M. Gould, M. E. Hribar, and J. Nocedal. On the solution of equality constrained quadratic programming problems arising in optimization. *SIAM J. Sci. Comput.*, 23(4):1376–1395, 2001.

[10] M. D. Gunzburger, L. S. Hou, and S. S. Ravindran. Analysis and approximation of optimal control problems for a simplified Ginzburg-Landau model of superconductivity. *Numer. Math.*, 77:243–268, 1997.

[11] M. D. Gunzburger, L. S. Hou, and T. P. Svobotny. Finite element approximations of an optimal control problem associated with the scalar Ginzburg-Landau equation. *Comput. Math. Appl.*, pages 123–131, 1991.

[12] M. Heinkenschloss and H. Nguyen. Domain decomposition preconditioners for linear–quadratic elliptic optimal control problems. Technical Report TR04–20, Dept. of Computational and Applied Mathematics, Rice U., 2004.

[13] M. Heinkenschloss and H. Nguyen. Neumann-Neumann domain decomposition preconditioners for linear–quadratic elliptic optimal control problems. *SIAM J. Sci. Comput.*, 28(3):1001–1028, 2006.

[14] M. Heinkenschloss and D. Ridzal. Solution of a class of quadratic programs arising in nonlinear programming using inexact linear system solves. Technical report, Dept. of Computational and Applied Math., Rice U., 2006.

[15] M. Heinkenschloss and L. N. Vicente. Analysis of inexact trust–region SQP algorithms. *SIAM J. Optim.*, 12:283–302, 2001.

[16] A. W. Leung. Positive solutions for systems of PDE and optimal control. *Nonlinear Anal.*, 47(2):1345–1356, 2001.

[17] X. Li and J Yong. *Optimal Control Theory for Infinite Dimensional Systems.* Systems & Control: Foundations & Applications. Birkhäuser-Verlag, Boston, Basel, Berlin, 1995.

[18] E. Prudencio, R. Byrd, and X.-C. Cai. Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems. *SIAM J. Sci. Comput.*, 27(4):1305–1328, 2006.

[19] E. Prudencio and X.-C. Cai. Parallel multi-level Lagrange-Newton-Krylov-Schwarz algorithms with pollution removing for PDE-constrained optimization. Technical report, U. of Colorado at Boulder, Dept. of Computer Science, Boulder, Colorado, 2006.

[20] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford University Press, Oxford, 1999.

[21] D. Ridzal. *Trust Region SQP Methods With Inexact Linear System Solves For Large-Scale Optimization.* PhD thesis, Dept. of Computational and Applied Mathematics, Rice U., Houston, TX, 2006.

[22] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition. Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, Cambridge, London, New York, 1996.

[23] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Computational Math., Vol. 34. Springer–Verlag, Berlin, 2004.

[24] F. Tröltzsch. *Optimale Steuerung partieller Differentialgleichungen. Grundlagen, Optimalitätsbedingungen und ihre Anwendungen.* Vieweg Verlag, Braunschweig, 2005.

Domain Decomposition Algorithms for Mortar Discretizations

Hyea Hyun Kim

National Institute for Mathematical Sciences, 385-16, Doryong-dong, Yuseong-gu, Daejeon 305-340, South Korea. hhk@nims.re.kr

Summary. Mortar discretizations have been developed for coupling different approximations in different subdomains, that can arise from engineering applications in complicated structures with highly non-uniform materials. The complexity of the mortar discretizations requires fast algorithms for solving the resulting linear systems. Several domain decomposition algorithms, that have been successfully applied to conforming finite element discretizations, have been extended to the linear systems of mortar discretizations. They are overlapping Schwarz methods, FETI-DP (Dual-Primal Finite Element Tearing and Interconnecting) methods, and BDDC (Balancing Domain Decomposition by Constraints) methods. The new result is that complete analysis, providing the optimal condition number estimate, has been done for geometrically non-conforming subdomain partitions and for problems with discontinuous coefficients. These algorithms were further applied to the two-dimensional Stokes and three-dimensional elasticity. In addition, a BDDC algorithm with an inexact coarse problem was developed.

1 Introduction

Mortar discretizations were introduced in [2] to couple different approximations in different subdomains so as to obtain a good global approximate solution. They are useful for modeling multi-physics, adaptivity, and mesh generation for three dimensional complex structures. The coupling is done by enforcing certain constraints on solutions across the subdomain interface using Lagrange multipliers. We call these constraints the mortar matching conditions.

The complexity of the discretizations requires fast algorithms for solving the resulting linear systems. We focus on extension of several domain decomposition algorithms, that have been successfully applied to conforming finite element discretizations, to solving such linear systems. They are overlapping Schwarz methods, FETI-DP (Dual-Primal Finite Element Tearing and Interconnecting) methods, and BDDC (Balancing Domain Decomposition by Constraints) methods, see Section 3 of [19], [5, 4], and [16, 17].

The new result is that complete analysis, providing the optimal condition number bound, was done for geometrically non-conforming subdomain partitions and for problems with discontinuous coefficients. These algorithms are further extended to the Stokes problem and three-dimensional elasticity. In addition, using an inexact solver for the coarse problem the BDDC method was extended to a three–level algorithm.

Throughout this paper, h_i and H_i denote the mesh size and the subdomain diameter, and C is a generic positive constant independent of the mesh parameters and problem coefficients.

2 Mortar Discretization

We consider a model elliptic problem,

$$\begin{aligned} -\nabla \cdot (\rho(x) \nabla u(x)) &= f(x), \quad x \in \Omega, \\ u(x) &= 0, \quad x \in \partial\Omega, \end{aligned} \tag{1}$$

where Ω is a polyhedral domain in $\mathbb{R}^3$, $f(x)$ is a square integrable function in Ω, $\rho(x)$ is a positive and bounded function. We decompose Ω into a non-overlapping subdomain partition $\{\Omega_i\}_i$, that can be geometrically non-conforming. In a geometrically non-conforming partition, a subdomain can intersect its neighbors in a part of a face, a part of an edge, or a vertex. This allows a subdomain partition that is not necessarily a triangulation of Ω. We then introduce a triangulation $\mathcal{T}_i$ to each subdomain Ω_i and denote by X_i the conforming piecewise linear finite element space associated to the triangulation $\mathcal{T}_i$. These triangulations can be non matching across the subdomain interface $\Gamma = \bigcup_{i,j}(\partial\Omega_i \cap \partial\Omega_j)$. We can select a set of subdomain faces of which union covers Γ, see [18, Section 4.1]. We then denote those faces $\{F_n\}_n$ and call them nonmortar faces.

A subdomain Ω_i, with a nonmortar face F_n as its face, can intersect more than one neighbors $\{\Omega_j\}_j$ through F_n. This gives a partition $\{F_{n(i,j)}\}_j$ to F_n, where $F_{n(i,j)}$ is the common part of Ω_i and Ω_j. We call $F_{n(i,j)}$ mortar faces. We note that the mortar faces can be only part of subdomain faces while nonmortar faces are a full subdomain face. On each nonmortar $F_n \subset \Omega$, we introduce a Lagrange multiplier space $M(F_n)$ based on the finite element space X_i, see [2, 22, 6] for the detailed construction.

We define a product space

$$X = \prod_i X_i,$$

and introduce a mortar matching condition on $(v_1, \cdots, v_N) \in X$

$$\int_{F_n} (v_i - \phi)\psi \, ds = 0, \quad \forall \psi \in M(F_n), \tag{2}$$

where $\phi = v_j$ on $F_{n(i,j)} \subset F_n$. A mortar finite element space is defined by

$$\widehat{X} = \{v \in X \,:\, v \text{ satisfies } (2)\}\,,$$

and mortar discretization is to approximate the solution u of (1) in the mortar finite element space $\widehat{X}$. The approximation error is given by

$$\sum_{i=1}^{N} \|u - u^h\|_{H^1(\Omega_i)}^2 \leq C \sum_{i=1}^{N} h_i^2 |\log(h_i)| \|u\|_{H^2(\Omega_i)}^2,$$

where u^h is the approximate solution, see [1]. The additional log factor does not appear when the subdomain partition is geometrically conforming.

3 An Overlapping Schwarz Method

To build an overlapping Schwarz preconditioner, we introduce two auxiliary partitions of Ω. They are an overlapping subregion partition $\{\widetilde{\Omega}_j\}_j$ and a coarse triangulation $\{T_k\}_k$ of Ω.

For each subregion $\widetilde{\Omega}_j$, we introduce a finite element space $\widetilde{X}_j$ as a subspace of $\widehat{X}$ in the following way. Among the nodes in the finite element space X, we define by genuine unknowns the nodes that are not contained in the interior of the nonmortar faces. The space $\widetilde{X}_j$ is then built from the basis functions of each genuine unknowns, that are supported in $\widetilde{\Omega}_j$. By assigning values of the basis functions at the nodes on the nonmortar faces using the mortar matching condition (2), we can obtain the resulting basis elements contained in $\widehat{X}$.

Similarly, we can construct a coarse finite element space $\widetilde{X}_0$ that belongs to $\widehat{X}$. Let X^H be the piecewise linear conforming finite element space associated to the coarse triangulation $\{T_k\}_k$. First we interpolate a function $v \in X^H$ to the produce space X using the nodal interpolant $I^h : X^H \rightarrow X$ such that

$$I^h(v) = (I_1^h(v), \cdots, I_N^h(v)),$$

where $I_i^h(v)$ denote the nodal interpolant of v to the space X_i. We then modify values of $I^h(v)$ at the nodes on the nonmortar faces using the mortar matching condition so that obtain the resulting interpolant $I^m(v)$ contained in $\widehat{X}$. The coarse finite element space $\widetilde{X}_0$ is given by

$$\widetilde{X}_0 = I^m(X^H) \subset \widehat{X}.$$

The two–level overlapping Schwarz algorithm consists of solving the local and coarse (when $j = 0$) problems; find $T_j u \in \widetilde{X}_j$ such that

$$a(T_j u, v_j) = a(u, v_j), \quad \forall v_j \in \widetilde{X}_j \; (j \geq 0).$$

For the overlapping Schwarz algorithm applied to the mortar discretization of the elliptic problem (1), we proved the condition number estimate, see [13].

Theorem 1. *We assume that the diameter of Ω_i is comparable to any coarse triangle T_k that intersects Ω_i and the diameter H_i of Ω_i satisfy $H_i \leq C\widetilde{H}_j$, where $\widetilde{H}_j$ is the diameter of subregion $\widetilde{\Omega}_j$ that intersects Ω_i. In addition, we assume that the mesh sizes of subdomains that intersect along a common face are comparable. We then obtain the condition number bound for the overlapping Schwarz algorithm,*

$$\kappa(\sum_{j=0}^{J} T_j) \leq C \max_{j,k} \left\{ \left(1 + \frac{\widetilde{H}_j}{\delta_j}\right) \left(1 + \log \frac{H_k}{h_k}\right) \right\},$$

where δ_j are the overlapping width of the subregion partition $\{\widetilde{\Omega}_j\}_j$, and H_k/h_k denote the number of nodes across subdomain Ω_k.

4 BDDC and FETI–DP Algorithms

In this section, we construct BDDC and FETI–DP algorithms for the mortar discretization. We first derive the primal form of the mortar discretization and then introduce a BDDC algorithm for solving the primal form. Secondly we introduce the dual form and build a FETI–DP algorithm that is closely related to the BDDC algorithm.

We separate unknowns in the finite element space X_i into interior and interface unknowns and after selecting appropriate primal unknowns among the interface unknowns we again decompose the interface unknowns into dual and primal unknowns,

$$X_i = X_I^{(i)} \times X_\Gamma^{(i)}, \quad X_\Gamma^{(i)} = W_\Delta^{(i)} \times W_\Pi^{(i)}, \tag{3}$$

where I, Γ, Δ, and Π denote the interior, interface, dual, and primal unknowns, respectively.

The primal unknowns are related to certain primal constraints selected from the mortar matching condition (2). They result in a coarse component of the BDDC preconditioner so that a proper selection of such constraints is important in obtaining a scalable BDDC algorithm. We consider $\{\psi_{ij,k}\}_k$, the basis functions in $M(F_n)$ that are supported in $\overline{F}_{n(i,j)}$, and introduce

$$\psi_{ij} = \sum_k \psi_{ij,k}.$$

We assume that at least one such basis function $\psi_{ij,k}$ exists for each $F_{n(i,j)} \subset F_n$. On each interface $F_{n(i,j)}$, we select the primal constraints for $(w_1, \cdots, w_N) \in X_\Gamma (= \prod_i X_\Gamma^{(i)})$ as

$$\int_{F_{n(i,j)}} (w_i - w_j)\psi_{ij} \, ds = 0. \tag{4}$$

For the case of a geometrically conforming partition, i.e., when $F_{n(i,j)}$ is a full face of two subdomains, the above constraints are the face average matching condition because $\psi_{ij} = 1$. We can change the variables to make the primal constraints explicit, see [14, Sec. 6.2], [15, Sec. 2.3], and [9, Sec. 2.2]. We then separate the unknowns in the space X_i as described in (3). We will also assume that all the matrices and vectors are written in terms of the new variables.

Throughout this paper, we use the notation V for the product space of local finite element spaces $V^{(i)}$. The same applies to the vector notations v and $v^{(i)}$. In addition, we use the notation $\widehat{V}$ for the subspace of V satisfying mortar matching condition (2) and the notation $\widetilde{V}$ for the subspace satisfying only the primal constraints (4). For example, we can represent the space

$$\widetilde{X}_{\Gamma} = \{w \in X_{\Gamma} \ : \ w \text{ satisfies the primal constraints (4)}\},$$

in the following way,

$$\widetilde{X}_{\Gamma} = W_{\Delta} \times \widehat{W}_{\Pi}.$$

We further decompose the dual unknowns into the part interior to the nonmortar faces and the other part to obtain

$$W_{\Delta} = W_{\Delta,n} \times W_{\Delta,m},$$

where n and m denote unknowns at nonmortar faces (open) and the other unknowns, respectively.

After enforcing the mortar matching condition (2) on functions in the space $\widetilde{X}_{\Gamma}$, we obtain the matrix representation,

$$B_n w_n + B_m w_m + B_{\Pi} w_{\Pi} = 0. \tag{5}$$

Here we enforced the mortar matching condition using a reduced Lagrange multiplier space, since the functions in the space $\widetilde{X}_{\Gamma}$ satisfy the primal constraints selected from the mortar matching condition (2). The reduced Lagrange multiplier space is obtained after eliminating one basis element among $\{\psi_{ij,k}\}_k$ for each $F_{ij} \subset F_l$ so that the matrix B_n in (5) is invertible. Therefore the unknowns w_n of the nonmortar part are determined by the other unknowns, w_m, and w_{Π}, which are called genuine unknowns. We define the space of genuine unknowns by

$$W_G = W_{\Delta,m} \times \widehat{W}_{\Pi}$$

and define the mortar map,

$$\widetilde{R}_{\Gamma} = \begin{pmatrix} -B_n^{-1} B_m & -B_n^{-1} B_{\Pi} \\ I & 0 \\ 0 & I \end{pmatrix}, \tag{6}$$

that maps the genuine unknowns in W_G into the unknowns in $\widetilde{X}_{\Gamma}$ which satisfy the mortar matching condition.

To derive the linear system of the mortar discretization, we introduce several matrices. The matrix $S_\Gamma^{(i)}$ is the local Schur complement matrix obtained from the local stiffness matrix $A^{(i)}$ by eliminating the subdomain interior unknowns,

$$S_\Gamma^{(i)} = A_{\Gamma I}^{(i)}(A_{II}^{(i)})^{-1}(A_{\Gamma I}^{(i)})^T = \begin{pmatrix} S_{\Delta\Delta}^{(i)} & (S_{\Pi\Delta}^{(i)})^t \\ S_{\Pi\Delta}^{(i)} & S_{\Pi\Pi}^{(i)} \end{pmatrix},$$

where Δ and Π stand for the blocks corresponding to dual and primal unknowns, respectively. We define extensions of the spaces by

$$W_G \xrightarrow{\widetilde{R}_\Gamma} \widetilde{X}_\Gamma \xrightarrow{\overline{R}_\Gamma} X_\Gamma,$$

where $\widetilde{R}_\Gamma$ is the mortar map in (6) and $\overline{R}_\Gamma$ is the product of restriction maps,

$$\overline{R}_\Gamma^{(i)} : \widetilde{X}_\Gamma \to X_\Gamma^{(i)}.$$

We next introduce the matrices S_Γ and $\widetilde{S}_\Gamma$, the block diagonal matrix and the partially assembled matrix at the primal unknowns, respectively, as

$$S_\Gamma = \mathrm{diag}_i(S_\Gamma^{(i)}), \quad \widetilde{S}_\Gamma = \overline{R}_\Gamma^t S_\Gamma \overline{R}_\Gamma.$$

The linear system of the mortar discretization is then written as:
find $u_G \in W_G$ such that

$$\widetilde{R}_\Gamma^t \widetilde{S}_\Gamma \widetilde{R}_\Gamma u_G = \widetilde{R}_\Gamma^t g_G, \tag{7}$$

where $g_G \in W_G$ is the part of genuine unknowns, i.e., the unknowns other than the nonmortar part, of $g \in X_\Gamma$, that is given by

$$g^{(i)} = f_\Gamma^{(i)} - A_{\Gamma I}^{(i)}(A_{II}^{(i)})^{-1} f_I^{(i)}.$$

Here $f^{(i)} = \begin{pmatrix} f_I^{(i)} \\ f_\Pi^{(i)} \end{pmatrix}$ is the local force vector. In the BDDC algorithm, we solve (7) using a preconditioner M^{-1} of the form,

$$M^{-1} = \widetilde{R}_{D,\Gamma}^t \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma},$$

where the weighted extension operator $\widetilde{R}_{D,\Gamma}$ is given by

$$\widetilde{R}_{D,\Gamma} = D\widetilde{R}_\Gamma = \begin{pmatrix} D_n & 0 & 0 \\ 0 & D_m & 0 \\ 0 & 0 & D_\Pi \end{pmatrix} \widetilde{R}_\Gamma.$$

Later, we will specify the weight D_n, D_m, and D_Π.

We now develop a FETI–DP algorithm closely related to the BDDC algorithm. In the FETI–DP algorithm, we solve the dual form of the mortar discretization that is derived from the constrained minimization problem,

$$\min_{w \in \widetilde{X}_\Gamma} \left\{ \frac{1}{2} w^t \widetilde{S}_\Gamma w - w^t \widetilde{g} \right\},$$

with w satisfying the mortar matching condition (5). The mixed form to the constrained minimization problem gives

$$\begin{aligned} \widetilde{S}_\Gamma w + B^t \lambda &= \widetilde{g}, \\ Bw &= 0, \end{aligned}$$

where $B = (B_n, B_m, B_\Pi)$. After eliminating w, we obtain the dual form,

$$B \widetilde{S}_\Gamma^{-1} B^t \lambda = B \widetilde{S}_\Gamma^{-1} \widetilde{g}. \tag{8}$$

We solve the equations of the dual form (8) iteratively using a preconditioner,

$$\widehat{F}_{DP}^{-1} = B_\Sigma \widetilde{S}_\Gamma B_\Sigma^t,$$

where

$$B_\Sigma^t = \Sigma B^t = \begin{pmatrix} \Sigma_n & 0 & 0 \\ 0 & \Sigma_m & 0 \\ 0 & 0 & \Sigma_\Pi \end{pmatrix} B^t.$$

As a result, we have obtained the two algorithms for solving the mortar discretization and we write them into

$$B_{DDC} = \widetilde{R}_{D,\Gamma}^t \widetilde{S}_\Gamma^{-1} \widetilde{R}_{D,\Gamma} \widetilde{R}_\Gamma^t \widetilde{S}_\Gamma \widetilde{R}_\Gamma, \; F_{DP} = B_\Sigma \widetilde{S}_\Gamma B_\Sigma^t B \widetilde{S}_\Gamma^{-1} B^t.$$

The convergence of the two algorithms depends on the condition number of B_{DDC} and F_{DP}. We now show a close connection between them and then provide weights D and Σ leading to scalable preconditioners. Let

$$P_\Sigma = B_\Sigma^t B, \; E_D = \widetilde{R}_\Gamma \widetilde{R}_{D,\Gamma}^t.$$

Theorem 2. *Assume that P_Σ and E_D satisfy*
1. $E_D + P_\Sigma = I$,
2. $E_D^2 = E_D, \; P_\Sigma^2 = P_\Sigma$,
3. $E_D P_\Sigma = P_\Sigma E_D = 0$.

Then the operators F_{DP} and B_{DDC} have the same spectra except the eigenvalues 0 and 1.

The same result was first shown by [17] and later by [15] for the conforming finite element discretizations. We are able to extend the result to the mortar discretizations.

The Neumann-Dirichlet preconditioner for the FETI-DP algorithms suggested by [10] was shown to be the most efficient for the problems with discontinuous coefficients, see [3]. The weight of the Neumann-Dirichlet preconditioner is given by

$$\Sigma_n = (B_n^t B_n)^{-1},\ \Sigma_m = 0,\ \Sigma_\Pi = 0, \tag{9}$$

and the condition number of the FETI-DP algorithm was shown to be

$$\kappa(F_{DP}) \leq C(1+\log(H/h))^2,$$

when the subdomain with smaller ρ_i is selected as the nonmortar side.

When the weight of the BDDC preconditioner is selected to be

$$D_n = 0,\ D_m = I,\ D_\Pi = I, \tag{10}$$

the E_D and P_Σ satisfy the assumptions in Theorem 2. Therefore, the BDDC algorithm equipped with the weight in (10) has the condition number bound,

$$\kappa(B_{DDC}) \leq C(1+\log(H/h))^2,$$

and the BDDC algorithm is as efficient as the FETI-DP algorithm.

5 Applications of the BDDC and FETI-DP Algorithms

The BDDC and FETI-DP algorithms introduced in the previous section can be generalized to the mortar discretizations of the Stokes problem and three dimensional compressible elasticity problems. For these cases, the selection of primal constraints is important in obtaining a scalable preconditioner.

We assume that the subdomain partition is geometrically conforming. We denote the common face (edge) of two subdomains Ω_i and Ω_j by F_{ij} in three (two) dimensions. An appropriate Lagrange multiplier space $M(F_{ij})$ is then provided for the nonmortar part of F_{ij}. We note that the space $M(F_{ij})$ contains the constant functions.

For the Stokes problem, we select the average matching condition across the interface as the primal constraints, namely,

$$\int_{F_{ij}} \mathbf{v}_i \, ds = \int_{F_{ij}} \mathbf{v}_j \, ds,$$

where F_{ij} is the common face (edge) of $\partial\Omega_i$ and $\partial\Omega_j$ in three (two) dimensions. For the elasticity problems, we select

$$\int_{F_{ij}} \mathbf{v}_i \cdot I_{M_{ij}}(\mathbf{r}_k) \, ds = \int_{F_{ij}} \mathbf{v}_j \cdot I_{M_{ij}}(\mathbf{r}_k) \, ds, \quad k = 1, \cdots, 6,$$

where $\mathbf{r}_k$ are the six rigid body motions and $I_{M_{ij}}(\mathbf{r}_k)$ is the nodal interpolant of $\mathbf{r}_k$ to the Lagrange multiplier space $M(F_{ij})$ provided for the nonmortar face F_{ij}.

With the selection of the primal constraints, we showed the condition number bound of the two algorithms

$$F_{DP} \le C\left(1+\log\frac{H}{h}\right)^2, \quad B_{DDC} \le C\left(1+\log\frac{H}{h}\right)^2,$$

when the weight are given by (9) and (10); see [11] and [7, 8]. The BDDC and FETI-DP algorithms of the elasticity can be extended to the geometrically nonconforming subdomain partitions as well. For such a case, the Lagrange multiplier space $M(F_{ij})$ is the span of basis elements ψ_l of $M(F_n)$ that are supported in $F_{n(i,j)}$. Here $F_n(\subset \partial\Omega_i)$ is the nonmortar face that is partitioned by its mortar neighbors $\{\Omega_j\}_j$.

We note that the BDDC preconditioner consists of solving local problems and the coarse problem,

$$\begin{aligned} M^{-1} &= \widetilde{R}_D^t \widetilde{S}_\Gamma^{-1} \widetilde{R}_D, \\ &= \widetilde{R}_D^t \begin{pmatrix} I & 0 \\ -S_{\Pi\Delta}S_{\Delta\Delta}^{-1} & I \end{pmatrix} \begin{pmatrix} S_{\Delta\Delta}^{-1} & 0 \\ 0 & F_{\Pi\Pi}^{-1} \end{pmatrix} \begin{pmatrix} I & -S_{\Delta\Delta}^{-1}S_{\Delta\Pi} \\ 0 & I \end{pmatrix} \widetilde{R}_D. \end{aligned}$$

As the increase of the number of subdomains, the cost for solving the coarse component becomes a bottleneck of the computation. By solving the coarse problem inexactly, we can speed up the total computational time.

BDDC algorithms with an inexact coarse problem were developed by [21, 20] for conforming finite element discretizations of elliptic problems in both two and three dimensions. The idea is to group subdomains into a subregion and to obtain a subregion partition. Using the additional level, we construct a BDDC preconditioner of the coarse component $F_{\Pi\Pi}$ in M^{-1}. The resulting preconditioner, called a three-level BDDC preconditioner, is given by

$$\overline{M}^{-1} = \widetilde{R}_D^t \overline{S}_\Gamma^{-1} \widetilde{R}_D,$$

where $\overline{S}_\Gamma^{-1}$ denotes the matrix that is the part $F_{\Pi\Pi}^{-1}$ of $\widetilde{S}_\Gamma^{-1}$ is replaced by a BDDC preconditioner using the additional subregion level. The condition number bound of the three–level BDDC algorithm was shown to be

$$\kappa(\overline{M}^{-1}\widetilde{R}_\Gamma^t \widetilde{S}\widetilde{R}_\Gamma) \le C\left(1+\log\frac{\widehat{H}}{H}\right)^2 \left(1+\log\frac{H}{h}\right)^2,$$

where $\widehat{H}$, H, and h denote the subregion diameters, subdomain diameters, and mesh sizes, respectively.

We obtain a subregion partition $\{\Omega^{(j)}\}_{j=1}^{N_c}$, where each subregion $\Omega^{(j)}$ is the union of N_j subdomains $\Omega_i^{(j)}$. An example of a subregion partition, that is

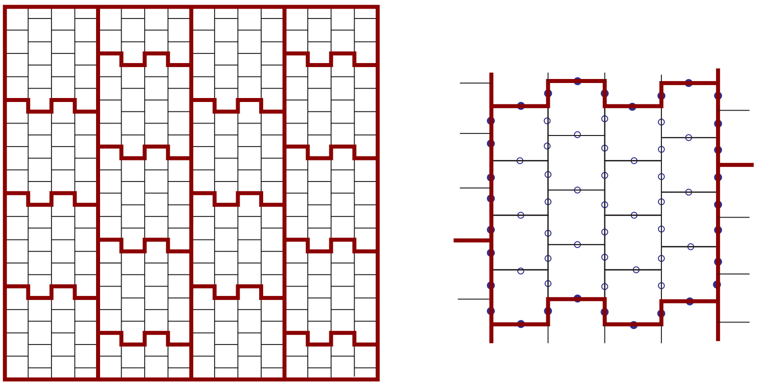


Fig. 1. A subregion partition (left) and unknowns at a subregion (right) when $\widehat{H}/H = 4$; small rectangles are subdomains in the left.

obtained from a geometrically non-conforming subdomain partition, is shown in Fig. 1.

In the subregion partition, we define faces as the intersection of two subregions and vertices (or edges) as the intersection of more than two subregions. Finite element spaces for the subregions are given by the primal unknowns of the two–level algorithm so that the subregion partition is equipped with a conforming finite element space, for which the unknowns match across the subregion interface. On this new level, the mortar discretization is no longer relevant. We can then develop the theory and algorithm for the subregion partition as in the two–level BDDC algorithm done for the conforming finite element discretization. Analysis and numerical computations of the three-level BDDC algorithm for mortar discretizations will be found in [12].

References

[1] F. Ben Belgacem. The mortar finite element method with Lagrange multipliers. *Numer. Math.*, 84(2):173–197, 1999.

[2] C. Bernardi, Y. Maday, and A.T. Patera. A new nonconforming approach to domain decomposition: the mortar element method. In *Nonlinear partial differential equations and their applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991)*, volume 299 of *Pitman Res. Notes Math. Ser.*, pages 13–51. Longman Sci. Tech., Harlow, 1994.

[3] Y.-W. Chang, H.H. Kim, and C.-O. Lee. Preconditioners for the Dual-Primal FETI methods on nonmatching grids: numerical study. *Comput. Math. Appl.*, 51(5):697–712, 2006.

[4] C.R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.

[5] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method. I. A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50(7):1523–1544, 2001.

[6] C. Kim, R.D. Lazarov, J.E. Pasciak, and P.S. Vassilevski. Multiplier spaces for the mortar finite element method in three dimensions. *SIAM J. Numer. Anal.*, 39(2):519–538, 2001.

[7] H.H. Kim. A FETI–DP algorithm for elasticity problems with mortar discretization on geometrically non–conforming partitions. Technical Report 882, Department of Computer Science, Courant Institute, New York University, 2006.

[8] H.H. Kim. A BDDC algorithm for mortar discretization of elasticity problems. *SIAM J. Numer. Anal.*, 2007. To appear.

[9] H.H. Kim, M. Dryja, and O.B. Widlund. A BDDC algorithm for problems with mortar discretization. Technical report, Department of Computer Science, Courant Institute, New York University, 2005.

[10] H.H. Kim and C.-O. Lee. A preconditioner for the FETI-DP formulation with mortar methods in two dimensions. *SIAM J. Numer. Anal.*, 42(5):2159–2175, 2005.

[11] H.H. Kim and C.-O. Lee. A preconditioner for the FETI-DP formulation of the Stokes problem with mortar methods. *SIAM J. Sci. Comp.*, 28(3):1133–1152, 2006.

[12] H.H. Kim and X. Tu. A three-level BDDC algorithm for mortar discretizations. *SIAM J. Numer. Anal.*, 2007. Submitted.

[13] H.H. Kim and O.B. Widlund. Two–level Schwarz algorithms with overlapping subregions for mortar finite elements. *SIAM J. Numer. Anal.*, 44(4):1514–1534, 2006.

[14] A. Klawonn and O.B. Widlund. Dual-Primal FETI methods for linear elasticity. *Comm. Pure Appl. Math.*, 59:1523–1572, 2006.

[15] J. Li and O. Widlund. FETI-DP, BDDC, and block Cholesky methods. *Internat. J. Numer. Methods Engrg.*, 66:250–271, 2006.

[16] J. Mandel and C.R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.*, 10(7):639–659, 2003.

[17] J. Mandel, C.R. Dohrmann, and R. Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.*, 54(2):167–193, 2005.

[18] D. Stefanica. *Domain Decomposition Methods for Mortar Finite Elements.* PhD thesis, Department of Computer Science, Courant Institute, New York Unversity, 2000.

[19] A. Toselli and O.B. Widlund. *Domain Decomposition Methods—Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin, 2005.

[20] X. Tu. Three-level BDDC in three dimensions. *SIAM J. Sci. Comput.*, 2007. To appear.

[21] X. Tu. Three-level BDDC in two dimensions. *Internat. J. Numer. Methods Engrg.*, 69:33–59, 2007.

[22] B.I. Wohlmuth. A mortar finite element method using dual spaces for the Lagrange multiplier. *SIAM J. Numer. Anal.*, 38(3):989–1012, 2000.

On the Multiscale Solution of Constrained Minimization Problems

Rolf Krause

Institute for Numerical Simulation
Wegelerstraße 6
D-53115 Bonn
krause@ins.uni-bonn.de

1 Introduction

For the constrained minimization of convex or non-convex functionals on the basis of multilevel or domain decomposition methods, different strategies have been proposed within the last decades. These include nonlinear and monotone multigrid methods, see [5, 9, 12, 16, 20], multilevel optimization strategies and multilevel Trust-Region methods, see [8, 21], nonlinear domain decomposition methods [1, 6, 22, 23], multigrid methods as linear solvers in the framework of interior point based methods, see [4, 24] and multigrid methods applied in the framework of primal-dual active set strategies or semi-smooth Newton methods, see [11] for the latter. For a nonlinear multigrid method for smooth problems we refer to [10]. We remark that the references given here are far from exhaustive and refer the reader to the references cited therein.

From the multiscale point of view, two features might be employed in order to distinguish between the different methods. The first one is the way the constraints are incorporated into the multiscale hierarchy. The second one is the way the nonlinearity is intertwined with the multiscale structure.

On the one hand, in case interior point methods, active set strategies or semi-smooth Newton methods are used as solution method, domain decomposition or multilevel methods are often used as an inner linear solver within an outer smooth or non-smooth iteration process. Then, the outer iteration provides the convergence of the iterates to a minimizer whereas the inner solver is only applied to linear problems. In order to accelerate the overall iteration process, often the arising linear subproblems are solved inexactly. In this case, the choice of the linear solution method can also effect the convergence of the overall nonlinear scheme significantly, since the approximate correction given by the iterative linear solver might provide a completely different descent direction then the solution of the linear system itself. As a consequence, even if the original nonlinear constrained minimization problem is reduced to a sequence of linear subproblems, the nonlinearity shows also up within the linear subproblems.

On the other hand, following nonlinear domain decomposition and multilevel strategies, the nonlinear iteration process can in contrast be carried out within the subspaces provided by the considered splitting, see [1, 14, 23]. In case of a multilevel method, for example, the nonlinearity might be evaluated on all levels of the multilevel hierarchy. The resulting information gathered on the multilevel hierarchy can then be used to provide faster convergence of the nonlinear iteration process. A possible drawback of this approach is that spurious coarse grid corrections might spoil the convergence of the nonlinear method, cf. [17]. A remedy can be found in adapting the multilevel decomposition to the nonlinearities by, e.g. using solution dependent interpolation operators and bilinear forms. Although this requires at least partially reassembling of the coarse grid stiffness matrices, the additional effort is easily justified by the resulting gain in robustness and convergence speed of the multilevel method.

2 Constrained Minimization

Let H be a Hilbert space and $\emptyset \neq \mathcal{K} \subset H$ a closed and convex subset. We consider the constrained minimization Problem: find $u \in \kappa$

$$J(u) \leq J(v)\,, \qquad v \in \mathcal{K}\,, \tag{1}$$

where $J\colon H \longrightarrow \mathbb{R}$ is a convex and l.s.c. functional. Under this assumptions, a minimizer exists, which is also unique of J is strictly convex, see, e.g. [7]. By introducing the characteristic functional

$$\chi_{\mathcal{K}}(v) = \begin{cases} 0\,, & \text{if } v \in \mathcal{K}\,, \\ \infty\,, & \text{else,} \end{cases}$$

the constraints can be translated into the non-smooth and nonlinear functional $\chi_{\mathcal{K}}$, leading to the unconstrained minimization problem: find $u \in H$

$$(J + \chi_{\mathcal{K}})(u) \leq (J + \chi_{\mathcal{K}})(v)\,, \qquad v \in H\,. \tag{2}$$

Since the resulting non-smooth energy $J + \chi_{\mathcal{K}}$ prevents the straight forward application of, e.g. a gradient method or Newton's method, often the functional $J + \chi_{\mathcal{K}}$ is replaced by a differentiable one, e.g. $J + \chi^{\alpha}_{\mathcal{K}}$, α a regularization parameter. This allows for applying Newton's method to the resulting first order conditions for a minimum

$$(J + \chi^{\alpha}_{\mathcal{K}})'(u^{\alpha})(v) = 0\,, \qquad v \in H\,. \tag{3}$$

A different and non-smooth approach can be found by formulating the necessary conditions for a minimizer of J as variational inequality. In this case, the energy functional J is generated by the H-elliptic bilinear form $a(\cdot, \cdot)$ and by the linear functional f on H as

$$J(u) = \frac{1}{2}a(u, u) - f(u)\,, \tag{4}$$

the minimization problem (1) can equivalently be reformulated as the variational inequality: find $u \in \mathcal{K}$

$$a(u, v - u) \geq f(v - u)\,, \qquad v \in \mathcal{K}\,, \tag{5}$$

see [7]. The advantage of the latter formulation is that the non-smooth structure of the minimization problem (1) is preserved. Numerical methods based on (5) therefore can be expected to give results with higher accuracy.

After discretization of (1) by, e.g. finite elements, we obtain the finite dimensional minimization problem: find $u^L \in \mathcal{K}^L$

$$J(u^L) \leq J(v)\,, \qquad v \in \mathcal{K}^L\,, \tag{6}$$

where the closed and convex set $\emptyset \neq \mathcal{K}^L \subset \mathcal{S}^L$ approximates $\mathcal{K}$ and $\mathcal{S}^L$ is a finite dimensional subspace of H. Here, the index L serves as discretization parameter. We remark that instead of solving the nonlinear problem (6) in finite dimensions it is also possible to apply, e.g. an interior point method in the function space H directly, see [24]. The approximate computation of the resulting Newton corrections then gives rise to linear subproblems, which can be solved by linear multigrid methods. Here, we do not follow this approach but rather focus on the efficient computation of a solution to the finite dimensional constrained minimization problem (6). This solution can be obtained by either applying, e.g. a semi-smooth Newton method or a primal dual active set strategy to the necessary first order conditions, cf. (5), or by attacking the minimization problem (6) directly. Consequently, a multigrid method can either be used as a solver or preconditioner for the linearized problem, or it can serve as a nonlinear solver by itself.

3 Low Frequency Representation of Constraints

Here, as an example for (1), let us consider a contact problem in elasticity. Subject to volume and surface forces, an elastic body is pressed against a rigid foundation which cannot be penetrated, see, e.g., Figure 1. The actual zone of contact γ_C depends on the sought deformations and is unknown in advance. We identify the elastic body in its reference configuration with the (polyhedral) domain $\Omega \subset \mathbb{R}^3$ and set as solution space $H = (H^1(\Omega))^3$. The boundary $\partial\Omega$ is decomposed into three disjoint parts, Γ_D, the Dirichlet boundary with $\text{meas}_2(\Gamma_D) > 0$, Γ_N, the Neumann boundary and Γ_C, the possible contact boundary. We assume $\overline{\gamma_C} \Subset \Gamma_C$. At Γ_C, we enforce the linearized non-penetration condition $\boldsymbol{u} \cdot \boldsymbol{n} \leq g$, cf. [13], with respect to the outer normal $\boldsymbol{n}$. Here, g is the distance in normal direction to the obstacle in the reference configuration. The normal and tangential displacements, respectively, are $u_n = \boldsymbol{u} \cdot \boldsymbol{n}$ and $\boldsymbol{u}_T = \boldsymbol{u} - u_n \cdot \boldsymbol{n}$. We use boldface symbols for tensor and vector quantities and the summation convention is enforced on indices running from $1, \ldots, 3$. The stresses $\boldsymbol{\sigma}$ are given by Hooke's law $\sigma_{ij}(\boldsymbol{u}) = E_{ijml}\, u_{l,m}$, where Hooke's tensor $(E_{ijml})_{i,j,l,m=1}^{3}$, $E_{ijlm} \in L^{\infty}(\Omega)$, $1 \leq i, j, l, m \leq 3$ is assumed to be sufficiently smooth, symmetric and uniformly positive definite and $\boldsymbol{\epsilon}(\boldsymbol{u}) = \frac{1}{2}(\nabla \boldsymbol{u} + (\nabla \boldsymbol{u})^T)$ is the linearized strain tensor. The minimization problem (1) now constitutes the elastic contact problem without friction, if we define the set of admissible displacements by

$$\mathcal{K} = \{\boldsymbol{u} \in H \,|\, \boldsymbol{u} \cdot \boldsymbol{n} \leq g \text{ on } \Gamma_C\} \tag{7}$$

and choose J to be the quadratic elastic energy

$$J(\boldsymbol{v}) = \frac{1}{2} a(\boldsymbol{v}, \boldsymbol{v}) - f(\boldsymbol{v}) = \frac{1}{2} \int\limits_{\Omega} \boldsymbol{\sigma}(\boldsymbol{v}) \colon \boldsymbol{\epsilon}(\boldsymbol{v}) \, dx - \int\limits_{\Omega} \boldsymbol{f} \boldsymbol{v} \, dx \,, \tag{8}$$

see [13]. Here $\boldsymbol{f} \in \boldsymbol{L}^2(\Omega)$ accounts for the volume forces and surface tractions. The finite dimensional minimization problem (6) is now obtained by discretizing by finite elements. To this end, let $\mathcal{T} = (\mathcal{T}^\ell)_{\ell=0}^{L}$ denote a family of nested and shape regular meshes with discretization parameter h^ℓ. Here, $L > 0$ is the index of the finest level and h^ℓ is the mesh-size of $\mathcal{T}^\ell$. The meshes may consist of tetrahedrons, hexahedrons, pyramids or prisms. We denote the set of all nodes of $\mathcal{T}^\ell$ by $\mathcal{N}^\ell$ and the nodes on the possible contact boundary Γ_C are $\mathcal{C}^\ell = \overline{\Gamma}_C \cap \mathcal{N}^\ell$. By $\mathcal{S}^\ell \subset \mathcal{S}^L$ we denote the spaces of first order Lagrangian finite elements on Level ℓ.

Multilevel methods for this type of problem have been considered by [3, 5, 12, 20] for scalar problems and by [16] for the system given above.

Construction of Subspaces and Coarse Level Energies

We first give the algorithmic formulation for a nonlinear and non-smooth multigrid method which has been implemented in the C++–toolbox ObsLib++, cf. [17].

Algorithm 1 (Non-smooth Multigrid Method)

```
(1) Initialize $\boldsymbol{u}_0^L$. For $k = 0, \ldots, k_{\max}$ do:
(2) Compute an approximate solution $\boldsymbol{c}^L$ of the problem: find $\boldsymbol{w}^L \in \mathcal{S}^L$, such that
        $(J + \chi_{\mathcal{K}^L})(\boldsymbol{u}_k^L + \boldsymbol{w}^L) \leq (J + \chi_{\mathcal{K}^L})(\boldsymbol{u}_k^L + \boldsymbol{w})\,, \quad \boldsymbol{v} \in \mathcal{S}^L.$
    Set $\bar{\boldsymbol{u}}^L = \boldsymbol{u}_k^L + \boldsymbol{c}^L$.
(3) For $\ell < L$ do:
    Choose subspace $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$, convex set $\mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}$, $\bar{\boldsymbol{u}}^l \in \mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}$ and functional $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L}$
    Coarse grid correction: find $\boldsymbol{c}^\ell \in \mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}$, such that
        $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L}(\bar{\boldsymbol{u}}^l + \boldsymbol{c}^\ell) \leq \mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L}(\bar{\boldsymbol{u}}^l + \boldsymbol{v})\,, \quad \boldsymbol{v} \in \mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}.$
(5) Set $\boldsymbol{u}_k^L = P^L(\bar{\boldsymbol{u}}^L + \sum_{\ell<L} \boldsymbol{c}^\ell)$
```

Here, in order to allow for an adaptation of the coarse grid basis to the actual iterate, we have replaced the multilevel decomposition induced by the spaces $\mathcal{S}^\ell$ by the subspaces $\{\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}\}_{\ell<L}$ which may depend on the smoothed iterate $\bar{\boldsymbol{u}}^L$ obtained after the leading fine grid smoothing (2) in Algorithm 1. The convex sets $\mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}$ provide a multilevel decomposition of $\mathcal{K}^L$, see [1, 16, 23]. By means of the mapping $P^L \colon \mathcal{S}^L \longrightarrow \mathcal{S}^L$, the feasibility of the iterates is ensured. Examples are global damping of the coarse grid corrections or line search. In case, the coarse grid corrections are feasible by construction of $\mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}$, P^L will be the identity. In case of a parallel multiscale method, P^L may also serve as a non-linear synchronization which is necessary for synchronizing the different iterates obtained on the different processors. For an example, we refer to Figure 2 and Table 3. Finally, on each Level $0 \leq \ell < L$, the correction in $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$ is computed with respect to the possibly level dependent convex functional $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L}$. This step requires either the restriction of the linear or nonlinear defect or the projection of the smoothed iterate $\bar{\boldsymbol{u}}^L$ onto $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$ in order to obtain the start iterate $\bar{\boldsymbol{u}}^\ell$. Concerning the construction of the coarse grid models, the straight forward approach would be to set $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L} = J + \chi_{\mathcal{K}^\ell}$. However, the characteristic functional $\chi_{\mathcal{K}^L}$ in general cannot be represented on the coarser grids $\ell < L$. As a consequence, coarse grid corrections originating from $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L} = J + \chi_{\mathcal{K}^\ell}$ might interfere in an undesirable way with $J + \chi_{\mathcal{K}^L}$, thus spoiling the convergence or efficiency of the multilevel method, see [17]. Therefore, a suitable multiscale representation of the non-smooth nonlinearities has to be constructed, which guarantees the nonlinear convergence of our multiscale method as well as their efficiency and robustness.

For the contact problem, the leading minimization step (2) in Algorithm 1 can be realized by applying a nonlinear Gauß-Seidel method. By means of the resulting smoothed iterate $\bar{\boldsymbol{u}}^L$, we can define the set

$$\mathcal{A}^L_{\bar{\boldsymbol{u}}^{\boldsymbol{L}}} = \{p \in \mathcal{C}^L \mid \bar{\boldsymbol{u}}^{\boldsymbol{L}}(p) \cdot \boldsymbol{n}(p) = g(p)\} \tag{9}$$

of active nodes on level L. In order to ensure the feasibility of the coarse grid corrections, they must at least vanish at all active nodes $p \in \mathcal{A}^L_{\bar{\boldsymbol{u}}^{\boldsymbol{L}}}$ in normal direction. In general, using the standard nodal multilevel basis this is not possible. We now show how suitable subspaces $\mathcal{X}^L_{\bar{\boldsymbol{u}}^L}$ can be obtained. Let λ^ℓ_p be the standard nodal hat function for $p \in \mathcal{N}^\ell$ and let $\{\boldsymbol{E}_i\}_{1 \leq i \leq 3}$ denote the Cartesian basis vectors of $\mathbb{R}^d$. We replace the standard nodal basis functions $\boldsymbol{\lambda}^\ell_p = (\lambda^\ell_p \cdot \boldsymbol{E}_1, \ldots, \lambda^\ell_p \cdot \boldsymbol{E}_d)^T$ of S^ℓ for $p \in \mathcal{C}^\ell$ by

$$\{\lambda^\ell_p \cdot \boldsymbol{e}_1(p), \ldots, \lambda^\ell_p \cdot \boldsymbol{e}_d(p)\}\,, \tag{10}$$

where $\{\boldsymbol{e}_i\}_{1 \leq i \leq 3}$ is an orthonormal basis associated with $p \in \mathcal{C}^L$ and with $\boldsymbol{e}_1(p) = \boldsymbol{n}(p)$. As a consequence of (10), the first component of the displacements at the nodes $p \in \mathcal{C}^\ell$ is always the displacement in normal direction. Let now $I^{\ell+1}_\ell \colon \mathcal{S}^l \to \mathcal{S}^{l+1}$ denote the interpolation operator with respect to the local transformation (10). The algebraic representation $\boldsymbol{I}^{\ell+1}_\ell = (\boldsymbol{i}_{pq})_{p \in \mathcal{N}^{l+1}, q \in \mathcal{N}^l}$ of $I^{\ell+1}_\ell$ is a rectangular matrix with the 3×3 blocks $\boldsymbol{i}_{pq} \in \mathbb{R}^{3\times 3}$. We note that due to (10) for $p \in \mathcal{C}^L$ the blocks $\boldsymbol{i}_{pq}$ in general are not diagonal, if the normals differ along Γ_C. Now, we introduce the sets $A^\ell \subset \mathcal{C}^\ell \times \{1, \dots, d\}$ of active degrees of freedom for each Level $\ell \leq L$. On Level L, we set $A^L = \{(p, 1) \mid p \in \mathcal{A}^L_{\bar{\boldsymbol{u}}^L}\}$. On the coarser levels, the spaces $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$ can be defined by removing the degrees of freedom in A^ℓ from the nodal basis of $\mathcal{S}^\ell$. Possible choices for the multiscale representation of the set A^L include now $(1 \leq i, j \leq 3)$

1. $A^\ell = \{(q, 1) \mid q \in \mathcal{C}^\ell \text{ and } \bar{\boldsymbol{u}}^L(q) \cdot \boldsymbol{n}(q) = g(q)\}$.
2. Set recursively for $\ell < L$: $A^\ell = \{(q, 1) \mid \exists (p, 1) \in A^{\ell+1} : (\boldsymbol{i}^{\ell+1}_\ell)^{11}_{pq} \neq 0\}$.
3. Set recursively for $\ell < L$: $A^{\ell-1} = \{(q, j) \mid \exists (p, i) \in A^{\ell+1} : (\boldsymbol{i}^{\ell+1}_\ell)^{ij}_{pq} \neq 0\}$.

In addition to 1—3, we employ truncated basis functions $\{\boldsymbol{\mu}^\ell_q\}_{q \in \mathcal{N}^\ell}$, see [16]. For $\ell < L$, they can be defined by

$$\boldsymbol{\mu}^\ell_q = \boldsymbol{\lambda}^\ell_q - \sum_{p \in \operatorname{int} \operatorname{supp} \lambda^\ell_q \cap \mathcal{A}^L_{\bar{\boldsymbol{u}}^L}} \omega_{qp} \, \lambda^L_p \cdot \boldsymbol{n}(p) \,,$$

where the weights ω_{qp} are such that for all active nodes $p \in \mathcal{A}^L_{\bar{\boldsymbol{u}}^L}$ it holds for $\ell \leq L$ that $\boldsymbol{\mu}^\ell_q(p) \cdot \boldsymbol{n}(p) = 0$. Thus, the resulting multilevel basis provides a multiscale representation of the active constraints $\mathcal{A}^L_{\bar{\boldsymbol{u}}^L}$ on all coarser levels $\ell < L$. We remark that the search directions $\boldsymbol{\mu}^\ell_q$ are never explicitly computed, since the corresponding stiffness matrix can be obtained recursively by modifying the interpolation operator and using local reassembling.

Global Convergence

Despite the coarse grid spaces, we also have to choose the coarse grid energies $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L}$ and the convex sets of feasible corrections $\mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}$. Using the multigrid method as nonlinear solver by itself, following the idea of monotone multigrid methods, see [14], global convergence is achieved by guaranteeing that during the multigrid iteration process the convex functional $J + \chi_{\mathcal{K}^L}$ always decreases. The minimizer of (6) is sought by successive minimization in direction of all basis functions of the subspaces $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$ originating from the truncated basis. In order to ensure the feasibility of the coarse grid corrections, inner approximations $\mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}$ of the set $\mathcal{K}^L$ are constructed for $\ell < L$. Choosing $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L} = J$, then the monotonicity of the iteration guarantees the global convergence of the resulting monotone multigrid method for contact problems, see [16]. In the following, we denote this method by M-MG.

For unconstrained convex minimization problems, [21] has shown the convergence of a multilevel optimization method if the coarse grid problems are solved "accurately enough". Then, it can be guaranteed that a descent direction is provided by the coarse grid corrections. In [8], Trust-Region strategies are intertwined with multilevel optimization methods. In all cases, the sufficient decrease of the functional J is used to ensure the convergence of the multilevel method. Let us remark that the convergence proof in [10] for a smooth nonlinear multigrid method is also based on a minimization property.

Influence of the Multilevel Splitting

As an alternative to using the multigrid method as nonlinear solution method, it can also be applied as linear solver or as a preconditioner within a nonlinear strategy as, e.g. a primal dual active set strategy. We therefore consider the influence of the multilevel decompositions $\mathcal{X}^{\ell}_{\bar{\boldsymbol{u}}^L}$ given above in the context of a monotone multigrid method as well as in the context of a primal dual active set strategy. Let us define the active set of $\boldsymbol{u}^L_k$ by $\mathcal{A}^L_k = \{p \in \mathcal{C}^L \,|\, s_n(p) + \rho(\boldsymbol{u}^L_k(p) \cdot \boldsymbol{n}(p) - g(p)) > 0\}$, where $s_n(p) = a(\boldsymbol{u}^L_k, \lambda^L_p \cdot \boldsymbol{n}(p)) - f(\lambda^L_p \cdot \boldsymbol{n}(p))$ are the discrete normal stresses and $\rho > 0$ is an algorithmic parameter. An inexact multigrid based primal-dual active set strategy can be obtained from Algorithm 1 by replacing $\chi_{\mathcal{K}^L}$ in step (2) by the characteristic functional of the set $\mathcal{X}^L_{u^L_{k-1}} = \{\boldsymbol{v} \in \mathcal{S}^L \,|\, \boldsymbol{v}(p) \cdot \boldsymbol{n}(p) = 0,\ p \in \mathcal{A}^L_{k-1}\}$ and by using the linear coarse grid corrections induced by setting $\mathcal{Q}^{\ell}_{\bar{\boldsymbol{u}}^L} = J$ and $\mathcal{X}^{\ell}_{\bar{\boldsymbol{u}}^L} = \mathcal{D}^{\ell}_{\bar{\boldsymbol{u}}^L}$. In each step k, the steps (2) and (3) in Algorithm 1 amount to the inexact solution of a linear sub-problem of the form: find $\boldsymbol{c} \in \mathcal{X}^L_{u^L_{k-1}}$, such that

$$a(\boldsymbol{c}, \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) - a(\boldsymbol{w}, \boldsymbol{v}), \qquad \boldsymbol{v} \in \mathcal{X}^L_{u^L_{k-1}}, \tag{11}$$

where $\boldsymbol{w} \in \mathcal{S}^L$ and $\boldsymbol{w}(p) \cdot \boldsymbol{n}(p) = g(p)$ for $p \in \mathcal{A}^L_{k-1}$.

Primal-dual active set strategies are known to converge superlinearly, if the initial iterate $\boldsymbol{u}^L_0$ is sufficiently close to the solution. If, in addition the stiffness matrix is an M-matrix and the linear systems (11) are solved exactly, also global convergence can be shown, see [11]. Global convergence can also be obtained using Trust-Region Strategies. As a matter of fact, in case the linear sub-problems (11) are solved only inexactly, the choice of the employed multilevel decomposition strongly influences the convergence of the overall nonlinear strategy. We illustrate this for a Hertzian contact problem in $3d$. Here, a sphere is pressed in z-direction against the rigid plane $\{z = 0\}$. The material parameters are $E = 10^5$ and $\nu = 0.3$ and we have $L = 5$ levels of adaptive refinement and 659.409 degrees of freedom on Level 5. In Table 1, the resulting numbers of $\mathcal{W}(3,3)$-cycles are shown for this multigrid based active set strategy. We use the coarse grid spaces induced by the active sets A^{ℓ} given on the previous page and the truncated basis as well as the globally convergent monotone multigrid method M-MG with the truncated basis. The iteration is stopped, if $\|u^L_{k+1} - u^L_k\|_a / \|u^L_k - u^L_{k-1}\|_a \leq 10^{-12}$, $\|u\|_a = a(u,u)^{1/2}$. The initial iterate $\boldsymbol{u}^L_0$ is given by random values in the interval $[-0.2, -0.1]$. For the definition of the set $\mathcal{K}^L$ we consider two different cases. Firstly, the case of constant normal direction at Γ_C, i.e. we take as normal direction a $\boldsymbol{n}(p) = (0,0,-1)^T$ for all $p \in \mathcal{C}^L$ ("equal normals"), and secondly, $\boldsymbol{n}(p)$ the outer normal at $p \in \mathcal{C}^L$ ("outer normals"). As can be seen from Table 1, for the case of the outer normals, the constraints at the interface are locally not decoupled and spurious corrections from the coarser grids can spoil the convergence. We emphasize that the truncated basis functions showed to provide the best nonlinear search directions with respect to both, efficiency and robustness. As a by product, they can also be used for the multilevel representation of Dirichlet values. The slightly higher iteration numbers for the monotone multigrid method show the influence of the multilevel decomposition of the set $\mathcal{K}^L$, since for M-MG the feasibility of the coarse grid corrections is enforced by the construction of the sets $\mathcal{D}^{\ell}_{\bar{\boldsymbol{u}}^L}$.

Summing up, regardless of using a multigrid method as inexact solver within a non-smooth solution method or as nonlinear solver by itself, the multilevel decomposition has to be adapted to the active constraints in order to provide a fast and robust method.

4 Low Frequency Representation of Nonlinearities

Approximation by Quadratic Functionals

We now consider the case where the functional to be minimized is non-quadratic and non-differentiable. As example we choose elastic contact with Tresca friction. The corresponding minimization problem, after discretization, is given by: find $u \in \mathcal{K}^L$

$$(J + j^L_{s_n})(\boldsymbol{u}^L) \leq (J + j^L_{s_n})(\boldsymbol{v}), \quad \boldsymbol{v} \in \mathcal{K}^L. \tag{12}$$

Here, J is the elastic energy (8) and the friction functional $j^L_{s_n}$ is given by

$$j^L_{s_n}(\boldsymbol{v}) = \sum_{p \in \mathcal{C}^L} \mathcal{F}\, |s_{p,n}|\, |\boldsymbol{v}_{p,T}|, \tag{13}$$

$|\cdot|$ the Euclidean norm in $\mathbb{R}^2$ and $s_n = (s_{p,n})_{p \in \mathcal{C}^L}$ are the prescribed scaled boundary stresses, and $\mathcal{F} > 0$ is the coefficient of friction. We write $u_{p,n} = u_n(p)$ and $\boldsymbol{u}_{p,T} = \boldsymbol{u}_T(p)$. Tresca's friction law induces a non-smooth relationship between tangential displacements and tangential stresses, cf. [13]. Taking into account the efficiency and robustness of SQP-methods, the construction of the coarse level functionals $\mathcal{Q}^{\ell}_{\bar{\boldsymbol{u}}^L}$ might be based on a quadratic approximation of $J + j^L_{s_n}$. However, since $j^L_{s_n}$ is non-differentiable, this turns out to be a non-trivial task. We therefore proceed as follows, see [15, 17]. After $\bar{\boldsymbol{u}}^L$ in Algorithm 1 has been obtained, the subsequent coarse grid corrections are restricted to a neighborhood $\mathcal{D}^L_{\bar{\boldsymbol{u}}^L}$ of $\bar{\boldsymbol{u}}^L$ where $\bar{\boldsymbol{u}}^L_T \neq 0$ and therefore the energy $J + j^L_{s_n}$ is smooth. In contrast to the Trust-Region techniques given in [8], here the neighborhood $\mathcal{D}^L_{\bar{\boldsymbol{u}}^L}$ is locally defined by box constraints. This allows us to construct the coarse grid models $\mathcal{Q}^{\ell}_{\bar{\boldsymbol{u}}^L}$ on the basis of the quadratic approximation

$$\begin{aligned}\mathcal{Q}_{\bar{\boldsymbol{u}}^L}(\boldsymbol{v}) = \frac{1}{2}\big(a(\boldsymbol{v},\boldsymbol{v}) + j''_{\bar{\boldsymbol{u}}^L}(\bar{\boldsymbol{u}}^L)(\boldsymbol{v},\boldsymbol{v})\big)& \\ - \big(f(\boldsymbol{v}) - j'_{\bar{\boldsymbol{u}}^L}(\bar{\boldsymbol{u}}^L)(\boldsymbol{v}) + j''_{\bar{\boldsymbol{u}}^L}(\bar{\boldsymbol{u}}^L)(\bar{\boldsymbol{u}}^L,\boldsymbol{v})\big)&\end{aligned} \tag{14}$$

of $J + j^L_{s_n}$ on $\mathcal{D}^L_{\bar{\boldsymbol{u}}^L}$, where we have set

$$j_{\bar{\boldsymbol{u}}^L}(\boldsymbol{v}) = \sum_{p \in \mathcal{B}^L} \mathcal{F}|s_{p,n}||\boldsymbol{v}_{p,T}|\,, \quad \boldsymbol{v} \in \mathcal{S}^L\,. \tag{15}$$

Here, $\mathcal{B}^L \subset \mathcal{C}^L$ denotes the set of all sliding nodes w.r.t. $\bar{\boldsymbol{u}}^L$, i.e. all nodes $p \in \mathcal{C}^L$ with $\bar{\boldsymbol{u}}^L_{p,T} \neq 0$. As coarse grid model we use $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L} = \mathcal{Q}_{\bar{\boldsymbol{u}}^L}$ from (14). For the construction of the subspaces $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$ we again use the truncated basis functions. At all sticky nodes, i.e. $p \in \mathcal{C}^L$ with $\bar{\boldsymbol{u}}_{p,T} = 0$, truncation is employed in all directions, such that $\boldsymbol{\mu}^\ell_q \cdot \boldsymbol{e}_i(p) = 0$ for $1 \leq i \leq 3$. Now setting $\mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L} = \{\boldsymbol{v} \in \mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L} \mid (\bar{\boldsymbol{u}}^L_{p,T}, \boldsymbol{v}_{p,T}) \geq 0\,, \quad p \in \mathcal{C}^L\}$ the global convergence of the resulting multigrid method can be shown, see [18]. Again, the convergence proof relies on the successive minimization of the frictional energy, but now the coarse grid functionals $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L}$ are different from the fine grid functional.

Since the sliding directions $\bar{\boldsymbol{u}}^L_{p,T}$ differ along Γ_C, we again equilibrate the constraints by applying a basis transformation as in (10), but now only in the tangential space span$\{\boldsymbol{e}_2(p), \boldsymbol{e}_3(p)\}$. This allows for a better representation of the sets $\mathcal{D}^\ell_{\bar{\boldsymbol{u}}^L}$ in $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$.

As an example, we consider an elastic block pressed onto a rigid plane. A coarse triangulation of a cube with eight hexahedrons is refined adaptively until $190,888$ elements are obtained on Level $L = 6$. In Figure 1, the resulting number of iterates of M-MG on Level 6 are shown if this additional basis transformation is applied (lower line) or not (upper line), again for the stopping criterion given above. As can be seen, adapting the spaces $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$ to the nonlinearity j_{s_n} improves the robustness and efficiency of the method. For details, we refer to [18]. As an additional example, Figure 1 shows a torus in contact with a rigid foundation and the tangential stresses for $\mathcal{F} = 0.3$ at the contact interface. As can be seen, the sharp interface between sliding and sticky nodes in the tangential stresses is perfectly resolved by our non-smooth minimization approach.

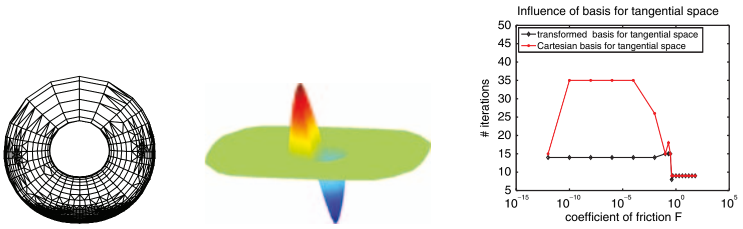


Fig. 1. Torus in contact with a rigid foundation *Left:* Deformed geometry. *Middle:* First component of the tangential stresses. *Right:* Block on plane: Robustness w.r.t to the coefficient of friction. Influence of coarse grid spaces $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$

Semi-smooth Approach

The regularized problem (3) motivates another possibility to ensure that the coarse grid corrections provide a descent direction for the minimization problem (6). To enforce the pointwise given constraints $u_n \leq g$ for our contact problem, one could use the classical logarithmic barrier function to obtain the smooth energy functional

$$(J + \chi_{\mathcal{K}^L_\mu})(\boldsymbol{u}) = J(\boldsymbol{u}) - \mu \sum_{p \in \mathcal{C}^L} \ln(g(p) + \varepsilon - u_n(p))\,, \tag{16}$$

$\mu > 0, \varepsilon \geq 0$ parameters. The disadvantage of this formulation is that ill-conditioning of the resulting Hessian may occur. Moreover, due to the regularization, the solution of the minimization problem (6) is only obtained in the limit $\mu \to 0$ and therefore some accuracy is lost. However, in the context of our nonlinear multigrid method, this approach can be used to construct the coarse grid energies $\mathcal{Q}^\ell_{\bar{\boldsymbol{u}}^L}$ for $\ell < L$ on the basis of the formulation (16). To this end, on Level L, the leading minimization step (2) in Algorithm 1 is done by means of a non-smooth method as, e.g. a nonlinear Gauß-Seidel method. Then, the spaces $\mathcal{X}^\ell_{\bar{\boldsymbol{u}}^L}$ are constructed using the truncated basis functions w.r.t (9). As coarse grid energies, we use the quadratic approximation (14) for the smooth energy (16). By means of this semi-smooth method, the coarse grid corrections are encouraged to stay within the feasible set, which gives rise to a "better" descent direction. In addition, the regularization does not influence the accuracy of the results, since it is only applied on the coarser grids. In Table 2, the resulting number of iterates for the contact problem from Figure 2 for $\mu = 10^{-4}$ and $\varepsilon = 10^{-7}$ are shown. Here, we compare the monotone multigrid method using the truncated basis functions with (M-MG) and without (NL-MG) enforcing the feasibility of the coarse grid corrections with the combined approach (C-MG). In order to stress the nonlinear iteration process, the components of the initial iterate $\boldsymbol{u}^L_0$ are chosen randomly in $[-100, 100]$. For the coarse grid problems, the respective method as algebraic multigrid method is used. As can be seen, the inner approximation of

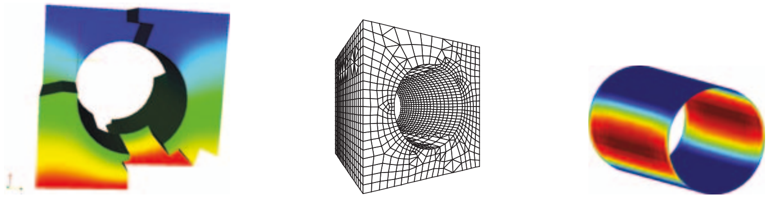


Fig. 2. Cube with hole in contact with a rigid cylinder inside *Left:* Parallel decomposition with 16 subdomains. *Middle:* Grid on Level $\ell = 1$. *Right:* Normal stresses.

Table 1. Iteration numbers illustrating the resolution of constraints for the different multilevel splittings given in Section 3

Level 5	Splitting 1	Splitting 2	Splitting 3	trc. Basis	M-MG
equal normals	15	34	34	15	17
outer normals	no conv.	no conv.	>100	15	25

Table 2. Non-smooth and combined non-smooth and regularization approach for randomly chosen initial iterate

Level	# it. M-MG	# it. NL-MG	# it C-MG	# dof	# contacts
1	34	34	34	5,016	228
2	44	24	23	22,326	854
3	70	140	35	142,146	3,226

Table 3. Iteration numbers illustrating the scalability for the parallelized non-smooth multigrid method M-MG. Nested iteration, example from Figure 2.

Level ℓ	#it. 1 Processor	#it. 2 Proc.	#it. 4 Proc.	#it. 8 Proc.	#it. 16 Proc.
2	16	17	17	17	16
3	17	18	18	18	18

the feasible set in M-MG requires additional iterations to identify the contact boundary. Setting $\mathcal{D}^{\ell}_{\bar{\boldsymbol{u}}^L} = \mathcal{X}^{\ell}_{\bar{\boldsymbol{u}}^L}$, as in NL-MG, in contrast to the previous section, here does not improve the convergence speed. However, the combined approach C-MG provides a good multilevel search strategy for bad initial iterates. We remark that in case of a better start iterate, all three strategies show similar iteration numbers. Our numerical experiments have been carried out in the framework of the finite element toolbox [2] and the C++–toolbox ObsLib++, see [17]. The hexahedral grids have been created using the Cubit grid generator, see [19].

Acknowledgement. We thank C. Groß for his help in implementing the UG/Cubit Interface and the anonymous referees for their remarks.

References

[1] L. Badea and J. Wang. An additive Schwarz method for variational inequalities. *Math. Comp.*, 69(232):1341–1354, 1999.

[2] P. Bastian, K. Birken, K. Johannsen, S. Lang, N. Neuß, H. Rentz–Reichert, and C. Wieners. UG – a flexible software toolbox for solving partial differential equations. *Comput. Vis. Sci.*, 1:27–40, 1997.

[3] V. Belsky. A multi–grid method for variational inequalities in contact problems. *Computing*, 51:293–311, 1993.

[4] M. Benzi, E. Haber, and L. R. Hansson. Multilevel algorithms for large scale interior point methods in bound constraint optimization. Technical report, Emory University, Atlanta, 2005.

[5] A. Brandt and C.W. Cryer. Multigrid algorithms for the solution of linear complementarity problems arising from free boundary problems. *SIAM J. Sci. Stat. Comput.*, 4:655–684, 1983.

[6] Z. Dostál, F.A.M. Gomes Neto, and S.A. Santos. Duality based domain decomposition with natural coarse space for variational inequalities. *J. Comput. Appl. Math.*, 126(1-2):397–415, 2000.

[7] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems.* Series in Computational Physics. Springer, New York, 1984.

[8] S. Gratton, A. Sartenaer, and P. Toint. Recursive trust-region methods for multiscale nonlinear optimization. *CERFACS*, 06(32), 05 2006. 06.

[9] W. Hackbusch and H.D. Mittelmann. On multi–grid methods for variational inequalities. *Numer. Math.*, 42:65–76, 1983.

[10] W. Hackbusch and A. Reusken. Analysis of a damped nonlinear multilevel method. *Numer. Math.*, 55:225–246, 1989.

[11] M. Hintermüller, K. Ito, and K. Kunisch. The primal dual active set strategy as a semismooth Newton method. *SIAM J. Optim.*, 13(3):865–888, 2003.

[12] R.H.W. Hoppe. Multigrid algorithms for variational inequalities. *SIAM J. Numer. Anal.*, 24:1046–1065, 1987.

[13] N. Kikuchi and J.T. Oden. *Contact Problems in elasticity.* SIAM, 1988.

[14] R. Kornhuber. *Adaptive Monotone Multigrid Methods for Nonlinear Variational Problems.* Teubner–Verlag, Stuttgart, 1997.

[15] R. Kornhuber. On constrained Newton linearization and multigrid for variational inequalities. *Numer. Math.*, 91:699–721, 2002.

[16] R. Kornhuber and R. Krause. Adaptive multigrid methods for Signorini's problem in linear elasticity. *Comput. Vis. Sci.*, 4:699–721, 2001.

[17] R. Krause. From inexact active set strategies to nonlinear multigrid methods. In P. Wriggers and U. Nackenhorst, editors, *Analysis and Simulation of Contact Problems*, volume 27. Springer, 2006.

[18] R. Krause. A non-smooth multiscale method for solving frictional two-body contact problems in $2d$ and $3d$ with multigrid efficiency. Technical Report INS–Preprint No. 604, INS, University of Bonn, 2006.

[19] Sandia National Laboratories. Cubit 9.1 mesh generation toolkit, 2004.

[20] J. Mandel. A multi–level iterative method for symmetric, positive definite linear complementarity problems. *Appl. Math. Optim.*, 11:77–95, 1984.

[21] S. G. Nash. A multigrid approach to discretized optimization problems. *Optim. Methods Softw.*, 14:99–116, 2000.

[22] X. C. Tai. Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. *Numer. Math.*, 93:755–786, 2003.

[23] X.-C. Tai and J. Xu. Global and uniform convergence of subspace correction methods for some convex optim. problems. *Math. Comp.*, 71:105–124, 2002.

[24] M. Weiser. Interior point methods in function space. Technical report, Zuse Institute Berlin (ZIB), Berlin, 2003.

Domain Decomposition Preconditioner for Anisotropic Diffusion

Yuri A. Kuznetsov

Department of Mathematics, University of Houston, 651 Philip G. Hoffman Hall,
Houston, TX 77204–3008, USA. kuz@math.uh.edu

Summary. We propose and investigate two-level preconditioners for the diffusion equations with anisotropic coefficients in model polyhedral domains. Preconditioners are based on a partitioning of the mesh in (x, y)-plane into non-overlapping subdomains and on a special coarsening algorithm in each of the mesh layers. The condition number of the preconditioned matrix does not depend on the coefficients in the diffusion operator. Numerical experiments confirm the theoretical results.

1 Introduction

In this paper, we propose and investigate a new approach to the construction of two-level preconditioners for the diffusion equation with anisotropic diffusion tensor. We consider the case of special polyhedral domains and special prismatic meshes. The diffusion tensor is assumed to be a diagonal matrix, and the simplest version of the finite volume method is used for the discretization of the diffusion equation. The choice of the domain, the meshes, and the discretization method are motivated by applications in reservoir simulation. We can also use for the discretization the polyhedral "div–const" mixed finite element method, see [4].

The paper is organized as follows. In Section 2, we describe the model problem and the matrices which arise in the simplest version of the finite volume method.

In Section 3, we propose a special coarsening procedure based on the partitioning of the mesh domain in (x, y)-plane into non-overlapping subdomains. This procedure is a special modification of the algorithm earlier proposed in [6] and utilized in [2]. We prove that the condition number of the preconditioned matrix is independent of the values of the coefficients in the diffusion equation, i.e. it does not depend on an anisotropy in the diffusion tensor.

An implementation algorithm in the form of a two-step iterative method is considered in Section 4. It is based on the idea of the matrix iterative methods in subspaces, see [3, 5]. The algorithm naturally leads to a coarse mesh system.

In Section 5, we design another two-level preconditioner which is much cheaper with respect to the arithmetic implementation cost than the previous one. Another advantage of the second preconditioner is that it allows a multilevel extension. Numerical results in Section 6 demonstrate the efficiency of the second preconditioner. They confirm the theoretical results in Sections 3 and 5.

2 Formulation of Model Problem

We consider the Neumann boundary value problem for the diffusion equation

$$\begin{aligned} -\nabla\cdot\big(a\nabla p\big) + c\,p = f \quad &\text{in} \quad \Omega \\ \big(a\nabla p\big)\cdot\boldsymbol{n} = 0 \quad &\text{on} \quad \partial\Omega \end{aligned} \tag{1}$$

where $\Omega = \Omega_{xy}\times(0;1)$ is a prismatic domain in $\mathbb{R}^3$, $\boldsymbol{n}$ is the outward unit normal to $\partial\Omega$, and Ω_{xy} is a polygon in the (x,y)-coordinate plane. An example of Ω is given in Figure 1. The diffusion tensor a is a diagonal 3×3 matrix with coinciding diagonal entries in (x,y)-plane, i.e. $a = \text{diag}\,\{a_{xy},\ a_{xy},\ a_z\}$, and c is a non-negative function. The domain Ω is partitioned into subdomains $\Omega_l = \Omega_{xy,l}\times(Z_B^{(l)};Z_T^{(l)})$ where $\Omega_{xy,l}$ are convex polygons, $0\le Z_B^{(l)} < Z_T^{(l)}\le 1$, $l=\overline{1,\ m}$, and m is a positive integer. We assume that a_{xy} and a_z are positive constants and c is a non-negative constant in each of the subdomains $\Omega_{xy,l}$, $l=\overline{1,\ m}$. We also assume that the coefficient c is positive in at least one subdomain Ω_l, $1\le l\le m$.

Let $\Omega_{xy,h}$ be a conforming polygonal mesh in Ω_{xy}, and Z_h be a partitioning of $[0;1]$ into segments $[z_{k-1};z_k]$, $k=\overline{1,\ n_z}$, where n_z is a positive integer. Then, $\Omega_h = \Omega_{xy,h}\times Z_h$ is a prismatic mesh in Ω. We assume that the mesh Ω_h is conforming with respect to the boundaries $\partial\Omega_l$ of subdomains Ω_l, $l=\overline{1,\ m}$. We also assume that the interfaces between neighboring cells in $\Omega_{xy,h}$ are always straight segments.

To discretize diffusion equation we utilize the simplest version of the finite volume method. In the case of uniform rectangular or hexagonal meshes $\Omega_{xy,h}$ this discretization is sufficiently accurate. We may also assume that $\Omega_{xy,h}$ is a Voronoi mesh. The simplest finite volume method results in the system of linear algebraic equations

$$K\,\bar{p} \;=\; \bar{f} \tag{2}$$

with a symmetric positive definite $\tilde{n}\times\tilde{n}$ matrix K where $\tilde{n} = \tilde{n}_{xy}\times n_z$ and $\tilde{n}_{xy}$ is the total number of polygonal cells in $\Omega_{xy,h}$. System (2) can be easily hybridized algebraically by introducing additional degrees-of-freedom (DOF) λ on the interfaces between all or selected neighboring mesh cells in $\Omega_{xy,h}$ as well as on the edges of cells in Ω_{xy} belonging to the boundary $\partial\Omega_{xy}$ of Ω_{xy}. In terms of old variables p and new variables λ the underlying system of linear algebraic equations can be written as follows:

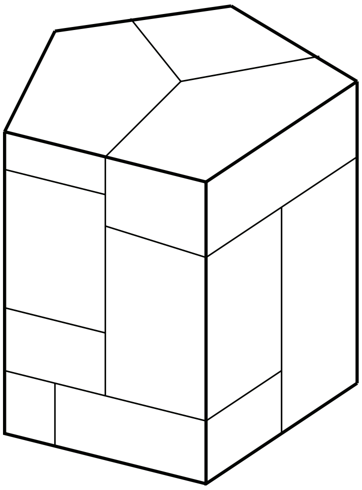


Fig. 1. An example of Ω partitioned into subdomains Ω_l, $l = \overline{1,\ m}$

$$A \begin{bmatrix} \bar{p} \\ \bar{\lambda} \end{bmatrix} \equiv \begin{pmatrix} A_p & A_{p\lambda} \\ A_{\lambda p} & A_{\lambda} \end{pmatrix} \begin{bmatrix} \bar{p} \\ \bar{\lambda} \end{bmatrix} = \overline{F}. \tag{3}$$

The matrix

$$K = A_p - A_{p\lambda} A_{\lambda}^{-1} A_{\lambda p} \tag{4}$$

in (2) is the Schur complement of A, and

$$\overline{F} = \begin{bmatrix} \bar{f} \\ 0 \end{bmatrix}. \tag{5}$$

The definition of λ is based on the observation that the three-point finite difference equation

$$\left(\frac{2a_1}{h_1} + \frac{2a_2}{h_2}\right) w + \left(p_2 - p_1\right) = 0 \tag{6}$$

for the flux equation $w + \partial p/\partial \xi = 0$ on the interface between two cells is equivalent to three equations

$$\begin{aligned} &\frac{2a_1}{h_1} w_1 - p_1 - \lambda = 0, \\ &\frac{2a_2}{h_2} w_2 + p_2 + \lambda = 0, \\ &w_1 - w_2 = 0 \end{aligned} \tag{7}$$

with $w = w_1 = w_2$. Then, we use the standard condensation procedure to derive system (3).

Let us present the matrix K as the sum of two matrices:

$$K \;=\; K_{xy} \;+\; K_z \tag{8}$$

where K_{xy} corresponds to the discretization of the operator

$$\mathcal{L}_{xy} \;=\; -\frac{\partial}{\partial x}\big(a_{xy}\frac{\partial}{\partial x}\big) \;-\; \frac{\partial}{\partial y}\big(a_{xy}\frac{\partial}{\partial y}\big) \tag{9}$$

and K_z corresponds to the discretization of the operator

$$\mathcal{L}_z \;=\; -\frac{\partial}{\partial z}\big(a_z\frac{\partial}{\partial z}\big) \;+\; c. \tag{10}$$

Then,

$$A \;=\; A_{xy} \;+\; A_z \tag{11}$$

where

$$A_{xy} \;=\; \begin{pmatrix} A_{xy,p} & A_{p\lambda} \\ A_{\lambda p} & A_{\lambda} \end{pmatrix}, \qquad A_z \;=\; \begin{pmatrix} K_z & 0 \\ 0 & 0 \end{pmatrix}, \tag{12}$$

and

$$K_{xy} \;=\; A_{xy,p} \;-\; A_{p\lambda}\, A_{\lambda}^{-1}\, A_{\lambda p} \tag{13}$$

is the Schur complement of the matrix A_{xy}.

We observe that with an appropriate permutation matrix P the matrix $PA_{xy}P^T$ is a block diagonal matrix, i.e.

$$P\,A_{xy}\,P^T \;=\; A_{xy}^{(1)} \;\oplus\; \cdots \;\oplus\; A_{xy}^{(n_z)} \tag{14}$$

where the submatrices $A_{xy}^{(k)}$ correspond to the above hybridized discretization of the operator $\mathcal{L}_{xy}$ in (9) in the mesh layers $\varOmega_{xy,h} \times [z_{k-1}; z_k]$, $k = \overline{1,\; n_z}$.

In the next section, we shall derive a preconditioner H for the matrix A in (3). Let us assume that A and B are symmetric and positive definite matrices and the inequalities

$$\alpha\,\big(B\,\bar{v},\; \bar{v}\big) \;\le\; \big(A\,\bar{v},\; \bar{v}\big) \;\le\; \beta\,\big(B\,\bar{v},\; \bar{v}\big) \tag{15}$$

hold for all $\bar{v} \in \mathbb{R}^n$ with some positive coefficients α and β, where n is the size of A. We present the matrix $H = B^{-1}$ in the block form similar to (3):

$$H \;=\; \begin{pmatrix} H_p & H_{p\lambda} \\ H_{\lambda p} & H_{\lambda} \end{pmatrix}. \tag{16}$$

Then, the inequalities

$$\frac{1}{\beta}\,\big(H_p\,\bar{q},\; \bar{q}\big) \;\le\; \big(K^{-1}\,\bar{q},\; \bar{q}\big) \;\le\; \frac{1}{\alpha}\,\big(H_p\,\bar{q},\; \bar{q}\big) \tag{17}$$

hold for all $\bar{q} \in \mathbb{R}^{\tilde{n}}$. Thus, with respect to estimates (15) the matrix H_p is not a worse preconditioner for the matrix K than the preconditioner $H = B^{-1}$ for the matrix A.

3 Two-Level Preconditioner

Let $\Omega_{xy,h}$ be partitioned into non-overlapping mesh subdomains $G_{h,s}$, $s = \overline{1,\ t}$, where t is a positive integer. We assume that this partitioning is conforming with respect to the boundaries of subdomains $\Omega_{xy,l}$, i.e. the boundaries of $\Omega_{xy,l}$ are subsets of the union of the boundaries of subdomains $G_{h,s}$, $s = \overline{1,\ t}$, $l = \overline{1,\ m}$. It follows from the above assumptions that the coefficients a_{xy} and a_z are positive constants $a_{xy}^{k,s}$ and $a_z^{k,s}$, respectively, and the coefficient c is a non-negative constant $c_{k,s}$ in each of the mesh subdomains $G_{h,s} \times [z_{k-1}; z_k]$, $s = \overline{1,\ t}$, $k = \overline{1,\ n_z}$.

We assume that the additional DOF λ are imposed only on the interface boundaries between mesh subdomains $G_{h,s}$ and $G_{h,s'}$, $s' \neq s$, $s, s' = \overline{1,\ t}$, and on the boundary $\partial\Omega_{xy}$. Then, assembling matrices N_s, $s = \overline{1,\ t}$, exist such that

$$A_{xy}^{(k)} \;=\; h_{z,k} \sum_{s=1}^{t} a_{xy}^{k,s}\, N_s\, A_{xy,s}\, N_s^T \tag{18}$$

where $h_{z,k} = z_k - z_{k-1}$, $k = \overline{1,\ n_z}$, and $A_{xy,s}$ represents the hybridized discretization of the operator $\mathcal{L}_{xy}$ in mesh subdomain $G_{h,s}$, $s = \overline{1,\ t}$. The matrices $A_{xy,s}$ are symmetric and positive semi-definite, and ker $A_{xy,s}$ (null-space of $A_{xy,s}$) is the span of $\bar{e}_s \in \mathbb{R}^{n_s}$ where $\bar{e}_s = \big(\ 1,\ \ldots\ ,\ 1\big)^T$ and n_s is the size of $A_{xy,s}$, $s = \overline{1,\ t}$.

Let D_s be a diagonal $n_s \times n_s$ matrix with positive entries on the diagonal, $1 \leq s \leq t$. Consider the eigenvalue problem

$$A_{xy,s}\, \bar{w} \;=\; \mu\, D_s\, \bar{w}, \quad \bar{w} \in \mathbb{R}^{n_s}. \tag{19}$$

Then the spectral decomposition of $A_{xy,s}$ is defined as follows:

$$A_{xy,s} \;=\; D_s\, W_s\, \Lambda_s\, W_s^T\, D_s \tag{20}$$

where

$$\Lambda_s \;=\; \mathrm{diag}\Big\{\mu_1^{(s)},\ \mu_2^{(s)},\ \cdots\ ,\ \mu_{n_s}^{(s)}\Big\} \tag{21}$$

is a diagonal matrix, and

$$W_s \;=\; \Big(\bar{w}_{s,1},\ \bar{w}_{s,2},\ \ldots\ ,\ \bar{w}_{s,n_s}\Big). \tag{22}$$

Here, $0 = \mu_1^{(s)} < \mu_2^{(s)} \leq \cdots \leq \mu_{n_s}^{(s)}$ are the eigenvalues, and $\bar{w}_{s,1},\ \bar{w}_{s,2},\ \ldots,$ $\bar{w}_{s,n_s}$ are the corresponding D_s-orthonormal eigenvectors. It is obvious that $\bar{w}_{s,1} = \sigma_s^{-1}\bar{e}_s$ with $\sigma_s = \big(D_s\bar{e}_s, \bar{e}_s\big)^{1/2}$.

Let us define the matrices

$$B_{xy,s} \;=\; \hat{\mu}_s \Big[\, D_s \;-\; D_s\, \bar{w}_{s,1}\, \bar{w}_{s,1}^T\, D_s \,\Big] \tag{23}$$

where $\hat{\mu}_s$ are arbitrary positive numbers, $s = \overline{1,\ t}$. It can be easily shown that the inequalities

$$\mu_2^{(s)} \left(B_{xy,s}\,\bar{v},\ \bar{v}\right) \;\le\; \hat{\mu}_s \left(A_{xy,s}\,\bar{v},\ \bar{v}\right) \;\le\; \mu_{n_s}^{(s)} \left(B_{xy,s}\,\bar{v},\ \bar{v}\right) \tag{24}$$

hold for all $\bar{v} \in \mathbb{R}^{n_s}$, $s = \overline{1,\ t}$.

We define the matrices

$$B_{xy}^{(k)} \;=\; h_{z,k} \sum_{s=1}^{t} a_{xy}^{k,s}\, N_s\, B_{xy,s}\, N_s^T\,, \tag{25}$$

$k = \overline{1,\ n_z}$, the matrix

$$B_{xy} \;=\; P^T \left(B_{xy}^{(1)} \;\oplus\; \cdots \;\oplus\; B_{xy}^{(n_z)}\right) P, \tag{26}$$

and, finally, the matrix

$$B \;=\; B_{xy} \;+\; A_z. \tag{27}$$

The matrix B in (27) may be considered as the first candidate to precondition the matrix A in (3).

It can be easily proved that for the matrix B in (27) inequalities (15) hold with

$$\alpha \;=\; \min\left\{1;\ \min_{1\le s\le t} \frac{\mu_2^{(s)}}{\hat{\mu}_s}\right\}, \quad \beta \;=\; \max\left\{1;\ \max_{1\le s\le t} \frac{\mu_{n_s}^{(s)}}{\hat{\mu}_s}\right\} \tag{28}$$

where $\mu_2^{(s)}$ and $\mu_{n_s}^{(s)}$ are the minimal non-zero and the maximal eigenvalues in (19), respectively.

Let $\mathrm{cond}_A(B^{-1}A)$ be the condition number of the matrix $B^{-1}A$ with respect to the norm generated by the matrix A. Then, the estimate

$$\mathrm{cond}_A\bigl(B^{-1}\,A\bigr) \;\le\; \nu \tag{29}$$

holds with $\nu = \beta/\alpha$ where the values of α and β are given in (28). We observe that the value of ν does not depend on the values of the coefficients a_{xy}, a_z, and c in diffusion equation (1) as well as on the mesh Z_h.

To define a proper diagonal matrix D_s in (19) we have to analyze the matrix $A_{xy,s}$ and the restriction of the mesh $\Omega_{xy,h}$ onto the subdomain $G_{h,s}$, $1 \le s \le t$. The matrices $A_{xy,s}$ and D_s can be presented in the 2×2 block form by

$$A_{xy,s} \;=\; \begin{pmatrix} A_p & A_{p\lambda} \\ A_{\lambda p} & A_\lambda \end{pmatrix}, \qquad D_s \;=\; \begin{pmatrix} D_p & 0 \\ 0 & D_\lambda \end{pmatrix} \tag{30}$$

where the index "s" in the blocks is omitted. Here, diagonal blocks A_p and D_p are associated with the cell-centered DOFs, and diagonal blocks A_λ and D_λ are associated with the interface DOFs. Let E be a polygonal cell in $G_{h,s}$.

Then, the diagonal entry of the matrix D_p in (30), associated with E, is equal to the area of E. The boundary of $G_{h,s}$ is the union of edges of polygonal cells E in $G_{h,s}$. We assigned with each of such edges one DOF in $\bar{\lambda}$, and with each DOF in $\bar{\lambda}$ we associate the length of the underlying edge in $G_{h,s}$. Moreover, the boundary of $G_{h,s}$ consists of the interfaces $\Gamma_{s,j}$, $j = \overline{1,\ l_s}$, between $G_{h,s}$ and neighboring subdomains $G_{h,s'}$, $s' \neq s$, as well as of the interfaces between $G_{h,s}$ and $\partial\Omega_{xy}$ where l_s is a positive integer. We assume that each of the interfaces is a simply connected subset of the boundary of $G_{h,s}$. We assign for each of the interfaces $\Gamma_{s,j}$ a positive number $d_{s,j}$, $j = \overline{1,\ l_s}$. We assume that for the interfaces $\Gamma_{s,j} = \Gamma_{s',j'}$ between neighboring subdomains $G_{h,s}$ and $G_{h,s'}$, $s' \neq s$, the values $d_{s,j}$ and $d_{s',j'}$ are equal to each other. Now, we define the diagonal entries of the matrix D_λ in (30). Let λ be a DOF in $\bar{\lambda}$ assigned for a segment γ belonging to interface $\Gamma_{s,j}$, $1 \leq j \leq l_s$. Then, the associated with λ the diagonal entry of D_λ is the product of the length of γ and $d_{s,j}$.

To derive estimates for α and β in (28), we assume that the mesh $\Omega_{xy,h}$ and the partitioning of $\Omega_{xy,h}$ into subdomains $G_{h,s}$, $s = \overline{1,\ t}$, are quasi-uniform and regular shaped. On the basis of the latter assumptions we introduce two parameters:

$$h_f \ = \ \tilde{n}_{xy}^{-1/2} \quad \text{and} \quad h_c \ = \ t^{-1/2}. \tag{31}$$

It is clear that h_f and h_c can be called as the fine mesh step size and the coarse mesh step size, respectively. We assume that $d_{s,j} = h_f$ in the definition of the diagonal entries of the submatrices D_λ in (30), $j = \overline{1,\ l_s}$, $s = \overline{1,\ t}$.

It can be proved that under the above assumptions the estimates

$$\begin{aligned} \min_{1\leq s\leq t} \mu_2^{(s)} &\geq c_1\, h_c^{-2} \\ \max_{1\leq s\leq t} \mu_{n_s}^{(s)} &\leq c_2\, h_f^{-2} \end{aligned} \tag{32}$$

hold, where c_1 and c_2 are positive constants independent of the mesh $\Omega_{xy,h}$ and subdomains $G_{h,s}$, $s = \overline{1,\ t}$.

Let us choose $\hat{\mu}_s = h_c^{-2}$, $s = \overline{1,\ t}$. Then, combining (28), (29), and (32) we get the following result.

Proposition 1. *Under assumptions made the estimate*

$$\text{cond}_A\big(B^{-1}A\big) \ \leq \ c_3\Big(\frac{h_c}{h_f}\Big)^2 \tag{33}$$

holds where c_3 is a positive constant independent of the coefficients a_{xy}, a_z, and c in (1), mesh Ω_h, and the subdomains $G_{h,s}$, $s = \overline{1,\ t}$.

Thus, the proposed preconditioner is robust with respect to the diffusion tensor but it is not optimal with respect to the mesh in the case $h_c \gg h_f$.

4 Implementation Algorithm

In this Section, we derive a solution algorithm for an algebraic system

$$B\,\bar{v} \;=\; \bar{g} \tag{34}$$

with the matrix B defined in (27) and a right hand side vector $\bar{g} \in \mathbb{R}^n$.

The solution algorithm is based on the splitting

$$B \;=\; B_0 \;-\; C_0 \tag{35}$$

of the matrix B into the matrices

$$B_0 \;=\; A_z \;+\; D \tag{36}$$

and

$$C_0 \;=\; P^T \left[C_0^{(1)} \;\oplus\; \ldots \;\oplus C_0^{(n_z)} \right] P. \tag{37}$$

Here,

$$D \;=\; P^T \left[\widetilde{D}_1 \;\oplus\; \ldots \;\oplus \widetilde{D}_{n_z} \right] P, \tag{38}$$

is a diagonal matrix with diagonal submatrices

$$\widetilde{D}_k \;=\; h_{z,k} \sum_{s=1}^{t} \hat{\mu}_s \, a_{xy}^{k,s} \, N_s \, D_s \, N_s^T, \tag{39}$$

$k = \overline{1,\; n_z}$, and

$$C_0 \;=\; D \;-\; B_{xy}, \tag{40}$$

where B_{xy} is defined in (25), (26).

The implementation algorithm consists of two steps. At the first step, we compute the solution vector of the system

$$B_0\,\bar{v}_1 \;=\; \bar{g}. \tag{41}$$

With a proper permutation matrix P_z the matrix $P_z B_0 P_z^T$ is a block diagonal matrix. Each diagonal block of this matrix is either a tridiagonal matrix (for p-variables) or a diagonal matrix (for λ-variables). The total number of blocks is equal to n_{xy}.

At the second step, we are looking for the vector

$$\bar{v}_2 \;=\; \bar{v}_1 \;+\; \bar{\eta} \tag{42}$$

where $\bar{\eta}$ is the solution vector of the system

$$B\,\bar{\eta} \;=\; -\big(B\,\bar{v}_1 \;-\; \bar{g}\big), \tag{43}$$

or of the equivalent system

$$B\,\bar{\eta} \;=\; \bar{\xi} \tag{44}$$

with the right hand side vector

$$\bar{\xi} \;=\; C_0\,B_0^{-1}\,\bar{g}. \tag{45}$$

It is obvious that the vector $\bar{v}_2$ in (42) is the solution of system (34).

The vector $\bar{\xi}$ in (45) belongs to the image of the matrix C_0. We observe that the rank of C_0 is equal to $t \times n_z$. It is much smaller than the size of system (44).

The crucial observation for the implementation algorithm is that the components of the solution vector $\bar{\eta}$ in (44) have a special structure. Namely, in mesh layer $\Omega_{xy,h} \times [z_{k-1}; z_k]$ all the components of the solution vector $\bar{\eta}$ corresponding to the interior of $G_{h,s}$, $1 \le s \le t$, are equal, and all the components of $\bar{\eta}$, corresponding to the interfaces $\Gamma_{s,j}$, $1 \le j \le l_s$, between neighboring subdomains $G_{h,s}$ and $G_{h,s'}$, $s' \ne s$, or between $G_{h,s}$ and the boundary of Ω_{xy}, are equal.

For instance, if $G_{h,s}$ is a polygon with six interfaces $\Gamma_{s,j}$, $j = \overline{1,\,6}$, then the components of the subvector of $\bar{\eta}$ assigned for this subdomain may take only seven different values.

In the matrix form, the above property of the vector $\bar{\eta}$ in (44) can be presented by the formula

$$\bar{\eta} \;=\; R\,\bar{\psi} \tag{46}$$

where $\bar{\psi} \in \mathbb{R}^{n_c}$ and R is an $n \times n_c$ matrix. Here, $n_c = n_{xy,c} \times n_z$ where $n_{xy,c}$ is equal to the total number of subdomains $G_{h,s}$ and different interfaces $\Gamma s, j$, $j = \overline{1,\,l_s}$, $s = \overline{1,\,t}$. It is clear that the matrix R has only one non-zero entry in each row, and this entry is equal to one. Thus, system (44) can be replaced by an equivalent system

$$B_c\,\bar{\psi} \;=\; \bar{\phi} \tag{47}$$

where

$$B_c \;=\; R^T\,B\,R \quad \text{and} \quad \bar{\phi} \;=\; R^T\,\bar{\xi}. \tag{48}$$

Here, B_c is said to be a coarse mesh matrix.

The above implementation algorithm can be presented in the form of the two-step iterative procedure: $\bar{v}_0 = 0$,

$$\begin{aligned} \bar{v}_1 &= \bar{v}_0 - B_0^{-1}\left(B\,\bar{v}_0 \;-\; \bar{g}\right), \\ \bar{v}_2 &= \bar{v}_1 - R\,B_c^{-1}\,R^T\left(B\,\bar{v}_1 \;-\; \bar{g}\right) \end{aligned} \tag{49}$$

where $v_2 = B^{-1}\bar{g}$ is the solution vector of system (34).

Let us introduce the matrix

$$T \;=\; \bigl(I \;-\; R\,B_c^{-1}\,R^T\,B\bigr)\,\bigl(I \;-\; B_0^{-1}\,B\bigr). \tag{50}$$

Then we get

$$\bar{v}_2 \;=\; \bigl(I \;-\; T\bigr)\,B^{-1}\,\bar{g} \tag{51}$$

where I is the identity $n \times n$ matrix. Because $\bar{v} = \bar{v}_2$ we get the formula

$$H \;\equiv\; B^{-1} \;=\; \bigl(I \;-\; T\bigr)\,B^{-1}. \tag{52}$$

It follows immediately that T is the null matrix.

5 A Better Two-Level Preconditioner

In this Section, we derive another preconditioner for the matrix A in (3) which is spectrally equivalent to preconditioner H in (52) but its implementation is much cheaper.

Let us complement iterative procedure (49) with one additional iteration step:

$$\bar{v}_3 \;=\; \bar{v}_2 \;-\; B_0^{-1}\,\bigl(B\,\bar{v}_2 \;-\; \bar{g}\bigr). \tag{53}$$

It is obvious that $\bar{v}_3 = \bar{v}_2$. Thus, we derived an alternative representation

$$B^{-1} \;=\; \Bigl[I \;-\; \bigl(I \;-\; B_0^{-1}\,B\bigr)\,T\Bigr]\,B^{-1} \tag{54}$$

for the matrix $H = B^{-1}$.

Let a matrix $\widehat{B}_c$ be spectrally equivalent to the matrix B_c in (48), i.e. the inequalities

$$q_1\,\bigl(\widehat{B}_c\bar{u},\;\bar{u}\bigr) \;\le\; \bigl(B_c\bar{u},\;\bar{u}\bigr) \;\le\; q_2\,\bigl(\widehat{B}_c\bar{u},\;\bar{u}\bigr) \tag{55}$$

hold for all $\bar{u} \in \mathbb{R}^{n_c}$ with positive constants q_1 and q_2 independent of the coefficients of the diffusion operator in (1) and the mesh Ω_h.

Let us introduce the matrix

$$\widehat{T} \;=\; \bigl(I \,-\, B_0^{-1}\,B\bigr)\,\bigl(I \,-\, q_3\,R\,\widehat{B}_c^{-1}\,R^T\,B\bigr)\,\bigl(I \,-\, B_0^{-1}\,B\bigr) \tag{56}$$

where q_3 is a positive constant independent of the coefficients in (1) and the mesh Ω_h, and satisfying the inequality $q_3 < 2/q_2$. Then, the matrix

$$\widehat{H} \;=\; I \;-\; \widehat{T} \tag{57}$$

is spectrally equivalent to the matrix H in (52).

To describe the derivation procedure for a matrix $\widehat{B}_c$ we consider a polygonal subdomain $G_{h,s}$ with interface boundaries $\Gamma_{s,j}$, $j = \overline{1,\;l_s}$, $1 \le s \le t$. In this case, we have

$$\sigma_s^2 \;=\; |G_{h,s}| \;+\; h_f \sum_{j=1}^{l_s} |\Gamma_{s,j}|. \tag{58}$$

Here, $|G_{h,s}|$ is the area of $G_{h,s}$ and $|\Gamma_{s,j}|$ is the length of $\Gamma_{s,j}$, $j = \overline{1,\ l_s}$, $1 \le s \le t$.

To define the matrix $\widehat{B}_c$, we replace each submatrix

$$R_s^T \left(D_s \;-\; D_s \,\bar{w}_{1,s}\, \bar{w}_{1,s}^T \, D_s \right) R_s \tag{59}$$

in the matrix B_c, where R_s is the underlying block in R, by the matrix

$$\frac{h_f\,|G_{h,s}|}{\sigma_s^2} \left(\widehat{D}_s \;-\; \check{D}_s \, Q_s \, \check{D}_s \right) \tag{60}$$

where $\widehat{D}_s = \mathrm{diag}\big\{ \sum\limits_{j=1}^{l_s} |\Gamma_{s,j}|, |\Gamma_{s,1}|, \ldots, |\Gamma_{s,l_s}| \big\}$, $\check{D}_s = \mathrm{diag}\big\{1, |\Gamma_{s,1}|, \ldots, |\Gamma_{s,l_s}|\big\}$, and

$$Q_s \;=\; \begin{pmatrix} 0 \; 1 \ldots 1 \\ 1 \; 0 \ldots 0 \\ \vdots \;\; \vdots \quad\;\; \vdots \\ 1 \; 0 \ldots 0 \end{pmatrix} \in \mathbb{R}^{(l_s+1)\times(l_s+1)}. \tag{61}$$

It can be proved that the matrix $\widehat{B}_c$ is spectrally equivalent to the matrix B_c with

$$q_1 \;=\; 1 \qquad \text{and} \qquad q_2 \;=\; 1 \;+\; \max_{1\le s\le t} \frac{h_f \sum\limits_{j=1}^{l_s} |\Gamma_{s,j}|}{|G_{h,s}|}. \tag{62}$$

Due to the regularity and quasiuniformity assumptions about the mesh $\Omega_{xy,h}$ and the partitioning of $\Omega_{xy,h}$ into subdomains $G_{h,s}$, $s = \overline{1,\ t}$, the value of q_2 in (62) is bounded from above by a positive constant c_4 which is independent of the coefficients a_{xy}, a_z, and c in (1) as well as of the mesh Ω_h. Thus, the matrix $\widehat{H}$ in (56), (57) with $q_3 < 2/c_4$ is spectrally equivalent to the matrix $H = B^{-1}$.

Numerical results in Section 6 are given for the PCG method with the preconditioner (56), (57) defined in this Section.

Let us denote the matrix $\widehat{B}_c$ by A_c, i.e. $A_c = \widehat{B}_c$. The matrix A_c can be presented as the 2×2 block matrix similar to presentation (3) for the matrix A:

$$A_c = \begin{pmatrix} A_{c,p} & A_{c,p\lambda} \\ A_{c,\lambda p} & A_{c,\lambda} \end{pmatrix} \tag{63}$$

where $A_{c,\lambda}$ is a diagonal matrix. It can be easily shown that the Schur complement

$$K_c = A_{c,p} - A_{c,p\lambda} A_{c,\lambda}^{-1} A_{c,\lambda p} \tag{64}$$

of the matrix A_c has the same structure as the original matrix K in (2).

Remark 1. The size of the matrix K_c in (64) is at least 2.5 times smaller than the size of the matrix B_c in (48). To this end, the Cholesky factorization of the matrix K_c is at least fifteen times cheaper than the same factorization of the matrix B_c. Thus, it can be shown that in the case $h_c \sim \sqrt{h_f}$ the PCG-method with the preconditioner $\widehat{H}$ proposed in this Section is more efficient than with the preconditioner H proposed in Sections 3 and 4.

Remark 2. Due to the structure of the matrix K_c in (64), we can design a two-level preconditioner $H_{c,p}$ for this matrix using the same coarsening technique. Replacing the matrix K_c^{-1} in the definition of $\widehat{H}$ by the matrix $H_{c,p}$ we get a three-level preconditioner.

Remark 3. The number of iterations of the PCG method with the proposed preconditioner is $O(h_f^{-1} h_c |\ln\, h_f|)$. The factorization of the matrices B_c and $\widehat{B}_c$ defined in (48) and (58)-(62), respectively, requires $O(h_c^{-6} h_f^{-1})$ arithmetic operations. Then, the solution of algebraic systems with factorized matrices B_c and $\widehat{B}_c^{-1}$ requires $O(h_c^{-4} h_f^{-1})$ arithmetic operations. The PCG method is faster for smaller values of h_c but implementation algorithms are more expensive. A reasonable choice is $h_c = \sqrt{h_f}$. In this case, the factorization of the matrices B_c and $\widehat{B}_c$ requires $O(h_f^{-4})$ arithmetic operations, and the implementation of the PCG with the factorized matrices B_c and $\widehat{B}_c$ requires $O(h_f^{-7/2} |\ln\ h_f|)$ arithmetic operations.

6 Numerical Results

To demonstrate the performance of the proposed two-level preconditioner we consider two examples. For both examples we compare the number of iterations and the total CPU time of the PCG-method with two-level preconditioner (TLP) and with the block Jacobi preconditioner (BJP). Both preconditioners are applied to system (2). The block Jacobi preconditioner is defined by

$$B_J = D_{xy} + K_z \tag{65}$$

where D_{xy} is the diagonal of the matrix K_{xy} in (8).

In the first example, the cubic domain Ω is partitioned into eight equal subcubes Ω_l, $l = \overline{1,\ 8}$. The coefficients a_z and c are equal to one. The coefficient a_{xy} is equal to one in four subdomains. In the other four subdomains the value of the coefficient a_{xy} is shown in Table 1. The distribution of two different values of the coefficient a_{xy} is based on the 3D-chess ordering of the subdomains. The mesh Ω_h is cubic with the mesh step size $h = 10^{-2}$. The square domain Ω_{xy} is partitioned into 100 square subdomains $G_{h,s}$, $s = \overline{1,\ 100}$. The coarse mesh matrix K_c is a block tridiagonal matrix (100 blocks, each block is a 10×10 matrix). The stopping criterion is to reduce the K-norm of the original error vector in 10^6 times.

Table 1. Variable $a_{xy}^{(2)}$, cubic mesh

	TLP $h_c = \sqrt{h_f}$		z-line BJP		Speed up TLP vs. BJP
$a_{xy}^{(2)}$	#it	CPU	#it	CPU	
10	64	23.2	984	212.	9.3
100	62	22.7	2336	491.	22.0
1000	61	22.2	6793	1450.	66.5

In the second example, $\Omega_{xy,h}$ is a uniform hexagonal mesh, and the shape of Ω_{xy} depends on the mesh. The domain Ω is again partitioned into eight subdomains as shown in Figure 2. The coefficients a_z and c are equal to one. The coefficient a_{xy} is equal to one in four subdomains and is equal to 100 in four others. The distribution of two values for a_{xy} is done in the 3D-chess order similar to Example 1. The mesh $\Omega_{xy,h}$ is partitioned into t identical subdomains $G_{h,s}$, $s = \overline{1,\ t}$, where t is equal to 36, 64, 100, and 144. In Table 2, the number of iterations and CPU time for the proposed two-level preconditioner on a sequence of meshes are given versus to the block Jacobi preconditioner.

Table 2. Sequence of hexagonal meshes $\Omega_{xy,h}$

	TLP $h_c = \sqrt{h_f}$		z-line BJP		Speed up TLP vs. BJP
$1/h_f$	#it	CPU	#it	CPU	
36	44	0.62	839	8.42	13.5
64	56	7.33	1463	90.3	12.3
100	68	25.5	2346	545.	21.4
144	81	116.	3391	2466.	21.3

The numerical results confirm the theoretical statements in Section 3: the number of iterations does not depend on the values of the coefficients, and the condition number of the matrix H_pK is proportional to $\left(h_c/h_f\right)^2$.

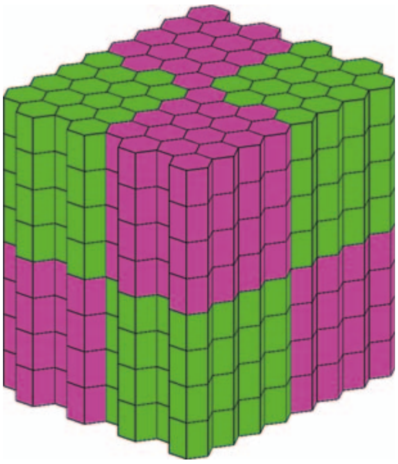


Fig. 2. Domain Ω with hexagonal mesh $\Omega_{xy,h}$

More detailed description of implementation algorithms as well as other results of numerical experiments can be found in [1].

Acknowledgement. This work was supported by ExxonMobil Upstream Research Company. The author is grateful to O. Boiarkine, I. Kapyrin, and N. Yavich for providing the numerical results for this paper. The author is also grateful to the referee for useful remarks.

References

[1] O. Boiarkine, I. Kapyrin, Yu. Kuznetsov, and N. Yavich. Numerical analysis of two-level preconditioners for diffusion equations with anisotropic coefficients. *Russian J. Numer. Anal. Math. Modelling*, 22(3), 2007. To appear.

[2] Yu. Kuznetsov and K. Lipnikov. An efficient iterative solver for a simplified poroelasticity problem. *East-West J. Numer. Math.*, 8(3):207–221, 2000.

[3] Yu. Kuznetsov and G. Marchuk. Iterative methods and quadratic functionals. In J.-L. Lions and G. Marchuk, editors, *Méthodes de l'Informatique–4*, pages 3–132, Paris, 1974. Dunod. In French.

[4] Yu. Kuznetsov and S. Repin. New mixed finite element method on polygonal and polyhedral meshes. *Russian J. Numer. Anal. Math. Modelling*, 18(3):261–278, 2003.

[5] Yu. A. Kuznetsov. Matrix iterative methods in subspaces. In *Proceedings of the International Congress of Mathematicians, Vol. 1, 2 (Warsaw, 1983)*, pages 1509–1521, Warsaw, 1984. PWN.

[6] Yu. A. Kuznetsov. Two-level preconditioners with projectors for unstructured grids. *Russian J. Numer. Anal. Math. Modelling*, 15(3-4):247–255, 2000.

Domain Decomposition of Constrained Optimal Control Problems for 2D Elliptic System on Networked Domains: Convergence and A Posteriori Error Estimates

Günter Leugering

Institut für Angewandte Mathematik, Lehrstuhl II, Universität Erlangen-Nürnberg
Martensstr.3 D-91058 Erlangen, Germany.
Guenter.Leugering@am.uni-erlangen.de

Summary. We consider optimal control problems for elliptic systems under control constraints on networked domains. In particular, we study such systems in a format that allows for applications in problems including membranes and Reissner-Mindlin plates on multi-link-domains, called networks. We first provide the models, derive first order optimality conditions in terms of variational equations and inequalities for a control-constrained linear-quadratic optimal control problem, and then introduce a non-overlapping iterative domain decomposition method, which is based on Robin-type interface updates at multiple joints (edges). We prove convergence of the iteration and derive a posteriori error estimates with respect to the iteration across the interfaces.

1 Introduction

Partial differential equations on networks or networked domains consisting of 1-d, 2-d and possibly 3-d sub-domains linked together at multiple joints, edges or faces, respectively, arise in many important applications, as in gas-, water-, traffic- or blood-flow in pipe-, channel-, road or artery networks, or in beam-plate structures, as well as in many micro-, meso- or macro-mechanical smart structures. The equations governing the processes on those multi-link domains are elliptic, parabolic and hyperbolic, dependent on the application. Problems on such networks are genuinely subject to sub-structuring by the way of non-overlapping domain decomposition methods. This remains true even for optimal control problems formulated for such partial differential equations on multi-link domains. While non-overlapping domain decompositions for unconstrained optimal control problems involving partial differential equations on such networks have been studied in depth in the monograph [5], such non-overlapping domain decompositions for problems with control constraints have not been discussed so far. This is the purpose of these notes. In order to keep matters simple but still provide some insight also in the modeling, we study elliptic systems only. The time-dependent case can also be handled, but is much more involved, see [5] for unconstrained problems. The ddm is in the spirit of [1]. See also [9] for a general reference, and [2] for vascular flow in artery networks, where Dirichlet-Neumann iterates are considered. See also [3] and [4] for ddm in the context of optimal control problems. Decomposition of optimality systems corresponding to state-constrained optimal control problems seem not to have been discussed so far. This is ongoing research of the author.

2 Elliptic Systems on 2-D Networks

As PDEs on networks are somewhat unusual, we take some effort in order to make the modeling and its scope more transparent. Unfortunately, this involves some notation. A *two-dimensional polygonal network* $\mathcal{P}$ in $\mathbb{R}^N$ is a finite union of nonempty subsets $\mathcal{P}_i$, $i \in \mathcal{I}$, such that

(i) each $\mathcal{P}_i$ is a simply connected open polygonal subset of a plane Π_i in $\mathbb{R}^N$;
(ii) $\bigcup_{i\in\mathcal{I}} \overline{\mathcal{P}}_i$ is connected;
(iii) for all $i, j \in \mathcal{I}$, $\overline{\mathcal{P}}_i \bigcap \overline{\mathcal{P}}_j$ is either empty, a common vertex, or a whole common side.

The reader is referred to [8], whose notation we adopt, for more details about such 2-d networks. For each $i \in \mathcal{I}$ we fix once and for all a system of coordinates in Π_i. We assume that the boundary $\partial\mathcal{P}_i$ of $\mathcal{P}_i$ is the union of a finite number of linear segments $\overline{\varGamma}_{ij}$, $j = 1, \ldots, N_i$. It is convenient to assume that $\varGamma_{ij}$ is open in $\partial\mathcal{P}_i$. The collection of all $\varGamma_{ij}$ are the *edges* of $\mathcal{P}$ and will be denoted by $\mathcal{E}$. An edge $\varGamma_{ij}$ corresponding to an $e \in \mathcal{E}$ will be denoted by $\varGamma_{ie}$ and the *index set* $\mathcal{I}_e$ of e is $\mathcal{I}_e = \{i | \, e = \varGamma_{ie}\}$. The *degree* of an edge is the cardinality of $\mathcal{I}_e$ and is denoted by $d(e)$. For each $i \in \mathcal{I}_e$ we will denote by ν_{ie} the unit outer normal to $\mathcal{P}_i$ along $\varGamma_{ie}$.

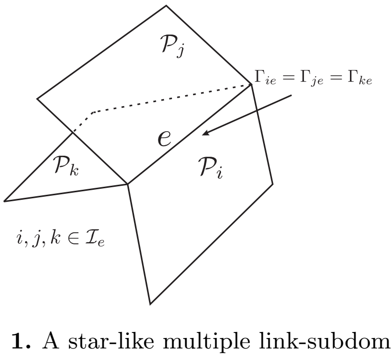

$\mathcal{P}_j$
$\Gamma_{ie} = \Gamma_{je} = \Gamma_{ke}$
e
$\mathcal{P}_k$
$\mathcal{P}_i$
$i, j, k \in \mathcal{I}_e$

Fig. 1. A star-like multiple link-subdomain

The coordinates of ν_{ie} in the given coordinate system of $\mathcal{P}_i$ are denoted by (ν_{ie}^1, ν_{ie}^2). We partition the edges of $\mathcal{E}$ into two disjoint subsets $\mathcal{D}$ and $\mathcal{N}$, corresponding respectively to edges along which Dirichlet conditions hold and along which Neumann or transmission conditions hold. The Dirichlet edges are assumed to be *exterior edges*, that is, edges for which $d(e) = 1$. The Neumann edges consist of exterior edges $\mathcal{N}^{\mathrm{ext}}$ and *interior edges* $\mathcal{N}^{\mathrm{int}} := \mathcal{N} \backslash \mathcal{N}^{\mathrm{ext}}$. Let $m \geq 1$ be a given integer. For a function $W : \mathcal{P} \mapsto \mathrm{I\!R}^m$, W_i will denote the restriction of W to $\mathcal{P}_i$, that is $W_i : \mathcal{P}_i \mapsto \mathrm{I\!R}^m : x \mapsto W(x)$. We introduce real $m \times m$ matrices $A_i^{\alpha\beta},\ B_i^\beta,\ C_i, \quad i \in \mathcal{I},\ \alpha, \beta = 1, 2$, where $A_i^{\alpha\beta} = (A_i^{\beta\alpha})^*, \quad C_i = C_i^*$ and where the * superscript denotes transpose. For sufficiently regular $W, \Phi : \mathcal{P} \mapsto \mathrm{I\!R}^m$ we define the symmetric bilinear form

$$a(W, \Phi) = \sum_{i \in \mathcal{I}} \int_{\mathcal{P}_i} [A_i^{\alpha\beta}(W_{i,\beta} + B_i^\beta W_i) \cdot (\Phi_{i,\alpha} + B_i^\alpha \Phi_i) + C_i W_i \cdot \Phi_i]\, dx, \quad (1)$$

where repeated lower case Greek indices are summed over 1,2. A subscript following a comma indicates differentiation with respect to the corresponding variable, e.g., $W_{i,\beta} = \partial W_i / \partial x_\beta$. The matrices $A_i^{\alpha\beta}$, B_i^β, C_i may depend on $(x_1, x_2) \in \mathcal{P}_i$ and $a(W, \Phi)$ is required to be $\mathcal{V}$-elliptic for an appropriate function space $\mathcal{V}$ specified below. We shall consider the variational problem

$$a(W, \Phi) = \langle F, \Phi \rangle_V, \quad \forall \Phi \in \mathcal{V},\ 0 < t < T, \quad (2)$$

where $\mathcal{V}$ is a certain space of test functions and F is a given, sufficiently regular function. The variational equation (2) obviously implies, in particular, that the W_i, $i \in \mathcal{I}$, formally satisfy the system of equations

$$\begin{aligned} -\frac{\partial}{\partial x_\alpha}[A_i^{\alpha\beta}(W_{i,\beta} + B_i^\beta W_i)] + (B_i^\alpha)^* A_i^{\alpha\beta}(W_{i,\beta} + B_i^\beta W_i) \\ + C_i W_i = F_i \ \text{ in } \mathcal{P}_i,\ i \in \mathcal{I}. \quad (3) \end{aligned}$$

To determine the space $\mathcal{V}$, we need to specify the conditions satisfied by W along the edges of $\mathcal{P}$. These conditions are of two types: *geometric edge conditions*, and *mechanical edge conditions*. As usual, the space $\mathcal{V}$ is then defined in terms of the geometric edge conditions. At a Dirichlet edge we set

$$W_i = 0 \ \text{ on } e \text{ when } \Gamma_{ie} \in \mathcal{D}. \quad (4)$$

Further, along each $e \in \mathcal{N}^{\mathrm{int}}$ we impose the condition

$$Q_{ie} W_i = Q_{je} W_j \ \text{ on } e \text{ when } \Gamma_{ie} = \Gamma_{je},\ e \in \mathcal{N}^{\mathrm{int}}, \quad (5)$$

where, for each $i \in \mathcal{I}_e$, Q_{ie} is a real, nontrivial $p_e \times m$ matrix of rank $p_e \leq m$ with p_e independent of $i \in \mathcal{I}_e$. If $p_e < m$ additional conditions *may* be imposed, such as

$$\Pi_{ie} W_i = 0 \ \text{ on } e,\ \forall\, i \in \mathcal{I}_e,\ e \in \mathcal{N}^{\mathrm{int}}, \quad (6)$$

where Π_{ie} is the orthogonal projection onto the kernel of Q_{ie}. (Note that (5) is a condition on only the components $\Pi_{ie}^{\perp}W_i$, $i \in \mathcal{I}_e$, where $\Pi_{ie}^{\perp}$ is the orthogonal projection onto the orthogonal complement in $\mathrm{I\!R}^m$ of the kernel of Q_{ie}.) For definiteness we always assume that (6) is imposed and leave to the reader the minor modifications that occur in the opposite case. Thus the geometric edge conditions are taken to be (4) - (6), and the space $\mathcal{V}$ of test functions consists of sufficiently regular functions $\Phi : \mathcal{P} \mapsto \mathrm{I\!R}^m$ that satisfy the geometric edge conditions. Formal integration by parts in (2) and taking proper variations shows that, in addition to (3), W_i must satisfy

$$\nu_{ie}^{\alpha} A_i^{\alpha\beta}(W_{i,\beta} + B_i^{\beta} W_i) = 0 \ \text{ on } e \text{ when } \Gamma_{ie} \in \mathcal{N}^{\text{ext}}. \tag{7}$$

For each $\Gamma_{ie} \in \mathcal{N}^{\text{int}}$ write $\Phi_i = \Pi_{ie}\Phi_i + \Pi_{ie}^{\perp}\Phi_i$, and let Q_{ie}^{+} denote the generalized inverse of Q_{ie}, that is Q_{ie}^{+} is a $m \times p_e$ matrix such that $Q_{ie}Q_{ie}^{+} = I_{p_e}$, $Q_{ie}^{+}Q_{ie} = \Pi_{ie}^{\perp}$. Then we deduce that

$$\sum_{i \in \mathcal{I}_e} (Q_{ie}^{+})^{*} \nu_{ie}^{\alpha} A_i^{\alpha\beta}(W_{i,\beta} + B_i^{\beta} W_i) = 0 \ \text{ on } e \text{ if } e \in \mathcal{N}^{\text{int}}. \tag{8}$$

Conditions (7) and (8) are called the *mechanical edge conditions*. We also refer to (5) and (8) as the geometric and mechanical *transmission conditions*, respectively. To summarize, the edge conditions are comprised of the geometric edge conditions (4) - (6), and the mechanical edge conditions (7), (8). The geometric transmission conditions are (5) and (6), while the mechanical transmission conditions are given by (8).

2.1 Examples

Example 1. (Scalar problems on networks) *Suppose that $m = 1$. In this case the matrices $A_i^{\alpha\beta}$, B_i^{α}, C_i, Q_{ie} reduce to scalars $a_i^{\alpha\beta}$, b_i^{α}, c_i, q_{ie}, where $a_i^{\alpha\beta} = a_i^{\beta\alpha}$. Set $A_i = \big(a_i^{\alpha\beta}\big), b_i = col\big(b_i^{\alpha}\big)$. The system (3) takes the form*

$$-\nabla \cdot (A_i \nabla W_i) + [-\nabla \cdot (A_i b_i) + b_i^{*} A_i b_i + c_i] W_i = F_i. \tag{9}$$

Suppose that all $q_{ie} = 1$. The geometric edge conditions (4), (5) are then W_i=0 on e when $e \in \mathcal{D}$, $W_i = W_j$ on e when $e \in \mathcal{N}^{int}$ while the mechanical edge conditions are

$$\sum_{i \in \mathcal{I}_e} [\nu_{ie} \cdot (A_i \nabla W_i) + (\nu_{ie} \cdot A_i b_i) W_i] = 0 \ \ \textit{on } e \textit{ when } e \in \mathcal{N}.$$

Example 2. (Membrane networks in $\mathrm{I\!R}^3$.) *In this case $m = N = 3$. For each $i \in \mathcal{I}$ set $\eta_{i3} = \eta_{i1} \wedge \eta_{i2}$, where η_{i1}, η_{i2} are the unit coordinate vectors in Π_i. Suppose that $B_i = C_i = 0$, $Q_{ie} = I_3$, where I_3 denotes the identity matrix with respect to the $\{\eta_{ik}\}_{k=1}^{3}$ basis. With respect to this basis the matrices $A_i^{\alpha\beta}$ are given by*

$$A_i^{11} = \begin{pmatrix} 2\mu_i + \lambda_i & 0 & 0 \\ 0 & \mu_i & 0 \\ 0 & 0 & \mu_i \end{pmatrix}, \quad A_i^{22} = \begin{pmatrix} \mu_i & 0 & 0 \\ 0 & 2\mu_i + \lambda_i & 0 \\ 0 & 0 & \mu_i \end{pmatrix},$$
$$A_i^{12} = \begin{pmatrix} 0 & \lambda_i & 0 \\ \mu_i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}, \quad A_i^{21} = \begin{pmatrix} 0 & \mu_i & 0 \\ \lambda_i & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Write $W_i = \sum_{k=1}^{3} W_{ik}\eta_{ik}$, $w_i = W_{i\alpha}\eta_{i\alpha}$, $\varepsilon_{\alpha\beta}(w_i) = \frac{1}{2}(W_{i\alpha,\beta} + W_{i\beta,\alpha})$, $\sigma_i^{\alpha\beta}(w_i) = 2\mu_i\varepsilon_{\alpha\beta}(w_i) + \lambda_i\varepsilon_{\gamma\gamma}(w_i)\delta^{\alpha\beta}$. *The bilinear form* (1) *may be written*

$$a(W, \Phi) = \sum_{i\in\mathcal{I}} \int_{\mathcal{P}_i} [\sigma_i^{\alpha\beta}(w_i)\varepsilon_{\alpha\beta}(\phi_i) + \mu_i W_{i3,\alpha}\Phi_{i3,\alpha}]\, dx$$

where $\Phi_i = \sum_{k=1}^{3} \Phi_{ik}\eta_{ik} := \phi_i + \Phi_{i3}\eta_{i3}$. *The geometric edge conditions* (4), (5) *are as above, but now in a vectorial sense. The mechanical edge conditions are obtained as usual.*

The corresponding system models the small, static deformation of a network of homogeneous isotropic membranes $\{\mathcal{P}_i\}_{i\in\mathcal{I}}$ *in* $I\!R^3$ *of uniform density one and Lamé parameters* λ_i *and* μ_i *under distributed loads* F_i, $i \in \mathcal{I}$; $W_i(x_1, x_2)$ *represents the displacement of the material particle situated at* $(x_1, x_2) \in \mathcal{P}_i$ *in the reference configuration. The reader is referred to [6] where this model is introduced and analyzed.*

Also networks of Reissner-Mindlin plates can be considered in this framework see [6, 5]. Networks of thin shells, such as Naghdi-shells or Cosserat-shells do not seem to have been considered in the literature. Such networks are subject to further current investigations.

2.2 Existence and Uniqueness of Solutions

In this section existence and uniqueness of solutions of the variational equation (2) are considered. It is assumed that the elements of the matrices $A_i^{\alpha\beta}$, B_i^{β}, C_i are all in $L^\infty(\mathcal{P}_i)$. For a function $\Phi : \mathcal{P} \mapsto \mathrm{I\!R}^m$ we denote by Φ_i the restriction of Φ to $\mathcal{P}_i$ and we set

$$\mathcal{H}^s(\mathcal{P}) = \{\Phi : \Phi_i \in \mathcal{H}^s(\mathcal{P}_i), \forall i \in \mathcal{I}\}$$
$$\|\Phi\|_{\mathcal{H}^s(\mathcal{P})} = \left(\sum_{i\in\mathcal{I}} \|\Phi_i\|^2_{\mathcal{H}^s(\mathcal{P}_i)} \right)^{1/2},$$

where $\mathcal{H}^s(\mathcal{P}_i)$ denotes the usual (vector) Sobolev space of order s on $\mathcal{P}_i$. Set $\mathcal{H} = \mathcal{H}^0(\mathcal{P})$ and define a closed subspace $\mathcal{V}$ of $\mathcal{H}^1(\mathcal{P})$ by

$$\begin{aligned}\mathcal{V} = \{\varPhi \in \mathcal{H}^1(\mathcal{P}) | \ \varPhi_i = 0 \text{ on } e \text{ when } \varGamma_{ie} \in \mathcal{D},\\ Q_{ie}\varPhi_i = Q_{je}\varPhi_j \text{ on } e \text{ when } \varGamma_{ie} = \varGamma_{je},\\ \varPi_{ie}\varPhi_i = 0 \text{ on } e \text{ when } e \in \mathcal{N}^{\text{int}},\ i \in \mathcal{I}\}.\end{aligned}$$

The space $\mathcal{V}$ is densely and compactly embedded in $\mathcal{H}$. It is assumed that $a(\varPhi, \varPhi)$ is elliptic on $\mathcal{V}$: there are constants $k \geq 0$, $K > 0$, such that

$$a(\varPhi, \varPhi) + k\|\varPhi\|^2_{\mathcal{H}} \geq K\|\varPhi\|^2_{\mathcal{H}^1(\mathcal{P})}, \quad \forall \varPhi \in \mathcal{V}. \tag{10}$$

Let $F = \{F_i\}_{i \in \mathcal{I}} \in \mathcal{H}$. It follows from standard variational theory and the Fredholm alternative that the variational equation (2) has a solution if and only if F is orthogonal in $\mathcal{H}$ to all solutions $W \in \mathcal{V}$ of $a(W, \varPhi) = 0$, $\forall \varPhi \in \mathcal{V}$. If it is known that $a(\varPhi, \varPhi) \geq 0$ for each $\varPhi \in \mathcal{V}$, the last equation has only the trivial solution if, and only if, (10) holds with $k = 0$.

3 The Optimal Control Problem

We consider the following optimal control problem.

$$\begin{cases} \min\limits_{f \in \mathcal{U}_{ad}} \frac{1}{2} \int\limits_{\mathcal{P}} \|W - W_d\|^2 dx + \sum\limits_{i \in \mathcal{I}} \sum\limits_{e \in \mathcal{N}_i^{ext}} \frac{\kappa}{2} \int\limits_{\varGamma_{ie}} \|f_{ie}\|^2 d\varGamma, \quad \text{subject to} \\ -\frac{\partial}{\partial x_\alpha}\left[A_i^{\alpha\beta}\big(W_{i,\beta} + B_i^\beta W_i\big)\right] + (B_i^\alpha)^* A_i^{\alpha\beta}\big(W_{i,\beta} + B_i^\beta W_i\big) \\ \qquad +C_i W_i = F_i \ \text{ in } \mathcal{P}_i, \\ W_i = 0 \ \text{ on } \varGamma_{ie} \text{ when } e \in \mathcal{D} \\ \nu_{ie}^\alpha A_i^{\alpha\beta}(W_{i,\beta} + B_i^\beta W_i) + \alpha_{ie} W_i = f_{ie} \ \text{ on } \varGamma_{ie} \text{ when } e \in \mathcal{N}^{\text{ext}} \\ \varPi_{ie} W_i = 0 \ \text{ on } \varSigma_{ie}, \forall i \in \mathcal{I}_e,\ e \in \mathcal{N}^{\text{int}} \\ Q_{ie} W_i = Q_{je} W_j \ \text{ on } \varGamma_{ie} \text{ when } \varGamma_{ie} = \varGamma_{je},\ e \in \mathcal{N}^{\text{int}} \\ \sum\limits_{i \in \mathcal{I}_e} (Q_{ie}^+)^* \nu_{ie}^\alpha A_i^{\alpha\beta}(W_{i,\beta} + B_i^\beta W_i) = 0 \ \text{ on } \varGamma_{ie} \text{ if } e \in \mathcal{N}^{\text{int}}. \end{cases} \tag{11}$$

where

$$\mathcal{U} = \prod_{i \in \mathcal{I}} \prod_{e \in \mathcal{N}_i^{\text{ext}}} L^2(\varGamma_{ie}) \tag{12}$$

and

$$\mathcal{U}_{ad} = \left\{ f \in \mathcal{U} : f_{ie} \in U_{ie},\ i \in \mathcal{I},\ e \in \mathcal{N}_i^{\text{ext}} \right\}, \tag{13}$$

where, in turn, the sets U_{ie} are all convex. Also notice that we added extra freedom on the external boundary, in order to allow for Robin-conditions. Instead of the strong form of this linear quadratic control constrained optimal control, we consider the weak formulation.

$$\begin{cases} \min\limits_{f \in \mathcal{U}_{ad}} \frac{1}{2} \int\limits_{\mathcal{P}} \|W - W_d\|^2 dx + \sum\limits_{i \in \mathcal{I}} \sum\limits_{e \in \mathcal{N}_i^{ext}} \frac{\kappa}{2} \int\limits_{\varGamma_{ie}} \|f_{ie}\|^2 d\varGamma, \quad \text{subject to} \\ a(W, \varPhi) + b\big(W, \varPhi\big) \\ \quad = \big(F, \varPhi\big)_{\mathcal{H}} + \sum\limits_{i \in \mathcal{I}} \sum\limits_{e \in \mathcal{N}_i^{\text{ext}}} \int_{\varGamma_{ie}} f_{ie} \cdot \varPhi_i \, d\varGamma, \quad \forall \varPhi \in \mathcal{V},\ 0 < t < T, \end{cases} \tag{14}$$

where

$$b(W,\Phi)=\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{N}_i^{\text{ext}}}\int_{\Gamma_{ie}}\alpha_{ie}W_i\cdot\Phi_i\,d\Gamma. \tag{15}$$

We may simplify the notation even further by introducing the inner product on the control space $\mathcal{U}$:

$$\langle f,\Phi\rangle_{\mathcal{U}}=\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{N}_i^{\text{ext}}}\int_{\Gamma_{ie}}f_{ie}\cdot\Phi_i d\Gamma$$

$$\begin{cases}\min\limits_{f\in\mathcal{U}_{ad}}\frac{1}{2}\|W-W_d\|^2+\frac{\kappa}{2}\|f\|^2_{\mathcal{U}}, \quad \text{subject to}\\ a(W,\Phi)+b\big(W,\Phi\big)=\langle f,\Phi\rangle+(F,\Phi)_{\mathcal{H}}, \quad \forall\Phi\in\mathcal{V}.\end{cases} \tag{16}$$

Existence, uniqueness of optimal controls and the validity of the following first order optimality condition follow by standard arguments.

$$\begin{cases}a(W,\Phi)+b(W,\Phi)=\langle f,\Phi\rangle+(F,\Phi)_{\mathcal{H}} \quad \forall\Phi\in\mathcal{V}\\ a(P,\Psi)+b(P,\Psi)=(W-W_d,\Psi), \quad \forall\Psi\in\mathcal{V}\\ \langle P+\kappa f,v-f\rangle\geq 0, \quad \forall v\in\mathcal{U}_{ad}\end{cases} \tag{17}$$

4 Domain Decomposition

Some preliminary material is required in order to properly formulate the subsystems in the decomposition. Let $\mathcal{H}_i$, $\mathcal{V}_i$, and $a_i(W_i,V_i)$ be the spaces associated with the bilinear form

$$a_i(W_i,\Phi_i)=\int_{\mathcal{P}_i}[A_i^{\alpha\beta}(W_{i,\beta}+B_i^{\beta}W_i)\cdot(\Phi_{i,\alpha}+B_i^{\alpha}\Phi_i)+C_iW_i\cdot\Phi_i]\,dx, \tag{18}$$

It is assumed that $a_i(\Phi_i,\Phi_i)\geq K_i\|\Phi_i\|^2_{\mathcal{H}^1(\mathcal{P}_i)},\forall\Phi_i\in\mathcal{V}_i$, for some constant $K_i>0$. We may then define a norm on $\mathcal{V}_i$ equivalent to the induced $\mathcal{H}^1(\mathcal{P}_i)$ by setting $\|\Phi_i\|_{\mathcal{V}_i}=\sqrt{a_i(\Phi_i,\Phi_i)}$. We identify the dual of $\mathcal{H}_i$ with $\mathcal{H}_i$ and denote by $\mathcal{V}_i^*$ the dual space of $\mathcal{V}_i$ with respect to $\mathcal{H}_i$. We define the continuous bilinear functional b_i on $\mathcal{V}_i$ by

$$b_i(W_i,\Phi_i)=\sum_{e\in\mathcal{N}_i^{\text{ext}}}\int_{\Gamma_{ie}}\alpha_{ie}W_i\cdot\Phi_i\,d\Gamma+\sum_{e\in\mathcal{N}_i^{\text{int}}}\int_{\Gamma_{ie}}\beta_eQ_{ie}W_i\cdot Q_{ie}\Phi_i\,d\Gamma. \tag{19}$$

where β_e is a positive constant independent of $i\in\mathcal{I}_e$. For each $i\in\mathcal{I}$ we consider the following local problems for functions W_i defined on $\mathcal{P}_i$:

$$\begin{aligned}&a_i(W_i,\Phi_i)+b_i(W_i,\Phi_i)\\&\quad=\big(F_i,\Phi_i\big)_{\mathcal{H}_i}+\sum_{e\in\mathcal{N}_i^{\text{ext}}}\int_{\Gamma_{ie}}f_{ie}\cdot\Phi_i\,d\Gamma+\sum_{e\in\mathcal{N}_i^{\text{int}}}\int_{\Gamma_{ie}}(g_{ie}+\lambda_{ie}^n)\cdot Q_{ie}\Phi_i\,d\Gamma,\\&\hspace{22em}\forall\Phi\in\mathcal{V}_i,\end{aligned} \tag{20}$$

where the inter-facial input λ_{ie}^n is to be specified below. We are going to consider the following local control-constrained optimal control problem.

$$\begin{cases} \min\limits_{f_i,g_i} J(f_i,g_i) := \frac{1}{2}\|W_i - W_{di}\|^2 + \frac{\kappa}{2} \sum\limits_{e\in\mathcal{N}_i^{\text{ext}}} \int\limits_{\Gamma_{ie}} |f_{ie}|^2 d\Gamma \\ \qquad + \sum\limits_{e\in\mathcal{N}_i^{\text{int}}} \frac{1}{2\gamma_e} \int\limits_{\Gamma_{ie}} |g_{ie}|^2 + |\gamma_e Q_{ie} W_i + \mu_{ie}^n|^2 d\Gamma \\ \qquad\qquad \text{subject to (20)},\ f_{ie} \in \mathcal{U}_{ad,i}, \end{cases} \tag{21}$$

where $g_{ie} \in L^2(\Gamma_{ie})$ serve as *virtual controls*. By standard arguments, we obtain the local optimality system:

$$\begin{cases} a_i(W_i,\Phi_i) + b_i(W_i,\Phi_i) = \big(F_i,\Phi_i\big)_{\mathcal{H}_i} + \sum\limits_{e\in\mathcal{N}_i^{\text{ext}}} \int\limits_{\Gamma_{ie}} f_{ie}\cdot\Phi_i\, d\Gamma \\ \qquad + \sum\limits_{e\in\mathcal{N}_i^{\text{int}}} \int\limits_{\Gamma_{ie}} (\lambda_{ie}^n - \gamma_e Q_{ie} P_i)\cdot Q_{ie}\Phi_i\, d\Gamma, \quad \forall \Phi_i \in \mathcal{V}_i, \\ a_i(P_i,\Phi_i) + b_i(P_i,\Phi_i) = (W_i - W_{d,i},\Phi) \\ \qquad + \sum\limits_{e\in\mathcal{N}_i^{\text{int}}} \int_{\Gamma_{ie}} (\mu_{ie}^n + \gamma_e Q_{ie} W_i)\cdot Q_{ie}\Phi_i\, d\Gamma, \quad \forall \Phi \in \mathcal{V}_i, \\ \quad \sum\limits_{e\in\mathcal{N}_i^{\text{ext}}} \int\limits_{\Gamma_{ie}} (\kappa f_{ie} + P_i)\cdot(\hat{f}_i - f_{ie}) d\Gamma \geq 0, \quad \forall f_i \in \mathcal{U}_{ad}\,. \end{cases} \tag{22}$$

We proceed to define update rules for $\lambda_{ie}^n, \mu_{ie}^n$ at the interfaces. To simplify the presentation, we introduce a 'scattering'-type mapping S_e for a given interior joint e:

$$S_e(u)_i := \frac{2}{d_e}\sum_{j\in\mathcal{I}_e} u_j - u_i, \quad i \in \mathcal{I}_e\,.$$

We obviously have $S_e^2 = Id$. We set

$$\begin{cases} \lambda_{ie}^{n+1} = S_e(2\beta_e Q_{\cdot e} W_\cdot^n + 2\gamma_e Q_{\cdot e} P_\cdot^n)_i - S_e(\lambda_\cdot^n)_i, \quad i \in \mathcal{I}_e\,, \\ \mu_{ie}^{n+1} = S_e(2\beta_e Q_{\cdot e} P_\cdot^n - 2\gamma_e Q_{\cdot e} W_\cdot^n)_i - S_e(\mu_\cdot^n)_i, \quad i \in \mathcal{I}_e\,. \end{cases} \tag{23}$$

If we assume convergence of the sequences $\lambda_{ie}^n, \mu_{ie}^n, W_i^n, P_i^n$, and if we use the properties of S_e in summing up the equations in (23) we obtain

$$\sum_i a_i(W_i,\Phi_i) = \sum_i \big(F_i,\Phi_i\big)_{\mathcal{H}_i} + \sum_i \sum_{e\in\mathcal{N}_i^{\text{ext}}} \int_{\Gamma_{ie}} f_{ie}\cdot\Phi_i\, d\Gamma, \tag{24}$$

for all $\Phi = \Phi|_{\mathcal{P}_i} \in \mathcal{V}$, and similarly for P_i. Thus the limiting elements $W_i, P_i,\ i \in \mathcal{I}$ satisfy the global optimality system. Therefore, by the domain decomposition method above we have decoupled the global optimality system into local optimality systems, which are the necessary condition for local optimal control problems. In other words, we decouple the globally defined optimal control problem into local ones of similar structure.

5 Convergence

We introduce the errors

$$\begin{cases} \widetilde{W}_i^n &= W_i^n - W_i|_{\mathcal{P}_i} \\ \widetilde{P}_i^n &= P_i^n - P_i|_{\mathcal{P}_i} \\ \tilde{f}_{ie}^n &= f_{ie}^n - f_{ie}, \end{cases} \tag{25}$$

where W_i^n, P_i^n and W_i, P_i solve the iterated system and the global one, respectively. By linearity, $\widetilde{W}_i^n, \widetilde{P}_i^n$ solve the systems

$$\begin{cases} a_i(\widetilde{W}_i^n, \Phi_i) + b_i(\widetilde{W}_i^n, \Phi_i) = \sum\limits_{e \in \mathcal{N}_i^{\text{ext}}} \int_{\Gamma_{ie}} \tilde{f}_{ie}^n \cdot \Phi_i \, d\Gamma \\ \qquad + \sum\limits_{e \in \mathcal{N}_i^{\text{int}}} \int_{\Gamma_{ie}} (\tilde{\lambda}_{ie}^n - \gamma_e Q_{ie} \widetilde{P}_i^n) \cdot Q_{ie} \Phi_i \, d\Gamma, \forall \Phi \in \mathcal{V}_i, \\ a_i(\widetilde{P}_i^n, \Phi_i) + b_i(\widetilde{P}_i^n, \Phi_i) = (\widetilde{W}_i, \Phi_i) \\ \qquad + \sum\limits_{e \in \mathcal{N}_i^{\text{int}}} \int_{\Gamma_{ie}} (\tilde{\mu_{ie}}^n + \gamma_e Q_{ie} \widetilde{W}_i^n) \cdot Q_{ie} \Phi_i \, d\Gamma, \forall \Phi \in \mathcal{V}_i, \end{cases} \tag{26}$$

and the variational inequality

$$\sum_{e \in \mathcal{N}_i^{\text{ext}}} \int_{\Gamma_{ie}} (\kappa f_{ie}^n + P_i^n) \cdot (\hat{f}_{ie} - f_{ie}^n) d\Gamma \geq 0, \quad \forall \hat{f}_{ie} \in \mathcal{U}_{ad,i}, \tag{27}$$

and a similar one for f_i. Upon choosing proper functions $\hat{f}_{ie}$ we obtain the inequality

$$\sum_{e \in \mathcal{N}_i^{\text{ext}}} \int_{\Gamma_{ie}} \widetilde{P}_i^n \cdot \tilde{f}_{ie}^n d\Gamma \leq -\kappa \sum_{e \in \mathcal{N}_i^{\text{ext}}} \int_{\Gamma_{ie}} |\tilde{f}_{ie}^n|^2 d\Gamma \, . \tag{28}$$

We now introduce the space $\mathcal{X}$ at the interfaces

$$\mathcal{X} = \prod_{i \in \mathcal{I}} \prod_{e \in \mathcal{N}_i^{\text{int}}} L^2(\Gamma_{ie}), \; X = (\lambda_{ie}, \mu_{ie}), \; i \in \mathcal{I}, e \in \mathcal{N}_i^{\text{int}}$$

together with the norm

$$\|X\|_{\mathcal{X}}^2 = \sum_{i \in \mathcal{I}} \sum_{e \in \mathcal{N}_i^{\text{int}}} \frac{1}{2\gamma_e} \int_{\Gamma_{ie}} |\tilde{\lambda}_{ie}|^2 + |\tilde{\mu}_{ie}|^2 d\Gamma \, .$$

The iteration map is now defined in the space $\mathcal{X}$ as follows:

$$\left\{ \begin{array}{cc} \mathcal{T}: & \mathcal{X} \to \mathcal{X} \\ \mathcal{T}X: & \{(S_e(2\beta_e Q_{\cdot e} W_{\cdot} + 2\gamma_e Q_{\cdot e} P_{\cdot})_i - S_e(\lambda_{\cdot e})_i); \\ & S_e((2\beta_e Q_{\cdot e} P_{\cdot} - 2\gamma_e Q_{\cdot e} W_{\cdot})_i - S_e(\mu_{\cdot e})_i) \end{array} \right\} . \tag{29}$$

We now consider the errors $\widetilde{X}^n$ and the norms of the iterates. Indeed, for the sake of simplicity, we assume that $\gamma_e, \beta_e, \alpha_e$ are independent of e. After considerable calculus, we arrive at

$$\begin{aligned}\|\mathcal{T}(\tilde{X})^n\|^2_{\mathcal{X}} = \|\tilde{X}^n\|^2_{\mathcal{X}} \quad & -\frac{2}{\gamma}\beta\sum_i\Big[a_i(\widetilde{P}_i^n,\widetilde{P}_i^n)+a_i(\widetilde{W}_i^n,\widetilde{W}_i^n)\Big]\\ & -\frac{2}{\gamma}\sum_{e\in\mathcal{N}_i^{\mathrm{ext}}}\int_{\Gamma_{ie}}|\widetilde{W}_i^n|^2+|\widetilde{P}_i^n|^2d\Gamma \\ & +\frac{2}{\gamma}\sum_{e\in\mathcal{N}_i^{\mathrm{ext}}}\int_{\Gamma_{ie}}\Big[\beta\tilde{f}_{ie}^n\cdot\widetilde{W}_i^n+\gamma\tilde{f}_{ie}^n\cdot\widetilde{P}_i^n\Big]d\Gamma\\ & +\frac{2}{\gamma}\beta\sum_i(\widetilde{W}_i^n,\widetilde{P}_i^n)-2(\widetilde{W}_i^n,\widetilde{W}_i^n)\,.\end{aligned}\tag{30}$$

We distinguish two cases: $\beta = 0$ and $\beta > 0$. In the first case we obtain using (28) the inequality

$$\|\mathcal{T}(\tilde{X})_i^n\|^2_{\mathcal{X}} \le \|\tilde{X}\|^2_{\mathcal{X}} - 2\kappa\sum_{e\in\mathcal{N}_i^{\mathrm{ext}}}\int_{\Gamma_{ie}}|\tilde{f}_{ie}|^2d\Gamma - 2\sum_i\|\widetilde{W}_i^n\|^2_{\mathcal{H}_i}\,. \tag{31}$$

Iterating (31) to zero we obtain the following result.

Theorem 1. *Let the parameters in* (30)*be independent of* e, *and let in particular* $\beta_e = \beta = 0$, $\forall e \in \mathcal{N}^{int}$ *and* $\alpha_e = \alpha = 0$, $\forall e \in \mathcal{N}^{ext}$ *then*

$$\begin{cases} i.) & \{\tilde{X}^n\}_n & \textit{is bounded}\\ ii.) & \widetilde{W}_i^n\to 0 & \textit{strongly in } L^2(\mathcal{P}_i)\\ iii.) & \tilde{f}_{ie}^n\to 0 & \textit{strongly in } L^2(\Gamma_{ie})\,.\end{cases}\tag{32}$$

While this result can be refined by exploiting the first statement further, it gives convergence in the L^2-sense, only. In order to obtain convergence in stronger norms also for the adjoint variable, we need to take positive Robin-boundary- and interface parameters α, β into account. We thus estimate (30) in that situation as follows.

$$\begin{aligned}\|\mathcal{T}\tilde{X}^n\|^2_{\mathcal{X}} \le \|\tilde{X}^n\|^2_{\mathcal{X}} & \\ -\frac{2\beta}{\gamma}\sum_i\Big\{ & a_i(\widetilde{P}_i^n,\widetilde{P}_i^n)+a(\widetilde{W}_i^n,\widetilde{W}_i^n)\\ & +(\frac{\gamma}{\beta}-\frac{1}{2\epsilon})\|\widetilde{W}_i^n\|^2-\frac{\epsilon}{2}\|\widetilde{P}_i^n\|^2\Big\}\end{aligned}\tag{33}$$

$$
-\frac{b}{\gamma}\sum_{e\in\mathcal{N}_i^{\rm ext}}(2\alpha-\frac{1}{\epsilon})\int\limits_{\Gamma_{ie}}|\widetilde{W}_i^n|^2d\Gamma-\frac{2\alpha\beta}{\gamma}\sum_{e\in\mathcal{N}_i^{\rm ext}}|\widetilde{P}_i^n|^2d\Gamma
$$
$$
-\frac{\beta}{\gamma}\sum_{e\in\mathcal{N}_i^{\rm ext}}(\frac{2\kappa\gamma}{\beta}-\epsilon)\int\limits_{e\in\mathcal{N}_i^{\rm ext}}|\tilde{f}_{ie}^n|^2d\Gamma\,.
$$

Theorem 2. *Let the parameters $\alpha_e=\alpha,\beta_e=\beta,\gamma_e=\gamma$ with $\alpha,\beta,\gamma>0$ such that $\frac{\gamma}{\beta}$ is sufficiently large. Then the iterates in* (33) *satisfy*

$$
\begin{cases}
i.)\quad a_i(\widetilde{P}_i^n,\widetilde{P}_i^n)\to 0\quad\forall i\\
ii.)\ \ a_i(\widetilde{W}_i^n,\widetilde{W}_i^n)\to 0\quad\forall i\\
iii.)\ \widetilde{P}_i^n|_{\Gamma_{ie}}\to 0\qquad in\ L^2(\Gamma_{ie}),\ i\in\mathcal{N}^{ext}\\
iv.)\ \ \tilde{f}_{ie}^n\to 0\qquad in\ L^2(\Gamma_{ie}),\ i\in\mathcal{N}^{ext}\,.
\end{cases}
\tag{34}
$$

6 A Posteriori Error Estimates

We are going to derive a posteriori error estimates with respect to the domain iteration, similar to those developed in [5] for unconstrained problems and single domains, as well as for time-dependent problems and time-and-space domain decompositions. The a posteriori error estimates derived in this section refer to the transmission conditions across multiple joints, only. A posteriori error estimates for problems without control and serial in-plane interfaces have first been described by [7]. To keep matters simple, we consider $\alpha_e=\beta_e=0$ and $\gamma_e=\gamma$. We consider the following error measure

$$
\sum_i\Big\{a_i(\widetilde{W}_i^{n+1},v_i)+a_i(\widetilde{W}_i^n,y_i)+a_i(\widetilde{P}_i^{n+1},u_i)+a_i(\widetilde{P}_i^n,z_i)\Big\}=
$$
$$
\sum_{e\in\mathcal{N}^{\rm ext}}\sum_{i\in\mathcal{I}_e}\int\limits_{\Gamma_{ie}}[\tilde{f}_{ie}^{n+1}\cdot v_id+\tilde{f}_i^n\cdot y_id]d\Gamma-\sum_i[(\widetilde{W}_i^{n+1},u_i)+(\widetilde{W}_i^n,z_i)]
$$
$$
+\sum_{e\in\mathcal{N}^{\rm int}}\sum_{i\in\mathcal{I}_e}\int\limits_{\Gamma_{ie}}\Big[\gamma Q_{ie}\widetilde{P}_i^n-\tilde{\lambda}_{ie}^n\Big]\cdot[S_e(Q_{\cdot e}v_\cdot)_i-Q_{ie}y_i]\,d\Gamma
$$
$$
+\sum_{e\in\mathcal{N}^{\rm int}}\sum_{i\in\mathcal{I}_e}\int\limits_{\Gamma_{ie}}\Big[\gamma Q_{ie}\widetilde{W}_i^n+\tilde{\mu}_{ie}^n\Big]\cdot[Q_{ie}z_i-S_e(Q_{\cdot e}u_\cdot)_i]\,d\Gamma
$$
$$
+\sum_{e\in\mathcal{N}^{\rm int}}\sum_{i\in\mathcal{I}_e}\int\limits_{\Gamma_{ie}}\Big[S_e(\gamma Q_{\cdot e}\widetilde{P}_\cdot^n)_i-\gamma Q_{\cdot e}\widetilde{P}_i^{n+1})Q_{ie}v_i\Big]\,d\Gamma
$$
$$
+\sum_{e\in\mathcal{N}^{\rm int}}\sum_{i\in\mathcal{I}_e}\int\limits_{\Gamma_{ie}}\Big[\gamma Q_{\cdot e}\widetilde{W}_i^{n+1}-S_e(\gamma Q_{\cdot e}\widetilde{W}_\cdot^n)_i\Big]\cdot Q_{ie}u_id\Gamma\,.
$$

We first choose $v_i=\widetilde{W}_i^{n+1},y_i=\widetilde{W}_i^n,u_i=\widetilde{P}_i^{n+1},z_i=\widetilde{P}_i^n$ and then $v_i=\widetilde{P}_i^{n+1},y_i=\widetilde{P}_i^n,u_i=-\widetilde{W}_i^{n+1},z_i=-\widetilde{W}_i^n$. Then, after substantial calculations and estimations we obtain the following error estimate, where the details may be found in a forthcoming publication.

Theorem 3. *Let $\beta_e = \alpha_e = 0$ for all e. There exists a positive number $C(\kappa, \gamma, \Omega)$ such that the total error satisfies the a posteriori error estimate*

$$\begin{aligned}\sum_i \Big\{ \|\widetilde{W}_i^{n+1}\|_{\mathcal{V}_i} + \|\widetilde{W}_i^{n}\|_{\mathcal{V}_i} + \|\widetilde{P}_i^{n+1}\|_{\mathcal{V}_i} + \|\widetilde{P}_i^{n}\|_{\mathcal{V}_i} \Big\} \\ \leq C(\kappa, \gamma, \Omega) \sum_{e \in \mathcal{N}_i^{int}} \sum_{i \in \mathcal{I}_e} \big\{ \|S_e(Q_{\cdot e} W_i^{n+1})_i - Q_{ie} W_i^n\|_{L^2(\Gamma_{ie})} \\ + \|S_e(Q_{\cdot e} P_i^{n+1})_i - Q_{ie} P_i^n\|_{L^2(\Gamma_{ie})} \big\} \,.\end{aligned}$$

References

[1] J.-D. Benamou. A domain decomposition method with coupled transmission conditions for the optimal control of systems governed by elliptic partial differential equations. *SIAM J. Numer. Anal.*, 33(6):2401–2416, 1996.

[2] L. Fatone, P. Gervasio, and A. Quarteroni. Numerical solution of vascular flow by heterogenous domain decomposition. In T. Chan et.al., editor, *12th International Conference on Domain Decomposition Methods*, pages 297–303. DDM.ORG, 2001.

[3] M. Heinkenschloss. A time-domain decomposition iterative method for the solution of distributed linear quadratic optimal control problems. *J. Comput. Appl. Math.*, 173/1:169–198, 2005.

[4] M. Heinkenschloss and H. Nguyen. Balancing Neumann-Neumann methods for elliptic optimal control problems. In R. Kornhuber et al., editor, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 589–596. Springer-Verlag, Berlin, Germany, 2005.

[5] J. E. Lagnese and G. Leugering. *Domain Decomposition Methods in Optimal Control of Partial Differential Equations.* Number 148 in ISNM. International Series of Numerical Mathematics. Birkhaeuser, Basel, 2004.

[6] J. E. Lagnese, G. Leugering, and E. J. P. G. Schmidt. *Modeling, Analysis and Control of Dynamic Elastic Multi-Link Structures.* Systems and Control: Foundations and Applications. Birkhäuser, Boston, 1994.

[7] G. Lube, L. Müller, and F. C. Otto. A nonoverlapping domain decomposition method for stabilized finite element approximations of the Oseen equations. *J. Comput. Appl. Math.*, 132(2):211–236, 2001.

[8] S. Nicaise. *Polygonal Interface Problems*, volume 39 of *Methoden und Verfahren der Mathematischen Physik.* Peter Lang, Frankfurt, 1993.

[9] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford University Press, 1999.

Challenges and Applications of Boundary Element Domain Decomposition Methods

Olaf Steinbach

Institute of Computational Mathematics, Graz University of Technology,
Steyrergasse 30, 8010 Graz, Austria. o.steinbach@tugraz.at

Summary. Boundary integral equation methods are well suited to represent the Dirichlet to Neumann maps which are required in the formulation of domain decomposition methods. Based on the symmetric representation of the local Steklov–Poincaré operators by a symmetric Galerkin boundary element method, we describe a stabilized variational formulation for the local Dirichlet to Neumann map. By a strong coupling of the Neumann data across the interfaces, we obtain a mixed variational formulation. For biorthogonal basis functions the resulting system is equivalent to nonredundant finite and boundary element tearing and interconnecting methods. We will also address several open questions, ideas and challenging tasks in the numerical analysis of boundary element domain decomposition methods, in the implementation of those algorithms, and their applications.

1 Introduction

Boundary element methods are well established approximation methods to solve exterior boundary value problems, or to solve partial differential equations with (piecewise) constant coefficients considered in complicated substructures and in domains with moving boundaries. For a state of the art overview on recent advances on mathematical aspects and engineering applications of boundary integral equation methods, see, for example, [15]. For more general partial differential equations, e.g. with nonlinear coefficients, the coupling of finite and boundary element methods seems to be an efficient tool to solve complex problems in complicated domains. For the formulation and for an efficient solution of the resulting systems of equations, domain decomposition methods are mandatory.

The classical approach to couple finite and boundary element methods is to use only the weakly singular boundary integral equation with single and double layer potentials, see, e.g., [1, 7], and [20]. In [3] a symmetric coupling of finite and boundary elements using the so–called hypersingular boundary integral operator was introduced. This approach was then extended to symmetric Galerkin boundary element methods, see, e.g., [5]. Appropriate preconditioned iterative strategies were later considered in [2], while quite general preconditioners based on operators of the opposite order were introduced in [18]. Boundary element tearing and interconnecting (BETI) methods were described in [10] as counterpart of FETI methods while in [9] these methods were combined with a fast multipole approximation of the local boundary integral operators involved. For an alternative approach to boundary integral domain decomposition methods see also [8].

Here we will give a quite general setting of tearing and interconnecting, or more general, hybrid domain decomposition methods. The local partial differential equation is rewritten as a local Dirichlet to Neumann map which can be realized either by domain variational formulations or by using boundary integral formulations. Since the related function spaces are fractional Sobolev spaces, one may ask for the right definition of the associated norms. It turns out that the used norms which are induced by the local single layer potential or its inverse allows for almost explicit spectral equivalence inequalities, and an appropriate stabilization of the singular Steklov–Poincaré operators. The modified Dirichlet to Neumann map is then used to obtain a mixed variational formulation allowing a weak coupling of the local Dirichlet data. However, staying with a globally conforming method and using biorthogonal basis functions we end up with the standard tearing and interconnecting approach as in FETI and in BETI.

The aim of this paper is to sketch some ideas to obtain advanced formulations in boundary integral domain decomposition methods, to propose to use special norms in the numerical analysis, and to state some challenging tasks in the implementation of fast boundary element domain decomposition algorithms to solve challenging problems from engineering and industry.

2 Boundary Integral Equation DD Methods

As a model problem we consider the Dirichlet boundary value problem of the potential equation,

$$-\mathrm{div}[\alpha(x)\nabla u(x)] \, = \, f(x) \quad \text{for } x \in \Omega, \quad u(x) = g(x) \quad \text{for } x \in \Gamma \qquad (1)$$

where $\Omega \subset \mathbb{R}^3$ is a bounded domain with Lipschitz boundary $\Gamma = \partial\Omega$. We assume that there is given a non–overlapping domain decomposition

$$\overline{\Omega} = \bigcup_{i=1}^{p} \overline{\Omega}_i, \quad \Omega_i \cap \Omega_j = \emptyset \quad \text{for } i \neq j, \quad \Gamma_i = \partial\Omega_i. \qquad (2)$$

The domain decomposition as considered in (2) may arise from a piecewise constant coefficient function $\alpha(x)$ due to the physical model, in particular we may assume $\alpha(x) = \alpha_i$ for $x \in \Omega_i$. However, to construct efficient solution strategies in parallel, one may also introduce a domain decomposition (2) when considering the Laplace or Poisson equation in a complicated three–dimensional structure. A challenging task is to find a domain decomposition (2) which is based on boundary informations only, i.e., without any additional volume meshes. Using ideas as used in fast boundary element methods, i.e. by a bisection algorithm it is possible to decompose a given boundary mesh into two separate surface meshes. While this step seems to be simple, the delicate task is the definition of the new interface mesh which should take care of the given geometric situation, i.e. one should avoid the intersection of the new interface with the original boundary. We have already applied this algorithm to find a suitable domain decomposition of the Lake St. Wolfgang domain as shown in Figure 1.

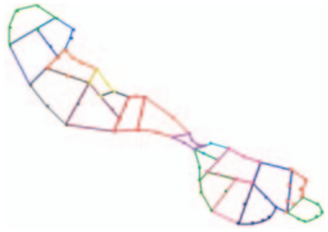


Fig. 1. Domain Decomposition of the Lake St. Wolfgang Domain.

It seems to be an open problem to find and to implement a robust algorithm for an automatic domain decomposition of complicated three–dimensional structures which is based on surface informations only. Such a tool is essentially needed when considering boundary element domain decomposition methods. Preliminary results on this topic will be published elsewhere [12]. A similar approach was already used in [6] to design an automatic domain decomposition approach for unstructured grids in three dimensions. There, the remeshing of the new interface is done within the splitting hyperplane without considering the robustness of the algorithm for complicated geometries.

Instead of the global boundary value problem (1) we now consider the local boundary value problems

$$-\alpha_i \varDelta u_i(x) = f_i(x) \quad \text{for } x \in \varOmega_i, \quad u_i(x) = g(x) \quad \text{for } x \in \varGamma_i \cap \varGamma \tag{3}$$

together with the transmission boundary conditions

$$u_i(x) = u_j(x), \quad \alpha_i t_i(x) + \alpha_j t_j(x) = 0 \quad \text{for } x \in \varGamma_{ij} = \varGamma_i \cap \varGamma_j, \tag{4}$$

where $t_i = n_i \cdot \nabla u_i$ is the exterior normal derivative of u_i on $\varGamma_i$. Since the solution u_i of the local boundary value problem (3) is given via the representation formula

$$u_i(x) \;=\; \frac{1}{4\pi}\int\limits_{\Gamma_i}\frac{t_i(y)}{|x-y|}ds_y - \frac{1}{4\pi}\int\limits_{\Gamma_i} u_i(y)\frac{\partial}{\partial n_y}\frac{1}{|x-y|}ds_y + \frac{1}{\alpha_i}\frac{1}{4\pi}\int\limits_{\Omega_i}\frac{f_i(y)}{|x-y|}dy$$

for $x \in \Omega_i$, it is sufficient to find the complete Cauchy data $[u_i, t_i]_{|\Gamma_i}$ which are related to the solutions u_i of the local boundary value problems (3). The appropriate boundary integral equations to derive a boundary integral representation of the involved Dirichlet to Neumann map are given by means of the Calderon projector

$$\begin{pmatrix} u_i \\ t_i \end{pmatrix} = \begin{pmatrix} \frac{1}{2}I - K_i & V_i \\ D_i & \frac{1}{2}I + K_i' \end{pmatrix}\begin{pmatrix} u_i \\ t_i \end{pmatrix} + \frac{1}{\alpha_i}\begin{pmatrix} \widetilde{N}_0 f_i \\ \widetilde{N}_1 f_i \end{pmatrix},$$

where V_i is the single layer potential, K_i is the double layer potential, D_i is the hypersingular boundary integral operator, and $\widetilde{N}_j f_i$ are some Newton potentials, respectively. Hence, we find the Dirichlet to Neumann map as

$$\alpha_i t_i(x) = \alpha_i (S_i u_i)(x) - (N_i f_i)(x) \quad \text{for } x \in \Gamma_i \tag{5}$$

with the Steklov–Poincaré operator

$$\begin{aligned}(S_i u_i)(x) &= V_i^{-1}(\frac{1}{2}I + K_i)u_i(x) &(6)\\ &= \left[D_i + (\frac{1}{2}I + K_i')V_i^{-1}(\frac{1}{2}I + K_i)\right]u_i(x) \quad \text{for } x \in \Gamma_i, &(7)\end{aligned}$$

and with the local Newton potential

$$N_i f_i \;=\; V_i^{-1}\widetilde{N}_0 f_i \;=\; (\frac{1}{2}I + K_i')V_i^{-1}\widetilde{N}_0 f_i - \widetilde{N}_i f_i \quad \text{on } \Gamma_i.$$

Replacing the partial differential equation in (3) by the related Dirichlet to Neumann map (5) this results in a coupled formulation to find the local Cauchy data $[u_i, t_i]_{|\Gamma_i}$ such that

$$\begin{aligned} \alpha_i t_i(x) &= \alpha_i (S_i u_i)(x) - (N_i f_i)(x) && \text{for } x \in \Gamma_i, \\ u_i(x) &= g(x) && \text{for } x \in \Gamma_i \cap \Gamma, \\ u_i(x) &= u_j(x) && \text{for } x \in \Gamma_{ij}, \\ \alpha_i t_i(x) + \alpha_j t_j(x) &= 0 && \text{for } x \in \Gamma_{ij}. \end{aligned} \tag{8}$$

In what follows we first have to analyze the local Steklov–Poincaré operators $S_i : H^{1/2}(\Gamma_i) \to H^{-1/2}(\Gamma_i)$. Since we are dealing with fractional Sobolev spaces $H^{\pm 1/2}(\Gamma_i)$ one may ask for appropriate norms to be used. It turns out that norms which are induced by the local single layer potentials V_i may be advantageous. In particular,

$$\|\cdot\|_{V_i} = \sqrt{\langle V_i \cdot, \cdot\rangle_{\Gamma_i}}, \qquad \|\cdot\|_{V_i^{-1}} = \sqrt{\langle V_i^{-1} \cdot, \cdot\rangle_{\Gamma_i}}$$

are equivalent norms in $H^{-1/2}(\varGamma_i)$ and in $H^{1/2}(\varGamma_i)$, respectively. With the contraction property of the double layer potential [19],

$$\|(\frac{1}{2}I + K_i)v_i\|_{V_i^{-1}} \le c_{K,i}\|v_i\|_{V_i^{-1}} \quad \text{for all } v_i \in H^{1/2}(\varGamma_i) \tag{9}$$

where the constant

$$c_{K,i} = \frac{1}{2} + \sqrt{\frac{1}{4} - c_1^{D_i} c_1^{V_i}} < 1$$

is only shape sensitive, we have

$$\|S_i v_i\|_{V_i} = \|(\frac{1}{2}I + K_i)v_i\|_{V^{-1}} \le c_{K,i}\, \|v_i\|_{V_i^{-1}} \quad \text{for all } v_i \in H^{1/2}(\varGamma_i)$$

as well as

$$\langle S_i v_i, v_i\rangle_{\varGamma_i} \ge (1 - c_{K,i})\, \|v_i\|^2_{V_i^{-1}} \quad \text{for all } v_i \in H^{1/2}(\varGamma_i), v_i \perp 1.$$

In particular, the constants form the non–trivial kernel of the local Steklov–Poincaré operators S_i, i.e., $S_i 1 = 0$ in the sense of $H^{-1/2}(\varGamma_i)$. To realize the related orthogonal splitting of $H^{1/2}(\varGamma_i)$ we introduce the natural density $w_{eq,i} \in H^{-1/2}(\varGamma_i)$ as the unique solution of the local boundary integral equation $V_i w_{eq,i} = 1$. Then we may define the stabilized hypersingular boundary integral operator $\widetilde{S}_i : H^{1/2}(\varGamma_i) \to H^{-1/2}(\varGamma_i)$ via the Riesz representation theorem by the bilinear form

$$\langle \widetilde{S}_i u_i, v_i\rangle_{\varGamma_i} = \langle S_i u_i, v_i\rangle_{\varGamma_i} + \beta_i \langle u_i, w_{eq,i}\rangle_{\varGamma_i} \langle v_i, w_{eq,i}\rangle_{\varGamma_i}, \quad \beta_i \in \mathbb{R}_+. \tag{10}$$

Theorem 1. *Let $\widetilde{S}_i : H^{1/2}(\varGamma_i) \to H^{-1/2}(\varGamma_i)$ be the stabilized Steklov–Poincaré operator as defined in (10). Then there hold the spectral equivalence inequalities*

$$c_1^{\widetilde{S}_i} \langle V_i^{-1} v_i, v_i\rangle_{\varGamma_i} \le \langle \widetilde{S}_i v_i, v_i\rangle_{\varGamma_i} \le c_2^{\widetilde{S}_i} \langle V_i^{-1} v_i, v_i\rangle_{\varGamma_i}$$

for all $v_i \in H^{1/2}(\varGamma_i)$ with positive constants

$$c_1^{\widetilde{S}_i} = \min\{1 - c_{K,i}, \beta_i \langle 1, w_{eq,i}\rangle_{\varGamma_i}\}, \quad c_2^{\widetilde{S}_i} = \max\{c_{K,i}, \beta_i \langle 1, w_{eq,i}\rangle_{\varGamma_i}\}.$$

Therefore, an optimal scaling is given for

$$\beta_i = \frac{1}{2\langle 1, w_{eq,i}\rangle_{\varGamma_i}}, \quad c_1^{\widetilde{S}_i} = 1 - c_{K,i}, \quad c_2^{\widetilde{S}_i} = c_{K,i}.$$

Hence, the Dirichlet to Neumann map (5) can be written in a modified variational formulation as

$$\alpha_i \langle t_i, v_i\rangle_{\varGamma_i} = \langle \widetilde{S}_i \widetilde{u}_i, v_i\rangle_{\varGamma_i} - \langle N_i f_i, v_i\rangle_{\varGamma_i} \quad \text{for all } v_i \in H^{1/2}(\varGamma_i) \tag{11}$$

when assuming the local solvability conditions

$$\alpha_i \langle t_i, 1\rangle_{\Gamma_i} + \langle N_i f_i, 1\rangle_{\Gamma_i} \,=\, 0\,. \tag{12}$$

In particular, inserting $v_i = 1$ into the modified Dirichlet to Neumann map (11), we obtain from the solvability condition (12)

$$0 \,=\, \alpha_i \langle t_i, 1\rangle_{\Gamma_i} + \langle N_i f_i, 1\rangle_{\Gamma_i} \,=\, \langle S_i \widetilde{u}_i, 1\rangle_{\Gamma_i} + \beta_i \langle \widetilde{u}_i, w_{eq,i}\rangle_{\Gamma_i} \langle 1, w_{eq,i}\rangle_{\Gamma_i}$$

and therefore the scaling condition

$$\langle \widetilde{u}_i, w_{eq,i}\rangle_{\Gamma_i} \,=\, 0 \tag{13}$$

due to

$$\langle S_i \widetilde{u}_i, 1\rangle_{\Gamma_i} \,=\, \langle \widetilde{u}_i, S_i 1\rangle_{\Gamma_i} \,=\, 0, \quad \langle 1, w_{eq,i}\rangle_{\Gamma_i} \,=\, \langle 1, V_i^{-1} 1\rangle_{\Gamma_i} \,>\, 0.$$

In fact, the scaling condition (13) is the natural characterization of functions $\widetilde{u}_i \in H^{1/2}(\Gamma_i)$ which are orthogonal to the constants in the sense of $H^{-1/2}(\Gamma_i)$. Hence, the local Dirichlet datum is given via

$$u_i \,=\, \widetilde{u}_i + \gamma_i, \quad \gamma_i \in \mathbb{R}.$$

Now, the coupled formulation (8) can be rewritten as

$$\begin{array}{rll} \alpha_i t_i(x) = \alpha_i (\widetilde{S}_i \widetilde{u}_i)(x) - (N_i f_i)(x) & & \text{for } x \in \Gamma_i, \\ \widetilde{u}_i(x) + \gamma_i = g(x) & & \text{for } x \in \Gamma_i \cap \Gamma, \\ \widetilde{u}_i(x) + \gamma_i = \widetilde{u}_j(x) + \gamma_j & & \text{for } x \in \Gamma_{ij}, \\ \alpha_i t_i(x) + \alpha_j t_j(x) = 0 & & \text{for } x \in \Gamma_{ij}, \\ \alpha_i \langle t_i, 1\rangle_{\Gamma_i} + \langle N_i f_i, 1\rangle_{\Gamma_i} = 0 & & \end{array} \tag{14}$$

where we have to find $\widetilde{u}_i \in H^{1/2}(\Gamma_i)$, $t_i \in H^{-1/2}(\Gamma_i)$, and $\gamma_i \in \mathbb{R}$, $i = 1, \ldots, p$. Hereby, the variational formulation of the modified Dirichlet to Neumann map reads: Find $\widetilde{u}_i \in H^{1/2}(\Gamma_i)$ such that

$$\alpha_i \langle \widetilde{S}_i \widetilde{u}_i, v_i\rangle_{\Gamma_i} - \alpha_i \langle t_i, v_i\rangle_{\Gamma_i} \,=\, \langle N_i f_i, v_i\rangle_{\Gamma_i} \tag{15}$$

is satisfied for all $v_i \in H^{1/2}(\Gamma_i)$, $i = 1, \ldots, p$. The Neumann transmission conditions in weak form are

$$\langle \alpha_i t_i + \alpha_j t_j, v_{ij}\rangle_{\Gamma_{ij}} \,=\, \int\limits_{\Gamma_{ij}} [\alpha_i t_i(x) + \alpha_j t_j(x)] v_{ij}(x) ds_x \,=\, 0 \tag{16}$$

for all $v_{ij} \in H^{1/2}(\Gamma_{ij})$. Taking the sum over all interfaces Γ_{ij}, this is equivalent to

$$\sum_{i=1}^{p} \alpha_i \langle t_i, v_{|\Gamma_i}\rangle_{\Gamma_i \setminus \Gamma} \,=\, 0 \quad \text{for all } v \in H^{1/2}(\Gamma_S), \tag{17}$$

where $\mathit{\Gamma}_S = \cup_{i=1}^p \mathit{\Gamma}_i$ is the skeleton of the given domain decomposition. The Dirichlet transmission conditions in (14) can be written as

$$\langle [\widetilde{u}_i + \gamma_i] - [\widetilde{u}_j + \gamma_j], \tau_{ij} \rangle_{\mathit{\Gamma}_{ij}} \, = \, 0 \quad \text{for all } \tau_{ij} \in \widetilde{H}^{-1/2}(\mathit{\Gamma}_{ij}) = [H^{1/2}(\mathit{\Gamma}_{ij})]', \tag{18}$$

while the Dirichlet boundary conditions in weak form read

$$\langle \widetilde{u}_i + \gamma_i, \tau_{i0} \rangle_{\mathit{\Gamma}_i \cap \mathit{\Gamma}} \, = \, \langle g, \tau_{i0} \rangle_{\mathit{\Gamma}_i \cap \mathit{\Gamma}} \quad \text{for all } \tau_{i0} \in \widetilde{H}^{-1/2}(\mathit{\Gamma}_i \cap \mathit{\Gamma}). \tag{19}$$

In addition we need to have the local solvability conditions

$$\alpha_i \langle t_i, 1 \rangle_{\mathit{\Gamma}_i} + \langle N_i f_i, 1 \rangle_{\mathit{\Gamma}_i} \, = \, 0. \tag{20}$$

The coupled variational formulation (15)–(20) is in fact a mixed (saddle point) domain decomposition formulation of the original boundary value problem (1). Hence we have to ensure a certain stability (LBB) condition to be satisfied, i.e., a stable duality pairing between the primal variables $\widetilde{u}_i$ and the dual Lagrange variable t_i along the interfaces $\mathit{\Gamma}_{ij}$. Note that we also have to incorporate the additional constraints (20) and their associated Lagrange multipliers γ_i. While the unique solvability of the continuous variational formulation (15)–(20) follows in a quite standard way, as, e.g. in [16], the stability of an associated discrete scheme is not so obvious. Clearly, the Galerkin discretization of the coupled problem (15)–(20) depends on the local trial spaces to approximate the local Cauchy data $[\widetilde{u}, t_i]$. In particular, the variational formulation (15)–(20) may serve as a starting point for Mortar domain decomposition or three–field formulations as well (see [16] and the references given therein). However, here we will consider only an approach which is globally conforming.

Let $S_h^1(\mathit{\Gamma}_S)$ be a suitable trial space on the skeleton $\mathit{\Gamma}_S$, e.g., of piecewise linear basis functions φ_k, $k = 1, \ldots, M$, and let $S_h^1(\mathit{\Gamma}_i)$ denote its restriction onto $\mathit{\Gamma}_i$, where the local basis functions are φ_k^i, $k = 1, \ldots, M_i$. In particular, $A_i \in \mathbb{R}^{M_i \times M}$ are connectivity matrices linking the global degrees of freedom $\underline{u} \in \mathbb{R}^M \leftrightarrow u_h \in S_h^1(\mathit{\Gamma}_S)$ to the local ones, $\underline{u}_i = A_i \underline{u} \in \mathbb{R}^{M_i} \leftrightarrow u_{h|\mathit{\Gamma}_i} \in S_h^1(\mathit{\Gamma}_i)$. Moreover, let $S_h^0(\mathit{\Gamma}_{ij})$ be some trial space to approximate the local Neumann data t_i and t_j along the interface $\mathit{\Gamma}_{ij}$, for example we may use piecewise constant basis functions ψ_s^{ij}. In the same way we introduce basis functions $\psi_s^0 \in S_h^0(\mathit{\Gamma})$ to approximate the Neumann data along the Dirichlet boundary $\mathit{\Gamma}$. The trial spaces $S_h^0(\mathit{\Gamma}_{ij})$ and $S_h^0(\mathit{\Gamma})$ define a global trial space $S_h^0(\mathit{\Gamma}_S)$ of piecewise constant basis functions ψ_ι implying $\lambda_h \in S_h^0(\mathit{\Gamma}_S) \leftrightarrow \underline{\lambda} \in \mathbb{R}^N$, i.e., we have $\lambda_{h|\mathit{\Gamma}_{ij}} \in S_h^0(\mathit{\Gamma}_{ij}) \leftrightarrow \underline{\lambda}_{ij} \in \mathbb{R}^{N_{ij}}$ and $\lambda_{h|\mathit{\Gamma}} \in S_h^0(\mathit{\Gamma}) \leftrightarrow \underline{\lambda}_0 \in \mathbb{R}^{N_0}$. For the global trial space

$$S_h^0(\mathit{\Gamma}_S) \, = \, \bigcup_{i<j} S_h^0(\mathit{\Gamma}_{ij}) \cup S_h^0(\mathit{\Gamma}) \, = \, \operatorname{span}\{\psi_\iota\}_{\iota=0}^N,$$

we define the restrictions $\psi_s^{ij} = r_\iota^{ij} \psi_\iota$ with $r_\iota^{ij} = 1$, $r_\iota^{ji} = -1$ for $i < j$, and $\psi_s^0 = r_\iota^0 \psi_\iota$, $r_\iota^0 = 1$ for $x \in \mathit{\Gamma}$. Hence we can also introduce a local mapping

$$\underline{t}_i = \frac{1}{\alpha_i} R_i \underline{\lambda} \in \mathbb{R}^{N_i} \quad \text{for } \underline{\lambda} \in \mathbb{R}^N$$

satisfying

$$R_i[s_i, \iota] = r_\iota^{ij} = 1, \quad R_j[s_j, \iota] = r_\iota^{ji} = -1$$

for $\iota = 1, \ldots, N, s_i = 1, \ldots, N_i, i < j$, and $R_i[s_i, \iota] = r_\iota^0 = 1$ for $x \in \mathit{\Gamma}$. For the associated approximations $t_{i,h} \in S_h^0(\mathit{\Gamma}_i) \leftrightarrow \underline{t}_i \in \mathbb{R}^{N_i}$, we then find

$$\alpha_i t_{i,h}(x) + \alpha_j t_{j,h}(x) \;=\; 0 \quad \text{for } x \in \mathit{\Gamma}_{ij},$$

i.e., the Neumann transmission conditions (16) are satisfied in a strong sense.

The Galerkin approximation of the Dirichlet transmission condition (18) can now be written as

$$\int\limits_{\mathit{\Gamma}_{ij}} \left\{ \left[\sum_{k=1}^{M_i} \widetilde{u}_{i,k} \varphi_k^i(x) + \gamma_i \right] - \left[\sum_{k=1}^{M_j} \widetilde{u}_{j,k} \varphi_k^j(x) + \gamma_j \right] \right\} \psi_\sigma^{ij}(x) ds_x \;=\; 0$$

for $\sigma = 1, \ldots, N_{ij}$, and $i < j$, or for $\iota = 1, \ldots, N$

$$\int\limits_{\mathit{\Gamma}_{ij}} \left\{ \left[\sum_{k=1}^{M_i} \widetilde{u}_{i,k} \varphi_k^i(x) + \gamma_i \right] r_\iota^{ij} \psi_\iota(x) + \left[\sum_{k=1}^{M_j} \widetilde{u}_{j,k} \varphi_k^j(x) + \gamma_j \right] r_\iota^{ji} \psi_\iota(x) ds_x \right\} = 0.$$

Correspondingly, the Galerkin discretization of the Dirichlet boundary condition (19) reads

$$\int\limits_{\mathit{\Gamma}_i \cap \mathit{\Gamma}} \left[\sum_{k=1}^{M_i} \widetilde{u}_{i,k} \varphi_k^i(x) + \gamma_i \right] r_\iota^0 \psi_\iota(x) ds_x \;=\; \int\limits_{\mathit{\Gamma}_i \cap \mathit{\Gamma}} g(x) r_\iota^0 \psi_\iota(x) ds_x.$$

Combining both the Galerkin discretization of the Dirichlet transmission and of the Dirichlet boundary conditions, we can write

$$\sum_{i=1}^{p} B_i \underline{\widetilde{u}}_i + G\underline{\gamma} \;=\; \underline{g} \tag{21}$$

where $B_i \in \mathbb{R}^{M \times M_i}$ are defined by

$$B_i[\iota, k] = \int\limits_{\mathit{\Gamma}_{ij}} \varphi_k^i(x) r_\iota^{ij} \psi_\iota(x) ds_x, \quad B_i[\iota, k] = \int\limits_{\mathit{\Gamma}_i \cap \mathit{\Gamma}} \varphi_k^i(x) r_\iota^0 \psi_\iota(x) ds_x.$$

In addition, the matrix $G = (G_1, \ldots, G_p) \in \mathbb{R}^{M \times p}$ and the vector $\underline{g} \in \mathbb{R}^M$ of the right hand side are defined correspondingly, i.e.

$$G_i[\iota, i] = \int\limits_{\mathit{\Gamma}_{ij}} r_\iota^{ij} \psi_\iota(x) ds_x, \quad G_i[\iota, i] = \int\limits_{\mathit{\Gamma}_i \cap \mathit{\Gamma}_i} r_\iota^0 \psi_\iota(x) ds_x.$$

In particular, we have $G_i = B_i \underline{1}_i$ where $\underline{1}_i \in \mathbb{R}^{M_i}$ is the coefficient vector which is related to the constant function $1 \in H^{1/2}(\Gamma_i)$. Moreover, from the solvability conditions (20) we obtain

$$G_i^\top \underline{\lambda} = q_i = -\langle N_i f_i, 1\rangle_{\Gamma_i} \quad \text{for } i = 1, \ldots, p.$$

The Galerkin discretization of the local Dirichlet to Neumann map (15) finally gives

$$\alpha_i \widetilde{S}_{i,h} \underline{\widetilde{u}}_i - B_i^\top \underline{\lambda} \;=\; \underline{f}_i,$$

where we have to approximate the exact stiffness matrix $\widetilde{S}_{i,h}$ including the local Steklov–Poincaré operator S_i, e.g., in the symmetric representation (7), by some boundary element discretization,

$$\widetilde{S}_{i,h} = D_{i,h} + (\frac{1}{2}M_{i,h}^\top + K_{i,h}^\top)V_{i,h}^{-1}(\frac{1}{2}M_{i,h} + K_{i,h}) + \beta_i \underline{a}_i \underline{a}_i^\top,$$

where the local boundary element matrices are given as

$$\begin{aligned} &D_{i,h}[\ell,k] = \langle D_i \varphi_k^i, \varphi_\ell^i\rangle_{\Gamma_i}, \;\; K_{i,h}[\nu,k] = \langle K_i \varphi_k^i, \vartheta_\nu^i\rangle_{\Gamma_i}, \\ &V_{i,h}[\nu,\mu] = \langle V_i \vartheta_\mu^i, \vartheta_\nu^i\rangle_{\Gamma_i}, \;\; M_{i,h}[\nu,k] = \langle \varphi_k^i, \vartheta_\nu^i\rangle_{\Gamma_i}, \quad a_{i,k} \;=\; \langle \varphi_k^i, w_{eq,i}\rangle_{\Gamma_i} \end{aligned}$$

for $k, \ell = 1, \ldots, M_i$, $\mu, \nu = 1, \ldots, \bar{N}_i$ where $\operatorname{span}\{\vartheta_\mu^i\}_{\mu=1}^{\bar{N}_i} \subset H^{-1/2}(\Gamma_i)$ is some local boundary element trial space to approximate the local Neumann data which are needed in the definition of the approximate Steklov–Poincaré operator. Note that the basis functions ϑ_μ^i can be defined in an almost arbitrary way, we only have to assume some approximation property to ensure convergence of the discrete scheme. The simplest choice would be to identify the basis functions ϑ_μ^i with ψ_s^{ij} which are defined along the skeleton. In an analogous manner, one may even define an approximate Steklov–Poincaré operator by using local finite elements, see, e.g., [11]. Summarizing the above, we end up with a global system of linear equations,

$$\begin{pmatrix} \alpha_1 \widetilde{S}_{1,h} & & & -B_1^\top & \\ & \ddots & & \vdots & \\ & & \alpha_p \widetilde{S}_{p,h} & -B_p^\top & \\ B_1 & \cdots & B_p & & G \\ & & & G^\top & \end{pmatrix} \begin{pmatrix} \underline{\widetilde{u}}_1 \\ \vdots \\ \underline{\widetilde{u}}_p \\ \underline{\lambda} \\ \underline{\gamma} \end{pmatrix} = \begin{pmatrix} \underline{f}_1 \\ \vdots \\ \underline{f}_p \\ \underline{g} \\ \underline{q} \end{pmatrix}. \tag{22}$$

The unique solvability of the linear system (22) and therefore of the coupled variational problem (15)–(20) follows from some stability (LBB) condition linking the local trial spaces $S_h^1(\Gamma_i)$ and $S_h^0(\Gamma_{ij})$ along the coupling interface Γ_{ij}. Here, we only consider the special case where the basis functions φ_k^i and ψ_s^{ij} are biorthogonal, i.e.

$$\int\limits_{\Gamma_{ij}} \varphi_k^i(x)\psi_s^{ij}(x)ds_x = \begin{cases} 1 & \text{for } s = k, \\ 0 & \text{for } s \neq k. \end{cases}$$

Then, the entries of the matrices B_i consist just of zeros and ±1 describing a nodal coupling of the associated primal degrees of freedom. In particular, the use of biorthogonal basis functions ensures the LBB condition which is related to the block matrices B_i in (22), see, e.g., [21]. Moreover, the use of biorthogonal basis functions to discretize the coupled variational problem (15)–(20) is equivalent to a redundant finite or boundary element tearing and interconnecting approach for a standard domain decomposition formulation, see, e.g., [11].

While for matching grids the described formulation is a conforming discretization scheme, it may be generalized to different local grids and different local trial spaces as well. This leads immediately to hybrid or mortar domain decomposition methods where the choice of local trial spaces is essential to ensure the local stability conditions, see, e.g., [21] and the references given therein. Since the approximation of the local Dirichlet to Neumann maps can be done by any available discretization scheme, the presented formulation allows the coupling of different discretization schemes such as finite and boundary element methods, and the coupling of locally different meshes and trial spaces. When considering a boundary element approximation of the Steklov–Poincaré operator

$$S_i u_i = [D_i + (\frac{1}{2}I + K_i')V_i^{-1}(\frac{1}{2}I + K_i)]u_i = D_i u_i + (\frac{1}{2}I + K_i')w_i$$

the local Neumann data $w_i = V_i^{-1}(\frac{1}{2}I + K_i)$ coincide with the Neumann data t_i as used in the coupled formulation (14). It seems to be an open problem how this relation can be used to find further advanced boundary element domain decomposition formulations, in particular to find more efficient preconditioned iterative strategies to solve the resulting linear systems of equations in parallel.

3 Conclusions

For the numerical analysis of standard boundary element methods see, for example, [17]. Since the discretization of non–local boundary integral operators with singular kernel functions leads to dense stiffness matrices, the use of fast boundary element methods is an issue. For an overview of those methods, and for the implementation and for the application of the Adaptive Cross Approximation approach, see [14]. Other possible fast boundary element methods are the Fast Multipole Method, see, e.g., [13] and the references therein, or Hierarchical Matrices, see, e.g., [4]. The iterative solution of the linear system (22) of the boundary element tearing and interconnecting approach can be done by a projected preconditioned conjugate gradient method in a special inner product since the system matrix has a two fold saddle point structure, see also [9], where we have also described appropriate preconditioning strategies. While the potential equation (1) is just a model problem, the methodology given in this paper can be extended to more advanced problems in a straightforward way, e.g., for problems in linear elastostatics, for almost incompressible materials and for the Stokes problem. More challenging are the handling of the Helmholtz equation or of the Maxwell system where more advanced formulations are needed to obtain boundary integral equations which are unique solvable for all wave numbers.

Acknowledgement. This work has been partially supported by the German Research Foundation (DFG) under the Grant SFB 404 Multifield Problems in Continuum Mechanics.

References

[1] F. Brezzi and C. Johnson. On the coupling of boundary integral and finite element methods. *Calcolo*, 16:189–201, 1979.

[2] C. Carstensen, M. Kuhn, and U. Langer. Fast parallel solvers for symmetric boundary element domain decomposition methods. *Numer. Math.*, 79:321–347, 1998.

[3] M. Costabel. Symmetric methods for the coupling of finite elements and boundary elements. In C. A. Brebbia, G. Kuhn, and W. L. Wendland, editors, *Boundary Elements IX*, pages 411–420, Berlin, 1987. Springer.

[4] W. Hackbusch, B. N. Khoromskij, and R. Kriemann. Direct Schur complement method by domain decomposition based on $\mathcal{H}$–matrix approximation. *Comput. Vis. Sci.*, 8:179–188, 2005.

[5] G. C. Hsiao and W. L. Wendland. Domain decomposition methods in boundary element methods. In R. Glowinski and et. al., editors, *Domain Decomposition Methods for Partial Differential Equations. Proceedings of the Fourth International Conference on Domain Decomposition Methods*, pages 41–49, Baltimore, 1990. SIAM.

[6] E. G. Ivanov, H. Andrä, and A. N. Kudryavtsev. Domain decomposition approach for automatic parallel generation of 3d unstructured grids. In P. Wesseling, E. Onate, and J. Periaux, editors, *Proceedings of the European Conference on Computational Fluid Dynamics ECCOMAS CFD 2006*, TU Delft, The Netherlands, 2006.

[7] C. Johnson and J. C. Nedelec. On coupling of boundary integral and finite element methods. *Math. Comp.*, 35:1063–1079, 1980.

[8] B. N. Khoromskij and G Wittum. *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface*, volume 36 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, 2004.

[9] U. Langer, G. Of, O. Steinbach, and W. Zulehner. Inexact data–sparse boundary element tearing and interconnecting methods. *SIAM J. Sci. Comput.*, 29:290–314, 2007.

[10] U. Langer and O. Steinbach. Boundary element tearing and interconnecting methods. *Computing*, 71:205–228, 2003.

[11] U. Langer and O. Steinbach. Coupled boundary and finite element tearing and interconnecting methods. In R. Kornhuber, R. Hoppe, J. Periaux, O. Pironneau, O. Widlund, and J. Xu, editors, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lecture Notes in Computational Science and Engineering*, pages 83–97. Springer, Heidelberg, 2004.

[12] G. Of and O. Steinbach. Automatic generation of boundary element domain decomposition meshes. In preparation, 2007.

[13] G. Of, O. Steinbach, and W. L. Wendland. Boundary element tearing and interconnecting methods. In R. Helmig, A. Mielke, and B. I. Wohlmuth, editors, *Multifield Problems in Solid and Fluid Mechanics*, volume 28 of *Lecture Notes in Applied and Computational Mechanics*, pages 461–490. Springer, Heidelberg, 2006.

[14] S. Rjasanow and O. Steinbach. *The Fast Solution of Boundary Integral Equations*. Mathematical and Analytical Techniques with Applications to Engineering. Springer, New York, 2007.

[15] M. Schanz and O. Steinbach, editors. *Boundary Element Analysis: Mathematical Aspects and Applications*, volume 29 of *Lecture Notes in Applied and Computational Mechanics*. Springer, Heidelberg, 2007.

[16] O. Steinbach. *Stability estimates for hybrid coupled domain decomposition methods*, volume 1809 of *Lecture Notes in Mathematics*. Springer, Heidelberg, 2003.

[17] O. Steinbach. *Numerical Approximation Methods for Elliptic Boundary Value Problems. Finite and Boundary Elements*. Texts in Applied Mathematics. Springer, New York, 2007.

[18] O. Steinbach and W. L. Wendland. The construction of some efficient preconditioners in the boundary element method. *Adv. Comput. Math.*, 9:191–216, 1998.

[19] O. Steinbach and W. L. Wendland. On C. Neumann's method for second order elliptic systems in domains with non–smooth boundaries. *J. Math. Anal. Appl.*, 262:733–748, 2001.

[20] W. L. Wendland. On asymptotic error estimates for combined BEM and FEM. In E. Stein and W. L. Wendland, editors, *Finite Element and Boundary Element Techniques from Mathematical and Engineering Point of View*, volume 301 of *CISM Courses and Lectures*, pages 273–333. Springer, Wien, New York, 1988.

[21] B. I. Wohlmuth. *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, volume 17 of *Lecture Notes in Computational Science and Engineering*. Springer, Berlin, 2001.

Part II

Minisymposia

MINISYMPOSIUM 1: Advanced Multigrid Methods for Systems of PDEs

Organizers: Panayot S. Vassilevski[1] and Ludmil T. Zikatanov[2]

[1] Lawrence Livermore National Laboratory, CASC, USA. panayot@llnl.gov
[2] Pennsylvania State University, Department of Mathematics, Center for Computational Mathematics and Applications, USA. ludmil@psu.edu

The main theme of the minisymposium was centered around the construction of advanced multigrid techniques for the solution of large scale linear systems that typically arise from the discretization of systems of PDEs. Examples of such PDEs are found in numerical models used in electromagnetics, flow simulation, and elasticity. In the present proceedings two papers are included; one deals with a multilevel hierarchical basis preconditioner for 3D elliptic problems discretized by the DG (discontinuous Galerkin) method, whereas the second one deals with a new auxiliary space preconditioning method for (semi–)definite time domain Maxwell ($H(\text{curl})$) problems.

Auxiliary Space AMG for $\boldsymbol{H}(\text{curl})$ Problems

Tzanio V. Kolev and Panayot S. Vassilevski

Center for Applied Scientific Computing Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, USA. {tzanio,panayot}@llnl.gov *

1 Introduction

The search for efficient preconditioners for $\boldsymbol{H}(\text{curl})$ problems on unstructured meshes has intensified in the last few years. The attempts to directly construct AMG (algebraic multigrid) methods had some success, see [10, 1, 6]. Exploiting available multilevel methods on auxiliary mesh for the same bilinear form led to efficient auxiliary mesh preconditioners to unstructured problems as shown in [7, 4]. A computationally more attractive approach was recently announced in [5]. Their method borrows the main tool from the above mentioned auxiliary mesh preconditioners, namely, the interpolation operator $\boldsymbol{\Pi}_h$ that maps functions from $\boldsymbol{H}(\text{curl})$ into the lowest order Nédélec finite element space $\boldsymbol{V}_h$. The method of [5] and its motivation are outlined in Section 2. In particular, we describe briefly their Nédélec space decomposition, which is the basis of the auxiliary space AMG preconditioners.

In the present paper we consider several options for constructing unstructured mesh AMG preconditioners for $\boldsymbol{H}(\text{curl})$ problems and report a summary of computational results from [8, 9]. In contrast to [5], we apply AMG directly to variationally constructed coarse-grid operators, and therefore no additional Poisson matrices are needed on input. We also consider variable coefficient problems, including some that lead to a singular matrix. Both types of problems are of great practical importance, and are not covered by the theory of [5].

The main Section 3 consists of an extensive set of numerical experiments that illustrate the behavior of various auxiliary space AMG preconditioners.

* This work was performed under the auspices of the U. S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48. UCRL-PROC-223798

2 The Auxiliary Spaces and Operators

We are interested in solving the following variational problem stemming from the definite Maxwell equations:

$$\text{Find } \mathbf{u} \in \boldsymbol{V}_h \; : \quad (\alpha \operatorname{curl} \mathbf{u}, \operatorname{curl} \mathbf{v}) + (\beta\, \mathbf{u}, \mathbf{v}) = (\mathbf{f}, \mathbf{v})\,, \quad \text{for all } \mathbf{v} \in \boldsymbol{V}_h\,. \tag{1}$$

Here we consider $\alpha > 0$ and $\beta \geq 0$ which are scalar coefficients, but extensions to (semi)definite tensors are possible. We allow β to be zero in part or the whole domain (in which case the resulting matrix is only semidefinite, and for solvability the right-hand side should be chosen to satisfy compatibility conditions). Let $\boldsymbol{A}_h$ be the stiffness matrix corresponding to (1), where $\boldsymbol{V}_h$ is the (lowest order) Nédélec space associated with a triangulation $\mathcal{T}_h$.

Let S_h be the space of continuous piecewise linear finite elements associated with the same mesh $\mathcal{T}_h$ as $\boldsymbol{V}_h$, and $\boldsymbol{S}_h$ be its vector counterpart. Let G_h and $\boldsymbol{\Pi}_h$ be the matrix representations of the mapping $\varphi \in S_h \mapsto \nabla\varphi \in \boldsymbol{V}_h$ and the nodal interpolation from $\boldsymbol{S}_h$ to $\boldsymbol{V}_h$, respectively. Note that G_h has as many rows as the number of edges in the mesh, with each row having two nonzero entries: $+1$ and -1 in the columns corresponding to the edge vertices. The sign depends on the orientation of the edge. Furthermore, $\boldsymbol{\Pi}_h$ can be computed based only on G_h and on the coordinates of the vertices of the mesh.

The auxiliary space AMG preconditioner for $\boldsymbol{A}_h$ is a "three-level" method utilizing the subspaces $\boldsymbol{V}_h$, $G_h S_h$, and $\boldsymbol{\Pi}_h \boldsymbol{S}_h$. Its additive form reads (cf. [11])

$$\Lambda_h^{-1} + G_h B_h^{-1} G_h^T + \boldsymbol{\Pi}_h \boldsymbol{B}_h^{-1} \boldsymbol{\Pi}_h^T\,, \tag{2}$$

where Λ_h is a smoother for $\boldsymbol{A}_h$, while B_h and $\boldsymbol{B}_h$ are efficient preconditioners for $G_h^T \boldsymbol{A}_h G_h$ and $\boldsymbol{\Pi}_h^T \boldsymbol{A}_h \boldsymbol{\Pi}_h$ respectively. Since these matrices come from elliptic forms, the preconditioner of choice is AMG (especially for unstructured meshes).

If β is identically zero, one can skip the subspace correction associated with G_h, in which case we get a two-level method.

The motivation for (2) is that any finite element function $\mathbf{u}_h \in \boldsymbol{V}_h$ allows for decomposition of the form (cf., [5]) $\mathbf{u}_h = \mathbf{v}_h + \boldsymbol{\Pi}_h \mathbf{z}_h + \nabla\varphi_h$ with $\mathbf{v}_h \in \boldsymbol{V}_h$, $\mathbf{z}_h \in \boldsymbol{S}_h$ and $\varphi_h \in S_h$ such that the following stability estimates hold,

$$h^{-1}\|\mathbf{v}_h\|_0 + \|\mathbf{z}_h\|_1 \leq C\, \|\operatorname{curl} \mathbf{u}_h\|_0 \quad \text{and} \quad \|\nabla\varphi_h\|_0 \leq C\, \|\mathbf{u}_h\|_0. \tag{3}$$

3 Numerical Experiments

In this section we present results from numerical experiments with different versions of the auxiliary space AMG method used as a preconditioner in PCG.

We set Λ_h^{-1} to be a sweep of symmetric Gauss-Seidel, and consider the following preconditioners:

{1} Multiplicative version of (2) with B_h and $\boldsymbol{B}_h$ implemented as one AMG V-cycle for $G_h^T \boldsymbol{A}_h G_h$ and $\boldsymbol{\Pi}_h^T \boldsymbol{A}_h \boldsymbol{\Pi}_h$ respectively.
{2} Additive preconditioner using the same components as {1} and extra smoothing.
{3} Multiplicative preconditioner with B_h and $\boldsymbol{B}_h$ implemented using a sweep of geometric multigrid for Poisson problems, as described in [5].
{4} Additive preconditioner using the same components as {3} and extra smoothing.
{5} The preconditioner {3} using AMG instead of geometric multigrid.

The AMG algorithm we used is a serial version of the BoomerAMG solver from the *hypre* library. For more details see [2].

We report the number of preconditioned conjugate gradient iterations with the above preconditioners and relative tolerance 10^{-6}, i.e. the iterations were stopped after the preconditioned residual norm was reduced by six orders of magnitude. In a few of the tests we also tried the corresponding two-level methods (using exact solution in the subspaces) and listed the iteration counts in parentheses following the V-cycle results.

3.1 Constant Coefficients

First we consider several simple constant coefficients problems with $\alpha = \beta = 1$. We test both two-dimensional triangular and three-dimensional tetrahedral meshes. The results are listed in Tables 1–6, where the following notation was used: ℓ is the refinement level of the mesh, N is the size of the problem, and n_1 to n_5 give the iteration count for each of the auxiliary space AMG preconditioners {1} to {5}. When available, the error in L^2 is also reported.

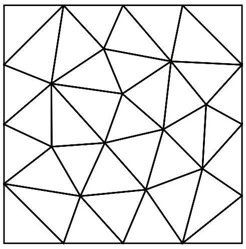
The experiments show that all considered solvers result in uniform and small number of iterations, which is in agreement with the theoretical results from [5, 9]. One can also observe that the multilevel results are very close in terms of number of iterations to the two-level ones.

Note that the first two methods (based on the original form) appear to work the same, independently of how complicated the geometry is. This is particularly interesting in the case of the third problem, where the assumption that the boundary is connected (needed to establish the decomposition in [5]) is violated. In contrast, the third and forth methods (based on Poisson subspace solvers) consistently result in higher number of iterations, and perform much worse on the third problem.

3.2 Variable Coefficients

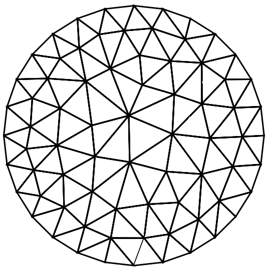
In Tables 7–8 we report results from a test where α and β are piecewise constant coefficients. Note that this case is not covered by the theory in [5]. However, the modifications to the Poisson-based preconditioners are straightforward, namely they assemble matrices corresponding to the bilinear forms

Table 1. Initial mesh and numerical results for the problem on a square.



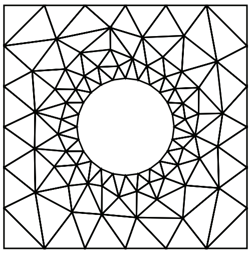
ℓ	N	n_1	n_2	n_3	n_4	$\|e\|_{L^2}$
2	896	4 (3)	9 (9)	10 (9)	16 (15)	0.011898
3	3520	4 (3)	10 (9)	11 (10)	17 (16)	0.005953
4	13952	4 (3)	10 (9)	12 (11)	18 (15)	0.002977
5	55552	4 (3)	10 (9)	13 (11)	18 (16)	0.001489
6	221696	4 (3)	10 (8)	13 (11)	18 (16)	0.000744
7	885760	5	10	13	18	0.000372
8	3540992	5	11	13	19	0.000186

Table 2. Initial mesh and numerical results for the problem on a disk.



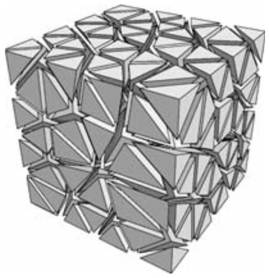
ℓ	N	n_1	n_2	n_3	n_4
1	736	4 (2)	9 (8)	7 (7)	11 (11)
2	2888	4 (3)	10 (9)	7 (7)	12 (11)
3	11440	4 (3)	10 (9)	7 (7)	12 (11)
4	45536	5 (3)	11 (9)	7 (7)	12 (11)
5	181696	5 (3)	11 (8)	8 (7)	12 (11)
6	725888	5	11	8	12
7	2901760	5	12	8	11

Table 3. Initial mesh and numerical results for the problem on a square with a circular hole.



ℓ	N	n_1	n_2	n_3	n_4
2	972	6 (3)	11 (9)	21 (20)	33 (31)
3	14976	6 (3)	12 (9)	23 (21)	33 (31)
4	59520	7 (3)	12 (9)	23 (17)	35 (23)
5	237312	6	13	24	35
6	947712	7	13	25	35
7	3787776	7	14	25	35

Table 4. Initial mesh and numerical results for the problem on a cube.

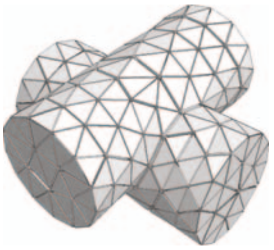


ℓ	N	n_1	n_2	n_3	n_4	$\|e\|_{L^2}$
0	722	3 (3)	9 (7)	6 (6)	11 (11)	0.6777
1	5074	4 (3)	10 (9)	9 (9)	16 (15)	0.3776
2	37940	5 (4)	11 (10)	12 (11)	20 (19)	0.2152
3	293224	5 (4)	11 (10)	14 (12)	22 (20)	0.1096
4	2305232	5	11	15	23	0.0549

Table 5. Initial mesh and numerical results for the problem on a ball.

ℓ	N	n_1	n_2	n_3	n_4
0	704	3 (3)	9 (7)	5 (5)	9 (9)
1	4669	4 (3)	10 (9)	7 (7)	13 (12)
2	37940	5 (4)	12 (10)	8 (8)	15 (14)
3	255700	6 (5)	13 (12)	9 (8)	15 (14)
4	1990184	7	14	10	16

Table 6. Initial mesh and numerical results for the problem on a union of two cylinders.



ℓ	N	n_1	n_2	n_3	n_4
0	1197	3 (3)	10 (8)	7 (7)	13 (13)
1	8248	4 (4)	11 (10)	10 (9)	17 (16)
2	60940	5 (5)	12 (11)	12 (11)	19 (18)
3	467880	6 (5)	12 (11)	13 (12)	21 (19)
4	3665552	6	13	14	22

$(\beta\nabla u, \nabla v)$ and $(\alpha\nabla\mathbf{u}, \nabla\mathbf{v}) + (\beta\mathbf{u}, \mathbf{v})$. Here we concentrated only on the multiplicative AMG methods.

For the problem illustrated in Table 7 (where the jumps are simple), we observe stable number of iterations with respect to both the mesh size and the magnitude of the jumps. Note that this setup was reported to be problematic for geometric multigrid in [3]. As before, the method based on the original form outperforms the one based on AMG Poisson subspace solvers.

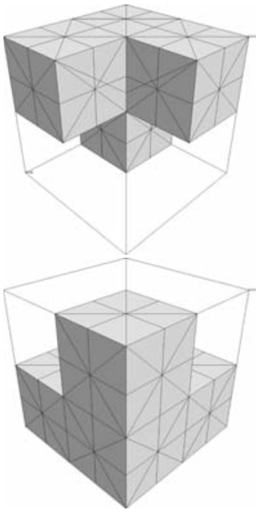
3.3 Singular Problems

Tables 9–10 present results for the problem corresponding to $\alpha = 1$, $\beta = 0$, i.e., to the bilinear form $(\operatorname{curl}\mathbf{u}, \operatorname{curl}\mathbf{v})$. In this case the matrix is singular, and the right-hand side, as well as the solution, belong to the space of discretely divergence free vectors (the kernel of G_h^T). Since $\beta = 0$, the solvers were modified to skip the correction in the space $G_h S_h$. This leads to a simpler preconditioner, which in additive form reads

$$\Lambda_h^{-1} + \boldsymbol{\Pi}_h \boldsymbol{B}_h^{-1} \boldsymbol{\Pi}_h^T. \tag{4}$$

The results in Tables 9–10 are quite satisfactory and comparable to those from Tables 3 and 6. This is not surprising, since (3) implies that any $[\mathbf{u}_h]$ in the factor space $\boldsymbol{V}_h/\nabla S_h$, has a representative $\tilde{\mathbf{u}}_h \in [\mathbf{u}_h]$, such that $\tilde{\mathbf{u}}_h = \mathbf{u}_h - \nabla\varphi_h = \mathbf{v}_h + \boldsymbol{\Pi}_h\mathbf{z}_h$ and

Table 7. Numerical results for the problem on a cube with α and β having different values in the shown regions (cf. [3]). Multiplicative preconditioner with AMG V-cycles in the subspaces.



ℓ	N	p								
		-8	-4	-2	-1	0	1	2	4	8
n_1 for $\alpha = 1, \beta \in \{1, 10^p\}$										
1	716	3	3	3	3	3	4	4	4	4
2	5080	4	4	4	4	4	4	5	6	6
3	38192	5	5	5	5	5	5	5	6	6
4	296032	5	5	5	5	5	5	6	6	6
5	2330816	5	5	5	5	5	6	6	6	6
n_1 for $\beta = 1, \alpha \in \{1, 10^p\}$										
1	716	6	6	5	4	3	4	4	4	4
2	5080	6	6	6	5	4	5	5	5	5
3	38192	7	7	7	5	5	5	6	6	6
4	296032	8	8	7	6	5	6	6	6	6
5	2330816	8	9	7	6	5	6	6	6	6

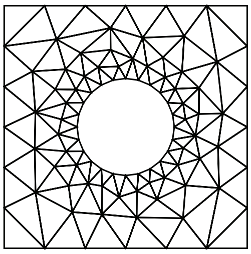
Table 8. Numerical results for the problem from Table 7 using multiplicative preconditioner with Poisson subspace solvers based on algebraic multigrid.

ℓ	N	p								
		-8	-4	-2	-1	0	1	2	4	8
n_5 for $\alpha = 1, \beta \in \{1, 10^p\}$										
1	716	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	6 (6)	6 (6)	5 (5)	5 (5)
2	5080	10 (10)	10 (10)	10 (10)	10 (10)	10 (10)	10 (10)	9 (9)	9 (9)	9 (9)
3	38192	11 (11)	11 (11)	11 (11)	11 (11)	11 (11)	11 (11)	10 (10)	11 (11)	12 (11)
4	296032	12 (12)	12 (12)	12 (12)	12 (12)	12 (12)	12 (12)	12 (12)	13 (13)	14 (13)
5	2330816	14	14	14	14	14	13	13	14	15
n_5 for $\beta = 1, \alpha \in \{1, 10^p\}$										
1	716	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	7 (7)	6 (6)
2	5080	10 (9)	9 (9)	10 (10)	10 (10)	10 (10)	10 (10)	10 (10)	10 (10)	9 (9)
3	38192	11 (11)	11 (11)	12 (12)	12 (12)	11 (11)	12 (12)	12 (12)	12 (12)	12 (12)
4	296032	13 (13)	13 (13)	14 (14)	14 (14)	12 (12)	15 (14)	15 (15)	15 (15)	14 (14)
5	2330816	15	15	16	16	14	16	17	17	17

$$h^{-1}\|\mathbf{v}_h\|_0 + \|\mathbf{z}_h\|_1 \le C\,\|\operatorname{curl}\tilde{\mathbf{u}}_h\|_0 = C\,\|\operatorname{curl}[\mathbf{u}_h]\|_0\,.$$

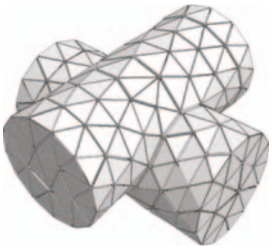
In Table 11 we also consider the important practical case when β is zero only in part of the region. For this test we used a preconditioner based on (2) instead of (4). Even though the problem is singular and β has jumps, the iterations counts are comparable to the case of constant coefficients. For example, the number of iterations for $\alpha = 1$, $\beta = 1$ given to the right of the table is almost identical to those when $\alpha = 1$, $\beta = 0$.

Table 9. Initial mesh and numerical results for the singular problem on a square with circular hole.



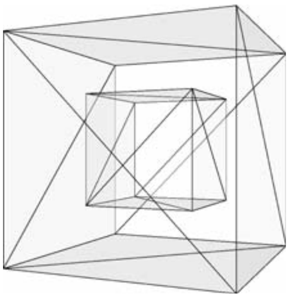
ℓ	N	n_1	n_2	n_3	n_4
2	972	7	12	19	28
3	14976	6	12	19	28
4	59520	7	12	19	28
5	237312	7	12	20	29
6	947712	7	11	20	29
7	3787776	7	12	21	29

Table 10. Initial mesh and numerical results for the singular problem on a union of two cylinders.



ℓ	N	n_1	n_2	n_3	n_4
0	1197	5	11	9	17
1	8248	6	13	12	19
2	60940	7	15	13	22
3	467880	7	15	14	23
4	3665552	8	15	15	23

Table 11. Initial mesh and numerical results for the problem on a cube with $\beta = 0$ outside the interior cube. Multiplicative preconditioner with AMG V-cycles in the subspaces.



ℓ	N	p									
		-8	-4	-2	-1	0	1	2	4	8	
n_1 for $\alpha = 1, \beta \in \{0, 10^p\}$											
1	485	2	3	3	2	4	2	3	3	3	(4)
2	3674	5	5	5	6	5	6	6	6	7	(6)
3	28692	8	7	8	7	7	8	8	10	10	(7)
4	226984	7	7	7	7	7	9	8	9	9	(7)
5	1806160	8	8	8	8	8	8	9	10	11	(8)

References

[1] P.B. Bochev, C.J. Garasi, J.J. Hu, A.C. Robinson, and R.S. Tuminaro. An improved algebraic multigrid method for solving Maxwell's equations. *SIAM J. Sci. Comput.*, 25(2):623–642, 2003.

[2] V.E. Henson and U.M. Yang. BoomerAMG: a parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41(1):155–177, 2002.

[3] R. Hiptmair. Multigrid method for Maxwell's equations. *SIAM J. Numer. Anal.*, 36(1):204–225, 1999.

[4] R. Hiptmair, G. Widmer, and J. Zou. Auxiliary space preconditioning in $H_0(\mathrm{curl}; \Omega)$. *Numer. Math.*, 103(3):435–459, 2006.

[5] R. Hiptmair and J. Xu. Nodal auxiliary space preconditioning in $H(curl)$ and $H(div)$ spaces. Technical Report 2006-09, ETH, Switzerland, 2006.

[6] J. Jones and B. Lee. A multigrid method for variable coefficient Maxwell's equations. *SIAM J. Sci. Comput.*, 27(5):1689–1708, 2006.

[7] Tz.V. Kolev, J.E. Pasciak, and P.S. Vassilevski. $H(curl)$ auxiliary mesh preconditioning, 2006. In preparation.

[8] Tz.V. Kolev and P.S. Vassilevski. Parallel H^1-based auxiliary space AMG solver for $H(curl)$ problems. Technical Report UCRL-TR-222763, LLNL, 2006.

[9] Tz.V. Kolev and P.S. Vassilevski. Some experience with a H^1-based auxiliary space AMG for $H(curl)$ problems. Technical Report UCRL-TR-221841, LLNL, 2006.

[10] S. Reitzinger and J. Schöberl. An algebraic multigrid method for finite element discretizations with edge elements. *Numer. Linear Algebra Appl.*, 9(3):223–238, 2002.

[11] J. Xu. The auxiliary space method and optimal multigrid preconditioning techniques for unstructured grids. *Computing*, 56(3):215–235, 1996.

A Multilevel Method for Discontinuous Galerkin Approximation of Three-dimensional Elliptic Problems

Johannes K. Kraus and Satyendra K. Tomar

Johann Radon Institute for Computational and Applied Mathematics, Altenberger Straße 69, A-4040 Linz, Austria.
{johannes.kraus,satyendra.tomar}@ricam.oeaw.ac.at

Summary. We construct optimal order multilevel preconditioners for interior-penalty discontinuous Galerkin (DG) finite element discretizations of 3D elliptic boundary-value problems. A specific assembling process is proposed which allows us to characterize the hierarchical splitting locally. This is also the key for a local analysis of the angle between the resulting subspaces. Applying the corresponding two-level basis transformation recursively, a sequence of algebraic problems is generated. These discrete problems can be associated with coarse versions of DG approximations (of the solution to the original variational problem) on a hierarchy of geometrically nested meshes. The presented numerical results demonstrate the potential of this approach.

1 Introduction

Discontinuous Galerkin (DG) finite element (FE) methods have gained much interest in the last decade due to their suitability for *hp*-adaptive techniques. They offer several advantages, e.g. the ease of treatment of meshes with hanging nodes, elements of varying shape and size, polynomials of variable degree, parallelization, preservation of local conservation properties, etc. An excellent overview and a detailed analysis of DG methods for elliptic problems can be found in [1]. Unfortunately, DG discretizations result in excessive number of degrees of freedom (DOF) as compared to their counter-part, i.e. the standard FE methods. Developing efficient preconditioning techniques, which yield fast iterative solvers, thus becomes of significant importance.

Optimal-order preconditioners obtained from recursive application of two-level FE methods have been introduced and extensively analyzed in the context of conforming methods, see e.g., [2, 3, 4]. For DG discretizations geometric multigrid (MG) type preconditioners and solvers for the linear system of equations have been considered in [6, 9]. However, our approach falls into the category of algebraic multilevel techniques. The method is obtained from recursive application of the two-level algorithm. A sequence of FE spaces is created using geometrically nested meshes. A specific splitting of the bilinear terms is proposed which results in an assembling process similar to that of the conforming methods. In this approach one avoids the projection onto a coarse (auxiliary) space [7, 13], where the auxiliary space is related to a standard Galerkin discretization, and instead, generates a sequence of algebraic problems associated with a hierarchy of coarse versions of DG approximations of the original problem.

The content of this paper is summarized as follows. In Section 2 we state our model problem and discuss the DG approximation. Discrete formulation and matrix assembly, based on the splitting proposed in [12], are parts of Section 3. In Section 4 we comment on the construction of a proper hierarchical basis transformation for the linear systems arising from DG discretization. The analysis of the angle between the induced subspaces is the subject of Section 5. Finally, numerical experiments are presented in Section 6.

2 Model Problem and DG Approximation

Consider a second order elliptic problem on a bounded Lipschitz domain $\Omega \subset \mathbb{R}^3$:

$$-\nabla \cdot (\underline{A}\,(x)\,\nabla u) = \underline{f}\,(x) \qquad \text{in } \Omega, \tag{1a}$$

$$u(x) = u_D \qquad \text{on } \Gamma_D, \tag{1b}$$

$$\underline{A}\nabla u \cdot \mathbf{n} = u_N \qquad \text{on } \Gamma_N. \tag{1c}$$

Here $\mathbf{n}$ is the exterior unit normal vector to $\partial\Omega \equiv \Gamma$. The boundary is assumed to be decomposed into two disjoint parts Γ_D and Γ_N, and the boundary data u_D, u_N are smooth. For the DG formulation below we shall need the existence of the traces of u and $\underline{A}\nabla u \cdot \mathbf{n}$ on the faces in Ω, and the solution u is assumed to have the required regularity. It is assumed that $\underline{A}$ is a symmetric positive definite matrix such that

$$c_1\,|\xi|^2 \leq \underline{A}\xi \cdot \xi \leq c_2\,|\xi|^2 \qquad \forall \xi \in \mathbb{R}^3.$$

Let $\mathcal{T}_h$ be a non-overlapping partition of Ω into a finite number of elements e. For any $e \in \mathcal{T}_h$ we denote its diameter by h_e and the boundary by ∂e. Let $\mathcal{F} = \bar{e}^+ \cap \bar{e}^-$ be a common face of two adjacent elements e^+, and e^-. Further, let $h = \max_{e\in\mathcal{T}_h} h_e$ denote the characteristic mesh size of the whole partition. The set of all the internal faces is denoted by $\mathcal{F}_0$, and $\mathcal{F}_D$ and $\mathcal{F}_N$ contain the faces of finite elements that belong to Γ_D and Γ_N, respectively. Finally, $\mathcal{F}$ is the set of all the faces, i.e., $\mathcal{F} = \mathcal{F}_0 \cup \mathcal{F}_D \cup \mathcal{F}_N$. We assume that the partition is shape-regular. We allow finite elements to vary in size and shape for local mesh adaptation and the mesh is not required to be conforming, i.e. elements may possess hanging nodes. Further, the face measure h_f is constant on each face $\mathcal{F} \in \mathcal{F}$ such that $h_f = |\mathcal{F}|^{\frac{1}{2}}, \quad \text{for } \mathcal{F} \in \mathcal{F}$.

On the partition $\mathcal{T}_h$ we define a broken Sobolev space:

$$\mathcal{V} := H^2(\mathcal{T}_h) = \{v \in L^2(\Omega) : v|_e \in H^2(e), \forall e \in \mathcal{T}_h\}.$$

Note that the functions in $\mathcal{V}$ **may not satisfy** any boundary condition. By

$$\mathcal{V}_h := \mathcal{V}_h(\mathcal{T}_h) = \{v \in L^2(\Omega) : v|_e \in P_r(e), \forall e \in \mathcal{T}_h\},$$

where P_r is the set of polynomials of degree $r \geq 1$, we define a finite dimensional subspace of $\mathcal{V}$. Obviously, $\mathcal{V}_h = \Pi_{e\in\mathcal{T}_h} P_r(e)$. For ease of notations in what follows, on $\mathcal{V}$ we introduce the following forms

$$(\underline{A}\nabla_h u_h, \nabla_h v_h)_{\mathcal{T}_h} := \sum_{e\in\mathcal{T}_h} \int_e \underline{A}\nabla_h u_h \cdot \nabla_h v_h dx, \quad \langle p, q\rangle_{\mathcal{F}^g} := \sum_{\mathcal{F}\in\mathcal{F}^g} \int_{\mathcal{F}} p \cdot q ds,$$

where $\mathcal{F}^g$ is one of the sets $\mathcal{F}$, $\mathcal{F}_0$, $\mathcal{F}_D$, $\mathcal{F}_N$ or any of their combinations.

Let us now recall the DG formulation for second order elliptic problems. In recent years a large number of DG FEM were developed for elliptic boundary value problems, for review see, e.g. [1] and the references therein. Below, we consider the standard interior penalty (IP) DG method, see, e.g., [1]. For the problem (1), the primal IP-DG formulation can be stated as follows:
Find $u_h \in \mathcal{V}$ such that

$$\mathcal{A}(u_h, v_h) = \mathcal{L}(v_h), \quad \forall v_h \in \mathcal{V}, \tag{2a}$$

where the bilinear form $\mathcal{A}(\cdot,\cdot) : \mathcal{V} \times \mathcal{V} \to \mathbb{R}$ and the linear form $\mathcal{L}(\cdot) : \mathcal{V} \to \mathbb{R}$ are defined by the relations

$$\begin{aligned}\mathcal{A}(u_h, v_h) =& (\underline{A}\nabla_h u_h, \nabla_h v_h)_{\mathcal{T}_h} + \alpha h_f^{-1} \langle [\![u_h]\!], [\![v_h]\!]\rangle_{\mathcal{F}_0\cup\mathcal{F}_D} \\ &- \langle \{\{\underline{A}\nabla_h u_h\}\}, [\![v_h]\!]\rangle_{\mathcal{F}_0\cup\mathcal{F}_D} - \langle [\![u_h]\!], \{\{\underline{A}\nabla_h v_h\}\}\rangle_{\mathcal{F}_0\cup\mathcal{F}_D}, \end{aligned} \tag{2b}$$

$$\mathcal{L}(v_h) = \int_\Omega f v_h dx + \alpha h_f^{-1} \langle u_D, v_h\rangle_{\mathcal{F}_D} - \langle u_D \mathbf{n}, \underline{A}\nabla_h v_h\rangle_{\mathcal{F}_D} + \langle u_N, v_h\rangle_{\mathcal{F}_N}. \tag{2c}$$

Here $\{\{\cdot\}\}$ and $[\![\cdot]\!]$ denote the trace operators for *average* and *jump*, respectively, and α is a parameter which is to be defined to guarantee the coercivity of the bilinear form $\mathcal{A}$, see e.g. [1, 11].

As usual, we assume that the Dirichlet boundary conditions are defined by a given function $u_D \in H^1(\Omega)$ in the sense that the trace of $u - u_D$ on Γ_D is zero. For the sake of simplicity, we also assume that u_D is such that the boundary condition can be exactly satisfied by the approximations used. For the coercivity, boundedness, and convergence properties of the bilinear form $\mathcal{A}$ the reader can refer, e.g., [1, 11].

3 Discrete Formulation and Matrix Assembly

The weak formulation (2) is transformed into a set of algebraic equations by approximating u_h and v_h using trilinear polynomials in each cubic element as

$$u_{e,h} = \sum_{j=0}^{7} \widetilde{u}_{e,j} \mathcal{N}_{e,j}(x)\,, \quad v_{e,h} = \sum_{j=0}^{7} \widetilde{v}_{e,j} \mathcal{N}_{e,j}(x)\,, \quad x \in e \subset \mathbb{R}^3. \tag{3}$$

Here $\widetilde{u}_{e,j} \in \mathbb{R}^8$ and $\widetilde{v}_{e,j} \in \mathbb{R}^8$ are the expansion coefficients of u_h and the test function v_h in the element e, respectively, and $\mathcal{N}_{e,j}$ are trilinear basis functions.

We now briefly show the computation of the element stiffness matrix. Consider a general element e with all its face internal. Let its neighboring elements, which share a face with this element, be denoted by e_1^+, e_2^+, e_3^+, e_4^+, e_5^+, and e_6^+. Here $\cdot^+$ represents the neighboring element and digits $1, \ldots, 6$ represent the face number with which the neighboring element is attached.

Using the definition of the trace operators $\{\{\cdot\}\}$ and $[\![\cdot]\!]$ and the specific splitting of the bilinear terms proposed in [12] the resulting elemental bilinear form reads

$$\begin{aligned}\mathcal{A}_e(u_h, v_h) = &\int_e \underline{A} \nabla_h u_h \cdot \nabla_h v_h dx - \frac{1}{2} \sum_{\mathrm{F}=1}^{6} \int_{\mathcal{F}_\mathrm{F}} \Big(\Big(v_e \mathbf{n}_e + v_{e_\mathrm{F}^+} \mathbf{n}_{e_\mathrm{F}^+} \Big) \cdot \underline{A} \nabla_h u_e \\ &+ \underline{A} \nabla_h v_e \cdot \Big(u_e \mathbf{n}_e + u_{e_\mathrm{F}^+} \mathbf{n}_{e_\mathrm{F}^+} \Big) \Big)\, ds \\ &+ \frac{\alpha h_f^{-1}}{2} \sum_{\mathrm{F}=1}^{6} \int_{\mathcal{F}_\mathrm{F}} \Big(v_e \mathbf{n}_e + v_{e_\mathrm{F}^+} \mathbf{n}_{e_\mathrm{F}^+} \Big) \cdot \Big(u_e \mathbf{n}_e + u_{e_\mathrm{F}^+} \mathbf{n}_{e_\mathrm{F}^+} \Big)\, ds.\end{aligned} \tag{4}$$

In this approach, the DOF of the element e are connected with only those DOF of its neighboring elements e_F^+ which are at the common face.

Now let $N = 8N_e$ denote the total number of DOF in the system. Using the polynomial approximation (3) into the weak form (2), with elemental bilinear form (4), we get the following linear system of equations

$$A\mathbf{x} = \mathbf{b}, \tag{5}$$

where $\mathbf{x} \in \mathbb{R}^N$, $A \in \mathbb{R}^{N \times N}$ with N_e^2 blocks of size 8×8, and $\mathbf{b} \in \mathbb{R}^N$, denote the vector of expansion coefficients, the global stiffness matrix, and the right hand side data vector, respectively.

4 Generalized Hierarchical Basis

In this section we discuss the two-level hierarchical basis (HB) transformation which is used in the construction of the multilevel preconditioner. Let us consider a hierarchy of partitions $\mathcal{T}_{h_\ell} \subset \mathcal{T}_{h_{\ell-1}} \subset \ldots \subset \mathcal{T}_{h_1} \subset \mathcal{T}_{h_0}$ of Ω, where the notation $\mathcal{T}_{h_k} = \mathcal{T}_h \subset \mathcal{T}_H = \mathcal{T}_{h_{k-1}}$ points out the fact that for any element e of the fine(r) partition $\mathcal{T}_h$ there is an element E of the coarse(r) mesh partition $\mathcal{T}_H$ such that $e \subset E$. For the construction of the preconditioner of the linear system (5) resulting from the IP-DG approximation of the basic problem (1) its DOF are partitioned into a *fine* and a *coarse* (sub-) set, indicated by the subscripts 1 and 2, respectively. The partitioning is induced by a regular mesh refinement at every level $(k-1) = 0, 1, \dots, \ell - 1$. In other words, by halving the mesh size, i.e., $h = H/2$, each element is subdivided into eight elements of similar shape, herewith producing the mesh at levels $k = 1, 2, \dots, \ell$. Hence, the linear system (5) can be represented in the 2×2 block form as

$$\begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} \begin{pmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix} \tag{6}$$

where $A_{21} = A_{12}^T$. By using the two-level transformation matrix

$$J = \begin{bmatrix} I_{11} & P_{12} \\ 0 & I_{22} \end{bmatrix}, \tag{7}$$

the system to be solved in the new basis has the representation

$$\widehat{A}\,\widehat{\mathbf{x}} = \widehat{\mathbf{b}}, \tag{8}$$

where $\widehat{A}$ and its submatrices $\widehat{A}_{11}$, $\widehat{A}_{12}$ $\widehat{A}_{21}$, $\widehat{A}_{22}$ are given by

$$\widehat{A} = J^T A\, J = \begin{bmatrix} \widehat{A}_{11} & \widehat{A}_{12} \\ \widehat{A}_{21} & \widehat{A}_{22} \end{bmatrix}, \tag{9a}$$

$$\widehat{A}_{11} = A_{11}, \quad \widehat{A}_{12} = A_{11}P_{12} + A_{12}, \quad \widehat{A}_{21} = P_{12}^T A_{11} + A_{21}, \tag{9b}$$

$$\widehat{A}_{22} = P_{12}^T A_{11} P_{12} + A_{21}P_{12} + P_{12}^T A_{12} + A_{22}. \tag{9c}$$

The vectors $\widehat{\mathbf{x}}$ and $\mathbf{b}$ are transformed according to $\mathbf{x} = J\,\widehat{\mathbf{x}}$ and $\widehat{\mathbf{b}} = J^T\mathbf{b}$, where

$$\mathbf{x}_1 = \widehat{\mathbf{x}}_1 + P_{12}\widehat{\mathbf{x}}_2, \quad \mathbf{x}_2 = \widehat{\mathbf{x}}_2, \tag{10a}$$

$$\widehat{\mathbf{b}}_1 = \mathbf{b}_1, \qquad \widehat{\mathbf{b}}_2 = P_{12}^T\mathbf{b}_1 + \mathbf{b}_2. \tag{10b}$$

If the interpolation matrix P_{12} in (7) is chosen in such a way that the matrix $\widehat{A}_{22}$ in (9c) corresponds to a coarse discretization of the original problem then P_{12} (and thus J) will constitute an HB transformation. In order to apply a local analysis, see the next Section, P_{12} is to be defined for a set of macro elements $\{E\}$ that covers the whole mesh. The general macro element we are using to define the local interpolation matrix P_E is simply the union of eight elements that share one vertex. The macro element accumulates 160 DOF, 32 of which define then an element on the next coarser level. The interpolation weights are simply taken to be 1/8 for the 8 interior fine DOF, 1/4 for the 48 fine DOF located at the face centers of the macro-element, and 1/2 for the 72 fine DOF associated with macro-element edges.

5 Local Estimation of the Constant in the CBS Inequality

It is known that the constant γ in the Cauchy-Bunyakowski-Schwarz (CBS) inequality, which is associated with the abstract angle between the two subspaces induced by the two-level hierarchical basis transformation, plays a key role in the derivation of optimal convergence rate estimates for two- and multilevel methods. Moreover, the value of the upper bound for $\gamma \in (0,1)$ is part of the construction of a proper stabilization polynomial in the linear algebraic multilevel iteration (AMLI) method, see [3, 4].

For the constant γ in the strengthened CBS inequality, the following relation holds

$$\gamma = \cos(\mathcal{V}_1, \, \mathcal{V}_2) = \sup_{u \in \mathcal{V}_1, \; v \in \mathcal{V}_2} \frac{\mathcal{A}(u,v)}{\sqrt{\mathcal{A}(u,u)\,\mathcal{A}(v,v)}}, \tag{11}$$

where $\mathcal{A}(\cdot,\cdot)$ is the bilinear form given by (2b). If $\mathcal{V}_1 \cap \mathcal{V}_2 = \{0\}$ then γ is strictly less than one. As shown in [8], the constant γ can be estimated locally over each macro-element $E \in \mathcal{T}_H$, i.e. $\gamma \leq \max\limits_{E} \gamma_E$, where

$$\gamma_E = \sup_{u \in \mathcal{V}_1(E), \; v \in \mathcal{V}_2(E)} \frac{\mathcal{A}_E(u,v)}{\sqrt{\mathcal{A}_E(u,u)\,\mathcal{A}_E(v,v)}}, \quad v \neq \text{const.}$$

The above mentioned spaces $\mathcal{V}_m(E), \;\; m = 1,2$, contain the functions from $\mathcal{V}_m$ restricted to E and $\mathcal{A}_E(u,v)$ corresponds to $\mathcal{A}(u,v)$ restricted to the macro element E.

Evidently, the global two-level stiffness matrix $\widehat{A}$ can be assembled from the macro-element two-level stiffness matrices $\widehat{A}_E$, which are obtained from assembling the element matrices for all elements e contained in E in the (local) hierarchical basis. In simplified notation this can be written as

$$\widehat{A} = J^T A \, J = \sum_{E \in \mathcal{T}_H} \widehat{A}_E = \sum_{E \in \mathcal{T}_H} J_E^T A_E \, J_E .$$

Like the global matrix, the local matrices are also of the following 2×2 block form

$$\widehat{A}_E = \begin{bmatrix} \widehat{A}_{E,11} & \widehat{A}_{E,12} \\ \widehat{A}_{E,21} & \widehat{A}_{E,22} \end{bmatrix} = J_E^T \begin{bmatrix} A_{E,11} & A_{E,12} \\ A_{E,21} & A_{E,22} \end{bmatrix} J_E, \tag{12}$$

where the local macro-element two-level transformation matrix J_E is defined by

$$J_E = \begin{bmatrix} I & P_E \\ 0 & I \end{bmatrix}, \tag{13}$$

and the transformation-invariant (local) Schur complement is given by

$$S_E = \widehat{A}_{E,22} - \widehat{A}_{E,21}\widehat{A}_{E,11}^{-1}\widehat{A}_{E,12} = A_{E,22} - A_{E,21}A_{E,11}^{-1}A_{E,12}. \tag{14}$$

In the present context the choice of P_E is based on simple averaging, see [11]. We know from the general framework of two-level block (incomplete) factorization methods that it suffices to compute the minimal eigenvalue $\lambda_{E;\min}$ of the generalized eigenproblem (cf. [8, Theorem 6])

$$S_E\mathbf{v}_{E,2} = \lambda_E\widehat{A}_{E,22}\mathbf{v}_{E,2}, \quad \mathbf{v}_{E,2} \perp (1,1,\ldots,1)^T, \tag{15}$$

in order to conclude the following upper bound for the constant γ in (11):

$$\gamma^2 \leq \max_{E\in\mathcal{T}_H}\gamma_E^2 = \max_{E\in\mathcal{T}_H}(1-\lambda_{E;\min}). \tag{16}$$

This relation then implies condition number estimates for the corresponding two-level preconditioner (of additive and multiplicative type), see, e.g., [2].

The analysis of multilevel methods obtained by recursive application of the two-level preconditioner necessitates the establishment of this kind of (local) bounds for each coarsening step since the two-level hierarchical basis transformation is also applied recursively. This requires the knowledge of the related (macro) element matrices on all coarse levels. For the hierarchical basis transformation, as described in detail in [11], we have a very simple recursion relation for the element matrices. This recursion relation shows that the sequence of (global) coarse-grid matrices can be associated with coarse-discretizations of the original problem but with an exponentially increasing sequence of stabilization parameters $\alpha^{(j)}$. In the following Lemma and Theorem we state the relation between the element matrices at successive levels and provide a local estimate for the CBS constant. For the proof the reader is referred to [11].

Lemma 1. *Let $\widehat{A}^{(\ell)} := (J^{(\ell)})^T A\, J^{(\ell)}$ denote the stiffness matrix from (5) in hierarchical basis, where $A = \sum_{e\in\mathcal{T}_h} A_e$ and $A_e = A_e(\alpha) =: A_e^{(0)}(\alpha)$ $\forall e \in \mathcal{T}_h$ denotes the element matrix. Let us further assume that A_e has the same representation over all the elements of the domain. Then, if one neglects the correction matrices related to the boundary conditions, the coarse-grid problem at level $(\ell - j)$, $j = 1,\ldots,\ell$, (involving the matrices $J^{(\ell)}, J^{(\ell-1)}, \ldots, J^{(\ell-j+1)}$) is characterized by the element matrix*

$$A_e^{(j)}(\alpha) = A_e(\alpha^{(j)}) = A_e(2^j\,\alpha). \tag{17}$$

In other words, the stabilization parameter α after j applications of the HB transformation equals $2^j\,\alpha$.

Theorem 1. *Consider the HB macro-element matrix $\widehat{A}_E^{(j)}(\alpha)$ associated with the eight elements defining the macro-element as a cube with side $2\,h_{\ell-j}$ where the element matrix $A_e^{(j)}(\alpha)$ from Lemma 1 is used in the standard way to assemble $A_E^{(j)}(\alpha)$. Then for the eigenvalues of (15) we have the lower bound*

$$\lambda_{E;\min}^{(j)} = \lambda_{E;\min}(\alpha^{(j)}) \geq \frac{1}{16}\left(1 - \frac{1}{\sqrt{2}\,\alpha^{(j)}}\right) = \frac{1}{16}\left(1 - \frac{1}{\sqrt{2}\,2^j\,\alpha}\right) \tag{18}$$

for all $\alpha \geq 3/2$, and thus the following relation holds for γ_E

$$\gamma_E \leq \sqrt{\frac{15}{16} + \frac{1}{16\sqrt{2}\,\alpha^{(j)}}}. \tag{19}$$

Remark 1. The bound (19) in Theorem 1 tells us that the condition number of the multiplicative preconditioner (with exact inversion of the A_{11}-block) can be stabilized using Chebyshev polynomials of degree five. However, as illustrated in the numerical examples below, this goal can also be achieved by employing four inner generalized conjugate gradient iterations.

6 Numerical Results

In this section we present numerical results which demonstrate the capabilities of the method. The computations are performed on Sun Fire V40z workstation with 4 AMD Opteron 852 CPUs (2.6GHz) with 32 GB RAM. For approximating u in all the examples we use trilinear elements i.e. linear shape functions for each of the variables x, y, and z. The stabilization parameter α is taken as 10. The pivot block in the multilevel preconditioner is approximated using incomplete LU (ILU) factorization based on a drop tolerance tol [12, 14]. For both the examples below we take Ω as a unit cube $(0,1) \times (0,1) \times (0,1)$.

Example 1. Consider the Poisson problem with homogeneous Dirichlet boundary conditions and choose f such that the analytic solution of the problem is given by $u = x\,(1-x)\,y\,(1-y)\,z\,(1-z)\exp{(2x+2y+2z)}$. The tolerance tol is taken as 10^{-2}.

Example 2. Consider the model problem (1) with homogeneous Dirichlet boundary conditions, $f = 1$, and the coefficient $\underline{A}$ as follows:

$$\underline{A} = \left\{ \begin{array}{ll} 1 \text{ in } & (I_1 \times I_1 \times I_1) \bigcup (I_2 \times I_2 \times I_1) \bigcup (I_2 \times I_1 \times I_2) \bigcup (I_1 \times I_2 \times I_2) \\ \varepsilon & \text{elsewhere} \end{array} \right\},$$

where $I_1 = (0, 0.5]$ and $I_2 = (0.5, 1)$, and $\varepsilon = \{0.1, 0.01, 0.001\}$. In this example the tolerance tol is chosen heuristically by relating it to the parameter ε as $\varepsilon \times 10^{-2}$.

For solving the linear system arising from various examples with varying h we employ the nonlinear algebraic multilevel iteration method (NLAMLI), see [5, 10, 12]. The stabilization of the condition number is achieved by using some fixed small number ν of inner generalized conjugate gradient (GCG) iterations. Here we choose $\nu = 4$ in all computations. The starting vector for the outer iteration is the zero vector and the stopping criteria is $\|r^{(n_{\text{it}})}\|/\|r^{(0)}\| \leq \delta =$

10^{-6}, where n_{it} is the number of iterations we report in the tables below. The coarsest mesh in all computations is of size 4×4×4 and has 512 DOF. The finer meshes for $1/h = 8, 16, 32, 64$ consist of $4096, \ldots, 2097152$ DOF, respectively.

Table 1. Numerical results

$1/h$	n_{it}	ρ	sec
8	27	0.59	0.47
16	27	0.60	4.76
32	27	0.60	44.27
64	27	0.60	422.09

(a) Example 1

	$\varepsilon = 0.1$		$\varepsilon = 0.01$		$\varepsilon = 0.001$	
$1/h$	n_{it}	ρ	n_{it}	ρ	n_{it}	ρ
8	25	0.57	25	0.56	25	0.56
16	28	0.61	28	0.60	28	0.61
32	30	0.62	29	0.62	30	0.62
64	30	0.62	29	0.62	30	0.63

(b) Example 2

In Table 1(a) we present the number of iterations, the average convergence factor ρ and the total CPU time (including the time for the construction of the preconditioner) for Example 1. We observe that the number of iterations is constant and the CPU time is proportional to the problem size which shows that the overall solution process is of optimal order of computational complexity. The same holds for Example 2, cf. Table 1(b). These results also indicate the robustness of the preconditioner with respect to the jumps in the coefficient $\underline{A}$.

Acknowledgement. The authors acknowledge the support by the Austrian Academy of Sciences.

References

[1] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001/02.

[2] O. Axelsson. *Iterative Solution Methods.* Cambridge University Press, Cambridge, 1994.

[3] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods. I. *Numer. Math.*, 56(2-3):157–177, 1989.

[4] O. Axelsson and P.S. Vassilevski. Algebraic multilevel preconditioning methods. II. *SIAM J. Numer. Anal.*, 27(6):1569–1590, 1990.

[5] O. Axelsson and P.S. Vassilevski. Variable-step multilevel preconditioning methods. I. Selfadjoint and positive definite elliptic problems. *Numer. Linear Algebra Appl.*, 1(1):75–101, 1994.

[6] S.C. Brenner and J. Zhao. Convergence of multigrid algorithms for interior penalty methods. *Appl. Numer. Anal. Comput. Math.*, 2(1):3–18, 2005.

[7] V.A. Dobrev, R.D. Lazarov, P.S. Vassilevski, and L.T. Zikatanov. Two-level preconditioning of discontinuous Galerkin approximations of second-order elliptic equations. *Numer. Linear Algebra Appl.*, 13(9):753–770, 2006.

[8] V. Eijkhout and P.S. Vassilevski. The role of the strengthened Cauchy-Buniakowskiĭ-Schwarz inequality in multilevel methods. *SIAM Rev.*, 33(3):405–419, 1991.

[9] J. Gopalakrishnan and G. Kanschat. A multilevel discontinuous Galerkin method. *Numer. Math.*, 95(3):527–550, 2003.

[10] J.K. Kraus. An algebraic preconditioning method for M-matrices: linear versus non-linear multilevel iteration. *Numer. Linear Algebra Appl.*, 9(8):599–618, 2002.

[11] J.K. Kraus and S.K. Tomar. A multilevel method for discontinuous Galerkin approximation of three-dimensional anisotropic elliptic problems. *Numer. Linear Algebra Appl.*, in press.

[12] J.K. Kraus and S.K. Tomar. Multilevel preconditioning of two-dimensional elliptic problems discretized by a class of discontinuous Galerkin methods. *SIAM J. Sci. Comput.*, to appear.

[13] L. Lazarov and S. Margenov. CBS constants for graph-Laplacians and application to multilevel methods for discontinuous Galerkin systems. *J. Complexity*, doi: 10.1016/j.jco.2006.10.003, 2006.

[14] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

MINISYMPOSIUM 2: Domain Decomposition Based on Boundary Elements

Organizers: Olaf Steinbach[1] and Wolfgang Wendland[2]

[1] Institute of Computational Mathematics, Graz University of Technology, Austria. o.steinbach@tugraz.at

[2] Institute of Applied Analysis and Numerical Simulation, University of Stuttgart, Germany. wendland@mathematik.uni-stuttgart.de

For non–overlapping domain decomposition methods the Steklov–Poincaré operator or its inverse play a decisive role to represent the Dirichlet to Neumann of the Neumann to Dirichlet map. In corresponding research during the last decades remarkable achievements have been made by using boundary integral and boundary element methods. In the minisymposium several aspects were discussed.

In particular for exterior boundary value problems the coupling of finite and boundary element methods seems to be mandatory. In the case of spherical domains one may use an explicit representation of the Dirichlet to Neumann map as it was considered in the talk of Yu Dehao on the natural boundary reduction and the domain decomposition method in unbounded domains. Coupled finite and boundary element tearing and interconnecting methods for nonlinear potential problems in unbounded regions were considered by C. Pechstein. Since the boundary integral formulation of Helmholtz and Maxwell boundary value problems may involve non–trivial eigensolutions, i.e. spurious modes, special care has to be taken. Therefore, R. Hiptmair presented a resonance free interface coupled BEM for the Maxwell system. A boundary element tearing and interconnecting approach for the numerical solution of variational inequalities was presented by Z. Dostál. Finally, A. Litvinenko discussed the use of hierarchical matrices as domain decomposition preconditioner for the skin problem.

Scalable BETI for Variational Inequalities

Jiří Bouchala, Zdeněk Dostál and Marie Sadowská

Department of Applied Mathematics, Faculty of Electrical Engineering
and Computer Science, VŠB-Technical University of Ostrava, 17. listopadu 15,
Ostrava-Poruba, 708 33, Czech Republic.
{jiri.bouchala,zdenek.dostal,marie.sadowska}@vsb.cz

Summary. We briefly review our first results concerning the development of scalable BETI based domain decomposition methods adapted to the solution of variational inequalities such as those describing the equilibrium of a system of bodies in mutual contact. They exploit classical results on the FETI and BETI domain decomposition methods for elliptic partial differential equations and our recent results on quadratic programming. The results of the numerical solution of a semicoercive model problem are given that are in agreement with the theory and illustrate the numerical scalability of our algorithm.

1 Introduction

The FETI (Finite Element Tearing and Interconnecting) domain decomposition method proposed by Farhat and Roux turned out to be one of the most successful methods for a parallel solution of linear problems described by elliptic partial differential equations and discretized by the finite element method (see [11]). Its key ingredient is a decomposition of the spatial domain into non-overlapping subdomains that are "glued" by Lagrange multipliers, so that, after eliminating the primal variables, the original problem is reduced to a small, typically equality constrained, quadratic programming problem that is solved iteratively. The time that is necessary for both the elimination and iterations can be reduced nearly proportionally to the number of the subdomains, so that the algorithm enjoys parallel scalability. Since then, many preconditioning methods were developed which guarantee also numerical scalability of the FETI methods (see, e.g., [15]). Recently Steinbach and Langer (see [13]) adapted the FETI method to the solution of problems discretized by the boundary element method. They coined their new BETI (Boundary Element Tearing and Interconnecting) method and proved its numerical scalability.

The FETI based results were recently extended to the solution of elliptic boundary variational inequalities, such as those describing the equilibrium of a system of elastic bodies in mutual contact. Using the so-called "natural coarse grid" introduced by Farhat, Mandel, and Roux (see [10]) and new algorithms for the solution of special quadratic programming problems (see [9, 4, 5]), Dostál and Horák modified the basic FETI algorithm and proved its numerical scalability also for the solution of variational inequalities (see [7]).

The latter algorithms turned out to be effective also for the solution of problems discretized by boundary elements (see [8, 2, 3]). In this paper, we review our BETI based algorithm for the solution of variational inequalities and report our theoretical results that guarantee the scalability of BETI with a natural coarse grid. Theoretical results are illustrated by numerical experiments.

2 Model Problem and Domain Decomposition

Let us consider the domain $\Omega = (0,1) \times (0,1)$ and let us denote $\Gamma_c = \{(0,y) :\ y \in [0,1]\}$ and $\Gamma_f = \partial\Omega \setminus \Gamma_c$. Moreover, let $f \in L^2(\Omega)$ satisfy

$$\int_\Omega f(x)\,\mathrm{d}x < 0 \tag{1}$$

and $g \in L^2(\Gamma_c)$. We shall look for a sufficiently smooth function u satisfying

$$-\triangle u = f \quad \text{in } \Omega, \qquad \frac{\partial u}{\partial n} = 0 \quad \text{on } \Gamma_f \tag{2}$$

together with the Signorini conditions

$$u - g \geq 0, \qquad \frac{\partial u}{\partial n} \geq 0, \qquad \frac{\partial u}{\partial n}(u-g) = 0 \quad \text{on } \Gamma_c. \tag{3}$$

Let us decompose the domain Ω into p non-overlapping subdomains,

$$\overline{\Omega} = \bigcup_{i=1}^{p} \overline{\Omega_i},\ \Omega_i \cap \Omega_j = \emptyset \quad \text{for } i \neq j,\ \Gamma_i = \partial\Omega_i,\ \Gamma_{i,j} = \Gamma_i \cap \Gamma_j,\ \Gamma_s = \bigcup_{i=1}^{p} \Gamma_i.$$

We assume that each subdomain boundary Γ_i is Lipschitz, and that for each subdomain Ω_i we have

$$\operatorname{diam} \Omega_i < 1. \tag{4}$$

We now reformulate the problem (2), (3) as a system of local subproblems

$$-\triangle u_i = f \quad \text{in } \Omega_i, \qquad \lambda_i = \frac{\partial u_i}{\partial n} = 0 \quad \text{on } \Gamma_i \cap \Gamma_f, \tag{5}$$

$$u_i - g \geq 0, \quad \lambda_i \geq 0, \quad \lambda_i(u_i - g) = 0 \quad \text{on } \Gamma_i \cap \Gamma_c \tag{6}$$

together with the so-called transmission conditions

$$u_i = u_j \quad \text{and} \quad \lambda_i + \lambda_j = 0 \quad \text{on } \Gamma_{i,j}. \tag{7}$$

We introduce the local single layer potential operator V_i, the double layer potential operator K_i, the adjoint double layer potential operator K_i', and the hypersingular boundary integral operator D_i defined by

$$\begin{aligned}
(V_i\lambda_i)(x) &= \int_{\Gamma_i} U(x,y)\lambda_i(y)\,\mathrm{d}s_y, \quad V_i: \ H^{-1/2}(\Gamma_i) \mapsto H^{1/2}(\Gamma_i),\\
(K_i u_i)(x) &= \int_{\Gamma_i} \frac{\partial}{\partial n_y} U(x,y) u_i(y)\,\mathrm{d}s_y, \quad K_i: \ H^{1/2}(\Gamma_i) \mapsto H^{1/2}(\Gamma_i),\\
(K_i'\lambda_i)(x) &= \int_{\Gamma_i} \frac{\partial}{\partial n_x} U(x,y)\lambda_i(y)\,\mathrm{d}s_y, \quad K_i': \ H^{-1/2}(\Gamma_i) \mapsto H^{-1/2}(\Gamma_i),\\
(D_i u_i)(x) &= -\frac{\partial}{\partial n_x}\int_{\Gamma_i} \frac{\partial}{\partial n_y} U(x,y) u_i(y)\,\mathrm{d}s_y, \quad D_i: \ H^{1/2}(\Gamma_i) \mapsto H^{-1/2}(\Gamma_i),
\end{aligned}$$

$x \in \Gamma_i$. The function U denotes the fundamental solution of the Laplace operator in $\mathbb{R}^2$ and it is defined by

$$U(x,y) = -\frac{1}{2\pi}\log\|x-y\| \quad \text{for } x,\, y \in \mathbb{R}^2.$$

From the assumption (4) it follows that the operator V_i is $H^{-1/2}(\Gamma_i)$-elliptic, and therefore its inversion is well-defined. Now let us define the local Dirichlet to Neumann map as

$$\lambda_i(x) = (S_i u_i)(x) - (N_i f)(x), \quad x \in \Gamma_i,$$

where S_i denotes the local Steklov-Poincaré operator given by

$$(S_i u_i)(x) = \left[D_i + (\frac{1}{2}I + K_i')V_i^{-1}(\frac{1}{2}I + K_i)\right] u_i(x), \quad x \in \Gamma_i,$$

and $N_i f$ denotes the local Newton potential given by

$$(N_i f)(x) = V_i^{-1}\left(N_{0,i} f\right)(x), \quad x \in \Gamma_i,$$

with $(N_{0,i}f)(x) = \int_{\Omega_i} U(x,y) f(y)\,\mathrm{d}y$. It can be further shown that the local Steklov-Poincaré operator $S_i: \ H^{1/2}(\Gamma_i) \mapsto H^{-1/2}(\Gamma_i)$ is bounded, symmetric, and semi-elliptic on $H^{1/2}(\Gamma_i)$. More details on the properties of the Steklov-Poincaré operator may be found, e.g., in [14].

3 Boundary Variational Formulation and Discretization

The boundary weak formulation of the problem (5), (6), (7) may be equivalently rewritten as the problem of finding $u \in \mathcal{K} = \big\{v \in H^{1/2}(\Gamma_s) : \ v - g \geq 0$ on $\Gamma_c\big\}$ such that

$$\begin{aligned} &\mathcal{J}(u) = \min\{\mathcal{J}(v) :\ v \in \mathcal{K}\}, \\ &\mathcal{J}(v) = \sum_{i=1}^{p} \left[\frac{1}{2} \int_{\Gamma_i} (S_i v|_{\Gamma_i})(x) v|_{\Gamma_i}(x) \,\mathrm{d}s_x - \int_{\Gamma_i} (N_i f)(x) v|_{\Gamma_i}(x) \,\mathrm{d}s_x \right]. \end{aligned} \tag{8}$$

The coercivity of the functional $\mathcal{J}$ follows (see [12]) from the condition (1). We shall now follow the technique of Langer and Steinbach (see [13]). Let us define the local boundary element space

$$Z_{i,h} = \text{span}\ \{\psi_k^i\}_{k=1}^{N_i} \subset H^{-1/2}(\Gamma_i)$$

to get suitable approximations $\widetilde{S}_i$ and $\widetilde{N}_i f$ of S_i and $N_i f$. The exact definitions and results on stability can be found, e.g., in [14]. Let us further define the boundary element space on the skeleton Γ_s and its restriction on Γ_i as

$$W_h = \text{span}\ \{\varphi_m\}_{m=1}^{M_0} \subset H^{1/2}(\Gamma_s) \quad \text{and} \quad W_{i,h} = \text{span}\ \{\varphi_m^i\}_{m=1}^{M_i} \subset H^{1/2}(\Gamma_i),$$

respectively. After the discretization of problem (8) by the Ritz method, we get the minimization problem

$$\begin{aligned} &J(\mathbf{u}) = \min\{J(\mathbf{v}) :\ \mathbf{v} \in \mathbb{R}^M,\ \mathsf{B}_I \mathbf{v} \le \mathbf{c}_I,\ \mathsf{B}_E \mathbf{v} = \mathbf{o}\}, \\ &J(\mathbf{v}) = \frac{1}{2} \mathbf{v}^T \widetilde{\mathsf{S}} \mathbf{v} - \mathbf{v}^T \widetilde{\mathbf{R}} \end{aligned} \tag{9}$$

with $M = \sum_{i=1}^{p} M_i$ and with a positive semidefinite block diagonal stiffness matrix $\widetilde{\mathsf{S}}$. The blocks of $\widetilde{\mathsf{S}}$ and the relevant blocks of $\widetilde{\mathbf{R}}$ are given by

$$\begin{aligned} &\widetilde{\mathsf{S}}_{i,h} = \mathsf{D}_{i,h} + (\frac{1}{2}\mathsf{M}_{i,h} + \mathsf{K}_{i,h})^T \mathsf{V}_{i,h}^{-1} (\frac{1}{2}\mathsf{M}_{i,h} + \mathsf{K}_{i,h}) \quad \text{and} \\ &\widetilde{\mathbf{R}}_{i,h} = \mathsf{M}_{i,h}^T \mathsf{V}_{i,h}^{-1} \mathbf{N}_{0,i,h}, \end{aligned}$$

respectively. The boundary element matrices and the vector $\mathbf{N}_{0,i,h}$ are defined by

$$\begin{aligned} &\mathsf{V}_{i,h}[l,k] = \left\langle V_i \psi_k^i, \psi_l^i \right\rangle_{L^2(\Gamma_i)}, \qquad &&\mathsf{M}_{i,h}[l,n] = \left\langle \varphi_n^i, \psi_l^i \right\rangle_{L^2(\Gamma_i)}, \\ &\mathsf{K}_{i,h}[l,n] = \left\langle K_i \varphi_n^i, \psi_l^i \right\rangle_{L^2(\Gamma_i)}, \qquad &&\mathsf{D}_{i,h}[m,n] = \left\langle D_i \varphi_n^i, \varphi_m^i \right\rangle_{L^2(\Gamma_i)}, \\ &\mathbf{N}_{0,i,h}[l] = \left\langle N_{0,i} f, \psi_l^i \right\rangle_{L^2(\Gamma_i)} \end{aligned}$$

for $k,\ l = 1, \dots, N_i$; $m,\ n = 1, \dots, M_i$ and $i = 1, \dots, p$. The inequality constraints are associated with the non-penetration condition across $\Gamma_i \cap \Gamma_c$, while the equality constraints arise from the continuity condition across the auxiliary interfaces $\Gamma_{i,j}$.

4 Dual Formulation and Natural Coarse Grid

We shall now use the duality theory to replace the general inequality constraints by the bound constraints. Let $\widetilde{\mathsf{S}}^+$ be a generalized inverse of $\widetilde{\mathsf{S}}$ satisfying $\widetilde{\mathsf{S}} = \widetilde{\mathsf{S}}\widetilde{\mathsf{S}}^+\widetilde{\mathsf{S}}$ and let R be a matrix whose columns span the kernel of

$\widetilde{\mathsf{S}}$. By introducing the Lagrange multipliers $\boldsymbol{\lambda}_I$ and $\boldsymbol{\lambda}_E$ associated with the inequalities and equalities, respectively, and denoting

$$\boldsymbol{\lambda} = \left[\boldsymbol{\lambda}_I^T,\ \boldsymbol{\lambda}_E^T\right]^T, \quad \mathsf{B} = \left[\mathsf{B}_I^T,\ \mathsf{B}_E^T\right]^T, \quad \text{and} \quad \mathbf{c} = \left[\mathbf{c}_I^T, \mathbf{o}^T\right]^T,$$

we can equivalently replace problem (9) by

$$\begin{aligned} &\Theta(\overline{\boldsymbol{\lambda}}) = \min\{\Theta(\boldsymbol{\lambda}) :\ \boldsymbol{\lambda}_I \geq \mathbf{o} \quad \text{and} \quad \widetilde{\mathsf{G}}\boldsymbol{\lambda} = \widetilde{\mathbf{e}}\}, &(10)\\ &\Theta(\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{\lambda}^T\mathsf{F}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\widetilde{\mathbf{d}}, \quad \mathsf{F} = \mathsf{B}\widetilde{\mathsf{S}}^{+}\mathsf{B}^T, \quad \widetilde{\mathbf{d}} = \mathsf{B}\widetilde{\mathsf{S}}^{+}\widetilde{\mathbf{R}} - \mathbf{c}, &\\ &\widetilde{\mathsf{G}} = \mathsf{R}^T\mathsf{B}^T, \quad \widetilde{\mathbf{e}} = \mathsf{R}^T\widetilde{\mathbf{R}}. & \end{aligned}$$

The solution $\mathbf{u}$ of (9) then may be evaluated by

$$\mathbf{u} = \widetilde{\mathsf{S}}^{+}(\widetilde{\mathbf{R}} - \mathsf{B}^T\overline{\boldsymbol{\lambda}}) + \mathsf{R}\boldsymbol{\alpha} \quad \text{and} \quad \boldsymbol{\alpha} = (\mathsf{R}^T\widetilde{\mathsf{B}}^T\widetilde{\mathsf{B}}\mathsf{R})^{-1}\mathsf{R}^T\widetilde{\mathsf{B}}^T(\widetilde{\mathbf{c}} - \widetilde{\mathsf{B}}\widetilde{\mathsf{S}}^{+}(\widetilde{\mathbf{R}} - \mathsf{B}^T\overline{\boldsymbol{\lambda}})),$$

where $\widetilde{\mathsf{B}} = [\widetilde{\mathsf{B}}_I^T,\ \mathsf{B}_E^T]^T$ and $\widetilde{\mathbf{c}} = [\widetilde{\mathbf{c}}_I^T, \mathbf{o}^T]^T$, and the matrix $[\widetilde{\mathsf{B}}_I, \widetilde{\mathbf{c}}_I]$ is formed by the rows of $[\mathsf{B}_I, \mathbf{c}_I]$ corresponding to the positive entries of $\overline{\boldsymbol{\lambda}}_I$. Now let us denote by T a regular matrix such that the matrix $\mathsf{G} = \mathsf{T}\widetilde{\mathsf{G}}$ has orthonormal rows. Then (see, e.g., [6]) problem (10) is equivalent to the following problem

$$\begin{aligned} &\Lambda(\overline{\boldsymbol{\lambda}}) = \min\{\Lambda(\boldsymbol{\lambda}) :\ \boldsymbol{\lambda}_I \geq -\widetilde{\boldsymbol{\lambda}}_I \ \text{ and } \ \mathsf{G}\boldsymbol{\lambda} = \mathbf{o}\}, &(11)\\ &\Lambda(\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{\lambda}^T\mathsf{PFP}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T\mathsf{P}\mathbf{d}, \quad \mathsf{P} = \mathsf{I} - \mathsf{Q}, \quad \mathsf{Q} = \mathsf{G}^T\mathsf{G}, &\\ &\mathbf{d} = \widetilde{\mathbf{d}} - \mathsf{F}\widetilde{\boldsymbol{\lambda}}, \quad \widetilde{\boldsymbol{\lambda}} = \mathsf{G}^T\mathbf{e}, \quad \mathbf{e} = \mathsf{T}\widetilde{\mathbf{e}}. & \end{aligned}$$

The matrices P and Q are orthogonal projectors on the kernel of G and the image of G^T, respectively, and they define the so-called natural coarse grid.

The key ingredient in the next development is the observation that there are positive constants C_1, C_2 such that

$$C_1\|\mathsf{P}\boldsymbol{\lambda}\|^2 \leq \boldsymbol{\lambda}^T\mathsf{PFP}\boldsymbol{\lambda} \leq C_2 H/h. \tag{12}$$

This nontrivial estimate is a corollary of two well-known results. The first one is the classical estimate of Farhat, Mandel, and Roux (see [10]) which gives that if F_{FETI} and P_{FETI} denote the matrices arising by an application of the above procedures to the problem discretized by sufficiently regular finite element grid with the discretization and decomposition parameters h and H, respectively, then there are positive constants C_3, C_4 such that the spectrum $\sigma(\mathsf{F}_{FETI}|\mathrm{Im}\mathsf{P}_{FETI})$ of the restriction of F_{FETI} to $\mathrm{Im}\mathsf{P}_{FETI}$ satisfies

$$\sigma(\mathsf{F}_{FETI}|\mathrm{Im}\mathsf{P}_{FETI}) \subseteq [C_3,\ C_4 H/h].$$

The second result is due to Langer and Steinbach, in particular, Lemma 3.3 of [13] which guarantees that $\mathsf{F}|\mathrm{Im}\mathsf{P}$ is spectrally equivalent to $\mathsf{F}_{FETI}|\mathrm{Im}\mathsf{P}_{FETI}$. Combining these two results, it is possible to prove (12). We shall give more details elsewhere.

5 Algorithms and Optimality

To solve the bound and equality constrained problem (11), we use our recently proposed algorithms MPRGP by Dostál and Schöberl (see [9]) and SMALBE (see [4, 5]). The SMALBE, a variant of the augmented Lagrangian method with adaptive precision control, enforces the equality constraints by the Lagrange multipliers generated in the outer loop, while auxiliary bound constrained problems are solved approximately in the inner loop by MPRGP, an active set based algorithm which uses the conjugate gradient method to explore the current face, the fixed steplength gradient projection to expand the active set, the adaptive precision control of auxiliary linear problems, and the reduced gradient with the optimal steplength to reduce the active set. The unique feature of SMALBE with the inner loop implemented by MPRGP when used to (11) is the rate of convergence in bounds on spectrum of the regular part of the Hessian of Λ (see [5]). Combining this result with the estimate (12), we get that if H/h is bounded, then there is a bound on the number of multiplications by the Hessian of Λ that are necessary to find an approximate solution $\overline{\boldsymbol{\lambda}}_{h,H}$ of (11) discretized with the decomposition parameter H and the discretization parameter h which satisfies

$$\|\mathbf{g}^P(\overline{\boldsymbol{\lambda}}_{h,H})\| \leq \varepsilon_1 \|\mathsf{P}_{h,H}\mathbf{d}_{h,H}\| \quad \text{and} \quad \|\mathsf{G}_{h,H}\overline{\boldsymbol{\lambda}}_{h,H}\| \leq \varepsilon_2 \|\mathsf{P}_{h,H}\mathbf{d}_{h,H}\|, \tag{13}$$

where $\mathbf{g}^P$ denotes the projected gradient, whose nonzero components are those violating the KKT conditions for (11) (see, e.g., [1]).

6 Numerical Experiment

Let $f(x,y) = -1$ for $(x,y) \in \Omega$ and $g(0,y) = \sqrt{1/4 - (y-1/2)^2} - 1$ for $y \in [0,1]$. We decompose Ω into identical square subdomains with the side length H. All subdomain boundaries Γ_i were further discretized by the same regular grid with the element size h. The spaces $W_{i,h}$ and $Z_{i,h}$ were formed by piecewise linear and piecewise constant functions, respectively. For the SMALBE algorithm we used the parameters $\eta = ||\mathsf{P}\mathbf{d}||$, $\beta = 10$, and $M = 1$. The penalty parameter ρ_0 and the Lagrange multipliers $\boldsymbol{\mu}^0$ for the equality constraints were set to $10\,\|\mathsf{PFP}\|$ and $\mathbf{o}$, respectively. For the MPRGP algorithm we used parameters $\overline{\alpha} = \|\mathsf{PFP} + \rho_k \mathsf{Q}\|^{-1}$ and $\Gamma = 1$. Our initial approximation $\boldsymbol{\lambda}^0$ was set to $-\widetilde{\boldsymbol{\lambda}}$. The stopping criterion of the outer loop was chosen as

$$\left\|\mathbf{g}^P(\boldsymbol{\lambda}^k, \boldsymbol{\mu}^k, \rho_k)\right\| \leq 10^{-4}\,\|\mathsf{P}\mathbf{d}\| \quad \text{and} \quad \left\|\mathsf{G}\boldsymbol{\lambda}^k\right\| \leq 10^{-4}\,\|\mathsf{P}\mathbf{d}\|\,.$$

The results of our numerical experiments are given in Table 1. We conclude that the scalability may be observed in the solution of realistic problems.

7 Comments and Conclusions

We combined the BETI methodology with preconditioning by the "natural coarse grid" to develop a scalable algorithm for the numerical solution of

Table 1. Performance with the constant ratio $H/h = 32$.

h	H	primal dim.	dual dim.	outer iter.	CG iter.
1/64	1/2	512	197	2	48
1/128	1/4	2048	915	2	58
1/256	1/8	8192	3911	2	52
1/512	1/16	32768	16143	2	45

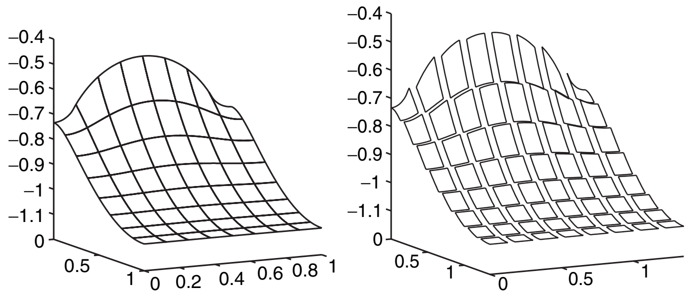


Fig. 1. Solution of the model problem with $h = 1/256$ and $H = 1/8$. On the right we emphasize the particular local solutions.

variational inequalities. The algorithm may be used for the solution of both coercive and semicoercive contact problems. Though we have restricted our exposition to a model variational inequality, our arguments are valid also for 2D and 3D contact problems of elasticity.

Acknowledgement. The research was supported by the grants GA CR 201/07/0294, AS CR 1ET400300415, and the Ministry of Education of the Czech Republic No. MSM6198910027.

References

[1] D.P. Bertsekas. *Nonlinear Optimization.* Athena Scientific-Nashua, 1999.

[2] J. Bouchala, Z. Dostál, and M. Sadowská. Solution of boundary variational inequalities by combining fast quadratic programming algorithms with symmetric BEM. In *Advances in Boundary Integral Methods, The Fifth UK Conference on Boundary Integral Methods*, pages 221–228, University of Liverpool, 2005.

[3] J. Bouchala, Z. Dostál, and M. Sadowská. Duality based algorithms for the solution of multidomain variational inequalities discretized by BEM. 2006. Submitted.

[4] Z. Dostál. Inexact semimonotonic augmented Lagrangians with optimal feasibility convergence for convex bound and equality constrained quadratic programming. *SIAM J. Numer. Anal.*, 43(1):96–115, 2005.

[5] Z. Dostál. An optimal algorithm for bound and equality constrained quadratic programming problems with bounded spectrum. *Computing*, 78:311–328, 2006.

[6] Z. Dostál and D. Horák. Scalability and FETI based algorithm for large discretized variational inequalities. *Math. Comput. Simulation*, 61:347–357, 2003.

[7] Z. Dostál and D. Horák. Theoretically supported scalable FETI for numerical solution of variational inequalities. *SIAM J. Numer. Anal.*, 45(2):500–513, 2007.

[8] Z. Dostál, J. Malík, A. Friedlander, and S.A. Santos. Analysis of semicoercive contact problems using symmetric BEM and augmented Lagrangians. *Engineering Analysis with Boundary Elements*, 18:195–201, 1996.

[9] Z. Dostál and J. Schöberl. Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Comput. Optim. Appl.*, 30:23–43, 2005.

[10] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115:365–387, 1994.

[11] C. Farhat and F.-X. Roux. An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems. *SIAM J. Sci. Comput.*, 13:379–396, 1992.

[12] I. Hlaváček, J. Haslinger, J. Nečas, and J. Lovíšek. *Solution of Variational Inequalities in Mechanics.* Springer - Verlag Berlin, 1988.

[13] U. Langer and O. Steinbach. Boundary element tearing and interconnecting methods. *Computing*, 71:205–228, 2003.

[14] O. Steinbach. *Stability Estimates for Hybrid Coupled Domain Decomposition Methods, Lecture Notes in Mathematics*, volume 1809. Springer - Verlag Berlin Heidelberg, 2003.

[15] A. Toselli and O.B. Widlund. *Domain Decomposition Methods—Algorithms and Theory.* Springer, Berlin, 2005.

Domain Decomposition Based $\mathcal{H}$-Matrix Preconditioners for the Skin Problem

Boris N. Khoromskij and Alexander Litvinenko

Max Planck Institute for Mathematics in the Sciences, Inselstr. 22, Leipzig, 04103,
Germany, bokh@mis.mpg.de, litvinen@tu-bs.de

Summary. In this paper we propose and analyse a new hierarchical Cholesky ($\mathcal{H}$-Cholesky) factorization based preconditioner for iterative solving the elliptic equations with highly jumping coefficients arising in the so-called skin-modeling problem in 3D [8]. First, we construct the block-diagonal approximation to the FE stiffness matrix, which is well suited to the "perforated" structure of the coefficients. We apply the $\mathcal{H}$-Cholesky factorization of this block-diagonal matrix as a preconditioner in the PCG iteration. It is shown that the new preconditioner is robust with respect to jumps in the coefficients and it requires less storage and computing time than the standard $\mathcal{H}$-Cholesky factorization.

1 Introduction

In papers [1, 9, 11] the authors successfully applied the PCG, GMRES, BiCGStab iterations with $\mathcal{H}$-matrix based preconditioners to different types of second order elliptic differential equations. In some cases $\mathcal{H}$-matrix inverse can be used even as a direct solver [6, 4]. In this paper we consider the elliptic equation with highly jumping coefficients,

$$\begin{array}{ll} -div(\alpha(x)\nabla u) = f(x) & x \in \Omega \subset \mathbb{R}^d,\ d = 2, 3, \\ u = 0 & x \in \gamma, \\ \frac{\partial u}{\partial n} = g & x \in \Gamma \setminus \gamma, \end{array} \tag{1}$$

where $\Gamma = \partial\Omega$, $\gamma \subset \partial\Omega$ corresponds to a piece of the boundary with the Dirichlet boundary condition. This equation was used for the numerical modeling of the so-called skin problem that describes penetration of drugs through the skin (cf. [8]). To simplify the model we choose Ω as a fragment with 8 cells Ω_c and the lipid layer Ω_l (see Fig. 2), where $\Omega = \Omega_c \cup \overline{\Omega}_l$, $\Omega_c = \cup_{i=1}^{8} \Omega_{c,i}$, Ω_l is a closed set. Fig. 1 (right) shows cells and the lipid layer in between. Typical feature for the skin problem is the highly jumping coefficients: the penetration coefficient inside the cells is very small, $\alpha(x) = \varepsilon \in [10^{-5} - 10^{-3}]$, but it is relatively large in the lipid layer ($\alpha(x) = 1$). In this problem the

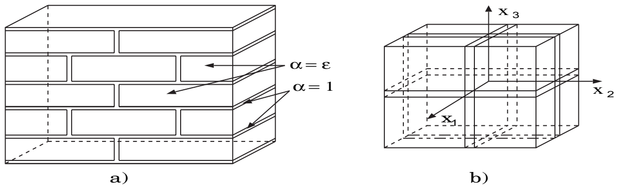


Fig. 1. (left) A skin fragment consists of the lipid layer and disjoint cells. **(right)** The simplified model of a skin fragment contains 8 cells with the lipid layer in between. $\Omega = [-1, 1]^3$, $\alpha(x) = \varepsilon$ inside the cells and $\alpha(x) = 1$ in the lipid layer.

Dirichlet boundary condition describes the presence of drugs on the boundary γ of the skin fragment. The nonzero Neumann condition on $\Gamma \backslash \gamma$ specifies the penetration through the surface $\Gamma \backslash \gamma$. The right-hand side in (1) presents external forces.

It is known that for problems with jumping coefficients (see (1)) the condition number $cond(A)$ of the FE stiffness matrix A is proportional to $h^{-2} \sup_{x,y \in \Omega} \alpha(x)/\alpha(y)$, where $\alpha(x)$ denotes the jumping coefficient and h is the step size of a finite element scheme. In the case of a large condition number one requires the efficient preconditioner W, so that $cond(W^{-1}A) \simeq 1$.

The rest of this paper is structured as follows. In Section 2 we describe the FEM discretization of (1). We recall the main idea of the $\mathcal{H}$-matrix techniques in Section 3. Section 4 describes the new preconditioner and presents the condition number estimates. Numerics for the 3D model problem is discussed in Section 5.

2 Discretization by FEM

We choose a rectangular quasi-uniform triangulation τ_h which is compatible with the lipid layer, i.e., $\tau_h := \tau_h^l \cup \tau_h^c$, where τ_h^l is a triangulation of the lipid layer and τ_h^c is a triangulation of cells (see Fig. 2). In the presented example Ω_l contains only two grid layers.

Let $V_h := span\{b_1, ..., b_n\}$ be the set of piecewise linear functions with respect to τ_h such that

$$V_h \subset H^1_{0,\gamma}(\Omega) := \{u \in H^1(\Omega) : \quad u|_\gamma = 0\}, \tag{2}$$

where b_j, $j \in I_\Omega := \{1, ..., n\}$, is the set of corresponding hat-functions. The related Galerkin discretization of the problem (1) reads as:

$$\text{find } u_h \in V_h \text{, so that } a(u_h, v) = c(v) \text{ for all } v \in V_h \tag{3}$$

with respective bilinear form a and linear functional c given by

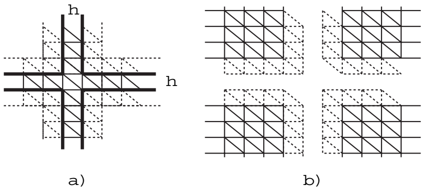


Fig. 2. a) A 2D grid of the lipid layer of width h (bounded by bold lines). b) Fragments of four cells ($\alpha = \varepsilon$). The finite elements restricted by dotted lines in (a) and (b) are needed for constructing the stiffness matrices A_{11} and A_{22}, respectively.

$$a(u,v) = \int_\Omega \alpha(x)(\nabla u, \nabla v)dx, \quad c(v) := \int_\Omega f v dx + \int_{\Gamma\backslash\gamma} g v d\Gamma. \tag{4}$$

The system of linear algebraic equations for the coefficients vector $\mathbf{u}$ reads as

$$A_\varepsilon \mathbf{u} = \mathbf{c}, \text{ where } A_\varepsilon = \{a(b_j, b_i)\}_{i,j\in I_\Omega} \in \mathbb{R}^{n\times n}, \, \mathbf{c} = \{c(b_i)\}_{i\in I_\Omega} \in \mathbb{R}^n. \tag{5}$$

The lipid layer Ω_l between the cells specifies the natural decomposition of Ω. The thickness of this layer is proportional to the step size h. Note that with the proper ordering of the index set I_Ω, we can represent the global stiffness matrix in the following block form

$$A_\varepsilon = \begin{bmatrix} A_{11} & \varepsilon A_{12} \\ \varepsilon A_{21} & \varepsilon A_{22} \end{bmatrix}, \quad A_{11} = A_0 + \varepsilon B_{11}. \tag{6}$$

Here A_{11}, εA_{22} are the stiffness matrices which correspond to the lipid layer and to the rest of the domain, respectively, εA_{12} and εA_{21} are coupling matrices. In turn, A_0 discretizes the Neumann problem in Ω_l with homogeneous Neumann data on the inner boundaries $\partial\Omega_c$. In the case $\varepsilon = 1$, the matrix A_ε corresponds to the discrete Laplace operator.

In the following we focus on a construction of the efficient preconditioner to the matrix A_ε, while the analysis of the discretization accuracy remains beyond the scope of the present paper. However, the error analysis can be based on the standard FEM theory under certain regularity assumptions.

3 Hierarchical Matrices

The hierarchical matrices ($\mathcal{H}$-matrices) (cf. [3]) provide the efficient data-sparse representation of fully-populated matrices arising in a wide range of FEM/BEM applications. The main idea of $\mathcal{H}$-matrices is to approximate certain subblocks $R \in \mathbb{R}^{n\times m}$ of a given matrix by the rank-k matrices, i. e., $R \cong AB^T$, with $A \in \mathbb{R}^{n\times k}$ and $B \in \mathbb{R}^{m\times k}$, $k \ll \min(n,m)$. The storage requirement for both matrices A and B is $k(n+m)$ instead of $n \cdot m$ for the matrix R. The advantage of the $\mathcal{H}$-matrix technique is that the complexity of the $\mathcal{H}$-matrix addition, multiplication and inversion is $\mathcal{O}(kn\log^q n)$, $q = 1, 2$ (see [3, 5]). Let I be an index set. To build an $\mathcal{H}$-matrix $M \in \mathbb{R}^{I\times I}$ one needs an admissible block partitioning (see Fig. 3) built on a block cluster tree $T_{I\times I}$ by means of an admissibility condition (see [3, 2]). The admissible block partitioning indicates which blocks can be approximated by low-rank matrices.

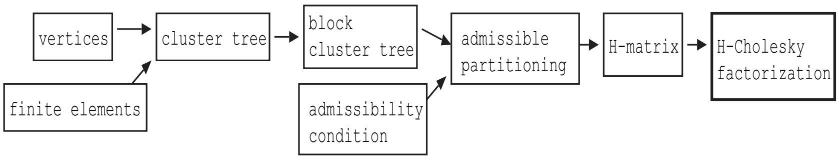


Fig. 3. The scheme of building an $\mathcal{H}$-matrix and its $\mathcal{H}$-Cholesky factorization.

Definition 1. *We define the set of* $\mathcal{H}$*-matrices with the maximal rank* k *as* $\mathcal{H}(T_{I\times I}, k) := \{M \in \mathbb{R}^{I\times I} \mid rank(M\,|_b) \le k,$ *for all admissible leaves* b *of* $T_{I\times I}\}$.

Suppose that $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix} = \begin{bmatrix} L_{11} & 0 \\ L_{21} & L_{22} \end{bmatrix} \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix}$. The algorithm that computes the $\mathcal{H}$-LU factorization (cf. [10, 1]) is the following:

1. compute L_{11} and U_{11} as $\mathcal{H}$-LU factorization of A_{11};
2. compute U_{12} from $L_{11}U_{12} = A_{12}$ (recursive block forward substitution);
3. compute L_{21} from $L_{21}U_{11} = A_{21}$ (recursive block backward substitution);
4. compute L_{22} and U_{22} as $\mathcal{H}$-LU factorization of $L_{22}U_{22} = A_{22} - L_{21}U_{12}$.

Note that all the steps are executed approximately via truncation to the class of $\mathcal{H}$-matrices.

4 Block $\mathcal{H}$-LU Preconditioner $\widetilde{W}_2$

Let us introduce the $\mathcal{H}$-Cholesky factorization of the following symmetric matrices

$$A_\varepsilon = \begin{bmatrix} A_{11} & \varepsilon A_{12} \\ \varepsilon A_{21} & \varepsilon A_{22} \end{bmatrix} \cong L_1 L_1^T =: \widetilde{W}_1,\; W_2 := \begin{bmatrix} A_{11} & 0 \\ 0 & \varepsilon A_{22} \end{bmatrix} \cong L_2 L_2^T =: \widetilde{W}_2. \quad (7)$$

$\mathcal{H}$-Cholesky factorization $L_1 L_1^T$ was successfully applied in [1, 9]. As a new preconditioner we use the $\mathcal{H}$-Cholesky factorization of W_2, which we denote by $\widetilde{W}_2$. Examples of the $\mathcal{H}$-Cholesky factors L_1 and L_2 are shown in Fig. 4.

Remark 1. Note that $\widetilde{W}_2^{-1/2} A_\varepsilon \widetilde{W}_2^{-1/2} = L_2^{-T} A_\varepsilon L_2^{-1}$, i.e., $\widetilde{W}_2^{-1/2} A_\varepsilon \widetilde{W}_2^{-1/2}$ is positive definite and symmetric (the same holds for $\widetilde{W}_1$). Thus, for solving the initial problem (5) one may apply the PCG method with preconditioner $\widetilde{W}_2$.

Lemma 1. *For a symmetric and positive definite matrix* $A = \begin{bmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{bmatrix}$ *and any vectors* u_1 *and* u_2 *of the respective size it holds that*

$$|(A_{12}u_2, u_1)| \leq (A_{11}u_1, u_1)^{1/2} \cdot (A_{22}u_2, u_2)^{1/2}.$$

Lemma 2. *For any* $u \in \mathbb{R}^n$ *we have* $(A_\varepsilon u, u) \leq 2(W_2 u, u)$ *with* W_2 *defined by (7).*

Proposition 1. *Let the component* u_2 *be discrete harmonic extension of* u_1 *into* $\Omega_c \subset \Omega$. *Then* $\exists c_1 > 0 \, : \, c_1(A_{11}u_1, u_1) \geq (A_{22}u_2, u_2)$ *with* c_1 *independent of* h.

Proof. This estimate corresponds to the case of the Laplace operator in Ω and can be verified by using standard properties of the harmonic elliptic extension operator and trace estimates. Applying Theorem 4.1.3 in [12], where we set $\Omega_1 = \Omega_l$ and $\Omega_2 = \Omega_c$, leads to the desired bound with constant c_1 depending only on the geometry.

Lemma 3. *Assume that Proposition 1 holds. Then* $\exists \varepsilon_0 \in (0, 1)$ *such that* $\forall \varepsilon \in (0, \varepsilon_0]$ *we have* $((A_\varepsilon - cW_2)u, u) \geq 0$ *with* $c = \frac{1-\varepsilon}{1+c_1\varepsilon}$, *where* c_1 *is defined in Proposition 1.*

Proof. We choose $c = \frac{1-\varepsilon}{1+c_1\varepsilon}$ and then apply Lemma 1 to obtain

$$\begin{aligned}
((A_\varepsilon - cW_2)u, u) &= (1-c)(A_{11}u_1, u_1) + \varepsilon(1-c)(A_{22}u_2, u_2) + 2\varepsilon(A_{12}u_2, u_1) \\
&\geq (1-c)(A_{11}u_1, u_1) + \varepsilon(1-c)(A_{22}u_2, u_2) - 2\varepsilon|(A_{12}u_2, u_1)| \\
&\geq (1-c)((A_{11}u_1, u_1) + \varepsilon(A_{22}u_2, u_2)) - \varepsilon((A_{11}u_1, u_1) + (A_{22}u_2, u_2)) \\
&\geq (1-c-\varepsilon)(A_{11}u_1, u_1) - c\varepsilon(A_{22}u_2, u_2).
\end{aligned}$$

Now Proposition 1 ensures

$$((A_\varepsilon - cW_2)u, u) \geq (A_{11}u_1, u_1)(1 - c - \varepsilon - c_1 c\varepsilon) = 0.$$

Thus, $((A_\varepsilon - cW_2)u, u) \geq 0$ holds with $c = \frac{1-\varepsilon}{1+c_1\varepsilon}$.

Notice that the constant $c = \frac{1-\varepsilon}{1+c_1\varepsilon}$ depends on the geometry of Ω.

In Fig. 4 (right) one can see two blocks on the first level of the $\mathcal{H}$-Cholesky factorization of W_2. The first block corresponds to the lipid layer Ω_l, the second one (with 8 subblocks) corresponds to 8 cells. The problems inside the cells can be treated in parallel.

Remark 2. The set of all nodal points in the lipid layer (Fig. 1 (right)) can be decomposed into 12 parts $\overline{\Omega}_l = \cup_{i=1}^{12} \Omega_{l,i}$, which will lead to further simplifications.

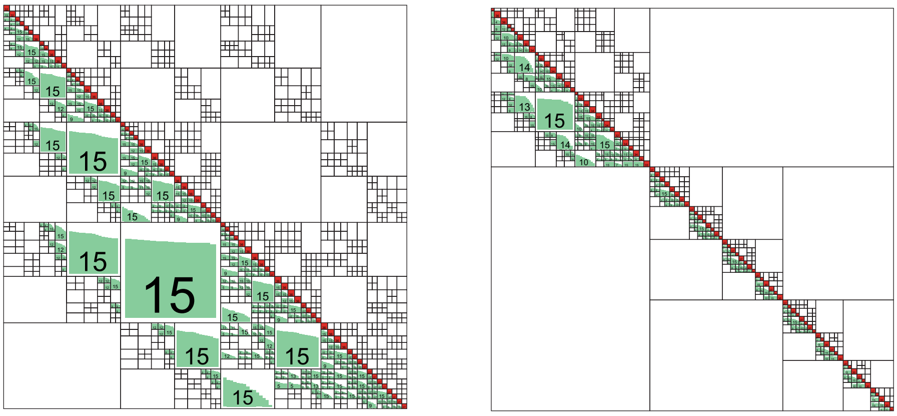


Fig. 4. $\mathcal{H}$-Cholesky factors of the global stiffness matrix A_ε (**left**) and of the block matrix W_2 (**right**). The dark blocks $\in \mathbb{R}^{36\times36}$ are dense matrices, the gray blocks are low-rank matrices and the white blocks are zero ones. The steps in the grey blocks show the decay of the singular values in a logarithmic scale. The numbers inside the subblocks indicate the local ranks.

5 Numerical Tests

In this Section we present numerical results for the 3D Dirichlet problem (see [7] for the 2D case). Figure 2 explains the discretization of the lipid layer. For simplicity the width of the lipid layer is chosen as h, but it can be a multiple of h.

Table 1 gives the theoretical estimates on the sequential and parallel complexities of $\widetilde{W}_1$ and $\widetilde{W}_2$.

Table 1. Computational complexities of the preconditioners $\widetilde{W}_1$ and $\widetilde{W}_2$. The number of processors is p. The number of degrees of freedom in the lipid layer is n_I (handled by one processor) and the number of dofs on each processor is $n_0 := \frac{n-n_I}{p-1}$.

	Sequential Complexity	Parallel Complexity
$\widetilde{W}_1$	$\mathcal{O}(n\log^2 n)$	$\mathcal{O}(n\log^2 n)$
$\widetilde{W}_2$	$\mathcal{O}(n_I\log^2 n_I)+\mathcal{O}((n-n_I)\log^2(n-n_I))$	$\max\{\mathcal{O}(n_I\log^2 n_I)+\mathcal{O}(n_0\log^2 n_0)\}$

Remark 3. The sparsity constant C_{sp} is an important $\mathcal{H}$-matrix parameter that effects all $\mathcal{H}$-matrix complexity estimates (see [3, 5]). The smaller C_{sp} the better complexity bound is. For instance, for the problem with 45^3 dofs $C_{sp}(A_\varepsilon) = 108$, while $C_{sp}(W_2) = 30$. For the model geometry with a larger number of cells the difference between the sparsity constants will be even more significant.

Table 2. Comparison of $\widetilde{W}_1$ and $\widetilde{W}_2$ in 3D, 40^3 dofs, $\|A_\varepsilon \mathbf{u} - \mathbf{c}\| = 10^{-8}$, $\varepsilon = 10^{-5}$.

rank k	time $t^{(1)}$, $t^{(2)}$	storage $S^{(1)}$, $S^{(2)}$	♯iter$^{(1)}$,$^{(2)}$
1	34.6, 18.7	2e+2, 1e+2	69, 99
2	81.3, 35	3.8e+2, 1.8e+2	46, 91
4	220.5, 81.5	7.5e+2, 3.5e+2	17, 60
6	565.7, 149	1.1e+3, 5.1e+2	11, 74

Table 2 illustrates storage requirements (denoted by $S^{(1)}$, $S^{(2)}$ and measured in MB) and computing times (denoted by $t^{(1)}$, $t^{(2)}$ and measured in sec) for the preconditioners $\widetilde{W}_1$ and $\widetilde{W}_2$, respectively, depending on the $\mathcal{H}$-matrix rank k. The columns $t^{(1)}$, $t^{(2)}$ contain the total computing times for setting up the preconditioners $\widetilde{W}_1$ and $\widetilde{W}_2$ and for performing the PCG iterations. The columns iter$^{(1)}$ and iter$^{(2)}$ show the number of PCG iterations in both cases (see [7] for more details). One can see that the preconditioner $\widetilde{W}_2$ requires less storage ($S^{(1)} > S^{(2)}$) and less computing time ($t^{(1)} > t^{(2)}$). Notice that the computation with a smaller rank k in the $\mathcal{H}$-matrix arithmetic (but with a larger number of iterations) leads to a better performance than in the case with a larger k (but with a smaller number of iterations). Table 3 illustrates linear-logarithmic scaling of the computational time and storage in the problem size n (with fixed maximal rank $k = 5$). Choosing the smaller maximal rank k leads to almost linear complexities. Table 4 shows the number of iterations and the computing times depending on the coefficient α. The number of iterations is relatively large since we use the low-rank $\mathcal{H}$-matrix approximation with $k = 1$. The computing time in the case of $\widetilde{W}_2$ is in a factor two smaller than in the case of $\widetilde{W}_1$. This factor is getting larger for problems with increasing number of cells.

Table 3. Dependence of the computing time and storage requirements on the problem size, $\|A_\varepsilon \mathbf{u} - \mathbf{c}\| = 10^{-8}$, max. rank= 5 (see Def. 1).

♯dofs	time $t^{(2)}$(sec.)	memory $S^{(2)}$(Mb)	♯iter$^{(2)}$
2200	0.27	5.4	33
15600	7.9	90	59
91100	119.4	1007	98

Table 4. The number of iterations and the computing times depending on the coefficient α for 40^3 dofs, $\|A_\varepsilon \mathbf{u} - \mathbf{c}\| = 10^{-8}$, $k = 1$.

ε	1	10^{-1}	10^{-2}	10^{-4}	10^{-6}	10^{-8}
♯ iter$^{(1)}$, iter$^{(2)}$	86, 89	77, 100	79, 113	79, 113	82, 116	85, 120
$t^{(1)}$, $t^{(2)}$ sec.	70, 33	67, 35	63, 37	65, 37	67, 37	67, 38

We conclude that the preconditioner $\widetilde{W_2}$ is efficient and robust with respect to the small coefficient $\alpha = \varepsilon$ characterising the skin problem (see Tables 1, 4). It requires less storage and computing time than $\widetilde{W_1}$ (see Tables 2, 3). The preconditioner $\widetilde{W_2}$ becomes more efficient for problems with increasing number of cells. The possible disadvantage of W_2 could be a relatively large number of PCG iterations, but it is compensated by their low computational cost (see Table 2).

Acknowledgement. The authors wish to thank Prof. Dr. Hackbusch for valuable comments as well as Dr. Börm and Dr. Grasedyck for the assistance in the usage of HLIB (see [2]).

References

[1] M. Bebendorf. Hierarchical LU decomposition-based preconditioners for BEM. *Computing*, 74(3):225–247, 2005.

[2] L. Grasedyck and S. Börm. $\mathcal{H}$-matrix library: www.hlib.org.

[3] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. I. Introduction to $\mathcal{H}$-matrices. *Computing*, 62(2):89–108, 1999.

[4] W. Hackbusch. Direct domain decomposition using the hierarchical matrix technique. In *Domain Decomposition Methods in Science and Engineering*, pages 39–50. Natl. Auton. Univ. Mex., México, 2003.

[5] W. Hackbusch and B.N. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic. II. Application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.

[6] W. Hackbusch, B.N. Khoromskij, and R. Kriemann. Direct Schur complement method by domain decomposition based on $\mathcal{H}$-matrix approximation. *Comput. Vis. Sci.*, 8(3-4):179–188, 2005.

[7] B.N Khoromskij and A. Litvinenko. Domain decomposition based $\mathcal{H}$-matrix preconditioner for the 2D and 3D skin problem. Technical Report 95, Max-Planck Institute for Math. in the Sciences, 2006.

[8] B.N. Khoromskij and G. Wittum. *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface*, volume 36 of *Lecture Notes in Computational Science and Engineering*. Springer-Verlag, Berlin, 2004.

[9] S. Le Borne and L. Grasedyck. $\mathcal{H}$-matrix preconditioners in convection-dominated problems. *SIAM J. Matrix Anal. Appl.*, 27(4):1172–1183, 2006.

[10] M. Lintner. The eigenvalue problem for the 2D Laplacian in $\mathcal{H}$-matrix arithmetic and application to the heat and wave equation. *Computing*, 72(3-4):293–323, 2004.

[11] A. Litvinenko. *Application of Hierarchical Matrices for Solving Multiscale Problems.* PhD thesis, Leipzig University, 2006.

[12] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford University Press, Oxford, 1999.

MINISYMPOSIUM 3: Domain Decomposition in Coupled Engineering Phenomena with Multiple Scales

Organizers: Richard Ewing[1], Oleg Iliev[2], and Raytcho D. Lazarov[1]

[1] Texas A&M University, Department of Mathematics, USA.
richard-ewing@tamu.edu, lazarov@math.tamu.edu
[2] Fraunhofer ITWM, FB Mathematik, Kaiserslautern, Germany.
iliev@itwm.fhg.de

The intent of the minisymposium was to discuss the state of the art and the perspectives in approximation and solution strategies related to Domain Decomposition methods for coupled phenomena in physics and engineering that involve multiple models and/or multiple space and time scales. Six from the presented eight talks were devoted to various aspects of the algorithms for solving multiscale problems. These works reflected the common roots of domain decomposition methods and some of the recent approaches for solving multiscale problems, such as Multiscale Finite Element Method, Multiscale Finite Volume Method, etc. The talks covered a wide spectrum of problems related to domain decomposition algorithms for coupled problems, construction and theoretical analysis of a number of algorithm for multiscale problems, and their applications to engineering and industrial problems. The last two talks were devoted to Discontinuous Galerkin Method, which is considered to be suitable for multiphysics problems because of its potential for coupling different discretizations PDE or system of PDEs, as well as for coupling different types of physical models.

The intensive discussions after the talks and during the breaks contributed to creating a nice working atmosphere and to productive exchange of new ideas.

Class of Preconditioners for Discontinuous Galerkin Approximations of Elliptic Problems

Paola F. Antonietti[1] and Blanca Ayuso[2]

[1] Dipartimento di Matematica, Università di Pavia, via Ferrata 7, 27100 Pavia, Italy. paola.antonietti@unipv.it
[2] Istituto di Matematica Applicata e Tecnologie Informatiche–CNR,via Ferrata 7, 27100 Pavia, Italy. blanca@imati.cnr.it

Summary. We present a class of Schwarz preconditioners for discontinuous Galerkin approximations of elliptic problems. We provide a unified framework for the construction and analysis of two-level methods which share the features of the classical Schwarz techniques for conforming finite element discretizations. Numerical experiments confirming the theoretical results are also included.

1 Introduction

Domain decomposition (DD) methods provide powerful preconditioners for the iterative solution of the large algebraic linear systems of equations that arise in finite element approximations of partial differential equations. Many DD algorithms can conveniently be described and analyzed as Schwarz methods, and, if on the one hand a general theoretical framework has been previously developed for classical conforming discretizations (see, e.g., [7]), on the other hand, only few results can be found for discontinuous Galerkin (DG) approximations (see, e.g., [6, 4, 2, 1]). Based on discontinuous finite element spaces, DG methods have become increasing popular thanks to their great flexibility for providing discretizations on matching and non-matching grids and their high degree of locality. In this paper we present and analyze, in the unified framework based on the *flux formulation* proposed in [3], a class of Schwarz preconditioners for DG approximations of second order elliptic problems. Schwarz methods for a wider class of DG discretizations are studied in [2, 1]. The issue of preconditioning non-symmetric DG approximations is also discussed. Numerical experiments to asses the performance of the proposed preconditioners and validate our convergence results are presented.

2 Discontinuous Galerkin Methods for Elliptic Problems

We consider the following model problem

$$-\Delta u = f \quad \text{in } \Omega\,, \quad u = 0 \quad \text{on } \partial\Omega\,, \tag{1}$$

where $\Omega \subset \mathbb{R}^d$, $d = 2, 3$, is a convex polygon or polyhedron and f a given function in $L^2(\Omega)$. Let $\mathcal{T}_h$ be a shape-regular *quasi-uniform* partition of Ω into disjoint open elements T (with diameter h_T), where each T is the affine image of a fixed master element $\widehat{T}$, i.e., $T = F_T(\widehat{T})$, and where $\widehat{T}$ is either the open unit d-simplex or the d-hypercube in $\mathbb{R}^d$, $d = 2, 3$. We define the mesh size h by $h = \max_{T \in \mathcal{T}_h} h_T$. We denote by $\mathscr{E}^I$ and $\mathscr{E}^B$ the sets of all interior and boundary faces of $\mathcal{T}_h$, respectively, and set $\mathscr{E} = \mathscr{E}^I \cup \mathscr{E}^B$. For a given approximation order $\ell_h \geq 1$, we define the discontinuous finite element spaces $V_h = \{v \in L^2(\Omega) : v|_T \circ F_T \in \mathcal{M}^{\ell_h}(\widehat{T}) \ \forall T \in \mathcal{T}_h\}$ and $\boldsymbol{\Sigma}_h = [V_h]^d$, where $\mathcal{M}^{\ell_h}(\widehat{T})$ is either the space of polynomials of degree at most ℓ_h on $\widehat{T}$, if $\widehat{T}$ is the reference d-simplex, or the space of polynomials of degree at most ℓ_h in each variable on $\widehat{T}$, if $\widehat{T}$ is the reference d-hypercube.

For any internal face $e \in \mathscr{E}^I$ shared by two adjacent elements $T^{\pm}$ with outward normal unit vectors $\mathbf{n}^{\pm}$, we define the jump and *weighted* average operators, with $\delta \in [0, 1]$, by:

$$\begin{aligned} [\![\boldsymbol{\tau}]\!] &= \boldsymbol{\tau}^+ \cdot \mathbf{n}^+ + \boldsymbol{\tau}^- \cdot \mathbf{n}^-, & [\![v]\!] &= v^+\mathbf{n}^+ + v^-\mathbf{n}^-, & e &\in \mathscr{E}^I, \\ \{\!\{\boldsymbol{\tau}\}\!\}_\delta &= \delta\boldsymbol{\tau}^+ + (1-\delta)\boldsymbol{\tau}^-, & \{\!\{\mathbf{v}\}\!\}_\delta &= \delta\mathbf{v}^+ + (1-\delta)\mathbf{v}^-, & e &\in \mathscr{E}^I, \end{aligned} \tag{2}$$

where $\boldsymbol{\tau}^{\pm}$ and $v^{\pm}$ denote the traces on $\partial T^{\pm}$ taken from the interior of $T^{\pm}$ of the (regular enough) functions $\boldsymbol{\tau}$ and v. On a boundary face $e \in \mathscr{E}^B$, we set

$$[\![\boldsymbol{\tau}]\!] = \boldsymbol{\tau} \cdot \mathbf{n}, \quad [\![v]\!] = v\,\mathbf{n}, \quad \{\!\{\boldsymbol{\tau}\}\!\}_\delta = \boldsymbol{\tau}\,, \quad \{\!\{v\}\!\}_\delta = v, \quad e \in \mathscr{E}^B. \tag{3}$$

For $\delta = 1/2$ we write $\{\!\{\cdot\}\!\}$ in lieu of $\{\!\{\cdot\}\!\}_{1/2}$.

The DG discretization based on the flux formulation proposed in [3] is defined by introducing an auxiliary variable $\boldsymbol{\sigma} = \nabla u$ and rewriting problem (1) as a first order system of equations. Further elimination of $\boldsymbol{\sigma}$, gives the *primal formulation* of DG methods:

$$\text{find } u_h \in V_h \text{ such that } A_h(u_h, v_h) = \int_\Omega f v_h \,\mathrm{d}x \qquad \forall\, v_h \in V_h\,. \tag{4}$$

Adopting the convention $\int_{\mathscr{E}} v_h \,\mathrm{d}s = \sum_{e \in \mathscr{E}} \int_e v_h \,\mathrm{d}s$, $A_h(\cdot, \cdot)$ is given by

$$\begin{aligned} A_h(u_h, v_h) = &\int_\Omega \nabla_h u_h \cdot \nabla_h v_h \,\mathrm{d}x + \int_{\mathscr{E}} [\![\widehat{u} - u_h]\!] \cdot \{\!\{\nabla_h v_h\}\!\} \,\mathrm{d}s \\ &+ \int_{\mathscr{E}^I} \{\!\{\widehat{u} - u_h\}\!\} [\![\nabla_h v_h]\!] \,\mathrm{d}s - \int_{\mathscr{E}} \{\!\{\widehat{\boldsymbol{\sigma}}\}\!\} \cdot [\![v_h]\!] \,\mathrm{d}s - \int_{\mathscr{E}^I} [\![\widehat{\boldsymbol{\sigma}}]\!] \{\!\{v_h\}\!\} \,\mathrm{d}s\,, \end{aligned} \tag{5}$$

where $\widehat{u}$ and $\widehat{\boldsymbol{\sigma}}$ are the scalar and vector numerical fluxes and ∇_h denotes the elementwise application of the operator ∇. By defining the numerical fluxes $\widehat{u}$ and $\widehat{\boldsymbol{\sigma}}$ as suitable linear combinations of averages and jumps of u_h and $\boldsymbol{\sigma}_h$, we obtain different DG methods (see Table 1 for the choices considered in this work). The stability is achieved by penalizing the jumps of u_h over each face $e \in \mathscr{E}$. Therefore, $A_h(\cdot,\cdot)$ contains the stabilization term $\mathcal{S}^h(\cdot,\cdot)$ defined by

$$\mathcal{S}^h(u,v) = \sum_{e\in\mathscr{E}} \int_e \alpha\, h_e^{-1} [\![u]\!] \cdot [\![v]\!] \, \mathrm{d}s \quad \forall\, u,v \in V_h\,,$$

where h_e is the diameter of the face $e \in \mathscr{E}$. Here $\alpha \geq 1$ is a parameter (independent of the mesh size) that, for all but the LDG and NIPG methods, has to be chosen large enough to ensure the coercivity of the bilinear form. From now on, we drop the subindex h from the finite element functions. In matrix notation, problem (4) is written as the linear system $\mathbf{Au} = \mathbf{f}$.

Table 1. Numerical fluxes on interior faces.

Method	$\widehat{u}(u_h)$	$\widehat{\boldsymbol{\sigma}}(\boldsymbol{\sigma}_h, u_h)$	Symmetry
SIPG	$\{\!\{u_h\}\!\}$	$\{\!\{\nabla_h u_h\}\!\} - \alpha\, h_e^{-1} [\![u_h]\!]$	Yes
SIPG(δ)	$\{\!\{u_h\}\!\}_{(1-\delta)}$	$\{\!\{\nabla_h u_h\}\!\}_\delta - \alpha\, h_e^{-1} [\![u_h]\!]$	Yes
NIPG	$\{\!\{u_h\}\!\} + [\![u_h]\!] \cdot \mathbf{n}_T$	$\{\!\{\nabla_h u_h\}\!\} - \alpha\, h_e^{-1} [\![u_h]\!]$	No
IIPG	$\{\!\{u_h\}\!\} + 1/2\, [\![u_h]\!] \cdot \mathbf{n}_T$	$\{\!\{\nabla_h u_h\}\!\} - \alpha\, h_e^{-1} [\![u_h]\!]$	No
LDG	$\{\!\{u_h\}\!\} - \boldsymbol{\beta} \cdot [\![u_h]\!]$	$\{\!\{\boldsymbol{\sigma}_h\}\!\} + \boldsymbol{\beta} \cdot [\![\boldsymbol{\sigma}_h]\!] - \alpha\, h_e^{-1} [\![u_h]\!]$	Yes

Remark 1. The results we present here apply to more general elliptic equations with possibly smooth variable coefficients, and remain valid for more general partitions (non necessarily matching).

3 Non-Overlapping Schwarz Methods

We consider three level of *nested* partitions of the domain Ω satisfying the previous assumptions: a subdomain partition $\mathcal{T}_{N_s}$ made of N_s non-overlapping subdomains Ω_i, a coarse partition $\mathcal{T}_H$ (with mesh size H) and a fine partition $\mathcal{T}_h$ (with mesh size h). For each subdomain $\Omega_i \in \mathcal{T}_{N_s}$ we denote by $\mathscr{E}_i$ the set of all faces of $\mathscr{E}$ belonging to $\overline{\Omega}_i$, and set $\mathscr{E}_i^I = \{e \in \mathscr{E}_i \,:\, e \subset \Omega_i\}$, $\mathscr{E}_i^B = \{e \in \mathscr{E}_i \,:\, e \subset \partial\Omega_i \cap \partial\Omega\}$. The set of all (internal) faces belonging to the skeleton of the subdomain partition will be denoted by Γ, i.e., $\Gamma = \bigcup_{i=1}^{N_s} \Gamma_i$ with $\Gamma_i = \{e \in \mathscr{E}_i^I \,:\, e \subset \partial\Omega_i\}$. For $i = 1, \ldots, N_s$, we define the local spaces by $V_h^i = \{u \in L^2(\Omega_i) \,:\, v|_T \circ F_T \in \mathcal{M}^{\ell_h}(\widehat{T}) \;\forall T \in \mathcal{T}_h,\, T \subset \Omega_i\}$ and $\mathbf{\Sigma}_h^i = [V_h^i]^d$, and the *prolongation* operators $R_i^T \,:\, V_h^i \longrightarrow V_h$ as the classical inclusion operators from V_h^i to V_h. For vector-valued functions R_i^T is defined componentwise. We observe that $V_h = R_1^T V_h^1 \oplus \ldots \oplus R_{N_s}^T V_h^{N_s}$ and $\mathbf{\Sigma}_h = R_1^T \mathbf{\Sigma}_h^1 \oplus \cdots \oplus R_{N_s}^T \mathbf{\Sigma}_h^{N_S}$. The restriction operators R_i, are defined as the transpose of R_i^T with respect to the L^2–inner product.

Local solvers: we consider the DG approximation of the problem:

$$-\Delta u_i = f|_{\Omega_i} \quad \text{in } \partial\Omega_i, \quad u_i = 0 \quad \text{on } \Omega_i \ .$$

In view of (5), the local bilinear forms $A_i : V_h^i \times V_h^i \longrightarrow \mathbb{R}$ are defined by

$$A_i(u_i, v_i) = \int_{\Omega_i} \nabla_h u_i \cdot \nabla_h v_i \, \mathrm{d}x + \int_{\mathscr{E}_i} [\![\widehat{u}_i - u_i]\!] \cdot \{\!\{\nabla_h v_i\}\!\} \, \mathrm{d}s$$
$$+ \int_{\mathscr{E}_i^I} \{\!\{\widehat{u}_i - u_i\}\!\} [\![\nabla_h v_i]\!] \, \mathrm{d}s - \int_{\mathscr{E}_i} \{\!\{\widehat{\boldsymbol{\sigma}}_i\}\!\} \cdot [\![v_i]\!] \, \mathrm{d}s - \int_{\mathscr{E}_i^I} [\![\widehat{\boldsymbol{\sigma}}_i]\!] \{\!\{v_i\}\!\} \, \mathrm{d}s,$$

where $\widehat{u}_i$ and $\widehat{\boldsymbol{\sigma}}_i$ are the *local* numerical fluxes. On $e \in \mathscr{E}_i^I$, $\widehat{u}_i$ and $\widehat{\boldsymbol{\sigma}}_i$ are defined as the numerical fluxes $\widehat{u}$, $\widehat{\boldsymbol{\sigma}}$ of the global DG method on interior faces, and on $e \in \mathscr{E}_i^B \cup \Gamma_i$ as $\widehat{u}$ and $\widehat{\boldsymbol{\sigma}}$ on boundary faces. Note that, each $e \in \Gamma_i$ is a boundary face for the local partition but an interior face for the global partition. From the definition of $A_i(\cdot,\cdot)$, and taking into account the different definition (2)-(3) of the average operator on interior and boundary faces (implying that $\{\!\{R_i^T v_i\}\!\}_\delta = \delta v_i$ *but* $\{\!\{v_i\}\!\}_\delta = v_i$ on $e \in \Gamma_i$), it follows that we are using *approximate* local solvers, that is, $A_h(R_i^T u_i, R_i^T u_i) \leq \omega A_i(u_i, u_i)$, with $\omega \neq 1$.

Coarse solver: for all $u_0, v_0 \in V_h^0 = \{v_H \in L^2(\Omega) : v_H|_T \in \mathcal{M}^{\ell_H}(T) \ \forall T \in \mathcal{T}_H\}$, with $0 \leq \ell_H \leq \ell_h$, the coarse solver $A_0 : V_h^0 \times V_h^0 \longrightarrow \mathbb{R}$ is defined by $A_0(u_0, v_0) = A_h(R_0^T u_0, R_0^T v_0)$, where R_0^T is the classical injection operator.

Remark 2. We notice that, in all the previously proposed Schwarz methods (see, e.g., [6, 4]) *exact* local solvers were employed.

3.1 Schwarz Methods: Variational and Algebraic Formulations

For $i = 0, \ldots, N_s$, and for all $v_i \in V_h^i$, we define the projection operators $\widetilde{P}_i : V_h \longrightarrow V_h^i$ by $A_i(\widetilde{P}_i u, v_i) = A_h(u, R_i^T v_i)$ and set $P_i = R_i^T \widetilde{P}_i : V_h \longrightarrow V_h$. The additive and multiplicative Schwarz operators we consider are defined by

$$P_{ad} = \sum_{i=0}^{N_s} P_i, \quad P_{mu} = I - (I - P_{N_s})(I - P_{N_s - 1}) \cdots (I - P_0),$$

respectively, where $I : V_h \longrightarrow V_h$ is the identity operator. We also define the error propagation operator $E_{N_s} = (I - P_{N_s})(I - P_{N_s-1}) \cdots (I - P_0)$ and observe that $P_{mu} = I - E_{N_s}$. The Schwarz methods can be written as the product of a suitable preconditioners, namely $\mathbf{B}_{ad}$ or $\mathbf{B}_{mu}$, and $\mathbf{A}$. In fact, for $i = 0, \ldots, N_s$, it is straightforward to note that the matrix representation of the projection operators P_i, is given by $\mathbf{P}_i = \mathbf{R}_i^T \mathbf{A}_i^{-1} \mathbf{R}_i \mathbf{A}$. Then,

$$\mathbf{P}_{ad} = \sum_{i=0}^{N_s} \mathbf{P}_i = \mathbf{B}_{ad}\mathbf{A}, \qquad \mathbf{P}_{mu} = \mathbf{I} - (\mathbf{I} - \mathbf{P}_{N_s}) \cdots (\mathbf{I} - \mathbf{P}_0) = \mathbf{B}_{mu}\mathbf{A}.$$

The additive Schwarz operator P_{ad} is symmetric for all symmetric DG approximations, while, the multiplicative operator P_{mu} is non symmetric (see [1] for a symmetrized version). Therefore, suitable iterative methods have to be used for solving the resulting linear systems: for the former case we use the *conjugate gradient* method while for the latter case we use the *generalized minimal residual* (GMRES) linear solver.

4 Convergence Results

In this section we present the convergence results for the proposed two-level Schwarz methods. We refer to [2, 1] for their proofs and further discussions on the convergence analysis. In what follows N_c denotes the maximum number of adjacent subdomains a given subdomain can have, and C is a positive constant independent of the mesh size.

Theorem 1. *Let $A_h(\cdot,\cdot)$ be the bilinear form of a symmetric DG method given in Table 1. Then, the condition number of P_{ad}, $\kappa(P_{ad})$, satisfies*

$$\kappa(P_{\mathrm{ad}}) \leq C\,\alpha\,\frac{H}{h}(1+\omega(1+N_c))\,. \tag{6}$$

Remark 3. Note that Theorem 1 shows that $\kappa(P_{\mathrm{ad}})$ depends linearly on the penalty parameter α.

The multiplicative operator is non-symmetric and in Theorem 2, we show that the energy norm of the error propagation operator is strictly less than one.

Theorem 2. *Let $A_h(\cdot,\cdot)$ be the bilinear form of a symmetric DG method given in Table 1. Then, there exists $\widetilde{\alpha} > 0$ such that if $\alpha \geq \widetilde{\alpha}$*

$$\|E_{N_s}\|_A^2 = \sup_{\substack{u\in V_h\\ u\neq 0}} \frac{A_h(E_{N_s}u, E_{N_s}u)}{A_h(u,u)} \leq 1 - \frac{2-\omega}{C\,\alpha\,(1+2\,\omega^2(N_c+1)^2)}\frac{h}{H} < 1.$$

Theorem 2 also guarantees that the multiplicative Schwarz method can be accelerated with the GMRES linear solver (see [5]).

Remark 4. As in the classical Schwarz theory, our convergence result for P_{mu} relies upon the hypothesis that $\omega \in (0,2)$. Since we are using approximate local solvers, we need a technical assumption on the size of the penalty parameter to guarantee $\omega \in (0,2)$. Nevertheless, we wish to stress that the assumed size of $\widetilde{\alpha}$ is moderate (see [1] for details).

Remark on Schwarz methods for the non-symmetric NIPG and IIPG approximations. In Table 2, we numerically demonstrate that the minimum eigenvalue of the *symmetric part* of the additive and multiplicative operators, denoted by $\lambda_{\min}(P_{ad})$ and $\lambda_{\min}(P_{mu})$, respectively, might be negative. As a consequence, the [5] GMRES convergence theory cannot be applied to explain the observed optimal performance of the proposed preconditioners (see Sect. 5).

Table 2. NIPG method ($\alpha = 1$) : $\ell_h = \ell_H = 1$, $N_s = 16$, Cartesian grids. Minimum eigenvalue of the *symmetric part* of P_{ad} (left) and P_{mu} (right).

$H \downarrow$ $h \rightarrow$	h_0	$h_0/2$	$h_0/4$	$h_0/8$
H_0	0.06	-0.16	-0.31	-0.40
$H_0/2$	0.64	0.01	-0.26	-0.40
$H_0/4$	-	0.63	-0.02	-0.27
$H_0/8$	-	-	0.62	-0.05

(a) $\lambda_{\min}(P_{ad})$

$H \downarrow$ $h \rightarrow$	h_0	$h_0/2$	$h_0/4$	$h_0/8$
H_0	0.16	-0.09	-0.27	-0.38
$H_0/2$	1.00	0.01	-0.21	-0.38
$H_0/4$	-	1.00	0.09	-0.20
$H_0/8$	-	-	1.00	-0.03

(b) $\lambda_{\min}(P_{mu})$

5 Numerical Results

We take $\Omega = (0,1) \times (0,1)$ and we choose f so that the exact solution of problem (1) (with non-homogeneous boundary conditions) is given by $u(x,y) = \exp(xy)$. The subdomain partitions consist of N_s squares, $N_s = 4, 16$ (see Fig. 1 for $N_s = 4$). We consider both matching and non-matching Cartesian grids (see Fig. 1 where the initial coarse and fine non-matching grids are depicted. The corresponding matching grids are obtained by gluing together all the elements that have at least a hanging-node). We denote by H_0 and h_0 the corresponding initial coarse and fine mesh sizes, respectively, and we consider n successive global uniform refinements of these initial grids so that the resulting mesh sizes are $H_n = H_0/2^n$ and $h_n = h_0/2^n$, with $n = 0, 1, 2, 3$. The tolerance is set to 10^{-9}.

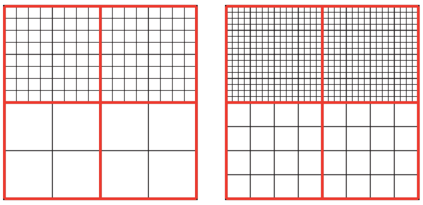


Fig. 1. Sample of a $N_s = 4$ subdomain partition of $\Omega = (0,1) \times (0,1)$ with the initial coarse (left) and fine (right) non-matching meshes.

We first address the scalability of the proposed additive Schwarz method, i.e., the independence of the convergence rate of the number of subdomains. In Table 3 we compare the condition number estimates for the SIP method ($\alpha = 10$) with $\ell_h = \ell_H = 1$ obtained on non-matching Cartesian grids (see Fig. 1) with $N_s = 4, 16$. As stated in Theorem 1, our preconditioner seems to be insensitive on the number of subdomains, and the expected convergence rates are clearly achieved.

Table 3. SIPG method ($\alpha = 10$): $\ell_h = \ell_H = 1$, non-matching Cartesian grids.

$H \downarrow\ h \rightarrow$	h_0	$h_0/2$	$h_0/4$	$h_0/8$
H_0	31.4	65.9	137.2	277.8
$H_0/2$	6.3	32.8	67.1	137.0
$H_0/4$	-	6.3	33.0	67.1
$H_0/8$	-	-	6.4	33.0
$\kappa(\mathbf{A})$	4.3e3	1.7e4	7.0e4	2.8e5

(a) $\kappa(P_{ad})$: $N_s = 4$

$H \downarrow\ h \rightarrow$	h_0	$h_0/2$	$h_0/4$	$h_0/8$
H_0	29.3	63.3	133.9	272.9
$H_0/2$	6.1	31.5	65.5	135.8
$H_0/4$	-	6.4	32.8	66.9
$H_0/8$	-	-	6.4	33.0
$\kappa(\mathbf{A})$	4.3e3	1.7e4	7.0e4	2.8e5

(b) $\kappa(P_{ad})$: $N_s = 16$

In Table 4 we compare the GMRES iteration counts obtained with our additive and multiplicative Schwarz methods. More precisely, the results reported in Table 4 have been obtained on Cartesian grids with the LDG method ($\alpha = 1$, $\beta = (0.5, 0.5)^T$), by using $\ell_h = 2$, $\ell_H = 1$ and $N_s = 16$. The crosses in the last row of Table 4 mean that the GMRES fails to converge due to its large memory requirements. In both cases we observe optimal convergence rates (we note however, that for the multiplicative preconditioner, the hypothesis on the size of α required in Theorem 2 is not satisfied).

Table 4. LDG method ($\alpha = 1$, $\beta = (0.5, 0.5)^T$): $\ell_h = 2$, $\ell_H = 1$, Cartesian grids.

$H \downarrow\ h \rightarrow$	h_0	$h_0/2$	$h_0/4$	$h_0/8$
H_0	49	68	95	128
$H_0/2$	33	46	64	88
$H_0/4$	-	33	47	65
$H_0/8$	-	-	34	48
#iter($\mathbf{A}$)	112	210	403	x

(a) $\mathbf{B}_{ad}\mathbf{Au} = \mathbf{B}_{ad}\mathbf{f}$: $N_s = 16$

$H \downarrow\ h \rightarrow$	h_0	$h_0/2$	$h_0/4$	$h_0/8$
H_0	22	30	40	53
$H_0/2$	14	17	23	32
$H_0/4$	-	12	16	21
$H_0/8$	-	-	10	13
#iter($\mathbf{A}$)	112	210	403	x

(b) $\mathbf{B}_{mu}\mathbf{Au} = \mathbf{B}_{mu}\mathbf{f}$: $N_s = 16$

Finally, we present some numerical computations carried out with the non-symmetric NIPG method ($\alpha = 1$). In Table 5 we compared the GMRES iteration counts obtained with $\ell_h = \ell_H = 1$ on Cartesian grids and by preconditioning with the proposed additive and multiplicative Schwarz preconditioners. Clearly, the GMRES applied to the preconditioned systems converges in a finite number of steps and, as in the symmetric case, the iteration counts remain unchanged whenever we decrease both the fine and the coarse mesh keeping their ratio constant. In all the cases addressed, the multiplicative Schwarz method seems to be approximately twice faster than the additive preconditioner.

Table 5. NIPG ($\alpha = 1$): GMRES iteration counts, $\ell_h = \ell_H = 1$, Cartesian grids.

$H \downarrow\ h \rightarrow$	h_0	$h_0/2$	$h_0/4$	$h_0/8$
H_0	25	26	29	36
$H_0/2$	14	21	24	28
$H_0/4$	-	14	20	23
$H_0/8$	-	-	14	19
#iter($\mathbf{A}$)	33	61	117	227

(a) $\mathbf{B}_{ad}\mathbf{Au} = \mathbf{B}_{ad}\mathbf{f}$: $N_s = 16$

$H \downarrow\ h \rightarrow$	h_0	$h_0/2$	$h_0/4$	$h_0/8$
H_0	12	13	16	20
$H_0/2$	1	9	11	14
$H_0/4$	-	1	8	10
$H_0/8$	-	-	1	7
#iter($\mathbf{A}$)	33	61	117	227

(b) $\mathbf{B}_{mu}\mathbf{Au} = \mathbf{B}_{mu}\mathbf{f}$: $N_s = 16$

References

[1] P.F. Antonietti and B. Ayuso. Multiplicative Schwarz methods for discontinuous Galerkin approximations of elliptic problems. Technical report, IMATI-CNR 10-PV, 2006. Submitted to *Math. Model. Numer. Anal.*

[2] P.F. Antonietti and B. Ayuso. Schwarz domain decomposition preconditioners for discontinuous Galerkin approximations of elliptic problems: non-overlapping case. *Math. Model. Numer. Anal.*, 41(1):21–54, 2007.

[3] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001/02.

[4] S.C. Brenner and K. Wang. Two-level additive Schwarz preconditioners for C^0 interior penalty methods. *Numer. Math.*, 102(2):231–255, 2005.

[5] S.C. Eisenstat, H.C. Elman, and M.H. Schultz. Variational iterative methods for nonsymmetric systems of linear equations. *SIAM J. Numer. Anal.*, 20(2):345–357, 1983.

[6] X. Feng and O.A. Karakashian. Two-level additive Schwarz methods for a discontinuous Galerkin approximation of second order elliptic problems. *SIAM J. Numer. Anal.*, 39(4):1343–1365, 2001.

[7] A. Toselli and O.B. Widlund. *Domain Decomposition Methods—Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, 2005.

Upscaling of Transport Equations for Multiphase and Multicomponent Flows

Richard Ewing[1], Yalchin Efendiev[1], Victor Ginting[2], and Hong Wang[3]

1. Department of Mathematics and Institute for Scientific Computation,
Texas A & M University, College Station, TX 77843-3404, USA,
richard-ewing@tamu.edu, efendiev@math.tamu.edu
2. Department of Mathematics, Colorado State University, Fort Collins, CO
80523-1874, USA, ginting@math.colostate.edu
3. Department of Mathematics, University of South Carolina, Columbia, SC 29208,
USA, hwang@math.sc.edu

Summary. In this paper we discuss upscaling of immiscible multiphase and miscible multicomponent flow and transport in heterogeneous porous media. The discussion presented in the paper summarizes the results of in *Upscaled Modeling in Multiphase Flow Applications* by Ginting et al. (2004) and in *Upscaling of Multiphase and Multicomponent Flow* by Ginting et al. (2006). Perturbation approaches are used to upscale the transport equation that has hyperbolic nature. Our numerical results show that these upscaling techniques give an improvement over the existing upscaled models which ignore the subgrid terms.

1 Introduction

The high degree of variability and multiscale nature of formation properties such as permeability pose significant challenges for subsurface flow modeling. Upscaling procedures are commonly applied to solve flow and transport equations in practice. On the fine (fully resolved) scale, the subsurface flow and transport of N components can be described in terms of an elliptic (for incompressible systems) pressure equation coupled to a sequence of $N-1$ hyperbolic (in the absence of dispersive and capillary pressure effects) conservation laws. Although there are various technical issues associated with subgrid models for the pressure equation, the lack of robustness of existing coarse scale models is largely due to the treatment of the hyperbolic transport equations. In this paper, we discuss the use of perturbation approaches for correcting the existing upscaled models for transport equations. Two-phase immiscible flow as well as miscible two-component flow are considered.

Previous approaches for the coarse scale modeling of transport in heterogeneous oil reservoirs include the use of pseudo relative permeabilities [2], the application of nonuniform or flow-based coarse grids [5], and the use of volume averaging and higher moments [7, 6]. Our methodology for subgrid upscaling of the hyperbolic (or convection dominated) equations uses volume averaging techniques. In particular, a perturbation analysis is employed to derive the macrodispersion that represents the effects of subgrid heterogeneities. The macrodispersion, in particular, can be written as time integration of a covariance between the velocity fluctuations and fine scale quantity that represents the length of fine scale trajectories. For the computation of fine scale quantities, we use detailed information that is contained in multiscale basis functions. We note that the resulting macrodispersion depends on the saturation due to the functional dependence of the velocities on it. Thus, a mere use of this macro-dispersion model would require saving the velocities for each time. We discuss a procedure to overcome the aforementioned impracticality by proposing a recursive relation relating the length of fine scale trajectories to the velocities.

2 Fine and Coarse Models

In this section, we briefly present mathematical models for two-phase immiscible and two-component miscible flow and transport. Because of the similarities of the governing equations, we present both models using the same equations (with some abuse of notations):

$$\begin{aligned} \nabla \cdot v &= q \\ S_t + v \cdot \nabla f(S) = (\tilde{S} - f(S))q, \quad v &= -d(S)k(x)\nabla p, \end{aligned} \tag{1}$$

where p is the pressure, S is the saturation (or concentration), $k(x)$ is a heterogeneous permeability field, v is the velocity field and q is the source term, and $\tilde{S}$ is the given saturation at the source term locations. The system is subject to some initial and boundary conditions. We will discuss upscaling techniques for (1). In further discussions, we refer to the first equation as the pressure equation (p is the pressure) and to the second equation as the saturation equation (S is the saturation or concentration).

For miscible two-component flow, $d(S) = \frac{1}{\mu(S)}$, where $\mu(S)$ is the viscosity function and has the form $\mu(S) = \frac{\mu(0)}{\left(1-S+M^{\frac{1}{4}}S\right)^4}$, and $f(S) = S$, where S is the concentration (will be referred as saturation in later discussions). For immiscible displacement of two-phase flow,

$$d(S) = \frac{k_{r1}(S)}{\mu_1} + \frac{k_{r2}(S)}{\mu_2}, \quad f(S) = \frac{k_{r1}(S)/\mu_1}{k_{r1}(S)/\mu_1 + k_{r2}(S)/\mu_2}. \tag{2}$$

Here $k_{ri}(S)$ ($i = 1, 2$) are relative permeabilities of phase i (e.g., water and oil), μ_i are viscosities of phase i.

Previous approaches for upscaling such systems are discussed by many authors; e.g., [1]. In most upscaling procedures, the coarse scale pressure equation is of the same form as the fine scale equation, but with an equivalent grid block permeability tensor k^* replacing k. For a given coarse scale grid block, the tensor k^* is generally computed through the solution of the pressure equation over the local fine scale region corresponding to the particular coarse block [4]. Coarse grid k^* computed in this manner has been shown to provide accurate solutions to the coarse grid pressure equation. We note that some upscaling procedures additionally introduce a different coarse grid functionality for d, though this does not appear to be essential in our formulation.

In this work, the proposed coarse model is the upscaling of the pressure equation to obtain the velocity field on the coarse grid and use it in saturation equation to resolve the concentration on the coarse grid. We will use multiscale finite element method. The key idea of the method is the construction of basis functions on the coarse grids such that these basis functions capture the small scale information on each of these coarse grids. The method that we use follows its finite element counterpart presented in [9]. The basis functions are constructed from the solution of the leading order homogeneous elliptic equation on each coarse element with some specified boundary conditions. Thus, if we consider a coarse element K that has d vertices, the local basis functions ϕ^i, $i = 1, \ldots, d$, are set to satisfy the following elliptic problem:

$$-\nabla \cdot (k \cdot \nabla \phi^i) = 0 \quad \text{in } K, \quad \phi^i = g^i \quad \text{on } \partial K, \tag{3}$$

for some functions g^i defined on the boundary of the coarse element K. Hou et al. [9] have demonstrated that a careful choice of boundary condition would guarantee the performance of the basis functions to incorporate the local information and, hence improve the accuracy of the method. In this paper, the function g^i for each i varies linearly along ∂K. Thus, for example, in case of a constant diagonal tensor, the solution of (3) would be a standard linear/bilinear basis function. We note that as usual we require $\phi^i(\xi_j) = \delta_{ij}$. Finally, a nodal basis function associated with the vertex ξ in the domain Ω are constructed from the combination of the local basis functions that share this ξ and zero elsewhere. These nodal basis functions are denoted by $\{\psi_\xi\}_{\xi \in Z_h^0}$.

We denote by V^h the space of our approximate pressure solution which is spanned by the basis functions $\{\psi_\xi\}_{\xi \in Z_h^0}$. A statement of mass conservation on a control volume V_ξ is formed from pressure equation, where now the approximate solution is written as a linear combination of the basis functions. To be specific, the problem now is to seek $p^h \in V^h$ with $p^h = \sum_{\xi \in Z_h^0} p_\xi \psi_\xi$ such that

$$\int_{\partial V_\xi} d(S) k \nabla p^h \cdot n \, dl = \int_{V_\xi} q \, dA, \tag{4}$$

for every control volume $V_\xi \subset \Omega$. Here n defines the normal vector on the boundary of the control volume, ∂V_ξ and S is the fine scale saturation field.

For the saturation equation, we will consider two different coarse models. We will present these models based on a perturbation technique, where the saturation, S, and the velocity v, on the fine scale are assumed to be the sum of their volume-averaged and fluctuating components,

$$v = \overline{v} + v', \quad S = \overline{S} + S', \quad f = \overline{f} + f'. \tag{5}$$

Here the overbar quantities designate the volume average of fine scale quantities over coarse blocks. For simplicity we will assume that the coarse blocks are rectangular which allows us to state that (cf [11]) $\overline{\nabla f} = \nabla \overline{f}$. Substituting (5) into the saturation equation and averaging over coarse blocks we obtain

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla \overline{f} + \overline{v' \cdot \nabla f'} = (\tilde{S} - \overline{f})q. \tag{6}$$

The term $\overline{v' \cdot \nabla f'}$ represents subgrid effects due to the heterogeneities of convection.

The first model is a simple/primitive model where subgrid term $\overline{v' \cdot \nabla f'}$ is ignored:

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla f(\overline{S}) = (\tilde{S} - f(\overline{S}))q. \tag{7}$$

This kind of upscaling technique in conjunction with the upscaling of absolute permeability is commonly used in applications (see e.g. [5]). The difference of our approach is that the coupling of the small scales is performed through the finite volume element formulation of the global problem and the small scale information of the velocity field can be easily recovered. Within this upscaling framework we use $\overline{S}$ instead of S in (4). If the saturation profile is smooth this approximation is of first order. In the coarse blocks where the discontinuities of S are present we need to modify the stiffness matrix corresponding to these blocks. The latter requires the values of the fine scale saturation. In our computation we will not do this and simply use $d(\overline{S})$ in (4).

To improve the primitive upscaled model, one can model the subgrid terms $\overline{v' \cdot \nabla f'}$. First, we briefly review the results for $f(S) = S$ and assume that the perturbations are small. Equation (6) becomes:

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla \overline{S} + \overline{v' \cdot \nabla S'} = (\tilde{S} - \overline{S})q. \tag{8}$$

The term $\overline{v' \cdot \nabla S'}$ represents subgrid effects due to the heterogeneities of convection. This term can be modeled using the equation for S' that is derived by subtracting (8) from the fine scale equation

$$\frac{\partial S^{'}}{\partial t} + \overline{v} \cdot \nabla S' + v' \cdot \nabla \overline{S} + v' \cdot \nabla S' = \overline{v' \cdot \nabla S'} - qS'.$$

This equation can be solved along the characteristics $dx/dt = \overline{v}$ by neglecting higher order terms. Carrying out the calculations in an analogous manner to the ones performed in [7] we can easily obtain the following coarse scale saturation equation:

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla \overline{S} = \nabla \cdot (D(x,t) \nabla \overline{S}(x,t)) + (\tilde{S} - \overline{S})q, \tag{9}$$

where $D(x,t)$ is the dispersive matrix coefficient, whose entries are written as $D_{ij}(x,t) = \left[\int_0^t \overline{v_i'(x) v_j'(x(\tau))} d\tau \right]$. Next it can be easily shown that the diffusion coefficient can be approximated up to the first order by $D_{ij}(x,t) = \overline{v_i'(x) L_j}$, where L_j is the displacement of the particle in j direction that starts at the point x and travels with velocity $-v$. The diffusion term in the coarse model for the saturation field (9) represents the effects of the small scales on the large ones. Note that the diffusion coefficient is a correlation between the velocity perturbation and the displacement. This is different from [7] where the diffusion is taken to be proportional to the length of the coarse scale trajectory. Using our upscaling methodology for the pressure equation we can recover the small scale features of the velocity field that allows us to compute the fine scale displacement.

For the nonlinear flux $f(S)$ we can use similar argument by expanding $f(S) = f(\overline{S}) + f_S(\overline{S})S^{'} + \cdots$. In this expansion we will take into account only linear terms and assume that the flux is nearly linear. This is similar to the linear case and the analysis can be carried out in an analogous manner. The resulting coarse scale equation has the form

$$\frac{\partial \overline{S}}{\partial t} + \overline{v} \cdot \nabla \overline{S} = \nabla \cdot f_S(\overline{S})^2 D(x,t) \nabla \overline{S}(x,t) + (\tilde{S} - f(\overline{S}))q, \tag{10}$$

where $D(x,t)$ is the macrodiffusion corresponding to the linear flow. This formulation has been derived within stochastic framework in [10]. We note that the higher order terms in the expansion of $f(S)$ may result in other effects which, to our best knowledge, have not been studied extensively. In [6] the authors use similar formulation though their implementation is different from ours.

We now turn our attention to the procedure of computing D_{ij}. Let $L_j(x,t)$, $j = 1,2$, be the trajectory length of the particle in x_j-direction that starts at point x computed as $L_j(x,t) = \int_0^t v_j'(x(\tau),\tau)\, d\tau$. Then $D_{ij}(x,t) \approx \overline{v_i'(x,t)\, L_j(x,t)}$. To show this relation we note

$$D_{ij}(x,t) = \overline{v_i'(x,t) \int_0^t v_j'(x(\tau),\tau)\, d\tau}. \tag{11}$$

We remark that since the velocity depends on (x,t), so is the trajectory in (11), i.e., we have $x(\tau) = r(\tau|x,t)$ with $x(t) = r(t|x,t) = x$. Now let $\tau = t_p < t$. We assume that t_p is close to t. Then we may decompose the time integration in (11) as the sum of two integrations, namely,

$$\int_0^t v_j'(r(\tau|x,t),\tau)\, d\tau = \int_0^{t_p} v_j'(r(\tau|x,t),\tau)\, d\tau + \int_{t_p}^t v_j'(r(\tau|x,t),\tau)\, d\tau = I_1 + I_2.$$

Suppose we denote by y_p the particle location at time t_p. Then $r(\tau|x,t) = r(\tau|y_p,t_p)$, $0 \leq \tau \leq t_p$. Thus, $I_1 = \int_0^{t_p} v'_j(r(\tau|y_p,t_p),\tau)\,d\tau = L_j(y_p,t_p)$. Furthermore, since we have assumed that t_p is close to t, the particle trajectory is still close to x, which gives $I_2 \approx (t-t_p)\,v'_j(x,t)$. By substituting these representations back to (11) we obtain our macrodispersion, where now we have $L_j(x,t) = L_j(y_p,t_p) + (t-t_p)\,v'_j(x,t)$. Thus the macrodispersion coefficient may be computed as

$$D_{ij}(x,t) \approx \overline{v'_i(x,t)\,L_j(y_p,t_p)} + (t-t_p)\,\overline{v'_i(x,t)\,v'_j(x,t)}.$$

This relation also shows us how to numerically compute D_{ij}. We note that the fluctuation components v'_i are obtained by subtracting the average $\overline{v_i}$ from v_i, where v_i is constructed from the informations embedded in the multiscale basis functions. Moreover, since $t_p < t$, $L_j(y_p,t_p)$ has been known. Thus we can compute the macrodispersion coefficients incrementally for each time level. This way, saving velocities information for all time levels may be avoided. The calculation of two-point correlations in spatial framework can produce oscillations. For this reason, the authors in [7] avoid computing two-point correlations and introduce some simplifications. In our simulations, we compute two-point correlation and smooth it to avoid the oscillation. In particular, the obtained macro-dispersion is monotone in time and reaches an asymptote.

3 Numerical Results

In this section we present numerical results that give comparison between the fine and the primitive coarse model, and the coarse model with macrodispersion that accounts for the subgrid effects on the coarse grid. It is expected to see possible improvement on the coarse model performance using this extension. We consider a typical cross section in the subsurface, where the system length in the horizontal direction x (L_x) is greater than the formation thickness (L_z); in the results presented below, $L_x/L_z = 5$. The fine model uses 120×120 rectangular elements. The absolute permeability is set to be diag(k,k). All of the fine grid permeability fields used in this study are 120×120 realizations of prescribed overall variance (σ^2) and correlation structure. The fields were generated using GSLIB algorithms [3] with a spherical covariance model [3], for which we specify the correlation lengths l_x and l_z, which are normalized by the system length in the corresponding direction. The coarse models use 12×12 elements which is a uniform coarsening of the fine grid description. In the examples presented below, we consider *side to side flow* flow. More precisely, we fix pressure and saturation ($S = 1$) at the inlet edge of the model ($x = 0$) and zero pressure at the outlet ($x = L_x$). The top and bottom boundaries are closed to flow.

We follow the standard practice for solving the two-phase immiscible flow as well as miscible two-component flow which is known as the *implicit pressure* *explicit saturation method.* For each time step, the pressure equation is solved first where the dependence of the elliptic equation on the saturation uses the values from the previous time level. The Darcy Law is used to compute the flux. Then the saturation equation is solved explicitly using these computed flux as input. We note that in our upscaled model, the pressure equation is solved by the multiscale finite volume element presented above, while the saturation equation is solved on the coarse grid by standard finite volume difference.

Results are presented in terms of fractional flow of displaced fluid (F, defined as fraction of the displaced fluid in the total produced fluid) versus pore volumes injected (PVI). PVI is analogous to dimensionless time and is defined as qt/V_p where q is the total volumetric flow rate, t is dimensional time and V_p is the total pore volume of the system. Figure 1 shows typical results from multicomponent miscible displacement. It uses an anisotropic field of $l_x = 0.20$, $l_z = 0.02$. In all plots, the solid line represents the fine model run on 120×120 elements which serves as a reference solution. The dashed line represents the primitive coarse model ($D = 0$), while the dotted line represents the coarse model with macrodispersion (with D). All coarse models are run on the 12×12 elements. In this figure we show how the performance of our coarse model varies with respect to the mobility ratio, M, and the overall variance of the permeability, σ. The left plot corresponds to the coarse model using $M = 2$ and $\sigma = 1.5$, the right plot corresponds to $M = 5$ and $\sigma = 1.5$. In all these cases we see that the addition of the macrodispersion to our coarse model improves the prediction of the breakthrough. Similar improvement has been observed in two-phase immiscible flow as well as in saturation contours. Due to page limitation, we do not include these results in the paper.

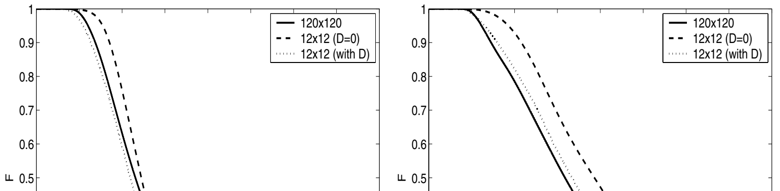


Fig. 1. Comparison of fractional flow of displaced fluid at the production edge for side to side flow. All coarse models are run on 12×12 elements. The permeability has correlation lengths $l_x = 0.20$, $l_z = 0.02$. Left: $M = 2$, $\sigma = 1.5$, Right: $M = 5$, $\sigma = 1.5$.

Summarizing the results, we see that the correction to primitive upscaled saturation equation using perturbation techniques gives an improvement. When the flux, $f(S)$, is a linear function, we do not need to perform linearization of the fluxes and the errors are only due to perturbations of the velocity field. The latter can be controlled by choosing adaptive grid or using adaptive coordinate system. In particular, our results presented in [8] show that in pressure-streamline coordinate system, perturbation techniques work better because the grid is adapted to the flow. In the presence of sharp fronts, one can use subgrid models away from these fronts and follow the front dynamics separately. This approach is also implemented in [8] in pressure-streamline coordinate system and we have observed further improvement in the performance of the method.

References

[1] M.A. Christie. Upscaling for reservoir simulation. *J. Pet. Tech.*, pages 1004–1010, 1996.

[2] N.H. Darman, G.E. Pickup, and K.S. Sorbie. A comparison of two-phase dynamic upscaling methods based on fluid potentials. *Comput. Geosci.*, 6:5–27, 2002.

[3] C.V. Deutsch and A.G. Journel. *GSLIB: Geostatistical Software Library and User's Guide.* Oxford University Press, New York, 2nd edition, 1998.

[4] L.J. Durlofsky. Numerical calculation of equivalent grid block permeability tensors for heterogeneous porous media. *Water Resour. Res.*, 27:699–708, 1991.

[5] L.J. Durlofsky, R.C. Jones, and W.J. Milliken. A nonuniform coarsening approach for the scale up of displacement processes in heterogeneous media. *Adv. in Water Res.*, 20:335–347, 1997.

[6] Y.R. Efendiev and L.J. Durlofsky. Numerical modeling of subgrid heterogeneity in two phase flow simulations. *Water Resour. Res.*, 38(8):1128, 2002.

[7] Y.R. Efendiev, L.J. Durlofsky, and S.H. Lee. Modeling of subgrid effects in coarse scale simulations of transport in heterogeneous porous media. *Water Resour. Res.*, 36:2031–2041, 2000.

[8] Y.R. Efendiev, T.Y. Hou, and T. Strinopoulos. Multiscale simulations of porous media flows in flow-based coordinate. *Comput. Geosci.*, 2006. submitted.

[9] T.Y. Hou and X.H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134:169–189, 1997.

[10] P. Langlo and M.S. Espedal. Macrodispersion for two-phase, immisible flow in porous media. *Adv. in Water Res.*, 17:297–316, 1994.

[11] W. Zijl and A. Trykozko. Numerical homogenization of two-phase flow in porous media. *Comput. Geosci.*, 6(1):49–71, 2002.

MINISYMPOSIUM 4: Domain Decomposition Methods Motivated by the Physics of the Underlying Problem

Organizers: Martin J. Gander[1] and Laurence Halpern[2]

[1] Section de Mathématiques, Université de Genève, Switzerland.
martin.gander@math.unige.ch
[2] LAGA, Université Paris 13, France. halpern@math.univ-paris13.fr

Domain decomposition methods are a powerful tool to handle very large systems of equations. They can however also be used to couple different physical models or approximations, which one might want to do for various reasons: in fluid structure coupling for example, the physical laws in the fluid differ from the physical laws in the structure, and a domain decomposition method could naturally take this into account. Even if the physical model is the same, one might want to use a simplified equation in part of the domain, where certain effects are negligible, like for example in aerodynamics, to save computation time. Or one could simply want to use a much coarser mesh, like in combustion away from the flame front, which again could be taken naturally into account by a domain decomposition method that can handle non-matching grids, possibly in space and time.

In the first paper, Gander, Halpern, Labbé and Santugini present an optimized Schwarz waveform relaxation algorithm for the parallel solution in space-time of the equations of ferro-magnetics in the micro-magnetic model. The algorithm uses Robin transmission conditions, and a numerical study of the dependence of the optimized parameters on the physical properties of the problem is presented.

In the second paper, Halpern and Japhet present a space-time decomposition method for heterogeneous problems, with transmission conditions adapted to the heterogeneity of the physical model. The algorithm is of Schwarz waveform relaxation type with time windows, and discretized using a discontinuous Galerkin method in time, and a classical finite element discretization in space. The performance of this new method is illustrated by numerical experiments.

In the third paper, Halpern and Szeftel present a quasi-optimal Schwarz waveform relaxation algorithm for the one dimensional Schrödinger equation. This algorithm uses non-local interface conditions in time, which are exact for the case of a constant potential. If the potential is not constant, a frozen coefficient approach is used. The authors compare the performance of this new method to the performance of a classical Schwarz waveform relaxation method, and an optimized one with Robin conditions.

In the fourth paper, Haynes, Huang and Russel present an innovative Moving Mesh Schwarz Waveform relaxation method. In this method, evolution problems are decomposed in space, and on each subdomain, a moving mesh method is used to solve the subdomain problem on a given time window, before information is exchanged across the interfaces between subdomains. The method uses classical Dirichlet transmission conditions, and is both adaptive in space, with the moving mesh method, and in time, with a step size control. Its performance is well illustrated by numerical experiments for the viscous Burgers equation.

An Optimized Schwarz Waveform Relaxation Algorithm for Micro-Magnetics

Martin J. Gander[1], Laurence Halpern[2], Stéphane Labbé[3], and Kévin Santugini-Repiquet[1]

[1] University of Geneva, Section de mathématiques, Case Postale 64, 1211 Genève 4, Switzerland. {Kevin.Santugini,Martin.Gander}@math.unige.ch

[2] Institut Galilée, Université Paris 13, Département de mathématiques, 99 av J-B Clément, 93430 Villetaneuse, France. halpern@math.univ-paris13.fr

[3] Laboratoire de mathématiques Bât 405, Université Paris 11, 91405 Orsay, France. Stephane.Labbe@math.u-psud.fr

Summary. We present an optimized Schwarz waveform relaxation algorithm for the parallel solution in space-time of the equations of ferro-magnetics in the micro-magnetic model. We use Robin transmission conditions, and observe fast convergence of the discretized algorithm. We show numerically the existence of an optimal parameter in the Robin condition, and study its dependence on the various physical and numerical parameters.

1 Introduction

Over the last decades, ferro-magnetics has been the subject of renewed interest due to its omnipresence in industrial applications, and the need for correctly predicting the behavior of ferro-magnets, which is best achieved by numerical simulations, see the historical introduction in [11] and [1]. Since micro-magnetic simulations are very costly, we present in this paper an optimized Schwarz waveform relaxation algorithm for the micro-magnetic equation. These algorithms have the advantage of independent adaptive discretizations per subdomain both in space and time, and they are naturally parallel, see [6, 7, 4, 5]. We present a numerical analysis of the algorithm with Robin transmission conditions applied to the equation of ferro-magnetics for a two subdomain decomposition, and study the dependence of the optimal parameter on the various physical and numerical parameters.

2 The Micro-Magnetic Model

Let Ω be a bounded open set in $\mathbb{R}^3$ filled with a ferromagnetic material. The magnetic state of the material is given by its magnetization vector $\boldsymbol{m} \in \mathbb{R}^3$, vanishing outside Ω, with the non-convexity constraint

$$|\boldsymbol{m}| = 1 \text{ a.e. in } \Omega. \tag{1}$$

The behavior of $\boldsymbol{m}$ is modeled by the Landau-Lifschitz equation,

$$\frac{\partial \boldsymbol{m}}{\partial t} = \mathcal{L}(\boldsymbol{m}) := -\boldsymbol{m} \times \boldsymbol{h}(\boldsymbol{m}) - \alpha \boldsymbol{m} \times (\boldsymbol{m} \times \boldsymbol{h}(\boldsymbol{m})), \tag{2}$$

where $\alpha > 0$ is the dissipation parameter. As a first step toward real computations, we include only the exchange interaction, which is local, and produces a magnetic excitation $\boldsymbol{h}(\boldsymbol{m}) = A \triangle \boldsymbol{m}$, where $A > 0$ is the exchange constant. The equation in $\Omega \times (0, T)$ is subject to homogeneous Neumann boundary conditions on the boundary of Ω, i.e. $\frac{\partial \boldsymbol{m}}{\partial \nu} = \mathbf{0}$, where $\boldsymbol{\nu}$ is the unit outward normal on the boundary. Equation (2) is often used to compute the steady states of the magnetization field; for more information, see [8].

3 Optimized Schwarz Waveform Relaxation Algorithm

We decompose the domain Ω into p non-overlapping subdomains $(\widetilde{\Omega}_i)_{i=1\ldots p}$, $\bigcup_{i=1}^{p} \overline{\widetilde{\Omega}_i} = \overline{\Omega}$. We then derive from this non-overlapping decomposition an overlapping one by choosing $(\Omega_i)_{i=1\ldots p}$ such that $\widetilde{\Omega}_i \subset \Omega_i$ and $\bigcup_{i=1}^{p} \Omega_i = \Omega$.

We define the interfaces and exterior boundary by

$$\Gamma_{ij} = \partial\Omega_i \cap \overline{\widetilde{\Omega}_j}, \qquad \Gamma_i^e = \partial\Omega_i \cap \partial\Omega.$$

An optimized Schwarz waveform relaxation algorithm computes for $n = 1, 2, \ldots$ the iterates $(\boldsymbol{m}_i^n)_{1 \le i \le p}$ defined by

$$\begin{aligned}
\frac{\partial \boldsymbol{m}_i^n}{\partial t} &= \mathcal{L}(\boldsymbol{m}_1^n) && \text{in } \Omega_i \times (0, T),\\
\boldsymbol{m}_i^n(\cdot, 0) &= \boldsymbol{m}_0 && \text{on } \Omega_i,\\
\mathcal{B}_{ij} \boldsymbol{m}_i^n &= \mathcal{B}_{ij} \boldsymbol{m}_j^{n-1} && \text{on } \Gamma_{ij} \times (0, T),\\
\frac{\partial \boldsymbol{m}_i^n}{\partial \boldsymbol{\nu}} &= \mathbf{0} && \text{on } \Gamma_i^e \times (0, T),
\end{aligned} \tag{3}$$

where the $\mathcal{B}_{ij}$ are linear operators.

In [2, 5], several strategies for choosing these boundary operators are proposed both in the case with and without overlap, and a complete convergence analysis is provided for a linear advection-diffusion equation. Here, because of the non-linearity, such an analysis is not yet available, and we use for our numerical study for the non-overlapping case Robin transmission conditions, which are robust and easy to implement, *i.e.* $\mathcal{B}_{ij} = \frac{\partial}{\partial \boldsymbol{\nu}} + \beta_{ij}\mathrm{I}$, where β_{ij} is a positive real number to be chosen optimally for best convergence.

4 Discretization

We use in space a finite difference discretization on a regular rectangular grid, where the Laplace operator can be approximated by the standard five point finite difference stencil. Since we use cell-centered nodes, the boundaries and the interfaces are halfway in between two nodes, and hence values there can be approximated using the mean of the two adjacent nodes, denoted by A and B in Figure 1, whereas the normal derivative can be approximated by a finite difference between the same nodes A and B.

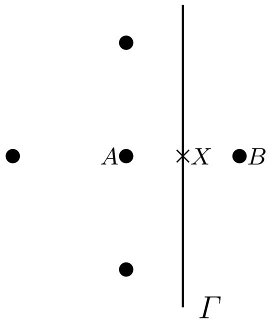


Fig. 1. Position of the interface in our cell-centered finite difference discretization.

The Robin condition $\partial_{\boldsymbol{\nu}} \boldsymbol{m} + \beta \boldsymbol{m} = g$ at point X in Figure 1 is thus discretized by $\frac{\boldsymbol{m}_B - \boldsymbol{m}_A}{\Delta x} + \beta \frac{\boldsymbol{m}_A + \boldsymbol{m}_B}{2} = g$, where Δx is the space step-size. This yields $\boldsymbol{m}_B = (2\Delta x g + (2 - \beta \Delta x)\boldsymbol{m}_A)/(2 + \beta \Delta x)$, which is then used to complete the missing value at the node B in the five-point finite difference stencil centered at A in Figure 1.

For the time discretization, we use the explicit second order scheme from [9, 10],

$$\boldsymbol{m}_{i+1} = \boldsymbol{m}_i + \Delta t \boldsymbol{F}(\boldsymbol{m}_i) + \frac{\Delta t^2}{2} \mathrm{D}\boldsymbol{F}(\boldsymbol{m}_i) \cdot \boldsymbol{F}(\boldsymbol{m}_i), \tag{4}$$

where D is the differentiation operator and

$$\boldsymbol{F}(\boldsymbol{m}_i) = -\boldsymbol{m}_i \times \boldsymbol{h}(\boldsymbol{m}_i) - \alpha \boldsymbol{m}_i \times (\boldsymbol{m}_i \times \boldsymbol{h}(\boldsymbol{m}_i)).$$

To satisfy the non-convexity constraint (1), we renormalize the magnetization after each time step. Our implementation can use an optimized time step per subdomain in order to maximize energy dissipation and speed up convergence to the steady state, and thus the algorithm is truly non-conforming in time. To study however the convergence to the discrete solution on the entire domain Ω numerically, we use in the sequel fixed time steps in the subdomains.

5 Numerical Study of the Algorithm

We consider a squared ferromagnetic thin plate of dimension $3.68e{-}6 \times 3.68e{-}6 \times \Delta x$, with the parameter $A = 4.06e{-}18$. We also fix the parameter α to be $\frac{1}{2}$.

We divide the domain into two subdomains, as shown in Figure 2. In this non-overlapping case, $\Gamma_{12} = \Gamma_{21} = \Gamma$, and we consider the case $\beta_{12} = \beta_{21} = \beta$ only. We discretize the problem as shown in Section 4, and we first compute the discrete solution on the entire domain. We then measure the relative error between the mono-domain solution and each iteration of the optimized Schwarz waveform relaxation algorithm in the l_h^2 norm,

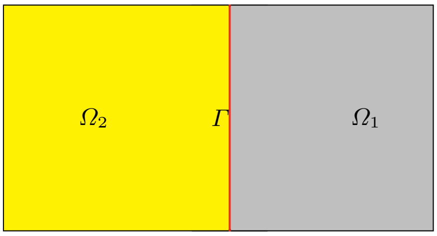


Fig. 2. The two subdomain decomposition used for our numerical experiments.

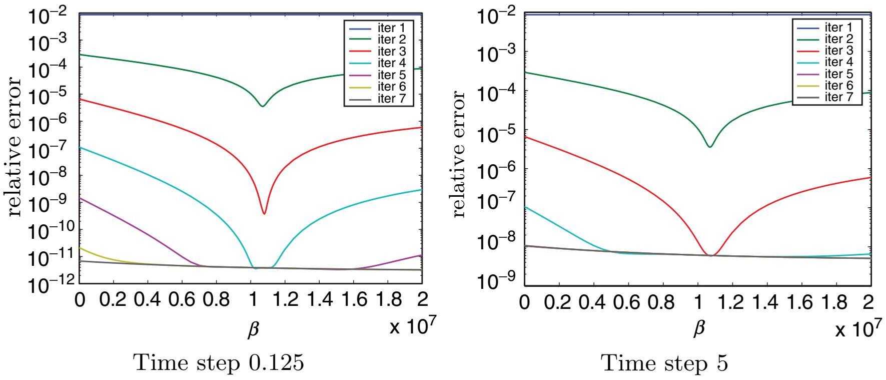
$$\|\boldsymbol{u}\|_{l_h^2}^2 = \sum_{i=1}^{N} |\omega_i| |\boldsymbol{u}_i|^2.$$

Since (ω_i) is a rectangular mesh, $|\omega_i| = |\Delta x|^3$.

We mesh the ferromagnetic domain with a space step of $\Delta x = 1.84e{-}7$, which yields a $20 \times 20 \times 1$ mesh. A numerical experiment over long time shows that for the physical parameters chosen above, the equilibrium state has not yet been reached at $T = 30000$.

We first study the convergence behavior of the optimized Schwarz waveform relaxation algorithm. The simulations presented here are done without overlap. We choose a final time of $T = 500$, and perform a series of computations for fixed Δt, for various values of the parameter β. In Figure 3, we show for the time discretization steps $\Delta t = 0.125$ and $\Delta t = 5$ the relative error curves for a sequence of iterates as a function of the parameter β in the Robin transmission condition. The algorithm is convergent, and there is a numerically optimal choice β_{opt} for β: in both cases, $\beta_{opt} \approx 1.05e{+}7$, which indicates that β_{opt} does not depend on the time step.

We now study the dependence of β_{opt} on the space discretization step Δx. As the step size increases, β_{opt} in the Robin transmission conditions decreases. The least squared best fit shown in Figure 4 gives $\ln \beta_{opt} \approx 1.03 - 0.97 \ln(\Delta x)$, which indicates that



Time step 0.125

Time step 5

Fig. 3. Convergence curves as function of the optimization parameter β for two different time steps.

$$\beta_{opt} \approx \frac{2.8}{\Delta x}. \tag{5}$$

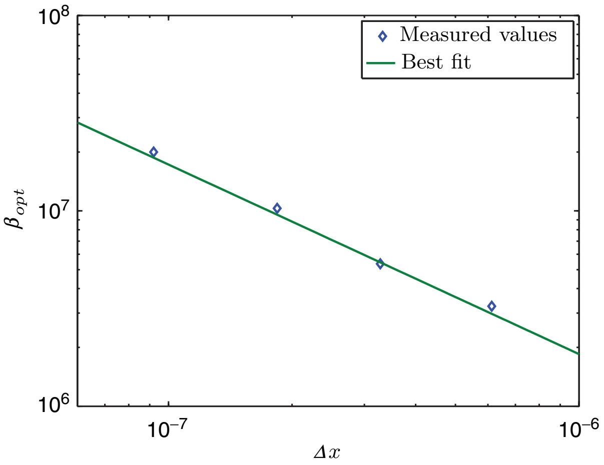
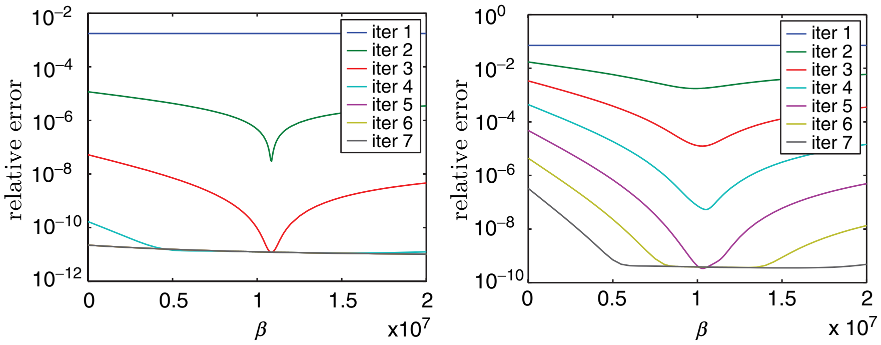


Fig. 4. β_{opt} as a function of the space discretization step

In the next sequence of numerical experiments, we study the dependence of β_{opt} on the physical parameters of the problem. In Figure 5, we show for the final times $T = 100$ and $T = 4000$ the relative error curves for a sequence of iterations as a function of the parameter β in the Robin transmission condition. These results, and a large sample of final times between 1 and 4000 indicate that β_{opt} does not seem to depend on the final time. This is somewhat



Final time 100 Final time 4000

Fig. 5. Convergence curves as function of the optimization parameter for two final times.

unexpected, since for the linear heat equation, β_{opt} depends on the final time, see [3].

The fact that β_{opt} does not depend on the final time of the simulation implies that β_{opt} does not depend on the physical parameter A in our case, since dividing the entire equation by A shows that A can be interpreted as a scaling factor for the final time. If however other exchange interactions were present, such as the demagnetization field, this scaling argument would not hold any more.

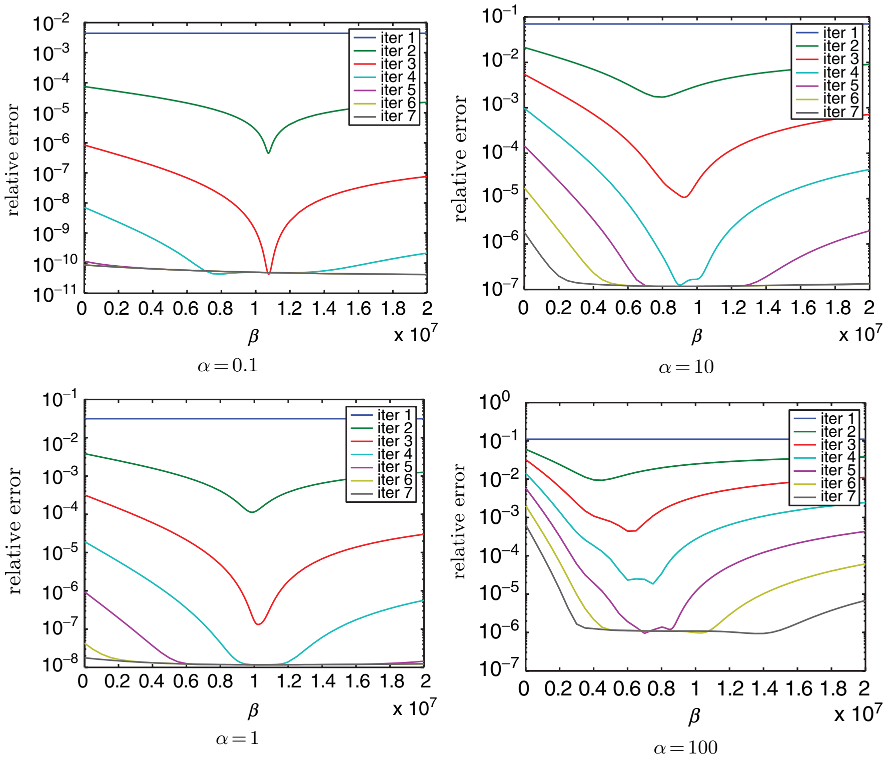
It remains to study the behavior of β_{opt} when α varies. To this end, we compute the convergence curves with parameters $T = 200$ and $\Delta x = 1.84e{-}7$ for α ranging from 0.1 to 100. We present some of these results in Figure 6. For small α, the algorithm converges very well and the β_{opt} has an approximate value of $1.05e{+}7$. However as α increases, the value of β_{opt} varies as shown in Figure 7 and the optimal error increases, see Figure 7.

6 Conclusion

We presented an optimized Schwarz waveform relaxation algorithm for the equations of ferro-magnetics in the micro-magnetic model. We studied numerically the convergence behavior of the algorithm with Robin transmission conditions. This study revealed that the algorithm converges very fast: after only few iterations, an error reduction by a factor of 10^{-9} is achieved. Using extensive numerical results, we determined a heuristic formula for the value of β_{opt}, in the non-overlapping case,

$$\beta_{opt} \approx \frac{g(\alpha)}{\Delta x},$$

where $g(\alpha)$ is represented in Figure 7 on the left.



$\alpha = 0.1$

$\alpha = 10$

$\alpha = 1$

$\alpha = 100$

Fig. 6. Convergence curves for various choice of α

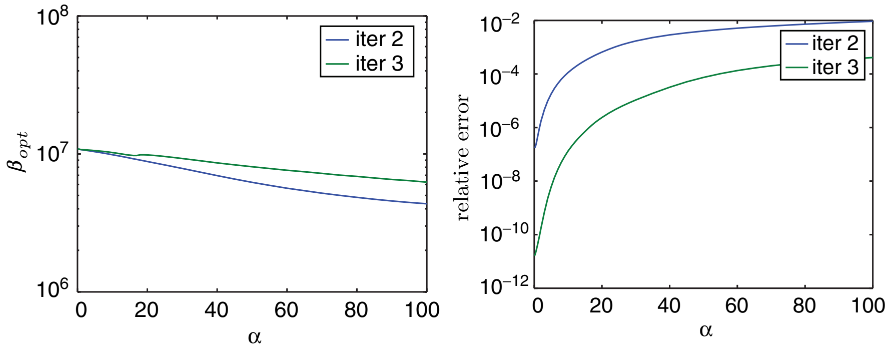


Fig. 7. β_{opt} and optimal error as functions of α

We are currently working on a convergence analysis of the algorithm, and the extension to other interaction terms; in particular adding the demagnetization field interaction is challenging, since it represents a global operator.

References

[1] A. Aharoni. *Introduction to the Theory of Ferromagnetism.* Oxford Science Publication, 1996.

[2] D. Bennequin, M.J. Gander, and L. Halpern. Optimized Schwarz waveform relaxation methods for convection reaction diffusion problems. Technical Report 24, Institut Galilée, Paris XIII, 2004.

[3] M.J. Gander and L. Halpern. Méthodes de relaxation d'ondes (SWR) pour l'équation de la chaleur en dimension 1. *C. R. Math. Acad. Sci. Paris*, 336(6):519–524, 2003.

[4] M.J. Gander and L. Halpern. Absorbing boundary conditions for the wave equation and parallel computing. *Math. Comp.*, 74(249):153–176, 2004.

[5] M.J. Gander and L. Halpern. Optimized Schwarz waveform relaxation methods for advection reaction diffusion problems. *SIAM J. Numer. Anal.*, 45(2):666–697, 2007.

[6] M.J. Gander, L. Halpern, and F. Nataf. Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation. In C-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, editors, *Eleventh International Conference of Domain Decomposition Methods*, pages 27–36. ddm.org, 1999.

[7] M.J. Gander, L. Halpern, and F. Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. *SIAM J. Numer. Anal.*, 41(5):1643–1681, 2003.

[8] L. Halpern and S. Labbé. La théorie du micromagnétisme. Modélisation et simulation du comportement des matériaux magnétiques. *Matapli*, 66:77–92, September 2001.

[9] S. Labbé. *Simulation Numérique du Comportement Hyperfréquence des Matériaux Ferromagnétiques.* PhD thesis, Université Paris 13, Décembre 1998.

[10] S. Labbé and P.-Y. Bertin. Microwave polarizability of ferrite particles. *J. Magnetism and Magnetic Materials*, 206:93–105, 1999.

[11] É. Trémolet de Lacheisserie, editor. *Magnétisme: Fondements*, volume I of *Collection Grenoble Sciences.* EDP Sciences, 2000.

Discontinuous Galerkin and Nonconforming in Time Optimized Schwarz Waveform Relaxation for Heterogeneous Problems

Laurence Halpern and Caroline Japhet

LAGA, Université Paris XIII, 99 Avenue J-B Clément, 93430 Villetaneuse, France,
{halpern,japhet}@math.univ-paris13.fr

Summary. We consider the question of domain decomposition for evolution problems with discontinuous coefficients. We design a method relying on four ingredients: extension of the optimized Schwarz waveform relaxation algorithms as described in [1], discontinuous Galerkin methods designed in [7], time windows, and a generalization of the projection procedure given in [6]. We so obtain a highly performant method, which retains the approximation properties of the discontinuous Galerkin method. We present numerical results, for a two-domains splitting, to analyze the time-discretization error and to illustrate the efficiency of the DGSWR algorithm with many time windows. This analysis is in continuation with the approach initiated in DD16 [2, 5], with applications in climate modeling, or nuclear waste disposal simulations.

1 Introduction

In order to be able to perform long time computations in highly discontinuous media, it is of importance to split the computation into subproblems for which robust and fast solvers can be used. This happens for instance in climate modeling, where heterogeneous climatic models must be run in parallel, or in nuclear waste disposal simulations, where different materials have different behaviors.

Optimized Schwarz waveform relaxation algorithms have proven to provide an efficient approach for convection-diffusion problems in one [1] and two dimensions [8]. The SWR algorithms are global in time, and therefore are well adapted to coupling models; they lead to fast and efficient solvers, and they allow for the use of non conforming space-time discretizations. Based on this approach, our final objective is to propose efficient algorithms with a high degree of accuracy, for heterogeneous advection-diffusion problems. The strategy we develop here is to split the time interval into time windows. In each window we will perform a small number of iterations of an optimized Schwarz waveform relaxation algorithm. The subdomain solver is the discontinuous

Galerkin methods in time, and classical finite elements in space. The coupling between the subdomains is done by the extension of a projection procedure written in [6].

After defining our model problem in Section 2, we recall in Section 3 the Schwarz waveform relaxation algorithm, with optimized transmission conditions of order 1 in time, as introduced in [2, 5]. The general discontinuous Galerkin formulation is given in Section 4. In Section 5, we introduce the discrete algorithm in time in the nonconforming case. The projection between arbitrary grids is performed by an efficient algorithm based on the method introduced in [6]. In Section 6, numerical results illustrate the validity of our approach, in particular the superconvergence result proved in [3] for the heat equation and homogeneous Dirichlet boundary conditions is valid.

2 Model Problem

We consider the advection-diffusion problem in $\Omega = (a, b)$:

$$\begin{aligned}
&\mathcal{L}u \equiv \frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(a(x)u) - \frac{\partial}{\partial x}(\nu(x)\frac{\partial u}{\partial x}) = f \text{ in } \Omega \times (0,T),\\
&u(\cdot,0) = u_0 \text{ in } \Omega,\\
&u(a,\cdot) = u(b,\cdot) = 0 \text{ in } (0,T).
\end{aligned} \tag{1}$$

The advection coefficient $a(x)$, and the viscosity $\nu(x)$ are in $L^\infty(\Omega)$. ν is bounded from below by a positive constant and we suppose here the advection coefficient to be positive. We are interested in a coupling procedure for a problem with discontinuities in the coefficients, and we suppose a and ν to be continuous in subregions $\Omega_j =]x_{j-1}, x_j[$ of Ω, but possibly discontinuous at interfaces x_j. We shall write

$$a_j^\pm = \lim_{x \to x_j^\pm} a(x), \quad \nu_j^\pm = \lim_{x \to x_j^\pm} \nu(x).$$

Problem (1) is equivalent to finding $\{u_j\}_{\ j=1,\dots,J}$ solutions of the advection-diffusion equation in each subdomain Ω_j, with the physical transmission conditions in $(0,T)$

$$u_j(x_j^-,\cdot) = u_{j+1}(x_j^+,\cdot), \quad (\nu_j^- \frac{\partial}{\partial x} - a_j^-)u_j(x_j^-,\cdot) = (\nu_j^+ \frac{\partial}{\partial x} - a_j^+)u_{j+1}(x_j^+,\cdot),$$

with $u_j = u_{|\Omega_j}$. In view of applications for long-time computations, we split the time domain into windows and we intend to design an algorithm which requires very few iterations per time window. This will be achieved with an optimized Schwarz waveform relaxation algorithm.

3 Optimized Schwarz Waveform Relaxation with Time Windows

The time interval is divided into time windows, $[0,T] = \cup_{N=0}^{Nw}[T_N, T_{N+1}]$. In each window we perform successively n_{N} iterations of the Schwarz waveform relaxation algorithm with $n_N \geq 2$ as small a possible, taking as initial value the final value of the last iterate of the algorithm in the previous window. Suppose $\{\tilde{u}_j\}$ is computed that way on $(0,T_N)$. We compute now $\{\tilde{u}_j\}$ on the next window by the algorithm, for $n = 1, ..., n_N$:

$$\mathcal{L}u_j^n = f \text{ in } \Omega_j \times (T_N, T_{N+1}),$$

with the initial value $u_j^n(\cdot, T_N) = \tilde{u}_j(\cdot, T_N^-)$, and the transmission conditions in (T_N, T_{N+1}),

$$(-\nu_{j-1}^+ \frac{\partial}{\partial x} + a_{j-1}^+ Id + S_j^-)u_j^n(x_{j-1}^+, \cdot) = (-\nu_{j-1}^- \frac{\partial}{\partial x} + a_{j-1}^- Id + S_j^-)u_{j-1}^{n-1}(x_{j-1}^-, \cdot),$$

$$(\nu_j^- \frac{\partial}{\partial x} - a_j^- Id + S_j^+)u_j^n(x_j^-, \cdot) = (\nu_j^+ \frac{\partial}{\partial x} - a_j^+ Id + S_j^+)u_{j+1}^{n-1}(x_j^+, \cdot).$$

It was proved in [5, 2] that a suitable choice of the operators $S_j^\pm$, leads to convergence in J iterations. However these "transparent" operators are not easy to handle, and we use instead transmission operators of the form:

$$S_j^- = \frac{p_j^- - a_j^-}{2} Id + \frac{q_j^-}{2} \frac{\partial}{\partial t}, \quad S_j^+ = \frac{a_j^+ + p_j^+}{2} Id + \frac{q_j^+}{2} \frac{\partial}{\partial t}. \tag{2}$$

The initial guesses on the interfaces have to be prescribed, this will be done on the discrete level in Section 6. We then define $\tilde{u}_j$ on (T_N, T_{N+1}) by $u_j^{n_N}$. In order to reduce the number of iterations, we need to make the convergence rate as small as possible. This can be achieved by choosing carefully the parameters $p_j^\pm$ and $q_j^\pm$ such as to minimize the local convergence rate, *i.e.* between two subdomains. Details for the optimization on the theoretical level for continuous coefficients can be found in [1], and for preliminary results in this case see [5]. For positive coefficients $p_j^\pm$ and $q_j^\pm$, the convergence of the algorithm can be proved by the method of energy.

4 Time Discontinuous Galerkin Method

We introduce the discretization of a subproblem in one time window $I = (T_N, T_{N+1})$ and one interval Ω_j. The subproblem at step n of the SWR procedure for an internal subdomain is to find v such that

$$\begin{cases} \mathcal{L}v = f & \text{in } \Omega_j \times I, \\ v(\cdot, T_N) = v_0 & \text{in } \Omega_j, \\ (-\nu^+ \frac{\partial}{\partial x} + \beta^- Id + \gamma^- \frac{\partial}{\partial t})\ v(x_{j-1}^+, \cdot) = g^- & \text{in } I, \\ (\nu^- \frac{\partial}{\partial x} + \beta^+ Id + \gamma^+ \frac{\partial}{\partial t})\ v(x_j^-, \cdot) = g_j^+ & \text{in } I. \end{cases} \tag{3}$$

Subproblems at either end of the interval have one boundary condition replaced by a Dirichlet boundary condition. Let $V_j = H^1(\Omega_j)$. This problem has the weak formulation: find v in $\mathcal{C}^0(0,T;L^2(\Omega_j)) \cap L^2(0,T;H^1(\Omega_j))$ such that $v(\cdot,T_N) = v_0$ and

$$((\frac{dv}{dt}(t),\varphi)) + b(v(t),\varphi) = (f(t),\varphi) + g^-(t)\varphi(x_{j-1}) + g^+(t)\varphi(x_j), \quad \forall \varphi \in V_j,$$

with $(\cdot,\cdot)$ the scalar product in $L^2(\Omega_j)$, and for φ,ψ in V_j:

$$\begin{cases} ((\varphi,\psi)) = (\varphi,\psi) + \gamma^-\varphi(x_{j-1})\psi(x_{j-1}) + \gamma^+\varphi(x_j)\psi(x_j), \\ b(\varphi,\psi) = (\nu(x)\dfrac{\partial\varphi}{\partial x},\dfrac{\partial\psi}{\partial x}) + (a(x)\varphi,\dfrac{\partial\psi}{\partial x}) \\ \qquad\qquad + \beta^-\varphi(x_{j-1})\psi(x_{j-1}) + \beta^+\varphi(x_j)\psi(x_j). \end{cases}$$

For positive $\beta^\pm$ and $\gamma^\pm$, this problem is well-posed in suitable Sobolev spaces, see [1]. For the discretization of (3) in time, we use the discontinuous Galerkin method [7] which is a Galerkin method with discontinuous piecewise polynomials of degree $q \geq 0$ defined as follows. Let $\mathcal{T}$ be a decomposition of I into $I = \cup_{k=1}^K I_k$ with $I_k = [t_{k-1},t_k]$, and $\Delta t_k = t_k - t_{k-1}$. For any space V, we define

$$\mathbf{P}_q(V) = \{\varphi : I \to V,\ \varphi(t) = \sum_{i=0}^{q} \varphi_i t^i,\ \varphi_i \in V\}$$
$$\mathcal{P}_q(V,\mathcal{T}) = \{\varphi : I \to V,\ \varphi_{|I_k} \in \mathbf{P}_q(V),\ 1 \leq k \leq K\}.$$

As the functions in $\mathcal{P}_q(V,\mathcal{T})$ may be discontinuous at the mesh points t_k, we define $\varphi^{k,\pm} = \varphi(t_k \pm 0)$. The discontinuous Galerkin Method, as formulated in [7], defines recursively on the intervals I_k, an approximate solution U of (4) in $\mathcal{P}_q(V_j,\mathcal{T})$, by

$$\begin{cases} U^{0,-} = v_0, \\ \forall\varphi \in \mathbf{P}_q(V_j): \quad \displaystyle\int_{I_k} [((\frac{dU}{dt},\varphi)) + b(U,\varphi)]\, dt + ((U^{k-1,+},\varphi^{k-1,+})) = \\ \displaystyle\int_{I_k} [(f(t),\varphi(t)) + g^-(t)\varphi(x_{j-1}) + g^+(t)\varphi(x_j)] dt + ((U^{k-1,-},\varphi^{k-1,+})). \end{cases} \tag{4}$$

Theorem 1. *For $f \in L^\infty(0,T;L^2(\Omega_j))$, $g^\pm \in L^\infty(0,T)$, and $\beta^\pm,\gamma^\pm > 0$, equation (4) has a unique solution $U \in \mathcal{P}_q(V_j,\mathcal{T})$. Moreover, for sufficiently smooth v, we have the error estimate:*

$$\|v - U\|_{L^\infty(I;L^2(\Omega_j))} \leq C^i C_q^s (\max_{1\leq k\leq K} L_k) \|\Delta t^{q+1} v^{(q+1)}\|_{L^\infty(I;L^2(\Omega_j))}, \tag{5}$$

where Δt is the local time step defined in I_k by $\Delta t(s) = \Delta t_k$, C_q^s is a stability constant related to the dG-discretization, independent of T, u, Δt, and U. C^i is an interpolation constant depending only on q, and $L_k = 1 + log(t_k/\Delta t_k)$.

Proof. Let $H = L^2(\Omega_j) \times \mathbb{R} \times \mathbb{R}$ with inner product $(\cdot,\cdot)_H$ defined for $U_1 = (\varphi_1, \alpha_1^-, \alpha_1^+)$ and $U_2 = (\varphi_2, \alpha_2^-, \alpha_2^+)$ in H by $(U_1, U_2)_H = (\varphi_1, \varphi_2) + \gamma_j^- \alpha_1^- \alpha_2^- + \gamma_j^+ \alpha_1^+ \alpha_2^+$. Let $D(A) = \{U = (\varphi, \alpha^-, \alpha^+),\ \varphi \in H^2(\Omega_j),\ \varphi(x_{j-1}) = \alpha^-,\ \varphi(x_j) = \alpha^+\}$. Let $A : D(A) \subset H \to H$ defined by

$$A = \begin{pmatrix} -\dfrac{\partial}{\partial x}(\nu \dfrac{\partial}{\partial x}) + \dfrac{\partial}{\partial x}(a\ Id) & 0 & 0 \\ \dfrac{\nu_{j-1}^+}{\gamma_j^-}\dfrac{\partial}{\partial x} & \dfrac{\beta_j^-}{\gamma_j^-} & 0 \\ \dfrac{\nu_j^-}{\gamma_j^+}\dfrac{\partial}{\partial x} & 0 & \dfrac{\beta_j^+}{\gamma_j^+} \end{pmatrix}$$

Then, the proof of Theorem 1 is based on the theoretical result in [4], since A is the infinitesimal generator of an analytic, uniformly bounded semi-group.

Remark 1. An analogous result holds in higher dimension and for general boundaries, as soon as they do not intersect.

5 The Discontinuous Galerkin Schwarz Waveform Relaxation

Let us consider the case where the time steps are different in the subdomains: in each Ω_j, let $\mathcal{T}_j$ be a partition of the time interval into $I = \cup_{k=1}^K I_k^j$ with $I_k^j = [t_{k-1}^j, t_k^j]$. Then, we need a projection procedure to transfer the boundary values from one domain to his two neighbors. We now define the precise procedure in domain Ω_j. Let $(g_j^{-,n-1}, g_j^{+,n-1})$ be given in $\mathcal{P}_q(\mathbb{R}, \mathcal{T}_j)$. Then, one iteration of the SWR method consists in the following steps:

$$\begin{tikzcd}
g_j^{-,n-1} \in \mathcal{P}_q(\mathbb{R},\mathcal{T}_j) \arrow[dr] & & g_j^{+,n-1} \in \mathcal{P}_q(\mathbb{R},\mathcal{T}_j) \arrow[dl] \\
& U_j^n \in \mathcal{P}_q(V_j,\mathcal{T}_j) \arrow[dl] \arrow[dr] & \\
\tilde{g}_j^{-,n} = (\nu_{j-1}^+ \partial_x - a_{j-1}^+ + S_{j-1}^+) U_j^n(x_{j-1}^+, \cdot) & & \tilde{g}_j^{+,n} = (-\nu_j^- \partial_x + a_j^- - S_{j+1}^-) U_j^n(x_j^-, \cdot) \\
\tilde{g}_j^{-,n} \in \mathcal{P}_q(\mathbb{R},\mathcal{T}_j) \arrow[d] & & \tilde{g}_j^{+,n} \in \mathcal{P}_q(\mathbb{R},\mathcal{T}_j) \arrow[d] \\
g_{j-1}^{+,n} = P_{j,j-1}\tilde{g}_j^{-,n} \in \mathcal{P}_q(\mathbb{R},\mathcal{T}_{j-1}) & & g_{j+1}^{-,n} = P_{j,j+1}\tilde{g}_j^{+,n} \in \mathcal{P}_q(\mathbb{R},\mathcal{T}_{j+1})
\end{tikzcd}$$

U_j^n is the solution of (4) in Ω_j with coefficients $\beta_j^\pm$ and $\gamma_j^\pm$ given by (2), and data $(g_j^{-,n-1}, g_j^{+,n-1})$. $P_{i,j}$ is the orthogonal L^2 projection on $\mathcal{P}_q(\mathbb{R}, \mathcal{T}_j)$, restricted to $\mathcal{P}_q(\mathbb{R}, \mathcal{T}_i)$.

Note that the computations in different subdomains on the same time window (T_N, T_{N+1}) can be done in parallel. One could even think of using multigrid in time or asynchronous algorithm.

6 Numerical Results

In order to see the effect of the coupling of discontinuous Galerkin with the domain decomposition algorithm, we perform numerical simulations with two subdomains only. We choose ν and a to be constant in each subdomain. For the space discretization, we replace V_j by a finite-dimensional subspace V_j^h (standard $\mathbb{P}_1$ finite element space) of V_j in (4).
Our computations are performed with $q = 1$ in the discontinuous Galerkin method. In that case a superconvergence result was proved in [3] for the heat equation and homogeneous Dirichlet boundary conditions: under suitable assumptions on the space and time steps, the accuracy is of order 3 in time at the discrete time levels t_k: let $\|\cdot\|_{k,j} = \|\cdot\|_{L^\infty(I_k;L^2(\Omega_j))}$,

$$\begin{aligned}&\|v(t_k) - U^{k,-}\|_{L^2(\Omega_j)}\\ &\qquad\le C_k \max_{1\le k\le K}\left\{\min_{0\le\ell\le 3} \Delta t_k^\ell \|\partial_t^{(\ell)} v\|_{k,j} + \min_{1\le\ell\le 2} h^\ell \|D^\ell v\|_{k,j}\right\}, \qquad (6)\end{aligned}$$

where h is the mesh size, $C_k = C(L_k)^{\frac{1}{2}}$ with C a constant independent of T, v, Δt, and U. Our numerical results will illustrate both estimates (5) and (6).

In the sequel, we denote by "1-window converged solution", the iterate of the optimized Schwarz waveform relaxation algorithm in one time window (the whole time interval), for which the residual on the boundary (*i*.e. $\|g_j^{\pm,n} - g_j^{\pm,n-1}\|$) is smaller than 10^{-8}.

6.1 An Example of Multidomain Solution with Time Windows

In this part, we consider Problem (1) on $\Omega =]0,6[$ with $f \equiv 0$, and the final time is $T = 2$. The initial value is $u_0 = e^{-3(2.5-x).^2}$. Ω is split into two subdomains $\Omega_1 = (0,3)$ and $\Omega_2 = (3,6)$. In each subdomain the advection and viscosity coefficients are constant, equal to $a_1 = 0.1$, $\nu_1 = 0.2$, $a_2 = \nu_2 = 1$. The mesh size is $h = 0.06$ for each subdomain. The number of time grid points in each window is $n_1 = 61$ for Ω_1, and $n_2 = 25$ for Ω_2. We denote by "6-windows solution", the approximate solution computed using 6 time windows, with $n_{\rm N} = 2$ iterations of the optimized Schwarz waveform relaxation algorithm in each time window. The initial guess $g_j^{\pm,0}$ on the interface in time window (T_N, T_{N+1}) is given at time T_N by the exact discrete value in the previous window, and is taken to be constant on the time interval. In Figure 1 we observe that at the final time $T = 2$, the 6-windows solution and the 1-window converged solution cannot be distinguished. Since the cost of the computation is much less with time windows, this validates the approach.

6.2 Error Estimates

Constant Coefficients

We first analyze the error in (4) for constant coefficients $a \equiv 1, \nu \equiv 1$. The exact solution is $u(x,t) = \cos(x)\cos(t)$, in $[-\pi/2, \pi/2] \times [0, 2\pi]$. The interface is at $x = 1$. The mesh size is $h = \pi \cdot 10^{-4}$, for each subdomain. The time steps are initially $\Delta t_1 = 2\pi/4$ in Ω_1 and $\Delta t_2 = 2\pi/6$ in Ω_2, and thereafter divided by 2 several times. Let $\Delta t = \max(\Delta t_1, \Delta t_2)$. Figure 2 shows the norms of the error involved in the estimates (5) and (6). The numerical results fit the theoretical estimates.

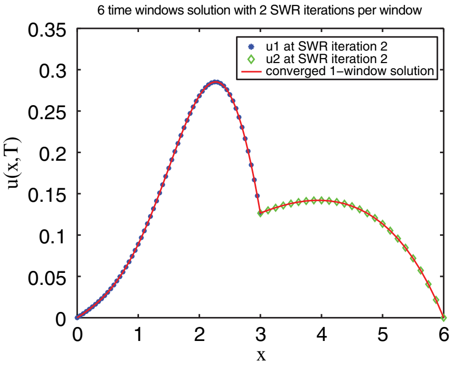


Fig. 1. 1-window converged solution (solid line) and 6-windows solution (star line for Ω_1 and diamond line for Ω_2), at time t=T=2.

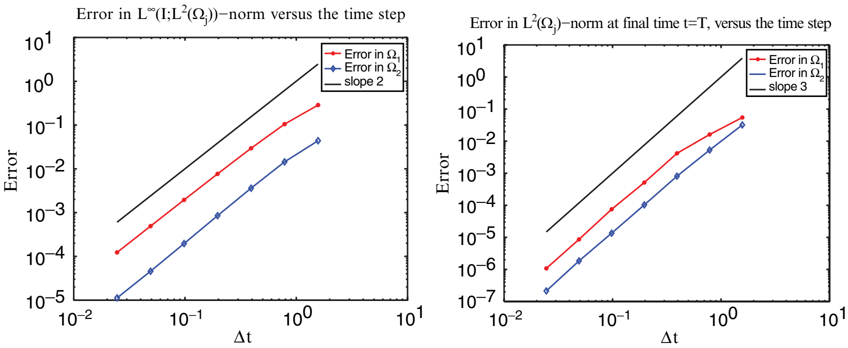


Fig. 2. Error in $L^\infty(I; L^2(\Omega_j))$-norm (on the left) and in $L^2(\Omega_j)$-norm at the final time $t = T$ (on the right), versus the time step Δt, for Ω_1 (star line) and for Ω_2 (diamond line), in the case of constant coefficients.

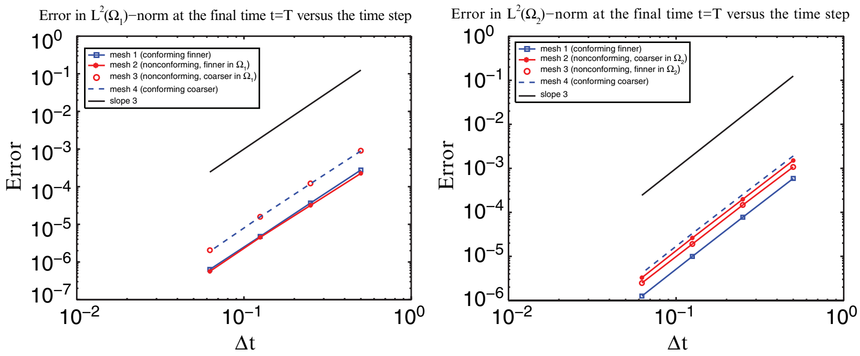


Fig. 3. $L^2(\Omega_j)$ error at the final time $t = T$ versus the time step Δt, for Ω_1 (on the left) and Ω_2 (on the right), for the meshes *Mesh 1,2,3,4* , in the case of discontinuous coefficients.

Discontinuous Coefficients

We consider the configuration in Section 6.1 with one time window. We observe the error between the 1-window converged solution and a reference solution (the 1-window converged solution on a very fine space-time grid), versus the time step. The mesh size is $h = 3 \cdot 10^{-4}$ for each subdomain. We consider four initial meshes in time

- *Mesh 1*: a uniform conforming finner grid with $\Delta t_1 = \Delta t_2 = T/6$,
- *Mesh 2*: a nonconforming grid with $\Delta t_1 = T/6$ and $\Delta t_2 = T/4$,
- *Mesh 3*: a nonconforming grid with $\Delta t_1 = T/4$ and $\Delta t_2 = T/6$,
- *Mesh 4*: a uniform conforming coarser grid with $\Delta t_1 = \Delta t_2 = T/4$.

Thereafter Δt_j, $j = 1, 2$, is divided by two at each computation. Figure 3 shows the error versus the time step $\Delta t = \max(\Delta t_1, \Delta t_2)$, for these four meshes, in Ω_1 (on the left) and in Ω_2 (on the right). The results show that the $L^2(\Omega_j)$ error at discrete time points tends to zero at the same rate as Δt^3, and this fits with the error estimate (6). On the other hand, we observe that the two curves corresponding to the nonconforming meshes are between the curves of the conforming meshes. We obtain similar results for the $L^\infty(I; L^2(\Omega_j))$ error, which fits with (5).

7 Conclusions

We have proposed a new numerical method for the advection-diffusion equation with discontinuous coefficients. It relies on the splitting of the time interval into time windows. In each window a few iterations of a Schwarz waveform relaxation algorithm are performed by a discontinuous Galerkin method, with projection of the time-grids on the interfaces of the spacial subregions. We have shown numerically that our method preserves the order of the discontinuous Galerkin method.

References

[1] D. Bennequin, M.J. Gander, and L. Halpern. Optimized Schwarz Waveform Relaxation for convection reaction diffusion problems. Technical Report 24, LAGA, Université Paris 13, 2004.

[2] E. Blayo, L. Halpern, and C. Japhet. Optimized Schwarz Waveform Relaxation algorithms with nonconforming time discretization for coupling convection-diffusion problems with discontinuous coefficients. In *Domain Decomposition Methods in Science and Engineering XVI*, pages 267–274. Springer, Berlin, 2007.

[3] K. Eriksson and C. Johnson. Adaptive finite element methods for parabolic problems II: optimal error estimates in $L_\infty L_2$ and $L_\infty L_\infty$. *SIAM J. Numer. Anal.*, 32(3):706–740, 1995.

[4] K. Eriksson, C. Johnson, and S. Larsson. Adaptive finite element methods for parabolic problems VI: analytic semigroups. *SIAM J. Numer. Anal.*, 35(4):1315–1325, 1998.

[5] M. Gander, L. Halpern, and M. Kern. Schwarz Waveform Relaxation method for advection–diffusion–reaction problems with discontinuous coefficients and non-matching grids. In *Domain Decomposition Methods in Science and Engineering XVI*, pages 283–290, Berlin, 2007. Springer.

[6] M.J. Gander, L. Halpern, and F. Nataf. Optimized Schwarz Waveform Relaxation method for the one dimensional wave equation. *SIAM J. Numer. Anal.*, 41(5), 2003.

[7] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Cambridge University Press, New York, 1987.

[8] V. Martin. An optimized Schwarz Waveform Relaxation method for unsteady convection diffusion equation. *Appl. Numer. Math.*, 52(4):401–428, 2005.

Optimized and Quasi-optimal Schwarz Waveform Relaxation for the One Dimensional Schrödinger Equation

Laurence Halpern[1] and Jérémie Szeftel[2]

[1] LAGA, Institut Galilée, Université Paris XIII, 93430 Villetaneuse, France. halpern@math.univ-paris13.fr

[2] Department of Mathematics, Princeton University, Fine Hall, Washington Road, Princeton NJ 08544-1000, USA. jszeftel@math.princeton.edu

Summary. We design and study Schwarz Waveform relaxation algorithms for the linear Schrödinger equation with a potential in one dimension. We show that the overlapping algorithm with Dirichlet exchanges of informations on the boundary is slowly convergent, and we introduce two new classes of algorithms: the optimized Robin algorithm and the quasi-optimal algorithm. Numerical results illustrate the great improvement of these methods over the classical algorithm.

1 Introduction

We investigate the design of domain decomposition algorithms for the linear Schrödinger equation with a real potential V, in one space dimension:

$$\begin{cases} i\partial_t u(t,x) + \partial_x^2 u(t,x) + V(x)u(t,x) = 0,\ t \geq 0,\ x \in \mathbb{R}, \\ u(0,x) = u_0(x). \end{cases} \tag{1}$$

This equation is an important model in quantum mechanics, in electromagnetic wave propagation, and in optics (Fresnel equation). To our knowledge, there is no study prior to the present work on domain decomposition methods for the Schrödinger equation.

We first introduce the classical algorithm, with overlapping subdomains, exchanging Dirichlet data on the boundaries. Its slow convergence emphasizes the need for new algorithms.

The key point of these new algorithms is to notice that the convergence in two iterations is obtained when using transparent boundary operators as transmission operators between the subdomains, even in the non-overlapping case. However, these operators are not available for a general potential. Thus, we introduce a quasi-optimal algorithm using the transparent operators corresponding to the value of the potential on the boundary. We also study the possibility of using simpler transmission conditions on the boundary, of complex Robin type.

We then introduce a discretization of the Robin algorithm and a discretization of the quasi-optimal algorithm.

We finally illustrate the results through numerical simulations, for various types of potentials, like constant, barrier, or parabolic. We show how slow the convergence is with Dirichlet Schwarz Waveform Relaxation (SWR), and how the optimized SWR greatly improves the convergence. We also show, that the best results by far are obtained by the discrete quasi-optimal algorithm.

Remark 1. For a more detailed study, we refer the reader to [6].

2 Classical Schwarz Waveform Relaxation

Let $\mathcal{L} := i\partial_t + \partial_x^2 + V(x)$. We decompose the spatial domain $\Omega = \mathbb{R}$ into two overlapping subdomains $\Omega_1 = (-\infty, L)$ and $\Omega_2 = (0, \infty)$, with $L > 0$. The overlapping Schwarz waveform relaxation algorithm consists in solving iteratively subproblems on $\Omega_1 \times (0,T)$ and $\Omega_2 \times (0,T)$, using as a boundary condition at the interfaces $x = 0$ and $x = L$ the values obtained from the previous iteration. The algorithm is thus for iteration index $k = 1, 2, \ldots$ given by

$$\begin{cases} \mathcal{L}u_1^k = f \text{ in } \Omega_1 \times (0,T), \\ u_1^k(\cdot, 0) = u_0 \text{ in } \Omega_1, \\ u_1^k(L, \cdot) = u_2^{k-1}(L, \cdot), \end{cases} \qquad \begin{cases} \mathcal{L}u_2^k = f \text{ in } \Omega_2 \times (0,T), \\ u_2^k(\cdot, 0) = u_0 \text{ in } \Omega_2, \\ u_2^k(0, \cdot) = u_1^{k-1}(0, \cdot). \end{cases} \tag{2}$$

Using the Fourier transform in time, we easily compute the convergence factor of the classical algorithm in the case where the potential V is constant:

$$\Theta(\tau, L) = exp\left[-\left(\frac{-\tau + V + \sqrt{1 + (\tau - V)^2}}{2}\right)^{1/2} L\right], \tag{3}$$

where τ is the time frequency.

The convergence factor in (3) tends to 1 when the overlap L tends to 0, as all overlapping Schwarz methods do. But it also tends to 1 when τ tends to infinity, which differs from what happens for wave equations [5] or parabolic equations [3]. This deterioration of the convergence factor for high frequencies suggests a poor performance of the classical algorithm for the Schrödinger equation. This is confirmed by the numerical results. In figure 1, we present the exact solution and the approximate solution computed with the classical algorithm at various times for the free Schrödinger equation ($V = 0$). The results are displayed after 200 iterations of the algorithm. We take two subdomains $\Omega_1 = (-5, 4\Delta x)$ and $\Omega_2 = (0, 5)$, and the step sizes are $\Delta t = 0.00125$, $\Delta x = 0.0125$.

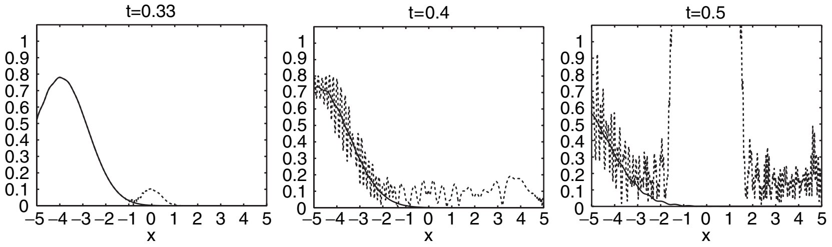


Fig. 1. Exact solution (solid) and approximate solution computed with the classical algorithm at iteration 200 (dashed). **(a)** $t = 0.33$, **(b)** $t = 0.4$, **(c)** $t = 0.5$

Although the algorithm works well for times up to $t = 0.3$, it then deteriorates so that the approximate solution becomes extremely oscillating and does not approximate the exact solution at all. Since this bad behavior happens after 200 iterations, this clearly demonstrates that one should avoid the classical algorithm when computing the Schrödinger equation. This also motivates the need for new algorithms, which we investigate in the next sections.

3 Optimal Schwarz Waveform Relaxation Algorithm

When V is constant, it is possible to compute the optimal algorithm. Let $\mathcal{S}_1$ and $\mathcal{S}_2$ be linear operators acting only in time. We introduce the algorithm

$$\begin{cases} \mathcal{L}u_1^k = f \text{ in } \Omega_1 \times (0,T), \\ u_1^k(\cdot,0) = u_0 \text{ in } \Omega_1, \\ (\partial_x + \mathcal{S}_1)u_1^k(L,\cdot) \\ \qquad\qquad = (\partial_x + \mathcal{S}_1)u_2^{k-1}(L,\cdot), \end{cases} \quad \begin{cases} \mathcal{L}u_2^k = f \text{ in } \Omega_2 \times (0,T), \\ u_2^k(\cdot,0) = u_0 \text{ in } \Omega_2, \\ (\partial_x + \mathcal{S}_2)u_2^k(0,\cdot) \\ \qquad\qquad = (\partial_x + \mathcal{S}_2)u_1^{k-1}(0,\cdot). \end{cases} \tag{4}$$

We define the symbol σ_j of $\mathcal{S}_j(\partial_t)$ by $\sigma_j(\tau) = \mathcal{S}_j(i\tau)$. Using Fourier transform in time, we can prove that the algorithm (4) converges to the solution u of (1) in two iterations independently of the size of the overlap $L \geq 0$, if and only if the operators $\mathcal{S}_1$ and $\mathcal{S}_2$ have the corresponding symbols

$$\sigma_1 = (\tau - V)^{1/2}, \quad \sigma_2 = -(\tau - V)^{1/2} \tag{5}$$

with

$$(\tau - V)^{1/2} = \begin{cases} \sqrt{\tau - V} \text{ if } \tau \geq V, \\ -i\sqrt{-\tau + V} \text{ if } \tau < V. \end{cases} \tag{6}$$

For variable potentials, the optimal operators are in general not at hand. We present here and will compare two approximations of those. The first one is to use a “frozen coefficients” variant of these operators. The second one is to replace them by a constant, obtaining “Robin type” transmission conditions, and to optimize them by minimizing the convergence factor in the constant case.

4 The Quasi-optimal Algorithm

We use as transmission operators the optimal operators for the constant potential equal to the value of V on the interface. The quasi-optimal algorithm is thus for iteration index $k = 1, 2, \dots$ given by

$$\begin{cases} \mathcal{L}u_1^k = f \text{ in } \Omega_1 \times (0,T), \\ u_1^k(\cdot,0) = u_0 \text{ in } \Omega_1, \\ (\partial_x + \sqrt{-i\partial_t - V(L)})u_1^k(L,\cdot) = \\ (\partial_x + \sqrt{-i\partial_t - V(L)})u_2^{k-1}(L,\cdot), \end{cases} \quad \begin{cases} \mathcal{L}u_2^k = f \text{ in } \Omega_2 \times (0,T), \\ u_2^k(\cdot,0) = u_0 \text{ in } \Omega_2, \\ (\partial_x - \sqrt{-i\partial_t - V(0)})u_2^k(0,\cdot) = \\ (\partial_x - \sqrt{-i\partial_t - V(0)})u_1^{k-1}(0,\cdot) \end{cases} \tag{7}$$

where $\sqrt{-i\partial_t - V(x)}$ is the operator acting only in time with symbol given by (6). Though being not differential, this operator is still easy to use numerically [2].

We call the algorithm (7) quasi-optimal, since it is optimal for a constant potential. Even for a non constant potential V, we are able to prove its convergence when there is no overlap, *i.e.* $L = 0$, and when $T = +\infty$ in the following spaces:

$$(H^{1/4}(0,T,L^2(\Omega_1)) \cap H^{-1/4}(0,T,H^1(\Omega_1))) \\ \times (H^{1/4}(0,T,L^2(\Omega_2)) \cap H^{-1/4}(0,T,H^1(\Omega_2))).$$

The proof is based on energy estimates and follows an idea from [7], which has widely been used since (see [4, 8] for steady problems, [5] for evolution equations). Here, the additional difficulty is to deal with the nonlocal operator $\sqrt{-i\partial_t - V(0)}$.

5 The Algorithm with Robin Transmission Conditions

A simple alternative to the previous approach is to use Robin transmission conditions, *i.e.* to replace the optimal operators $\mathcal{S}_j$ by $\mathcal{S}_1 = -\mathcal{S}_2 = -ipI$ where p is a real number, which gives the algorithm

$$\begin{cases} \mathcal{L}u_1^k = f \text{ in } \Omega_1 \times (0,T), \\ u_1^k(\cdot,0) = u_0 \text{ in } \Omega_1, \\ (\partial_x - ip)u_1^k(L,\cdot) \\ \qquad = (\partial_x - ip)u_2^{k-1}(L,\cdot), \end{cases} \quad \begin{cases} \mathcal{L}u_2^k = f \text{ in } \Omega_2 \times (0,T), \\ u_2^k(\cdot,0) = u_0 \text{ in } \Omega_2, \\ (\partial_x + ip)u_2^k(0,\cdot) \\ \qquad = (\partial_x + ip)u_1^{k-1}(0,\cdot). \end{cases} \tag{8}$$

Remark 2. This algorithm is not the usual Robin algorithm as the constant ip used here is complex, whereas the usual Robin algorithm uses a real constant.

Relying on energy estimates, we are able to prove the convergence even for a non constant potential V when there is no overlap, *i.e.* $L = 0$, and for any $p > 0$ in the following spaces:

$$L^\infty(0,T;L^2(\Omega_1)) \times L^\infty(0,T;L^2(\Omega_2)).$$

Of course, the convergence taking place for any $p > 0$, we will optimize the convergence rate with respect to $p > 0$ in order to accelerate the convergence.

6 Construction of the Discrete Algorithms

For the Robin algorithm, we use a finite volume discretization. In the interior, it produces the Crank-Nicolson scheme, widely used in the linear and nonlinear computations for the Schrödinger equation, whereas the Robin transmission conditions are naturally taken into account. This idea was first introduced in [5] for the wave equation in one dimension.

For the discretization of the quasi-optimal algorithm, we also use the Crank-Nicolson scheme on the interior. Here, the main task is to discretize the nonlocal transmission condition. We thus have to discretize the operator $\sqrt{-i\partial_t + V}$. We use the discrete transparent boundary condition designed by Arnold and Ehrhardt precisely for the Crank-Nicolson scheme [2]. It is a discrete convolution:

$$\sqrt{-i\partial_t + V}U(0,n) \simeq \sum_{m=0}^{n} S(n-m)U(0,m),$$

where the convolution kernel $S(m)$ is given by a recurrence formula (see [2]).

Remark 3. Other choices of discrete transparent boundary conditions (for example the one designed in [1]) could be used to discretize the quasi-optimal algorithm.

7 Numerical Results

The physical domain is $(a,b) = (-5,+5)$. It is divided in two subdomains of equal size. Our algorithms are implemented the Gauss-Seidel way, *i.e.* we compute u_1 with g_L, then deduce g_0 by u_1 and give it to the right domain for the computation of u_2. Thus iteration $\#k$ in this section corresponds to the computation of u_1^{2k-1}, u_2^{2k} in the theoretical setting.

7.1 The Free Schrödinger Equation

In the case of the free Schrödinger equation, the quasi-optimal algorithm coincides with the optimal one and converges in two iterations as expected by the theory. It is thus the best algorithm, but we would still like to see how the Robin algorithm behaves and to compare it with the classical algorithm. We consider in Figure 2 an overlap of 2%. The error is the L^2 norm of the error on the boundary of Ω_2. We clearly see the great improvement.

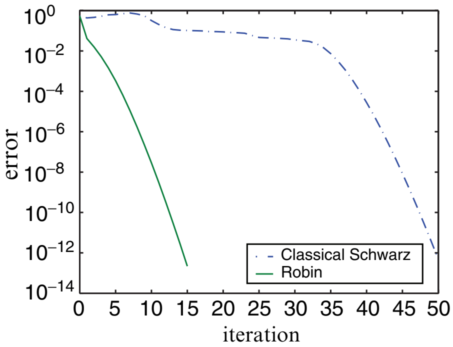


Fig. 2. Convergence history: comparison of the Dirichlet and optimized Robin Schwarz algorithm. $\delta = 2\%$.

7.2 Non Constant Potentials

We consider the interval $(-5, 5)$, with a final time $T = 1$, discretized with $\Delta x = 0.05$ and $\Delta t = 0.005$. The size of the overlap is $4\Delta x$. The potential is a barrier equal to 20 times the characteristic function of the interval $(-1, 1)$. In figure 3, we draw the convergence history for Dirichlet and Robin algorithms. In this case again, the Robin condition behaves much better than the Dirichlet condition.

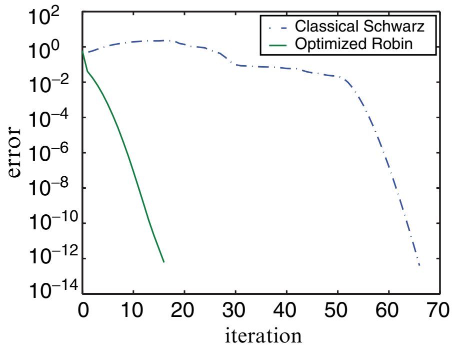


Fig. 3. Convergence history: comparison of the Dirichlet and optimized Robin Schwarz algorithm for a potential barrier. The overlap is equal to 4%.

The quasi-optimal algorithm is by far the most efficient. In all cases, even when the potential is not constant, the precision 10^{-12} is reached in at most five iterations with or without overlap. As an example, we show in Figure 4 the convergence history with an overlap of 8 grid points, for a parabolic potential, for various mesh sizes. The convergence does not depend on the mesh size.

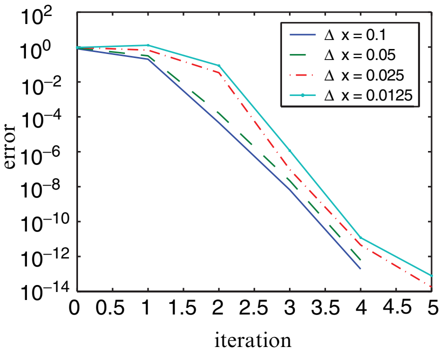


Fig. 4. Convergence history for the quasi-optimal Schwarz algorithm in presence of a parabolic potential

Finally, we present the exact solution and the approximate solution computed with the three algorithms at time $t = 0.9$ for a parabolic potential. The results are displayed after only three iterations of the algorithm. We take two subdomains $\Omega_1 = (-5, 4\Delta x)$ and $\Omega_2 = (0, 5)$, and the step sizes are $\Delta t = 0.0025$ and $\Delta x = 0.025$. As expected, the classical algorithm produces a highly oscillating solution. The Robin algorithm behaves far better and clearly approximates the exact solution. Finally, the quasi-optimal algorithm is the best as we can not distinguish between the exact and the approximate solution.

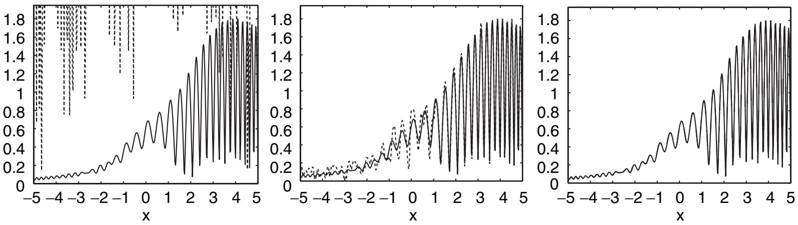


Fig. 5. Exact solution (solid) and approximate solution computed with the three algorithms after 3 iterations (dashed) at time $t = 0.9$. **(a)** Classical algorithm, **(b)** Robin algorithm, **(c)** quasi-optimal algorithm

Remark 4. The Robin algorithm is very sensitive to the value of $p > 0$. In our numerical experiments, we take the optimal value of p obtained for a constant potential V which is given by an explicit formula.

Remark 5. As predicted by the theory, our numerical results indicate that the Robin algorithm and the quasi-optimal algorithm both converge even without overlap unlike the classical algorithm.

8 Conclusion

We have presented here a general approach to design optimized and quasi-optimal domain decomposition algorithms for the linear Schrödinger equation with a potential in one dimension. It allows the use of any discretization, any time and space steps in the subdomains. These algorithms greatly improve the performances of the classical Schwarz relaxation algorithm. We intend to extend our analysis to the two-dimensional case in a close future.

References

[1] X. Antoine and C. Besse. Unconditionally stable discretization schemes of non-reflecting boundary conditions for the one-dimensional Schrödinger equation. *J. Comp. Phys.*, 188(1):157–175, 2003.

[2] A. Arnold and M. Ehrhardt. Discrete transparent boundary conditions for the Schrödinger equation. *Rivista di Matematica dell'Università di Parma*, 6(4):57–108, 2001.

[3] D. Bennequin, M.J. Gander, and L. Halpern. Optimized Schwarz waveform relaxation for convection reaction diffusion problems. Technical Report 2004-24, LAGA, Université Paris 13, 2004. http://www-math.math.univ-paris13.fr/prepub/pp2004/pp2004-24.html.

[4] B. Després. *Méthodes de Décomposition de Domaines pour les Problèmes de Propagation d'Ondes en Régime Harmonique*. PhD thesis, Université Paris IX Dauphine, 1991.

[5] M.J. Gander, L. Halpern, and F. Nataf. Optimal Schwarz waveform relaxation for the one dimensional wave equation. *SIAM J. Numer. Anal.*, 41:1643–1681, 2003.

[6] L. Halpern and J. Szeftel. Optimized and quasi-optimal schwarz waveform relaxation for the one dimensional Schrödinger equation. Technical report, CNRS, 2006. http://hal.ccsd.cnrs.fr/ccsd-00067733/en/.

[7] P.-L. Lions. On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 202–223. SIAM, Philadelphia, PA, 1990.

[8] F. Nataf, F. Rogier, and E. de Sturler. Optimal interface conditions for domain decomposition methods. Technical Report 301, CMAP (Ecole Polytechnique), 1994.

A Moving Mesh Method for Time–dependent Problems Based on Schwarz Waveform Relaxation

Ronald D. Haynes[1], Weizhang Huang[2], and Robert D. Russell[3]

[1] Acadia University, Wolfville, N.S., Canada. ronald.haynes@acadiau.ca
[2] University of Kansas, Lawrence, KS, USA. huang@math.ku.edu
[3] Simon Fraser University, Burnaby, B.C., Canada. rdr@cs.sfu.ca

1 Introduction

It is well accepted that the efficient solution of complex PDEs frequently requires methods which are adaptive in both space and time. Adaptive mesh methods for PDEs may be classified into one or more of the following broad categories:

- r–refinement: moving a fixed number of mesh points to difficult regions of the physical domain,
- p–refinement: varying the order of the numerical method to adapt to local solution smoothness,
- h–refinement: mesh refinement and derefinement, depending upon the local level of resolution.

These methods are applied in either a static fashion, where refining/coarsening or redistributing grids is done at fixed times during a simulation *or* in a dynamic fashion, where the solution and mesh are computed simultaneously.

In this paper we are interested in a class of spatially adaptive moving mesh PDE methods introduced in [17, 11] and [12]. Traditionally, moving mesh methods have been implemented in a (moving) method of lines framework — discretizing spatially and then integrating in time using a stiff initial value problem (IVP) solver. This approach propagates all unknowns (mesh and physical solution) forward in time using identical time steps. It is quite common, however, for problems with moving interfaces or singular behavior to have solution components which evolve on disparate scales in both space and time.

Our purpose is to introduce and explore a natural coupling of domain decomposition, in the Schwarz waveform context, and the spatially adaptive moving mesh PDE methods. This will allow the mesh and physical solution to be evolved according to local space and time scales.

2 Moving Mesh Methods

We consider the solution of a PDE of the form

$$u_t = \mathcal{L}(u) \qquad 0 < x < 1, \quad t > 0,$$

subject to appropriate initial and boundary conditions, where $\mathcal{L}$ denotes a spatial differential operator. The assumption is that the solution of this PDE has features which are difficult to resolve using a uniform mesh in the physical coordinate x. We seek, for fixed t, a one–to–one coordinate transformation

$$x = x(\xi, t) : [0, 1] \to [0, 1], \quad \text{with } x(0, t) = 0,\, x(1, t) = 1$$

such that $u(x(\xi, t), t)$ is sufficiently smooth that a simple (typically uniform) mesh $\xi_i, i = 0, 1, \ldots, N$ can be used to resolve solution features in the computational domain $\xi \in [0, 1]$. The mesh in the physical coordinate x is then specified from the coordinate transformation by $x_i(t) = x(\xi_i, t),\ i = 0, 1, \ldots, N$.

One standard way to perform adaptivity in space is to use the equidistribution principle (EP), introduced by [3]. We assume for the moment that a monitor function, $M = M(t, x)$, measuring the difficulty or error in the numerical solution, is given. Typically, its dependence on t and x is through the physical solution $u = u(t, x)$. Then, equidistribution requires that the mesh points satisfy

$$\int_{x_{i-1}}^{x_i} M(t, \tilde{x})\, d\tilde{x} = \frac{1}{N} \int_0^1 M(t, \tilde{x}) d\tilde{x} \quad \text{for } i = 1, ..., N,$$

or equivalently,

$$\int_0^{x(\xi_i, t)} M(t, \tilde{x})\, d\tilde{x} = \xi_i \int_0^1 M(t, \tilde{x}) d\tilde{x} \quad \text{for } i = 1, \ldots, N.$$

The continuous generalization of this is that

$$\int_0^{x(\xi, t)} M(t, \tilde{x})\, d\tilde{x} = \xi \theta(t), \tag{1}$$

where $\theta(t) \equiv \int_0^1 M(t, \tilde{x})\, d\tilde{x}$ (e.g., see [11]). It follows immediately from (1) that

$$\frac{\partial}{\partial \xi} \left\{ M(t, x(\xi, t)) \frac{\partial}{\partial \xi} x(\xi, t) \right\} = 0. \tag{2}$$

Note that (2) does not explicitly involve the node speed $\dot{x}$. This is generally introduced by relaxing the equation to require equidistribution at time $t + \tau$. A number of parabolic moving mesh PDEs (MMPDEs) are developed using somewhat subtle simplifying assumptions and their correspondence to various heuristically derived moving mesh methods is shown in [17] and [11, 12]. A particularly useful one is MMPDE5,

$$\dot{x} = \frac{1}{\tau M(t, x(\xi, t))} \frac{\partial}{\partial \xi} \left(M(t, x(\xi, t)) \frac{\partial x}{\partial \xi} \right). \qquad \text{(MMPDE5)}$$

The relaxation parameter τ is chosen in practice (cf. [10]) so that the mesh evolves at a rate commensurate with that of the solution $u(t, x)$.

A simple popular choice for $M(t, x)$ is the arclength-like monitor function

$$M(t, x) = \sqrt{1 + \frac{1}{\alpha} |u_x|^2}, \tag{3}$$

based on the premise that the error in the numerical solution is large in regions where the solution has large gradients. It is recommended in [10] (also see [1] and [18]) that the intensity parameter α be chosen as

$$\alpha = \left[\int_0^1 |u_x| dx \right]^2,$$

expecting that about one–half of mesh points are concentrated in regions of large gradients. We note that there are other choices for the monitor function for certain classes of problems, cf. [2] and [15].

Using the coordinate transformation $x = x(\xi, t)$ to rewrite the physical PDE in quasi-Lagrangian form, a moving mesh method is obtained by solving the coupled system

$$\begin{aligned} \dot{u} - u_x \dot{x} &= \mathcal{L}(u), \\ \dot{x} &= \frac{1}{\tau M} \left(M x_\xi \right)_\xi , \end{aligned} \tag{4}$$

where $\dot{u}$ is the total time derivative of u.

Initial and boundary conditions for the physical PDE come from the problem description. On a fixed interval the boundary conditions for the mesh can be specified as $\dot{x}_0 = \dot{x}_N = 0$. If the initial solution $u(x, 0)$ is smooth then it suffices to use a uniform mesh as the initial mesh. Otherwise, an adaptive initial mesh can be obtained by solving MMPDE5 for a monitor function computed based on the initial solution $u(x, 0)$ (cf. [13]).

A typical implementation (cf. [13]) to solve (4) involves spatial discretization and solution of a nonlinear system of ODEs with a stiff ODE solver like DASSL, see [16]. This becomes quite expensive in higher dimensions. Instead we use an alternating solution procedure where the mesh PDE is integrated over a time step for the new mesh and then the physical PDE(s) is integrated with available old and new meshes. The reader is referred to [14] for a detailed description of the alternating solution procedure.

3 The Schwarz Waveform Implementation

Schwarz waveform relaxation methods have garnered tremendous attention as a means of applying domain decomposition strategies to problems in both space and time. Convergence results for linear problems may be found in [7] and [4] and for nonlinear problems in [6]. There are several ways to implement Schwarz waveform relaxation and moving meshes together to design an effective solver. In [8] and [9] the classical Schwarz waveform algorithm is applied to the coupled system of mesh and physical PDEs. Specifically, if x_j and ξ_j denote the physical and computational meshes on each overlapping subdomain Ω_j and the physical solution on each subdomain is denoted by u_j then

$$\begin{aligned} \dot{u}_j^k - \frac{\partial u_j^k}{\partial x}\dot{x}_j^k &= \mathcal{L}(u_j^k) \\ \dot{x}_j^k &= \frac{1}{\tau M(t, x_j^k)} \frac{\partial}{\partial \xi}\left(M(t, x_j^k) \frac{\partial x_j^k}{\partial \xi} \right) \end{aligned} \tag{5}$$

is solved for $j = 1, \ldots, D$ and $k = 1, 2, \ldots$. The boundary values for u_j^k and x_j^k are obtained from the values $u_{j-1}^{k-1}, x_{j-1}^{k-1}$ and $u_{j+1}^{k-1}, x_{j+1}^{k-1}$ from the previous iteration on the respective boundaries of Ω_{j-1} and Ω_{j+1}. If this Schwarz iteration converges it will converge to the mono–domain solution for both the mesh and physical solution.

In this paper we propose an alternate strategy. We apply a Schwarz iteration solver to the physical PDE and obtain the solution by using a moving mesh method on each subdomain, which allows one to use standard moving mesh software. Instead of solving the coupled mesh and physical PDEs on each subdomain, we use the approach mentioned in the previous section and alternately solve for the physical solution and the mesh.

As in the fixed mesh case, the rate of convergence of the classical Schwarz iteration is improved as the size of the overlap is increased, with the faster convergence being offset by the increased computational cost per iteration. Things are further complicated, however, by the desire to isolate difficult regions of the solution from regions where there is little activity. As the overlap is increased more subdomains become "active" requiring smaller time steps in a larger proportion of the physical domain.

4 Numerical Results

In this section we highlight some particular aspects of the moving Schwarz method described in the previous section with the viscous Burgers' equation, a standard test problem for moving mesh methods. Specifically we solve $u_t = \epsilon u_{xx} - \frac{1}{2}(u^2)_x, u(0,t) = 1, u(1,t) = 0$ and $u(x,0) = c - \frac{1}{2}\tanh((x - x_0)/4\epsilon)$. For our experiments we choose $c = 1/2$, $x_0 = 1/10$, and $\epsilon \ll 1$. The solution is a traveling front of thickness $O(\epsilon)$ which moves to the right from x_0 at speed c.

In Figure 1 we illustrate the mesh trajectories generated by a moving mesh method on one domain. The plot shows the time evolution of all mesh

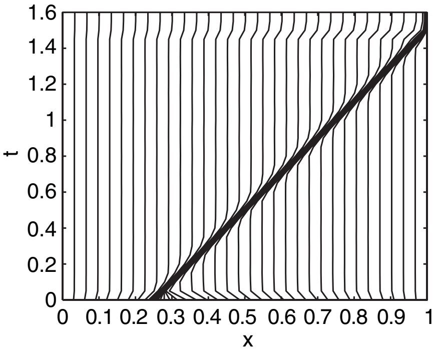
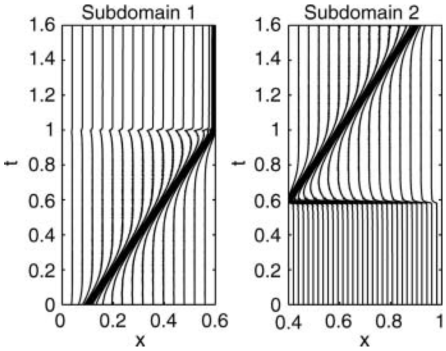


Fig. 1. Mesh trajectories generated on one domain.

points. Initially the mesh points are concentrated at the initial front location ($x_0 = 1/4$). As the solution evolves we see mesh points move in and out of the front location ensuring a sufficient resolution.

Figures 2 and 3 illustrate the meshes obtained using a two domain solution during the first two Schwarz iterations with 10% overlap. We see that the mesh points in subdomain one concentrate and follow the front until it passes into subdomain two. At that point the mesh in subdomain two, which was initially uniform, reacts and resolves the front until it reaches the right boundary. During the first Schwarz iteration the mesh points stay at the right boundary of subdomain one. The right boundary condition for subdomain one is incorrect during the first iteration and the solution presents itself as a layer. During the second iteration, however, the boundary condition issue has basically been resolved and the mesh in subdomain one returns to an essentially uniform state as the front moves into subdomain two.



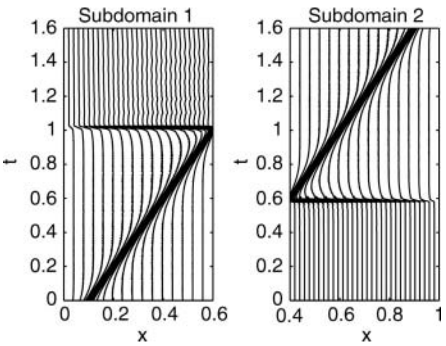


Fig. 2. Mesh trajectories generated on two domains during the first Schwarz iteration.

Fig. 3. Mesh trajectories generated on two domains during the second Schwarz iteration.

In Figures 4, 5, and 6 we show solutions of Burgers' equation using the moving Schwarz method on two subdomains with a 10% overlap. Each Figure shows the solution of the physical PDE on the left and the pointwise error on the right. The left plot of each figure shows the computed solution on subdomain one marked with circles and the solution on subdomain two marked with diamonds. The error plots are annotated in the same way. At $t = 0.8$ during the first Schwarz iteration (Figure 4) the solutions on each subdomain agree, at least qualitatively, with the one domain solution. By $t = 1.4$ (Figure 5), the front has moved across the subdomain boundary and the solution on subdomain one is not correct. This is to be expected since boundary data for subdomain one is incorrect during the first iteration. During the third Schwarz iteration (Figure 6), however, the solutions on both subdomains now agree with the one domain solution to within discretization error.

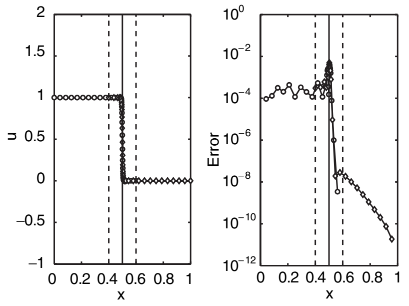


Fig. 4. Solution (left) and pointwise error (right) after Schwarz iteration 1 at $t = 0.8$.

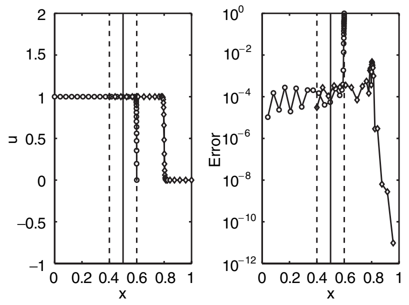


Fig. 5. Solution (left) and pointwise error (right) after Schwarz iteration 1 at $t = 1.4$.

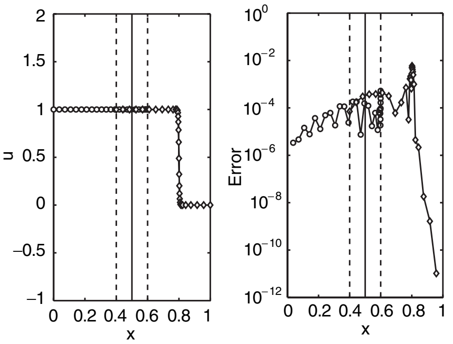


Fig. 6. Solution (left) and pointwise error (right) after Schwarz iteration 3 at $t = 1.4$.

In our experiments, total cpu time is increased as the overlap is increased. Convergence of the Schwarz iteration is rapid for small ϵ (the regime of interest for moving mesh methods), requiring only two or three iterations to reach discretization error. This is consistent with the theoretical results of [6]. Increasing the overlap for this model problem only serves to make each subdomain active for a larger portion of the time interval. Any improvement in the convergence rate is more than offset by the increased cpu time on each subdomain as the overlap is increased.

5 Conclusions

In this paper we propose a moving mesh Schwarz waveform relaxation method. In this approach, classical Schwarz waveform relaxation is applied to the physical PDE and a moving mesh method is used to facilitate the solution on each subdomain. In this way a solution is obtained which benefits both from the domain decomposition approach and the ability to dynamically refine meshes within each subdomain. A careful comparison with previous approaches [9] is ongoing. The benefits of such an approach are likely to be fully realized in two or more space dimensions. This is certainly the subject of current work and interest. The effects of higher order transmission conditions (cf. [5]) are also being studied in this context.

Acknowledgement. The authors would like to acknowledge the support of NSERC (Canada) under discovery grants A8781 and 311796 and by the NSF (USA) under grant DMS–0410545.

References

[1] G. Beckett and J. A. Mackenzie. Uniformly convergent high order finite element solutions of a singularly perturbed reaction-diffusion equation using mesh equidistribution. *Appl. Numer. Math.*, 39(1):31–45, 2001.

[2] C.J. Budd, W. Huang, and R.D. Russell. Moving mesh methods for problems with blow-up. *SIAM J. Sci. Comput.*, 17(2):305–327, 1996.

[3] C. de Boor. Good approximation by splines with variable knots. II. In *Conference on the Numerical Solution of Differential Equations (Univ. Dundee, Dundee, 1973)*, pages 12–20. Lecture Notes in Math., Vol. 363. Springer, Berlin, 1974.

[4] M. J. Gander and A. M. Stuart. Space–time continuous analysis of waveform relaxation for the heat equation. *SIAM J. Sci. Comput.*, 19(6):2014–2031, 1998.

[5] M.J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.

[6] M.J. Gander and C. Rohde. Overlapping Schwarz waveform relaxation for convection-dominated nonlinear conservation laws. *SIAM J. Sci. Comput.*, 27(2):415–439, 2005.

[7] E. Giladi and H.B. Keller. Space-time domain decomposition for parabolic problems. *Numer. Math.*, 93(2):279–313, 2002.

[8] R.D. Haynes. *The Numerical Solution of Differential Equations: Grid Selection for Boundary Value Problems and Adaptive Time Integration Strategies.* PhD thesis, Simon Fraser University, Burnaby, B.C. V5A 1S6, 2003.

[9] R.D. Haynes and R.D. Russell. A Schwarz waveform moving mesh method. *SIAM J. Sci. Comput.*, 2007. To Appear.

[10] W. Huang. Practical aspects of formulation and solution of moving mesh partial differential equations. *J. Comput. Phys.*, 171(2):753–775, 2001.

[11] W. Huang, Y. Ren, and R.D. Russell. Moving mesh methods based on moving mesh partial differential equations. *J. Comput. Phys.*, 113(2):279–290, 1994.

[12] W. Huang, Y. Ren, and R.D. Russell. Moving mesh partial differential equations (MMPDES) based on the equidistribution principle. *SIAM J. Numer. Anal.*, 31(3):709–730, 1994.

[13] W. Huang and R.D. Russell. A moving collocation method for solving time dependent partial differential equations. *Appl. Numer. Math.*, 20(1-2):101–116, 1996.

[14] W. Huang and R.D. Russell. Moving mesh strategy based on a gradient flow equation for two-dimensional problems. *SIAM J. Sci. Comput.*, 20(3):998–1015, 1999.

[15] W. Huang and W. Sun. Variational mesh adaptation. II. Error estimates and monitor functions. *J. Comput. Phys.*, 184(2):619–648, 2003.

[16] L.R. Petzold. A description of DASSL: a differential/algebraic system solver. In *Scientific computing (Montreal, Que., 1982)*, IMACS Trans. Sci. Comput., I, pages 65–68. IMACS, New Brunswick, NJ, 1983.

[17] Y. Ren and R.D. Russell. Moving mesh techniques based upon equidistribution, and their stability. *SIAM J. Sci. Statist. Comput.*, 13(6):1265–1286, 1992.

[18] J. M. Stockie, J. A. Mackenzie, and R. D. Russell. A moving mesh method for one-dimensional hyperbolic conservation laws. *SIAM J. Sci. Comput.*, 22(5):1791–1813, 2000.

MINISYMPOSIUM 5: FETI, Balancing, and Related Hybrid Domain Decomposition Methods

Organizers: Axel Klawonn[1], Olof B. Widlund[2], and Barbara Wohlmuth[3]

[1] University of Duisburg-Essen, Department of Mathematics, Germany. axel.klawonn@uni-due.de
[2] Courant Institute, New York University, USA. widlund@cims.nyu.edu
[3] University of Stuttgart,Institute for Applied Analysis and Numerical Simulation, Germany. wohlmuth@ians.uni-stuttgart.de

The FETI and Balancing domain decomposition algorithms form two important families of iterative methods. They have been implemented and tested for very large applications and have been used extensively in academia as well as in national laboratories in Europe and the United States. Over the years, these methods have been improved and in the FETI family, FETI–DP (dual-primal finite element tearing and interconnection) has proven to be a very robust algorithm which also, when carefully implemented, respects the memory hierarchy of modern parallel and distributed computing systems. This is essential for approaching peak floating point performance.

While the coarse component of these preconditioners typically only has a dimension which is a small multiple of the number of subdomains of the decomposition of the domain, it has become increasingly clear that it can become a bottleneck when the number of subdomains is very large. Solutions of this problem are quite nontrivial. Inexact, rather than exact, solvers of the coarse problem have been developed successfully; see the contribution by Klawonn et al. in which FETI-DP is applied to spectral elements. The problem can also be approached by introducing a third or even more levels. This is demonstrated for BDDC in the paper by Mandel et al..

The FETI–DP methods are very closely related to the BDDC (balancing domain decomposition by constraints) algorithms and, in fact, it has been established that the eigenvalues of the relevant operators essentially are the same, given a pair of methods defined by the same set of constraints; see the paper by Brenner. As demonstrated in the paper by Dostál, et al., the FETI–DP algorithms have also proven quite successful for difficult mechanical contact problems. Another extension from the original studies of linear elliptic problems and lower order finite elements is exemplified by the work by Klawonn et al., which demonstrates that these algorithms perform very well also for spectral element approximations. Still another extension of the BDDC methods is the study of Dryja et al., which develops and analyzes Discontinuous Galerkin Methods.

The basic ideas of the FETI algorithms have also inspired work on new iterative methods for boundary integral methods. The contribution by Of provides a sample of this important development.

Finally, there are two papers by Dohrmann et al., which represent a different development. They concern two issues. The first is the development of a new family of two-level overlapping Schwarz methods with traditional local solvers on overlapping subdomains but where the coarse level solver is inspired by those of iterative substructuring methods, i.e., methods which are based on a partition into non-overlapping subdomains. While the development of theory is just beginning, these methods have proven successful in a number of different applications. The second issue and paper concerns the extension of the theory for one of these methods to the case when the subdomains have quite irregular boundaries; so far these results are for two dimensions only.

We also note that the invited plenary talk by Hyea Hyun Kim concerned BDDC and FETI–DP algorithms for mortar finite elements.

A Functional Analytic Framework for BDDC and FETI-DP

Susanne C. Brenner

Department of Mathematics and Center for Computation and Technology,
Louisianan State University, Baton Rouge, LA 70803, USA.
brenner@math.lsu.edu

1 Introduction

In this paper we present a concise common framework for the BDDC algorithm (cf. [4, 8, 9]) and the FETI-DP algorithm (cf. [6, 5, 10]), using the mathematical language of function spaces, their dual spaces and quotient spaces, and operators. This abstract framework will be illustrated in terms of the following model problem.

Let $\Omega \subset \mathbb{R}^d$ ($d = 2$ or 3) be a bounded polyhedral domain subdivided into J nonoverlapping polyhedral subdomains $\Omega_1, \ldots, \Omega_J$, $\mathcal{T}$ be a triangulation of Ω aligned with the boundaries of the subdomains, and $V(\Omega) \subset H_0^1(\Omega)$ be the P_1 finite element space associated with $\mathcal{T}$. For simplicity, we assume that the subdomains are geometrically conforming, i.e., the intersection of the closures of two distinct subdomains is either empty, a common vertex, a common edge or a common face. The interface of the subdomains is $\Gamma = \bigcup_{j=1}^{J}(\partial\Omega_j \setminus \partial\Omega)$.

The model problem is to find $u \in V(\Omega)$ such that

$$a(u,v) = \sum_{j=1}^{J} a_j(u_j, v_j) = \int_\Omega f v \, dx \qquad \forall\, v \in V(\Omega), \tag{1}$$

where $u_j = u\big|_{\Omega_j}$, $v_j = v\big|_{\Omega_j}$,

$$a_j(u_j, v_j) = \alpha_j \int_{\Omega_j} \nabla u_j \cdot \nabla v_j \, dx,$$

α_j is a positive constant, and $f \in L_2(\Omega)$.

The rest of the paper is organized as follows. The common framework for BDDC and FETI-DP will be presented in Section 2, followed by a discussion of the additive Schwarz formulations of these algorithms in Section 3. Condition number estimates for BDDC and FETI-DP (applied to the model problem) are then sketched in Section 4. Throughout the paper we use $\langle \cdot, \cdot \rangle$ to denote the canonical bilinear form between a vector space V and its dual space V', and the superscript t to denote the transpose of an operator with respect to the canonical bilinear forms.

2 A Common Framework for BDDC and FETI-DP

After parallel subdomain solves, the model problem (1) is reduced to the *interface problem* of computing $u_\Gamma \in V(\Gamma)$ such that

$$a(u_\Gamma, v_\Gamma) = \int_\Omega f v_\Gamma \, dx \qquad \forall\, v_\Gamma \in V(\Gamma), \tag{2}$$

where $V(\Gamma) \subset V(\Omega)$ is the space of discrete harmonic functions whose members satisfy

$$a(v_\Gamma, w) = 0 \qquad \forall\, w \in V(\Omega) \text{ that vanish on } \Gamma.$$

The interface problem (2) is solved by the BDDC method through a preconditioned conjugate gradient algorithm. In the FETI-DP approach, it is first transformed to a dual-primal problem and then solved by a preconditioned conjugate gradient algorithm.

The first ingredient in the common framework for BDDC and FETI-DP is of course the space $V(\Gamma)$. The restriction of $V(\Gamma)$ to Ω_j gives the space $\mathcal{H}_j$ of discrete harmonic functions on Ω_j. The second ingredient is a space $\mathcal{H}_c \subset \mathcal{H}_1 \times \cdots \times \mathcal{H}_J$ of constrained piecewise discrete harmonic functions. The subdomain components of a function in $\mathcal{H}_c$ share certain average values (constraints) along the interface Γ. In particular, we have $V(\Gamma) \subset \mathcal{H}_c$. The constraints (shared averages) are chosen so that (i) the bilinear form $a(\cdot,\cdot)$ remains positive definite on $\mathcal{H}_c$, (ii) the bilinear form $a_j(\cdot,\cdot)$ is positive definite on the subspace of $\mathcal{H}_j$ whose members have vanishing constraints, and (iii) the preconditioned systems in BDDC and FETI-DP have good condition numbers.

Example For the two-dimensional model problem, the constraints that define $\mathcal{H}_c$ are the values of the subdomain components at the corners of the subdomains that are interior to Ω, i.e., $\mathcal{H}_c$ is the space of piecewise discrete harmonic functions that are continuous at these interior corners (cross points). For the three-dimensional model problem, the constraints are the averages along the edges of the subdomains that are interior to Ω, i.e., $\mathcal{H}_c$ is the space of piecewise discrete harmonic functions whose average along any interior edge is continuous across the subdomains sharing the edge.

The third ingredient of the framework is the Schur complement operator $\mathbb{S} : \mathcal{H}_c \longrightarrow \mathcal{H}_c'$ defined by

$$\langle \mathbb{S} v, w \rangle = a(v, w) \qquad \forall\, v, w \in \mathcal{H}_c.$$

Let $I_\Gamma : V(\Gamma) \longrightarrow \mathcal{H}_c$ be the natural injection. Then the interface problem (2) can be written as $S u_\Gamma = \phi_\Gamma$, where

$$S = I_\Gamma^t \mathbb{S} I_\Gamma \tag{3}$$

and $\langle \phi_\Gamma, v\rangle = \int_\Omega f v\,dx \qquad \forall\, v \in V(\Gamma)$.

In the FETI-DP approach, the interface problem (2) is transformed to the equivalent dual-primal problem of finding $(u^c, \lambda) \in \mathcal{H}_c \times [\mathcal{H}_c/V(\Gamma)]'$ such that

$$\begin{array}{lcll} \displaystyle\sum_{j=1}^{J} a_j(u_j^c, v_j) + \langle \lambda, Q_\Gamma v\rangle & = & \displaystyle\int_\Omega f v\,dx & \forall\, v \in \mathcal{H}_c \\ \langle \mu, Q_\Gamma u^c\rangle & = & 0 & \forall\, \mu \in [\mathcal{H}_c/V(\Gamma)]' \end{array} \tag{4}$$

where $Q_\Gamma : \mathcal{H}_c \longrightarrow \mathcal{H}_c/V(\Gamma)$ is the canonical projection, and $[\mathcal{H}_c/V(\Gamma)]'$ plays the role of the space of Lagrange multipliers that enforce the continuity of the constraints along Γ for functions in $\mathcal{H}_c$. Eliminating u^c from (4), we find $S^\dagger \lambda = Q_\Gamma \mathbb{S}^{-1}\phi_c$, where the operator $S^\dagger : [\mathcal{H}_c/V(\Gamma)]' \longrightarrow [\mathcal{H}_c/V(\Gamma)]$ is defined by

$$S^\dagger = Q_\Gamma \mathbb{S}^{-1} Q_\Gamma^t \tag{5}$$

and $\langle \phi_c, v\rangle = \int_\Omega f v\,dx \quad \forall v \in \mathcal{H}_c$.

The final ingredient of the framework is a operator P_Γ that projects $\mathcal{H}_c$ onto $V(\Gamma)$. We can then define the preconditioners $B_{BDDC} : V(\Gamma)' \longrightarrow V(\Gamma)$ and $B_{FETI\text{–}DP} : [\mathcal{H}_c/V(\Gamma)] \longrightarrow [\mathcal{H}_c/V(\Gamma)]'$ by

$$B_{BDDC} = P_\Gamma \mathbb{S}^{-1} P_\Gamma^t, \qquad B_{FETI\text{–}DP} = L_\Gamma^t \mathbb{S} L_\Gamma, \tag{6}$$

where the lifting operator $L_\Gamma : \mathcal{H}_c/V(\Gamma) \longrightarrow \mathcal{H}_c$ is given by

$$L_\Gamma(v + V(\Gamma)) = v - I_\Gamma P_\Gamma v \qquad \forall\, v \in \mathcal{H}_c. \tag{7}$$

Example For our model problems the projection operator P_Γ is defined by weighted averaging:

$$(P_\Gamma v)(p) = \Big(\frac{1}{\sum_{j\in\sigma_p} \alpha_j^\gamma}\Big) \sum_{\ell\in\sigma_p} \alpha_\ell^\gamma v_\ell(p) \qquad \forall\, p \in \mathcal{N}_\Gamma, \tag{8}$$

where $\mathcal{N}_\Gamma$ = the set of nodes on Γ, σ_p = the index set for the subdomains that share p as a common boundary node, and γ is any number $\geq 1/2$. The key property of this weighted averaging is that

$$\alpha_k \alpha_\ell^\gamma / (\sum_{j\in\sigma_p} \alpha_j^\gamma) \leq \alpha_\ell \qquad \forall\, k, \ell \in \sigma_p. \tag{9}$$

In summary, the system operators for the BDDC and FETI-DP methods and their preconditioners are defined in terms of the four ingredients $V(\Gamma)$, $\mathcal{H}_c$, $\mathbb{S}$ and P_Γ through (3) and (5)–(7).

It is easy to see that

$$P_\Gamma I_\Gamma = Id_{V(\Gamma)}, \quad Q_\Gamma L_\Gamma = Id_{\mathcal{H}_c/V(\Gamma)} \quad \text{and} \quad I_\Gamma P_\Gamma + L_\Gamma Q_\Gamma = Id_{\mathcal{H}_c}. \tag{10}$$

The following result (cf. [9, 7, 3]) on the spectra of $B_{BDDC}S$ and $B_{FETI\text{-}DP}S^\dagger$ follows from the three relations in (10).

Theorem 1. *It holds that* $\lambda_{\min}(B_{BDDC}S) \geq 1$*,* $\lambda_{\min}(B_{FETI\text{-}DP}S^\dagger) \geq 1$*, and* $\sigma(B_{BDDC}S)\backslash\{1\} = \sigma(B_{FETI\text{-}DP}S^\dagger)\backslash\{1\}$*. Furthermore, the multiplicity of any common eigenvalue different from 1 is identical for* $B_{BDDC}S$ *and* $B_{FETI\text{-}DP}S^\dagger$*.*

3 Additive Schwarz Formulations

The additive Schwarz formulations of B_{BDDC} and $B_{FETI\text{-}DP}$ involve the spaces $\overset{\circ}{\mathcal{H}}_j = \{v \in \mathcal{H}_j : E_j v \in \mathcal{H}_c\}$, where E_j is the trivial extension operator defined by

$$E_j v = \begin{cases} v & \text{on } \Omega_j \\ 0 & \text{on } \Omega \setminus \Omega_j \end{cases}, \tag{11}$$

and the Schur complement operators $S_j : \overset{\circ}{\mathcal{H}}_j \longrightarrow \overset{\circ}{\mathcal{H}}{}'_j$ defined by

$$\langle S_j v, w \rangle = a_j(v, w) \qquad \forall\, v, w \in \overset{\circ}{\mathcal{H}}_j.$$

Example $\overset{\circ}{\mathcal{H}}_j$ is precisely the space of discrete harmonic functions on Ω_j whose interface constraints are identically zero. For the 2D model problem these functions vanish at the corners of Ω_j. For the 3D model problem, they have zero averages along the edges of Ω_j. Note that $a_j(\cdot,\cdot)$ is positive definite on $\overset{\circ}{\mathcal{H}}_j$.

We can now introduce the coarse space

$$\mathcal{H}_0 = \{v \in \mathcal{H}_c : a_j(v_j, w_j) = 0 \quad \forall\, w_j \in \overset{\circ}{\mathcal{H}}_j,\ 1 \leq j \leq J\},$$

and define the Schur complement operator $S_0 : \mathcal{H}_0 \longrightarrow \mathcal{H}'_0$ by

$$\langle S_0 v, w \rangle = a(v, w) \qquad \forall\, v, w \in \mathcal{H}_0.$$

Lemma 1. *The inverse of* $\mathbb{S}$ *can be written as*

$$\mathbb{S}^{-1} = \sum_{j=0}^{J} E_j S_j^{-1} E_j^t, \tag{12}$$

where E_j *for* $1 \leq j \leq J$ *is defined in* (11) *and* $E_0 : \mathcal{H}_0 \longrightarrow \mathcal{H}_c$ *is the natural injection.*

Proof. Let $v \in \mathcal{H}_c$ be arbitrary. Then we have a unique decomposition $v = \sum_{k=0}^{J} E_k v_k$, where $v_0 \in \mathcal{H}_0$ and $v_k \in \overset{\circ}{\mathcal{H}}_k$ for $1 \le k \le J$, and

$$\begin{aligned}\Big[\sum_{j=0}^{J} E_j S_j^{-1} E_j^t\Big]\mathbb{S}v &= \Big[\sum_{j=0}^{J} E_j S_j^{-1} E_j^t\Big]\mathbb{S}\Big[\sum_{k=0}^{J} E_k v_k\Big]\\ &= \sum_{j=0}^{J} E_j S_j^{-1} E_j^t \mathbb{S} E_j v_j = \sum_{j=0}^{J} E_j v_j = v,\end{aligned}$$

where we have used the facts that $E_j^t \mathbb{S} E_k = 0$ if $j \ne k$ and $S_j = E_j^t \mathbb{S} E_j$.

It follows from (6) and (12) that

$$B_{BDDC} = \sum_{j=0}^{J} (P_\Gamma E_j) S_j^{-1} (P_\Gamma E_j)^t. \tag{13}$$

Let $\overset{\circ}{\mathcal{H}} = \sum_{j=1}^{J} E_j \overset{\circ}{\mathcal{H}}_j$ be the subspace of $\mathcal{H}_c$ whose members have zero interface constraints. Note that the lifting operator defined by (7) actually maps $\mathcal{H}_c/V(\Gamma)$ to $\overset{\circ}{\mathcal{H}}$, since the interface constraints of a function $v \in \mathcal{H}_c$ are preserved by the weighted averaging operator P_Γ. Therefore we can factorize L_Γ as

$$L_\Gamma = \overset{\circ}{I} \circ \overset{\circ}{L}_\Gamma,$$

where $\overset{\circ}{L}_\Gamma : \mathcal{H}_c/V(\Gamma) \longrightarrow \overset{\circ}{\mathcal{H}}$ is defined by the same formula in (7) and the operator $\overset{\circ}{I} : \overset{\circ}{\mathcal{H}} \longrightarrow \mathcal{H}_c$ is the natural injection. We can then write

$$B_{FETI\text{-}DP} = \overset{\circ}{L}{}_\Gamma^t (\overset{\circ}{I}{}^t \mathbb{S} \overset{\circ}{I}) \overset{\circ}{L}_\Gamma. \tag{14}$$

The following lemma can be established by arguments similar to those in the proof of Lemma 1.

Lemma 2. *We have*

$$\overset{\circ}{I}{}^t \mathbb{S} \overset{\circ}{I} = \sum_{j=1}^{J} R_j^t S_j R_j, \tag{15}$$

where $R_j : \overset{\circ}{\mathcal{H}} \longrightarrow \overset{\circ}{\mathcal{H}}_j$ *is the restriction operator.*

It follows from (14) and (15) that

$$B_{FETI\text{-}DP} = \sum_{j=1}^{J} (R_j \overset{\circ}{L}_\Gamma)^t S_j (R_j \overset{\circ}{L}_\Gamma). \tag{16}$$

The formulations (13) and (16) allow both algorithms to be analyzed by the additive Schwarz theory (cf. [2, 11] and the references therein).

4 Condition Number Estimates

In view of Theorem 1, the preconditioned systems in the BDDC and FETI-DP methods have similar behaviors. Here we will sketch the condition number estimates for the BDDC method applied to our model problem. Since we already know that $\lambda_{\min}(B_{BDDC}S) \geq 1$, it only remains to find an upper bound for $\lambda_{\max}(B_{BDDC}S)$ using the following formula from the theory of additive Schwarz preconditioners (cf. [2]):

$$\lambda_{\max}(B_{BDDC}S) = \max_{v \in V(\Gamma)\setminus\{0\}} \frac{\langle Sv, v\rangle}{\displaystyle\min_{\substack{v=\sum_{j=0}^{J} P_\Gamma E_j v_j \\ v_0 \in \mathcal{H}_0,\, v_j \in \overset{\circ}{\mathcal{H}}_j\ (1\leq j\leq J)}} \sum_{j=0}^{J}\langle S_j v_j, v_j\rangle} \tag{17}$$

Let w be a discrete harmonic function on a subdomain Ω_j and the geometric object $\mathcal{G}$ be either a corner c ($\dim\mathcal{G}=0$), an open edge e ($\dim\mathcal{G}=1$) or an open face f ($\dim\mathcal{G}=2$) of Ω_j. We will denote by $w_{\mathcal{G}}$ the discrete harmonic function that agrees with w at the nodes on $\mathcal{G}$ and vanishes at all other nodes. The following estimate (cf. [2, 11] and the references therein) is crucial for the condition number estimate of the model problem:

$$|w_{\mathcal{G}}|^2_{H^1(\Omega_j)} \leq C\Big(1+\ln\frac{H_j}{h_j}\Big)^{3-d+\dim\mathcal{G}} |w|^2_{H^1(\Omega_j)}, \tag{18}$$

where $d=2$ or 3, H_j is the diameter of Ω_j, and h_j is the mesh size of the quasi-uniform triangulation which is the restriction of $\mathcal{T}$ to Ω_j. We assume that w vanishes at one of the corners of Ω_j when $d=2$ and that w has zero average along one of the edges of Ω_j. (Henceforth we use C to denote a generic positive constant that can take different values at different occurrences.)

Furthermore, if $w \in V(\Gamma)$, then it follows from the equivalence of $|w|_{H^1(\Omega_j)}$ and $|w|_{H^{1/2}(\partial\Omega_j)}$ (cf. [2, 11]) that

$$|w_{\mathcal{G}}|_{H^1(\Omega_k)} \leq C|w_{\mathcal{G}}|_{H^1(\Omega_\ell)} \tag{19}$$

if Ω_k and Ω_ℓ share the common geometric object $\mathcal{G}$.

Let $v \in V(\Gamma)$ be arbitrary and $v=\sum_{j=0}^{J} P_\Gamma E_j v_j$ be any decomposition of v, where $v_0 \in \mathcal{H}_c$ and $v_j \in \overset{\circ}{\mathcal{H}}_j$ for $1 \leq j \leq J$. We want to show

$$\langle Sv, v\rangle \leq C\Big(1+\ln\frac{H}{h}\Big)^2 \sum_{j=0}^{J}\langle S_j v_j, v_j\rangle, \tag{20}$$

where $H/h = \max_{1\leq j\leq J}(H_j/h_j)$.

Observe first that

$$\langle Sv, v\rangle = \langle S\sum_{j=0}^{J} P_\Gamma E_j v, \sum_{k=0}^{J} P_\Gamma E_k v\rangle$$

$$\leq 2\Big[\langle SP_\Gamma E_0 v_0, P_\Gamma E_0 v_0\rangle + \langle S\sum_{j=1}^{J} P_\Gamma E_j v_j, \sum_{j=k}^{J} P_\Gamma E_k v\rangle\Big] \tag{21}$$

$$\leq C\sum_{j=0}^{J}\langle SP_\Gamma E_j v, P_\Gamma E_j v\rangle,$$

where we have used the fact that each v_j $(1 \leq j \leq J)$ only interacts with functions from a few subdomains. Therefore, it remains only to relate $\langle SP_\Gamma E_j v_j, P_\Gamma E_j v_j\rangle$ to $\langle S_j v_j, v_j\rangle$.

Let $w = v_0 - P_\Gamma E_0 v_0$. Then w vanishes at the corners of the subdomains when $d = 2$ and has zero averages along the edges of the subdomains when $d = 3$. We can write

$$w = \sum_{1\leq j\leq J}\Big(\sum_{\mathrm{c}\in\mathcal{C}_j} w_{\mathrm{c}} + \sum_{\mathrm{e}\in\mathcal{E}_j} w_{\mathrm{e}} + \sum_{\mathrm{f}\in\mathcal{F}_j} w_{\mathrm{f}}\Big)$$

where $\mathcal{C}_j$ (resp. $\mathcal{E}_j$ and $\mathcal{F}_j$) is the set of the corners (resp. edges and faces) of Ω_j ($\mathcal{C}_j = \emptyset = \mathcal{F}_j$ for $d = 2$), and apply (8), (9), (18) and (19) to obtain the estimate

$$\langle Sw, w\rangle \leq C\sum_{j=1}^{J}\Big(1 + \ln\frac{H_j}{h_j}\Big)^2 \alpha_j |v_0|^2_{H^1(\Omega_j)} \leq C\Big(1 + \ln\frac{H}{h}\Big)^2\langle S_0 v_0, v_0\rangle,$$

which together with the triangle inequality implies that

$$\langle SP_\Gamma E_0 v_0, P_\Gamma E_0 v_0\rangle \leq C\Big(1 + \ln\frac{H}{h}\Big)^2\langle S_0 v_0, v_0\rangle. \tag{22}$$

Similarly, we have the estimate

$$\langle SP_\Gamma E_j v_j, P_\Gamma E_j v_j\rangle \leq C\Big(1 + \ln\frac{H_j}{h_j}\Big)^2\langle S_j v_j, v_j\rangle \quad \text{for} \quad 1 \leq j \leq J. \tag{23}$$

The estimate (20) follows from (21)–(23).

We see from (20) that

$$\langle Sv, v\rangle \leq C\Big(1 + \ln\frac{H}{h}\Big)^2 \min_{\substack{v=\sum_{j=0}^{J} P_\Gamma E_j v_j \\ v_0\in\mathcal{H}_0,\, v_j\in\mathring{\mathcal{H}}_j\ (1\leq j\leq J)}} \sum_{j=0}^{J}\langle S_j v_j, v_j\rangle. \tag{24}$$

Combining (17) and (24) we have the estimate

$$\lambda_{\max}(B_{BDDC}S) \leq C\Big(1 + \ln\frac{H}{h}\Big)^2$$

and hence the following theorem on the condition number of $B_{BDDC}S$, which has also been obtained in [8] and [9] by a different approach.

Theorem 2. *For the model problem we have*

$$\kappa(B_{BDDC}S) = \frac{\lambda_{\max}(B_{BDDC}S)}{\lambda_{\min}(B_{BDDC}S)} \le C\Big(1 + \ln\frac{H}{h}\Big)^2,$$

where the positive constant C is independent of h_j, H_j, α_j and J.

Finally we remark that for the model problem the estimate

$$\kappa(B_{FETI\text{–}DP}S^{\dagger}) \le C\Big(1 + \ln\frac{H}{h}\Big)^2$$

follows from Theorem 1 and Theorem 2. A direct analysis of $B_{FETI\text{–}DP}S^{\dagger}$ by the additive Schwarz theory can also be found in [1].

Acknowledgement. The work in this paper was partially supported by the National Science Foundation under Grant No. DMS-03-11790. The author would also like to thank the Alexander von Humboldt Foundation for support through her Humboldt Research Award.

References

[1] S.C. Brenner. Analysis of two-dimensional FETI-DP preconditioners by the standard additive Schwarz framework. *ETNA*, 16:165–185, 2003.

[2] S.C. Brenner and L.R. Scott. *The Mathematical Theory of Finite Element Methods.* Springer-Verlag, New York-Berlin-Heidelberg, second edition, 2002.

[3] S.C. Brenner and Li-yeng Sung. BDDC and FETI-DP without matrices and vectors. *Comput. Methods Appl. Mech. Engrg.*, 2006.

[4] C.R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25:246–258, 2003.

[5] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, and D. Rixen. FETI-DP: A dual-primal unified FETI method, Part I: A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50:1523–1544, 2001.

[6] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.*, 7:687–714, 2000.

[7] J. Li and O. Widlund. FETI-DP, BDDC, and block Cholesky methods. *Internat. J. Numer. Methods Engrg.*, 66:250–271, 2005.

[8] J. Mandel and C.R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.*, 10:639–659, 2003.

[9] J. Mandel, C.R. Dohrmann, and R. Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.*, 54:167–193, 2005.

[10] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.

[11] A. Toselli and O.B. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Springer-Verlag, Berlin, 2005.

A Family of Energy Minimizing Coarse Spaces for Overlapping Schwarz Preconditioners

Clark R. Dohrmann[1], Axel Klawonn[2], and Olof B. Widlund[3]

[1] Sandia National Laboratories, Albuquerque, USA. crdohrm@sandia.gov
[2] Universität Duisburg-Essen, Essen, Germany. axel.klawonn@uni-due.de
[3] New York University, New York, USA. widlund@courant.nyu.edu

Summary. A simple and effective approach is presented to construct coarse spaces for overlapping Schwarz preconditioners. The approach is based on energy minimizing extensions of coarse trace spaces, and can be viewed as a generalization of earlier work by Dryja, Smith, and Widlund. The use of these coarse spaces in overlapping Schwarz preconditioners leads to condition numbers bounded by $C(1 + H/\delta)(1 + \log(H/h))$ for certain problems when coefficient jumps are aligned with subdomain boundaries. For problems without coefficient jumps, it is possible to remove the $\log(H/h)$ factor in this bound by a suitable enrichment of the coarse space. Comparisons are made with the coarse spaces of two other substructuring preconditioners. Numerical examples are also presented for a variety of problems.

1 Introduction

In order to introduce the subject of this paper, consider the linear system

$$Ax = b, \tag{1}$$

where A is a coefficient matrix, x is a vector of unknowns, and b is a known vector. The coarse space for x can be defined as the range of an interpolation matrix Φ. The vector of unknowns for overlapping subdomain i can be expressed as $R_i x$, where each row of the restriction matrix R_i has a single nonzero entry of unity. We can now express a two-level, additive, overlapping Schwarz preconditioner for A concisely as

$$M^{-1} = \Phi A_0^{-1} \Phi^T + \sum_{i=1}^{N} R_i^T A_i^{-1} R_i, \tag{2}$$

where N is the number of subdomains, and

$$A_0 = \Phi^T A \Phi, \qquad A_i = R_i A R_i^T. \tag{3}$$

Detailed introductions to overlapping Schwarz preconditioners can be found in [15] and [16]. If restriction matrices R_i are available, we see from (2) and (3) that the only missing ingredient for M^{-1} is the interpolation matrix Φ. The subject of this paper is an approach to constructing Φ.

If A in (1) originates from a finite element discretization of an elliptic partial differential equation, then Φ can be constructed using the shape functions of a coarser discretization. One obvious shortcoming of such an approach is that it requires an auxiliary finite element mesh. To address this shortcoming, *algebraic* approaches have been developed that do not require a second mesh. Examples of these include smoothed aggregation [1, 8] and partition of unity methods [12]. The approach presented here is also an algebraic approach and can be viewed as a generalization of earlier work in [6]; see also Section 5.4.3 of [16] for a description.

A common perception is that condition number bounds for iterative substructuring approaches are superior to those of their overlapping Schwarz counterparts for problems with large jumps in material properties. Although proofs are not provided here, it can be shown, under the usual assumptions for substructuring, that use of the subject coarse spaces in overlapping Schwarz preconditioners leads to condition number bounds that are competitive with iterative substructuring for certain problems. In addition, for problems with constant material properties, the coarse spaces can be enriched to give the classic bounds for two-level overlapping Schwarz preconditioners whose coarse spaces are based on coarse triangulations. We note that some other coarse spaces well suited for overlapping Schwarz preconditioners and problems with jumps in material properties can be found in [5, 13, 7, 14].

The paper is organized as follows. The subject approach for constructing coarse spaces is described in Section 2. Comparisons with two different substructuring preconditioners are given in Section 3. Some of the theoretical results available to date are summarized in Section 4. Section 5 provides numerical examples for the Poisson equation, elasticity, plate bending, and problems in $H(\mathrm{curl};\Omega)$.

2 Our Approach

Consider a finite element mesh, and let $\Omega_1, \ldots, \Omega_N$ denote a partitioning of its elements into nonoverlapping subdomains. Thus, each element is contained in exactly one subdomain. Decomposing a mesh into subdomains can be readily done using graph partitioning software.

Given a decomposition into nonoverlapping subdomains, the only other required input is a coarse matrix G. This matrix has the same number of rows as x in (1) and its number of columns is flexible. The most important feature of G is that its columns span the rigid body modes of each subdomain. We note that the coarse space in Algorithm 6.10 of [6] and Algorithm 5.16 of [16] is identical to the present one for the special case of scalar partial differential equations and G chosen as a vector with all entries equal to unity. Accordingly, we use the acronym GDSW for generalized Dryja, Smith, Widlund coarse space.

As in (1), let x denote the vector of degrees of freedom (dofs) for the original problem. Similarly, let x_Γ denote the vector of dofs in x shared by two or more subdomains. We then have $x_\Gamma = R_\Gamma x$, where each row of the restriction matrix R_Γ has exactly one nonzero entry of unity. The vector x_Γ can be expressed in partitioned form as

$$x_\Gamma = \sum_{j=1}^{M} R_{\Gamma_j}^T x_{\Gamma_j}, \tag{4}$$

where $x_{\Gamma_j} = R_{\Gamma_j} x_\Gamma$. As with the other subscripted R matrices, each row of R_{Γ_j} contains exactly one nonzero entry equal to unity. The partitioning in (4) is chosen such that all dofs in x_{Γ_j} are connected and common to the same set of subdomains. Thus, the dofs in x_{Γ_j} form an equivalence class.

The coarse approximation of x_{Γ_j} is expressed as

$$x_{\Gamma_{jc}} = G_{\Gamma_j} q_j \tag{5}$$

for some q_j, where the columns of G_{Γ_j} form a basis for the columns of $R_{\Gamma_j} R_\Gamma G$. Accordingly, from (4) and (5) the coarse approximation of x_Γ is given by

$$x_{\Gamma_c} = \sum_{j=1}^{M} R_{\Gamma_j}^T G_{\Gamma_j} q_j = \Phi_\Gamma q \tag{6}$$

for some q. The coarse space for the remaining dofs, not on subdomain boundaries, is obtained from energy minimizing extensions of x_{Γ_c} into subdomain interiors. We note that these extensions require either exact or approximate (with some care) solutions of subdomain problems with nonhomogeneous essential boundary conditions. All of these problems are local to each subdomain and can be solved in parallel. Notice that the support of coarse basis functions associated with G_{Γ_j} only includes those subdomains having Γ_j a part of their boundaries. Thus, the coarse basis functions have local support.

To obtain an explicit expression for the interpolation matrix Φ, define

$$x_c = R_\Gamma^T x_{\Gamma_c} + R_I^T x_I, \tag{7}$$

where R_I is a restriction matrix to subdomain interiors and x_I is the corresponding vector of interior dofs. Substituting (6) into (7) and minimizing the potential $x_c^T A x_c$ with respect to x_I then leads to

$$x_c = (R_\Gamma^T \Phi_\Gamma + R_I^T \Phi_I) q = \Phi q,$$

where

$$\Phi_I = -(R_I A R_I^T)^{-1} R_I A R_\Gamma^T \Phi_\Gamma .$$

3 Comparisons

In this section we make some broad comparisons with the coarse spaces for the BDD, [10] and BDDC, [2, 11] approaches. The results are summarized in Table 1. Regarding Point 3, the sparsity of the coarse stiffness matrix for BDD is not as nice as the other two because coupling can occur between nonadjacent subdomains. Of the three approaches compared, notice that the present one (GDSW) is the only one not requiring individual subdomain matrices. Concerning Point 8, the problem considered is a unit cube decomposed into N cubic subdomains. Notice that the coarse problem dimension is significantly larger for GDSW than for the other two approaches. We note, however, that the $9N$ figure for BDDC would be somewhat larger to effectively handle certain problems with large material property jumps. Regarding Point 9, we comment that special considerations must be made in order for BDD and BDDC to effectively handle nearly incompressible elasticity problems. In contrast, no special considerations are needed for GDSW. The primary reason for this can be linked to the large coarse space dimension of GDSW.

Table 1. Comparisons of coarse spaces for three different approaches. Results under the heading GDSW are for the present approach.

Point		BDD	BDDC	GDSW
1	well suited for elasticity problems	yes	yes	yes
2	well suited for plate bending problems	yes	yes	yes
3	nice coarse problem sparsity	no	yes	yes
4	individual subdomain matrices required	yes	yes	no
5	null space information required	yes	no	yes
6	simple multilevel extensions	no	yes	yes
7	theory for coefficient jumps	yes	yes	yes
8	3D elasticity coarse problem dimension	$6N$	$9N$	$36N$
9	well suited for nearly incompressible elasticity	yes	yes	yes

4 Theory

Theoretical results for two-level overlapping Schwarz preconditioners which use the subject coarse spaces have been obtained for the Poisson equation and for isotropic elasticity provided the Poisson ratio ν is bounded away from 1/2. Because of space limitations, additional details and proofs are given elsewhere [3]. Under the usual assumptions for substructuring given in Section 4.2 of [16], we have the condition number bound

$$\kappa(M^{-1}A) \leq C(1 + H/\delta)(1 + \log(H/h)), \tag{8}$$

provided the columns of the coarse matrix G of Section 2 span the rigid body modes of the problem operator. The constant C is independent of both the number of subdomains and possible jumps in material properties across subdomain boundaries. The term H/h is the ratio of the subdomain diameter to that of the elements and H/δ is the typical ratio of H and overlap widths. For problems without coefficient jumps, the $\log(H/h)$ term in (7) can be removed anytime the columns of G span all linear functions of the spatial coordinates.

For stable, mixed finite element formulations of elasticity that are based on continuous interpolation of displacement and discontinuous interpolation of pressure, the pressure dofs can be eliminated at the element level provided $\nu < 1/2$. Such an elimination process results in a finite element with only displacement dofs. Numerical results and initial theoretical work for problems that use such elements suggest that condition number bounds exist which are insensitive to ν being arbitrarily close to the incompressible limit of 1/2. The bound in (8), however, has an $(H/\delta)^3$ dependence [4].

The coarse spaces considered here have also proven useful in the analysis of overlapping Schwarz [3] and iterative substructuring [9] methods on irregular subdomains in two dimensions. Efforts are underway to extend these results to irregular subdomains in three dimensions.

Numerical results in the next section suggest that the coarse spaces also work well for plate bending and $H(\mathrm{curl};\Omega)$ problems in 2D, but we presently have no supporting theory. In addition, a suitable coarse space for $H(\mathrm{curl};\Omega)$ problems in 3D has not yet been identified.

5 Numerical Examples

Results are presented for unit square domains with homogeneous essential boundary conditions on all four sides. The stable $\mathbb{Q}_2 - \mathbb{P}_1$ element is used in the nearly incompressible elasticity examples. This element uses continuous biquadratic interpolation of displacement and discontinuous linear interpolation of pressure. Moreover, its pressure dofs are eliminated at the element level. The standard bilinear element $\mathbb{Q}_1$ is used for all the other elasticity and Poisson equation examples. The plate bending examples use the discrete Kirchoff triangular element and the lowest-order quadrilateral edge element is used for the $H(\text{curl}; \Omega)$ examples. Except for the $H(\text{curl}; \Omega)$ examples, the columns of the coarse matrix G described in Section 2 span the rigid body modes of the problem operator.

Equation (1) is solved to a relative residual tolerance of 10^{-8} for a random vector b using preconditioned conjugate gradients. In addition to numbers of iterations, condition number estimates obtained from the conjugate gradient iterations are also reported. The overlap width δ is defined as the minimum distance between a subdomain boundary and the boundary of its overlapping extension. Unless stated otherwise, the values of the elastic modulus and Poisson ratio ν are 1 and 0.3, respectively. The elasticity results are for plane strain conditions.

5.1 Poisson Equation, Compressible Elasticity, Plate Bending

Results for fixed values of H/h, H/δ, and increasing numbers of square subdomains are shown in Table 2. Good scalability with respect to the number of subdomains N is evident for all three problem types. We now fix $N = 16$ and $H/\delta = 4$ while increasing the ratio H/h. The slow growth in iterations and condition numbers shown in Table 3 is consistent with the estimate in (8). Results for a problem with the elastic modulus equal to σ in a square centered region of length 1/2 and equal to 1 elsewhere are shown in Table 4. Material property jumps are aligned with subdomain boundaries and there is no great sensitivity to σ in the numerical results.

Table 2. Iterations (iter) and condition number estimates (cond) for increasing numbers of subdomains N. Fixed values of $H/h = 8$ and $H/\delta = 4$ are used.

N	Poisson Equation		Linear Elasticity		Plate Bending	
	iter	cond	iter	cond	iter	cond
16	24	8.97	24	6.93	41	17.7
64	27	10.0	26	7.52	48	19.8
256	28	10.3	28	8.01	51	21.1
1024	30	10.4	29	8.28	55	21.7

5.2 Nearly Incompressible Elasticity

Results for a fixed value of H/δ are shown in Table 5 for three different values of the Poisson ratio ν. As noted earlier, the stable $\mathbb{Q}_2 - \mathbb{P}_1$ element is used. Good scalability with respect to the number of subdomains is evident for all three values

Table 3. Results for $N = 16$ and $H/\delta = 4$.

H/h	Poisson Equation		Linear Elasticity		Plate Bending	
	iter	cond	iter	cond	iter	cond
8	24	8.97	24	6.93	41	17.7
16	25	10.5	25	7.87	46	23.4
24	25	11.3	26	8.38	48	26.2
32	26	11.9	27	8.73	50	28.0
40	26	12.3	27	8.99	48	29.4

Table 4. Results for elastic modulus equal to σ in a square centered region and 1 elsewhere. Fixed values of $N = 16$, $H/h = 8$, and $H/\delta = 4$ are used.

σ	Poisson Equation		Linear Elasticity		Plate Bending	
	iter	cond	iter	cond	iter	cond
10^{-4}	23	6.93	24	6.02	38	13.7
10^{-2}	23	7.05	23	6.05	40	15.4
1	24	8.97	24	6.93	41	17.7
10^{2}	24	10.5	26	7.72	39	17.3
10^{4}	24	10.5	27	7.74	38	15.3

of ν. Table 5 also shows results for 16 subdomain and different values of H/h. As in the previous examples, the number of iterations and condition number estimates grow slowly as H/h increases. Notice in all the examples that the ratio H/δ has been fixed. Although the relevant numerical results are not presented here, we have observed a stronger dependence on H/δ than in (8) for problems with ν very close to 1/2.

Table 5. Plane strain results for $H/\delta = 4$.

$H/h = 8$							$N = 16$						
N	$\nu = 0.3$		$\nu = 0.4999$		$\nu = 0.49999$		H/h	$\nu = 0.3$		$\nu = 0.4999$		$\nu = 0.49999$	
	iter	cond	iter	cond	iter	cond		iter	cond	iter	cond	iter	cond
16	25	8.05	31	10.6	33	10.6	8	25	8.05	31	10.6	33	10.6
64	29	8.93	32	11.2	34	11.2	16	27	8.89	33	12.3	34	12.3
256	32	9.67	34	11.6	35	11.7	24	28	9.35	34	13.4	36	13.4
1024	33	10.1	34	11.7	35	11.7	32	28	9.67	35	14.1	36	14.1
4096	34	10.3	34	11.7	35	11.8	40	28	9.90	34	14.6	36	14.7

5.3 $\boldsymbol{H}(\mathbf{curl}; \boldsymbol{\Omega})$ Examples

We now consider examples for the bilinear form

$$a(\mathbf{u}, \mathbf{v}) = \int_{\boldsymbol{\Omega}} (\alpha(\nabla \times \mathbf{u}) \cdot (\nabla \times \mathbf{v}) + \beta \mathbf{u} \cdot \mathbf{v})\, \mathbf{dx},$$

where $\alpha \geq 0$, $\beta > 0$, and $\nabla \times \mathbf{u}$ denotes the curl of $\mathbf{u}$. We assume that edge element shape functions are scaled so that the integral of the tangential component along each edge of an element is unity. Assuming a consistent sign convention for each element edge of a subdomain edge, the matrix G is chosen as a vector with all entries equal to unity.

To simplify the computer implementation, an overlapping subdomain is chosen to include all edges a graph distance m or less from the edges of the nonoverlapping subdomain. Results for fixed values of β, H/h, and m are shown in Table 6 for different values of α and N. Similar results for increasing values of H/h are shown in Table 7. In contrast to the previous examples, monotonic growth of condition number estimates with H/h is not evident. This may be caused by our choice of overlapping subdomains, but the results are quite acceptable.

Table 6. $H(\text{curl}; \Omega)$ results for $H/h = 8$, $m = 1$, and $\beta = 1$.

N	$\alpha = 0.0$		$\alpha = 10^{-2}$		$\alpha = 1$		$\alpha = 10^2$		$\alpha = 10^4$	
	iter	cond	iter	cond	iter	cond	iter	cond	iter	cond
16	6	3.01	20	5.28	25	7.38	28	7.48	30	7.49
32	6	3.01	22	5.96	26	7.47	28	7.53	31	7.54
64	6	3.01	23	6.43	26	7.52	29	7.56	31	7.57
100	6	3.01	24	6.77	27	7.58	30	7.61	32	7.62

Table 7. $H(\text{curl}; \Omega)$ results for $N = 16$, $H/(mh) = 8$, and $\beta = 1$.

H/h	$\alpha = 0.0$		$\alpha = 10^{-2}$		$\alpha = 1$		$\alpha = 10^2$		$\alpha = 10^4$	
	iter	cond	iter	cond	iter	cond	iter	cond	iter	cond
8	6	3.01	20	5.28	25	7.38	28	7.48	30	7.49
16	4	3.00	21	5.61	25	7.46	28	7.52	30	7.53
24	4	3.00	21	5.76	25	7.39	27	7.41	29	7.45
32	4	3.00	21	5.85	25	7.47	26	7.52	29	7.53
40	3	3.00	21	5.88	25	7.30	27	7.45	29	7.49

6 Conclusions

A simple and effective approach to constructing coarse spaces for overlapping Schwarz preconditioners has been presented. Initial numerical and theoretical results suggest that it could be a viable approach for a variety of problem types. There remain several opportunities for future discovery and development from both a theoretical and practical point of view.

References

[1] M. Brezina and P. Vaněk. A black box iterative solver based on a two-level Schwarz method. *Computing*, 63:233–263, 1999.

[2] C. R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.

[3] C. R. Dohrmann, A. Klawonn, and O. B. Widlund. Domain decomposition for less regular subdomains: Overlapping Schwarz in two dimensions. Technical Report TR2007–888, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, March 2007.

[4] C. R. Dohrmann and O. B. Widlund. An overlapping Schwarz preconditioner for almost incompressible elasticity. In preparation, 2007.

[5] M. Dryja, M. V. Sarkis, and O. B. Widlund. Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.*, 72(3):313–348, 1996.

[6] M. Dryja, B. F. Smith, and O. B. Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal.*, 31(6):1662–1694, 1994.

[7] I. G. Graham, P. Lechner, and R. Scheichl. Domain decomposition for multiscale pdes. Technical report, Bath Institute for Complex Systems, 2006.

[8] E. W. Jenkins, C. E. Kees, C. T. Kelley, and C. T. Miller. An aggregation-based domain decomposition preconditioner for groundwater flow. *SIAM J. Sci. Comput.*, 23(2):430–441, 2001.

[9] A. Klawonn, O. Rheinbach, and O. B. Widlund. An analysis of a FETI-DP algorithm on irregular subdomains in the plane. Technical Report TR2007–889, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, April 2007.

[10] J. Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9:233–241, 1993.

[11] J. Mandel and C. R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.*, 10(7):639–659, 2003.

[12] M. Sarkis. Partition of unity coarse spaces and Schwarz methods with harmonic overlap. *Lecture Notes in Computational Science and Engineering*, 23:77–94, 2002.

[13] M. Sarkis. Partition of unity coarse spaces: Enhanced versions, discontinuous coefficients and applications to elasticity. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *Fourteenth International Conference on Domain Decomposition Methods*, 2003.

[14] R. Scheichl and E. Vainikko. Additive Schwarz and aggregation-based coarsening for elliptic problems with highly variable coefficients. Technical report, Bath Institute for Complex Systems, 2006.

[15] B. Smith, P. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.

[16] A. Toselli and O. Widlund. *Domain Decomposition Methods – Algorithms and Theory.* Springer Series in Computational Mathematics. Springer, 2005.

Extending Theory for Domain Decomposition Algorithms to Irregular Subdomains

Clark R. Dohrmann[1], Axel Klawonn[2], and Olof B. Widlund[3]

[1] Sandia National Laboratories, Albuquerque, USA. crdohrm@sandia.gov
[2] Universität Duisburg-Essen, Essen, Germany. axel.klawonn@uni-due.de
[3] Courant Institute, New York University, New York, USA.
widlund@cims.nyu.edu

1 Introduction

In the theory of iterative substructuring domain decomposition methods, we typically assume that each subdomain is quite regular, e.g., the union of a small set of coarse triangles or tetrahedra; see, e.g., [13, Assumption 4.3]. However, this is often unrealistic especially if the subdomains result from using a mesh partitioner. The subdomain boundaries might then not even be uniformly Lipschitz continuous. We note that existing theory establishes bounds on the convergence rate of the algorithms which are insensitive to even large jumps in the material properties across subdomain boundaries as reflected in the coefficients of the problem. The theory for overlapping Schwarz methods is less restrictive as far as the subdomain shapes are concerned, see e.g. [13, Chapter 3], but little has been known on the effect of large changes in the coefficients; see however [11] and recent work [6] and [12].

The purpose of this paper is to begin the development of a theory under much weaker assumptions on the partitioning. We will focus on a recently developed overlapping Schwarz method, see [4], which combines a coarse space adopted from an iterative substructuring method, [13, Algorithm 5.16], with local preconditioner components selected as in classical overlapping Schwarz methods, i.e., based on solving problems on overlapping subdomains. This choice of the coarse component will allow us to prove results which are independent of coefficient jumps. We note that there is an earlier study of multigrid methods [5] in which the coarsest component is similarly borrowed from iterative substructuring algorithms.

We will use nonoverlapping subdomains, and denote them by $\Omega_i, i = 1, \ldots, N$, as well as overlapping subdomains $\Omega'_j, j = 1, \ldots, N'$. The interface between the Ω_i will be denoted by Γ.

So far, complete results have only been obtained for problems in the plane. Although our results also hold for compressible plane elasticity, we will confine ourselves to scalar elliptic problems of the following form:

$$-\nabla \cdot (\rho(x) \nabla u(x)) \;=\; f(x), \quad x \in \Omega \subset \mathbb{R}^2, \tag{1}$$

with a Dirichlet boundary condition on a measurable subset $\partial\Omega_D$ of $\partial\Omega$, the boundary of Ω, and a Neumann condition on $\partial\Omega_N = \partial\Omega \setminus \partial\Omega_D$. The coefficient $\rho(x)$ is strictly positive and assumed to be a constant ρ_i for $x \in \Omega_i$. We use piecewise linear, continuous finite elements and triangulations with shape regular elements and assume that each subdomain is the union of a set of quasi uniform elements. The weak formulation of the elliptic problem is written in terms of a bilinear form,

$$a(u,v) \ := \ \sum_{i=1}^{N} a_i(u,v) \ := \ \sum_{i=1}^{N} \rho_i \int_{\Omega_i} \nabla u \cdot \nabla v dx.$$

Our study requires the generalization of some technical tools used in the proof of a bound of the convergence rate of this type of algorithm; see [3, 8]. Some of the standard tools are no longer available and we have to modify the basic line of reasoning in the proof of our main result. Three auxiliary results, namely a Poincaré inequality, a Sobolev-type inequality for finite element functions, and a bound for certain edge terms, will be required in our proof; see Lemmas 2, 3, and 4. We will work with John domains, see Section 2, and will be able to express our bounds on the convergence of our algorithm in terms of a few geometric parameters.

2 John Domains and a Poincaré Inequality

We first give a definition of a John domain; see [7] and the references therein.

Definition 1 (John domain). *A domain $\Omega \subset \mathbb{R}^n$ – an open, bounded, and connected set – is a John domain if there exists a constant $C_J \geq 1$ and a distinguished central point $x_0 \in \Omega$ such that each $x \in \Omega$ can be joined to it by a curve $\gamma : [0,1] \rightarrow \Omega$ such that $\gamma(0) = x,\ \gamma(1) = x_0$ and $dist(\gamma(t), \partial\Omega) \geq C_J^{-1}|x - \gamma(t)|$ for all $t \in [0,1]$.*

This condition can be viewed as a twisted cone condition. We note that certain snowflake curves with fractal boundaries are John domains and that the length of

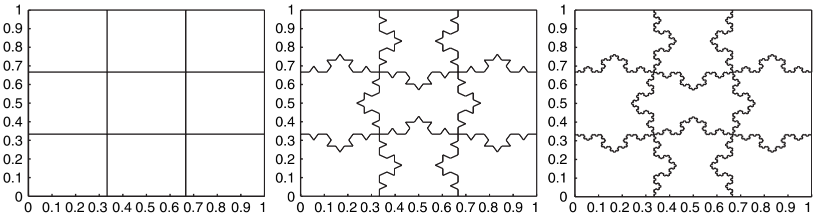


Fig. 1. The subdomains are obtained by first partitioning the unit square into smaller squares. We then replace the middle third of each edge by the other two edges of an equilateral triangle, increasing the length by a factor 4/3. The middle third of each of the resulting shorter edges is then replaced in the same way and this process is repeated until we reach the length scale of the finite element mesh.

the boundary of a John domain can be arbitrarily much larger than its diameter; see Figure 1.

In any analysis of any domain decomposition method with more than one level, we need a Poincaré inequality. This inequality is closely related to an isoperimetric inequality; see [10].

Lemma 1 (Isoperimetric inequality). *Let $\Omega \subset \mathbb{R}^n$ be a domain and let f be sufficiently smooth. Then,*

$$\inf_{c\in\mathbb{R}} \left(\int_\Omega |f-c|^{n/(n-1)}\, dx \right)^{(n-1)/n} \leq \gamma(\Omega,n) \int_\Omega |\nabla f|\, dx,$$

if and only if,

$$[\min(|A|,|B|)]^{1-1/n} \leq \gamma(\Omega,n)|\partial A \cap \partial B|. \qquad (2)$$

Here, $A \subset \Omega$ is arbitrary, and $B = \Omega \setminus A$; $\gamma(\Omega,n)$ is the best possible constant and $|A|$ is the measure of the set A, etc.

We note that the domain does not need to be star-shaped or Lipschitz. For two dimensions, we immediately obtain a standard Poincaré inequality by using the Cauchy-Schwarz inequality.

Lemma 2 (Poincaré's inequality). *Let $\Omega \subset \mathbb{R}^2$ be a domain. Then,*

$$\inf_{c\in\mathbb{R}} \|u-c\|^2_{L_2(\Omega)} \leq (\gamma(\Omega,2))^2|\Omega|\|\nabla u\|^2_{L_2(\Omega)}, \quad \forall u \in H^1(\Omega).$$

For $n=3$ such a bound is obtained by using Hölder's inequality several times. In Lemma 2, we then should replace $|\Omega|$ by $|\Omega|^{2/3}$. The best choice of c is $\bar{u}_\Omega$, the average of u over the domain.

Throughout, we will use a weighted $H^1(\Omega_i)-$norm defined by

$$\|u\|^2_{H^1(\Omega_i)} := |u|^2_{H^1(\Omega_i)} + 1/H_i^2\|u\|^2_{L_2(\Omega_i)} := \int_{\Omega_i} \nabla u \cdot \nabla u dx \; + \; 1/H_i^2 \int_{\Omega_i} |u|^2 dx.$$

Here, H_i is the diameter of Ω_i. The weight originates from a dilation of a domain with diameter 1. We will use Lemma 2 to remove L_2-terms from full H^1-norms.

3 The Algorithm, Technical Tools, and the Main Result

The domain $\Omega \subset \mathbb{R}^2$ is decomposed into nonoverlapping subdomains Ω_i, each of which is the union of finite elements, and with the finite element nodes on the boundaries of neighboring subdomains matching across the interface Γ, which is the union of the parts of the subdomain boundaries which are common to at least two subdomains. The interface Γ is composed of edges and vertices. An edge $\mathcal{E}^{ij}$ is an open subset of Γ, which contains the nodes which are shared by the boundaries of a particular pair of subdomains Ω_i and Ω_j. The subdomain vertices $\mathcal{V}^k$ are end points of edges and are typically shared by more than two; see [9, Definition 3.1] for more details on how these sets can be defined for quite general situations. We denote the standard finite element space of continuous, piecewise linear functions on Ω_i by $V^h(\Omega_i)$ and assume that these functions vanish on $\partial\Omega_i \cap \partial\Omega_D$.

We will view our algorithm as an additive Schwarz method, as in [13, Chapter 2], being defined in terms of a set of subspaces. To simplify the discussion, we will use exact solvers for both the coarse problem and the local ones. All that is then required for the analysis of our algorithm is an estimate of a parameter in a stable decomposition of any elements in the finite element space; see [13, Assumption 2.2 and Lemma 2.5]. Thus, we need to estimate C_0^2 in

$$\sum_{j=0}^{N'} a(u_j, u_j) \leq C_0^2 a(u,u), \quad \forall u \in V^h,$$

for some $\{u_j\}$, such that

$$u = \sum_{j=0}^{N'} R_j^T u_j, \quad u_j \in V_j.$$

Here $R_j^T \; : \; V_j \longrightarrow V^h$ is an interpolation operator from the space of the j-th subproblem, defined on Ω_j', into the space V^h. By using [13, Lemmas 2.5 and 2.10], we find that the condition number $\kappa(P_{ad})$ of the additive Schwarz operator can be bounded by $(N^C + 1)C_0^2$ where N^C is the minimal number of colors required to color the subdomains Ω_j' such that no pair of intersecting subdomains have the same color.

Associated with each space V_j is a projection P_j defined by

$$a(\tilde{P}_j u, v) = a(u,v), \; \forall v \in V_j, \text{ and } P_j = R_j^T \tilde{P}_j.$$

The additive Schwarz operator, the preconditioned operator used in our iteration, is

$$P_{ad} = \sum_{j=0}^{N'} P_j.$$

The coarse space V_0, which is described differently in [4], is spanned by functions defined by their values on the interface and extended as discrete harmonic functions into the interior of the subdomains Ω_i. The discrete harmonic extensions minimize the energy; see [13, Section 4.4]. There is one basis function, $\theta_{\mathcal{V}^k}(x)$, for each subdomain vertex; it is the discrete harmonic extension of the standard nodal basis function. There is also a basis function, $\theta_{\mathcal{E}^{ij}}(x)$, for each edge $\mathcal{E}^{ij}$, which equals 1 at all nodes on the edge and vanishes at all other interface nodes. The vertex and edge functions provide a partition of unity.

The local spaces $V_j, j = 1, \ldots N'$, are defined as

$$V_j \;=\; V^h(\Omega_j') \cap H_0^1(\Omega_j').$$

This is the standard choice as in [13, Chapter 3]. We assume that each Ω_j' has a diameter comparable to those of the subdomains Ω_i which intersect Ω_j'; we also assume that neighboring subdomains Ω_i and Ω_j have comparable diameters. The overlap between the subdomains is characterized by parameters δ_j, as in [13, Assumption 3.1]; δ_j is the minimum width of the subset Ω_{j,δ_j} of Ω_j' which is also covered by neighboring overlapping subdomains. We will assume that the width of Ω_{j,δ_j} is on the order of δ_j everywhere; our arguments can easily be extended to a more general case.

We can now formulate our main result, which is also valid for compressible elasticity with piecewise constant Lamé parameters, provided that the coarse space is enriched as in [4].

Theorem 1 (Condition number estimate). *Let $\Omega \subset \mathbb{R}^2$ be an arbitrary John domain with a shape regular triangulation. The condition number then satisfies*

$$\kappa(P_{ad}) \leq C\,(1 + H/\delta)(1 + \log(H/h))^2,$$

where $C > 0$ is a constant which only depends on the John and Poincaré parameters, the number of colors required for the overlapping subdomains, and the shape regularity of the finite elements.

Here, H/h is shorthand for $\max_i(H_i/h_i)$, as in many domain decomposition papers; h_i is the diameter of the smallest element of Ω_i. Similarly, H/δ is the largest ratio of H_i and the smallest of the δ_j of the subregions Ω'_j that intersect Ω_i.

The logarithmic factors of our main result can be improved to a first power if a sufficiently large subset of each subdomain edge is Lipschitz. If the coefficients do not have large jumps across the interface, the coarse space is suitably enriched, and the subregions satisfy [13, Assumption 4.3], we can eliminate the logarithmic factors altogether.

To prove this theorem, we need two auxiliary results, in addition to Poincaré's inequality. The first is a discrete Sobolev inequality:

Lemma 3 (Discrete Sobolev inequality).

$$\|u\|^2_{L_\infty(\Omega_i)} \leq C(1 + \log(H/h))\|u\|^2_{H^1(\Omega_i)}, \quad \forall u \in V^h(\Omega_i). \tag{3}$$

The constant $C > 0$ depends only on the John parameter and the shape regularity of the finite elements.

The inequality (3) is well-known in the theory of iterative substructuring methods. Proofs for domains satisfying an interior cone condition are given in [1] and in [2, Sect. 4.9].

The second important tool provides estimates of the edge functions.

Lemma 4 (Edge functions). *The edge function $\theta_{\mathcal{E}^{ij}}$ can be bounded as follows:*

$$\|\theta_{\mathcal{E}^{ij}}\|^2_{H^1(\Omega_i)} \leq C(1 + \log(H_i/h_i)), \tag{4}$$

and

$$\|\theta_{\mathcal{E}^{ij}}\|^2_{L_2(\Omega_i)} \leq CH_i^2(1 + \log(H_i/h_i)). \tag{5}$$

Proofs of Lemmas 3 and 4 are given in [3] and [8], respectively. We note that inequality (4) can be established using ideas similar to those in [13, Section 4.6.3]. The proof of inequality (5) requires a new idea. We note that a uniform L_2-bound holds for more regular edges or if all angles of the triangulation are acute.

4 Proof of Theorem 1

As in many other proofs of results on domain decomposition algorithms, we can work on one subdomain at a time. With local bounds, there are no difficulties in handling variations of the coefficients across the interface.

We recall that the coarse space is spanned by the $\theta_{\mathcal{V}^k}$, the discrete harmonic extensions of the restrictions of the standard nodal basis functions to Γ, and the edge functions $\theta_{\mathcal{E}^{ij}}$. The vertex basis functions have bounded energy, while, according to (4), the edge functions have an energy that grows in proportion to $(1+\log(H/h))$. The coarse space component $u_0 \in V_0$ in the decomposition of an arbitrary finite element function $u(x)$ is chosen as

$$u_0(x) \;=\; \sum_k u(\mathcal{V}^k)\theta_{\mathcal{V}^{ik}}(x) + \sum_{ij} \bar{u}_{\mathcal{E}^{ij}}\theta_{\mathcal{E}^{ij}}(x).$$

Here, $\bar{u}_{\mathcal{E}^{ij}}$ is the average of u over the edge. This interpolation formula is the two-dimensional analog of [13, Formula (5.13)] and it reproduces constants. In the case of regular edges, we can estimate the edge averages by using the Cauchy–Schwarz inequality and an elementary trace theorem. In our much more general case, we instead get two logarithmic factors by estimating the edge averages by $\|u\|_{L_\infty}$ and using Lemmas 3 and 4. The norms of the vertex terms of u_0 are bounded by one logarithmic factor. Replacing $u(x)$ by $u(x)-\bar{u}_{\Omega_i}$ and using Lemma 2, to remove the L_2-terms of the H^1-norms, we find that

$$|u_0|^2_{H^1(\Omega_i)} \le C(1+\log(H/h))^2|u|^2_{H^1(\Omega_i)},$$

and

$$a(u_0,u_0) \le C(1+\log(H/h))^2 a(u,u).$$

Similarly, we can prove

$$\|u-u_0\|^2_{L_2(\Omega_i)} \le C(1+\log(H/h))^2 H_i^2|u|^2_{H^1(\Omega_i)}. \tag{6}$$

In the case of regular subdomain boundaries, or if all angles of the triangulation are acute, no logarithmic factors are necessary in (6).

We now turn to the estimate related to the local spaces. Again, we will carry out the work on one subdomain Ω_i at a time. Let $w := u-u_0$ and define a local term in the decomposition by $u_j = I^h(\theta_j w)$. We will borrow extensively from [13, Sections 3.2 and 3.6]. Thus, I^h interpolates into V^h and the θ_j, supported in Ω_j', provide a partition of unity. These functions vary between 0 and 1 and their gradients are bounded by $|\nabla\theta_j| \le C/\delta_j$ and they vanish outside the areas of overlap.

We note only a fixed number of Ω_j' intersect Ω_i; we will only consider the contribution from one of them, Ω_j'. As in our earlier work, the only term that requires a careful estimate is $\nabla\theta_j w$. We cover the set Ω_{j,δ_j} with patches of diameter δ_j and note that on the order of H_i/δ_j of them will suffice. Just as in the proof of [13, Lemma 3.11], we have

$$\int_{\Omega_i} |\nabla\theta_j w|^2 \le C/\delta_j^2\big(\delta_j^2|w|^2_{H^1(\Omega_i)} + (H_i/\delta_j)\delta_j^2\|w\|^2_{H^1(\Omega_i)}\big).$$

The proof is completed by using (6) and the bound on the energy of u_0.

Acknowledgement. The authors wish to thank Professor Fanghua Lin of the Courant Institute for introducing us to John domains and Poincaré's inequality for very general domains and Dr. Oliver Rheinbach for producing Figure 1.

Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

This work was supported in part by the U.S. Department of Energy under contracts DE-FG02-06ER25718 and DE-FC02-01ER25482 and in part by National Science Foundation Grant DMS-0513251.

References

[1] J.H. Bramble, J.E. Pasciak, and A.H. Schatz. The construction of preconditioners for elliptic problems by substructuring. I. *Math. Comp.*, 47(175):103–134, 1986.

[2] S.C. Brenner and L. Ridgway Scott. *The Mathematical Theory of Finite Element Methods.* Springer-Verlag, New York, 2nd edition, 2002.

[3] C.R. Dohrmann, A. Klawonn, and O.B. Widlund. Domain decomposition for less regular subdomains: Overlapping Schwarz in two dimensions. Technical Report TR2007-888, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, March 2007.

[4] C.R. Dohrmann, A. Klawonn, and O.B. Widlund. A family of energy minimizing coarse spaces for overlapping Schwarz preconditioners. In these proceedings., 2007.

[5] M. Dryja, M.V. Sarkis, and O.B. Widlund. Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.*, 72(3):313–348, 1996.

[6] I.G. Graham, P. Lechner, and R. Scheichl. Domain decomposition for multiscale pdes. Technical report, Bath Institute for Complex Systems, 2006.

[7] P. Hajłasz. Sobolev inequalities, truncation method, and John domains. In *Papers on Analysis*, volume 83 of *Rep. Univ. Jyväskylä Dep. Math. Stat.*, pages 109–126. Univ. Jyväskylä, Jyväskylä, 2001.

[8] A. Klawonn, O. Rheinbach, and O.B. Widlund. An analysis of a FETI–DP algorithm on irregular subdomains in the plane. Technical Report TR2007-889, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, April 2007.

[9] A. Klawonn and O.B. Widlund. Dual-Primal FETI methods for linear elasticity. *Comm. Pure Appl. Math.*, 59:1523–1572, 2006.

[10] V. G. Maz'ja. Classes of domains and imbedding theorems for function spaces. *Soviet Math. Dokl.*, 1:882–885, 1960.

[11] M. Sarkis. Partition of unity coarse spaces: enhanced versions, discontinuous coefficients and applications to elasticity. In *Domain Decomposition Methods in Science and Engineering*, pages 149–158 (electronic). Natl. Auton. Univ. Mex., México, 2003.

[12] R. Scheichl and E. Vainikko. Additive Schwarz and aggregation-based coarsening for elliptic problems with highly variable coefficients. Technical report, Bath Institute for Complex Systems, 2006.

[13] A. Toselli and O.B. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Springer-Verlag, Berlin Heidelberg New York, 2005.

Scalable FETI Algorithms for Frictionless Contact Problems

Zdeněk Dostál[1], Vít Vondrák[1], David Horák[1], Charbel Farhat[2], and Philip Avery[2]

[1] Department of Applied Mathematics, Faculty of Electrical Engineering
and Computer Science, VŠB-Technical University of Ostrava, and
Center of Intelligent Systems and Structures, CISS - Institute of
Thermomechanics AVČR,
17. listopadu 15, Ostrava-Poruba, 708 33, Czech Republic.
{zdenek.dostal,vit.vondrak,david.horak}@vsb.cz

[2] Stanford University, Department of Mechanical Engineering and Institute for
Computational and Mathematical Engineering, Stanford, CA 94305, USA.
{cfarhat,pavery}@stanford.edu

Summary. We review our FETI based domain decomposition algorithms for the solution of 2D and 3D frictionless contact problems of elasticity and related theoretical results. We consider both cases of restrained and unrestrained bodies. The scalability of the presented algorithms is demonstrated on the solution of 2D and 3D benchmarks.

1 Introduction

The Finite Element Tearing and Interconnecting (FETI) method was originally proposed by Farhat and Roux (see [15]) as a parallel solver for problems described by elliptic partial differential equations. The computational domain is decomposed (teared) into non-overlapping subdomains that are "glued" by Lagrange multipliers, so that, after eliminating the primal variables, the original problem is reduced to a small, relatively well-conditioned, possibly equality constrained quadratic programming problem that is solved iteratively. The time that is necessary for both the elimination and iterations can be reduced nearly proportionally to the number of the subdomains, so that the algorithm enjoys parallel scalability. Since then, many preconditioning methods were developed which guarantee also numerical scalability of the FETI methods (see, e.g., [18]). The equality constraints can be avoided by using the Dual-Primal FETI method (FETI–DP) introduced by Farhat et al., see [13]. The continuity of the primal solution at crosspoints is implemented directly into the formulation of the primal problem by considering one degree of freedom per variable at each crosspoint. Across the rest of the subdomain interfaces, the continuity of the primal solution is once again enforced by Lagrange multipliers. After eliminating the primal variables, the problem is again reduced to a small, unconstrained, relatively well conditioned, strictly convex quadratic programming problem that is solved iteratively. An attractive feature of FETI–DP is that the local problems are nonsingular. Moreover, the conditioning of the resulting quadratic programming problem may be further improved by preconditioning, see [17], and the method performs better than the original FETI method on the fourth order problems.

Though the FETI and FETI-DP domain decomposition methods were originally developed for solving efficiently large-scale linear systems of equations arising from the discretization of the problems defined on a single domain Ω, it was soon observed that they can be even more efficient for the solution of multidomain contact problems, see e.g. [12, 5], and [1]. The reason is that the duality in this case not only reduces the original discretized problem to a smaller and better conditioned problem, but it also transforms the more general inequalities describing non-penetration into the bound constraints that can be treated much more efficiently. Moreover, since the FETI method treats naturally such subdomains, this approach is well suited for the solution of semicoercive contact problems with "floating" subdomains. These observations were soon confirmed by numerical experiments ([12, 5], and [1]). Recently, using new results in development of quadratic programming (see [11, 4]), the experimental evidence was supported by theory (see [3, 6, 9, 10]). There are also references to some other development of scalable algorithms for contact problems. See also [16] or the paper by Krause in this proceedings.

In this paper, we review our work related to the development of scalable algorithms for the solution of multibody contact problems by FETI–DP based methods with a special stress on the solution of 3D problems. For the sake of simplicity, we consider only the frictionless problems of linear elasticity with the linearized, possibly non-matching non-interpenetration conditions implemented by mortars, but the results may be exploited also for the solution of the problems with friction or large deformations with more sophisticated implementation of the kinematic constraints, see e.g. [8].

2 FETI and Contact Problems

Assuming that the bodies are assembled from the subdomains $\Omega^{(s)}$, the equilibrium of the system may be described as a solution $\mathbf{u}$ of the problem

$$\min j(\mathbf{v}) \quad \text{subject to} \quad \sum_{s=1}^{N_s} \mathsf{B}_I^{(s)} \mathbf{v}^{(s)} \le \mathbf{g}_I \quad \text{and} \quad \sum_{s=1}^{N_s} \mathsf{B}_E^{(s)} \mathbf{v}^{(s)} = \mathbf{o}, \tag{1}$$

where $j(\mathbf{v})$ is the energy functional defined by

$$j(\mathbf{v}) = \sum_{s=1}^{N_s} \frac{1}{2} {\mathbf{v}^{(s)}}^T \mathsf{K}^{(s)} \mathbf{v}^{(s)} - {\mathbf{v}^{(s)}}^T \mathbf{f}^{(s)},$$

$\mathbf{v}^{(s)}$ and $\mathbf{f}^{(s)}$ denote the admissible subdomain displacements and the subdomain vector of prescribed forces, $\mathsf{K}^{(s)}$ is the subdomain stiffness matrix, $\mathsf{B}^{(s)}$ is a block of the matrix $\mathsf{B} = \left[\mathsf{B}_I^T, \mathsf{B}_E^T\right]^T$ that corresponds to $\Omega^{(s)}$, and $\mathbf{g}_I$ is a vector collecting the gaps between the bodies in the reference configuration. The matrix B_I and the vector $\mathbf{g}_I$ arise from the nodal or mortar description of non-penetration conditions, while B_E describes the "gluing" of the subdomains into the bodies.

To simplify the presentation of basic ideas, we can describe the equilibrium in terms of the global stiffness matrix K_g, the vector of global displacements $\mathbf{u}_g$, and the vector of global loads $\mathbf{f}_g$. In the original FETI methods, FETI I and FETI II, we have

$$\mathsf{K}_g = \mathrm{diag}(\mathsf{K}^{(1)}, \dots, \mathsf{K}^{(\mathrm{N_s})}), \quad \mathbf{u}_{\mathrm{g}} = \begin{bmatrix} \mathbf{u}^{(1)} \\ \vdots \\ \mathbf{u}^{(N_s)} \end{bmatrix}, \quad \text{and} \quad \mathbf{f}_{\mathrm{g}} = \begin{bmatrix} \mathbf{f}^{(1)} \\ \vdots \\ \mathbf{f}^{(N_s)} \end{bmatrix},$$

where $\mathsf{K}^{(s)}$ is a positive definite or positive semidefinite matrix.

A distinctive feature of the FETI-DP method is that the continuity of the components of the displacement field at some "corner" interface nodes is not enforced by the Lagrange multipliers, but is achieved by defining the corner unknowns only at the global level, while defining all other displacement unknowns at the subdomain level. If the subscripts c and r are chosen to designate all the degrees of freedom that correspond to the corners and remainders, respectively, then the subdomain and global stiffness matrices have the form

$$\mathsf{K}^{(s)} = \begin{bmatrix} \mathsf{K}_{rr}^{(s)} & \mathsf{K}_{rc}^{(s)} \\ \mathsf{K}_{cr}^{(s)} & \mathsf{K}_{cc}^{(s)} \end{bmatrix} \quad \text{and} \quad \mathsf{K}_{\mathrm{g}} = \begin{bmatrix} \mathsf{K}_{rr}^{g} & \mathsf{K}_{rc}^{g} \\ \mathsf{K}_{cr}^{g} & \mathsf{K}_{cc}^{g} \end{bmatrix}, \quad \mathsf{K}_{\mathrm{rr}}^{\mathrm{g}} = \mathrm{diag}(\mathsf{K}_{\mathrm{rr}}^{(1)}, \dots, \mathsf{K}_{\mathrm{rr}}^{(\mathrm{N_s})}),$$

where $\mathsf{K}_{rr}^{g} = \mathrm{diag}(\mathsf{K}_{\mathrm{rr}}^{(1)}, \dots, \mathsf{K}_{\mathrm{rr}}^{(\mathrm{N_s})})$ is nonsingular and K_{cc}^{g} is a positive definite or semidefinite small matrix.

Whichever variant of the domain decomposition we use, the energy function reads

$$j(\mathbf{v}_g) = \frac{1}{2}\mathbf{v}_g^T \mathsf{K}_g \mathbf{v}_g - \mathbf{f}_g^T \mathbf{v}_g$$

and the vector of global displacements $\mathbf{u}_g$ solves

$$\min j(\mathbf{v}_g) \quad \text{subject to} \quad \mathsf{B}_I \mathbf{v}_g \le \mathbf{g}_I \quad \text{and} \quad \mathsf{B}_E \mathbf{v}_g = \mathbf{o}. \tag{2}$$

Alternatively, the global equilibrium my be described by the Karush-Kuhn-Tucker conditions (e.g. [2])

$$\mathsf{K}_g \mathbf{u}_g = \mathbf{f}_g - \mathsf{B}^T \boldsymbol{\lambda}, \quad \boldsymbol{\lambda}_I \ge \mathbf{o}, \quad \boldsymbol{\lambda}_I^T (\mathsf{B}_I \mathbf{u} - \mathbf{g}_I) = \mathbf{o}, \tag{3}$$

where $\mathbf{g} = \left[\mathbf{g}_E^T, \mathbf{o}^T\right]^T$, and $\boldsymbol{\lambda} = \left[\boldsymbol{\lambda}_I^T, \boldsymbol{\lambda}_E^T\right]^T$ denotes the vector of Lagrange multipliers which may be interpreted as the reaction forces. The problem (3) differs from the linear problem by the non-negativity constraint on the components of reaction forces $\boldsymbol{\lambda}_I$ and by the complementarity condition.

We can use the left equation of (3) and the sparsity pattern of K_g to eliminate the displacements. We shall get the problem to find

$$\max \Theta(\boldsymbol{\lambda}) \quad \text{s.t.} \quad \boldsymbol{\lambda}_I \ge \mathbf{o} \quad \text{and} \quad \mathsf{R}^T(\mathbf{f}_g - \mathsf{B}^T \boldsymbol{\lambda}) = \mathbf{o}, \tag{4}$$

where

$$\Theta(\boldsymbol{\lambda}) = -\frac{1}{2}\lambda^T \mathsf{B}\mathsf{K}_g^{\dagger}\mathsf{B}^T \boldsymbol{\lambda} + \boldsymbol{\lambda}^T(\mathsf{B}\mathsf{K}_g^{\dagger}\mathbf{f}_g - \mathbf{g}) - \frac{1}{2}\mathbf{f}_g \mathsf{K}_g^{\dagger}\mathbf{f}_g, \tag{5}$$

$\mathsf{K}_g^{\dagger}$ denotes a generalized inverse that satisfies $\mathsf{K}_g \mathsf{K}_g^{\dagger} \mathsf{K}_g = \mathsf{K}_g$, and R denotes the full rank matrix whose columns span the kernel of K_g. Recalling the FETI notation

$$\mathsf{F} = \mathsf{BK}_g^\dagger \mathsf{B}^T, \quad \mathbf{e} = \mathsf{R}^T \mathbf{f}_g, \quad \mathsf{G} = \mathsf{R}^T \mathsf{B}^T, \quad \mathsf{P} = \mathsf{G}^T(\mathsf{GG}^T)^\dagger \mathsf{G}, \quad \mathbf{d} = \mathsf{BK}_g^\dagger \mathbf{f}_g - \mathbf{g},$$

denoting $\mathsf{Q} = \mathsf{I} - \mathsf{P}$, and observing that $\mathsf{Q}\boldsymbol{\lambda} = \boldsymbol{\lambda}$ for any feasible $\boldsymbol{\lambda}$, we can modify (4) to

$$\min \theta(\boldsymbol{\lambda}) \quad \text{s.t.} \quad \boldsymbol{\lambda}_I \geq 0 \quad \text{and} \quad \mathsf{G}\boldsymbol{\lambda} = \mathbf{e}, \tag{6}$$

where

$$\theta(\boldsymbol{\lambda}) = \frac{1}{2}\boldsymbol{\lambda}^T \mathsf{QFQ}\boldsymbol{\lambda} - \boldsymbol{\lambda}^T \mathsf{Q}\ \mathbf{d}.$$

Alternatively, the Lagrange multipliers of the solution are determined by the KKT conditions for (4) which read

$$\mathsf{F}\boldsymbol{\lambda} - \mathbf{d} + \mathsf{G}^T\boldsymbol{\alpha} = \mathbf{o}, \quad \boldsymbol{\lambda}_I \geq \mathbf{o}, \quad \text{and} \quad \mathsf{G}\boldsymbol{\lambda} = \mathbf{e}. \tag{7}$$

For details concerning the matrices and parallel implementation, see e.g. in [8, 12], and [1].

3 Algorithms

We implemented two FETI based algorithms for the solution of contact problems, using the research software which is being developed in Stanford. The first one, FETI-DPC, is based on FETI-DP domain decomposition method. The algorithm uses the Newton-like method which solves the equilibrium equation (7) in Lagrange multipliers in the inner loop, while feasibility of each step is ensured in the outer loop by the primal and dual planning steps. The algorithm exploits standard FETI preconditioners, namely the Schur and lumped ones. The additional speedup of convergence is achieved by application Krylov type acceleration scheme. The algorithm exploits a globalization strategy in order to achieve monotonic global convergence.

The second algorithm is based on the TFETI domain decomposition (see [7]), a variant of the FETI-I domain decomposition method (see [14]), which treats all the boundary conditions by Lagrange multipliers, so that all the subdomains are floating, and their kernels are known a priori and can be used in construction of the natural coarse grid. It exploits our recently proposed algorithms MPRGP (Modified Proportioning with Reduced Gradient Projection) by Dostál and Schöberl (see [11]) and SMALBE (Semimonotonic Augmented Lagrangians for Bound and Equality constrained problems) (see [3, 4]). The SMALBE, a variant of augmented Lagrangian method with adaptive precision control for the solution of quadratic programming problems with bound and equality constraints, is applied to (6). It enforces the equality constraints by the Lagrange multipliers generated in the outer loop, while the auxiliary bound constrained problems are solved approximately in the inner loop by MPRGP, an active set based algorithm which uses the conjugate gradient method to explore the current face, the fixed steplength gradient projection to expand the active set, the adaptive precision control of auxiliary linear problems, and the reduced gradient with the optimal steplength to reduce the active set. The unique feature of SMALBE with the inner loop implemented by MPRGP when used to (6) is the rate of convergence in bounds on spectrum of the regular part of the Hessian of θ, so that using the classical results by Farhat, Mandel, and Roux (see [14]), the algorithm has been proved to be numerically scalable (see [6]).

4 Numerical Experiments

Algorithms described in this paper were tested and their results compared on two model contact problems.

The first 2D problem involves 6 rectangles in mutual contact as it is depicted in Figure 1 (left). The left rectangles are fixed on the left side (blue arrows) while the right ones are free and they are loaded (red arrows represent forces in opposite direction) such a way that the problem has unique solution. Each rectangle were further decomposed to the 4 subrectangles and therefore the original problem were decomposed to 24 subdomains (Figure 1 (middle)). The performance of the algorithms FETI-DPC and SMALBE is compared in Table 1. Outer iterations are used only in the case of SMALBE method while the number of subiterations is used only in methods FETI-DP. The number of dual plannings and primal plannings of FETI-DPC methods corresponds to the number of expansion and proportioning steps in the case of SMALBE method. Therefore they share the same column for each methods. The numbers on the left side of the slashes represent the number of iterations for 6 subdomains problem and the numbers on the right sides represent the number of iterations for 24 subdomains problem. The resulting deformation with distribution of the stresses are depicted in Figure 1 (right).

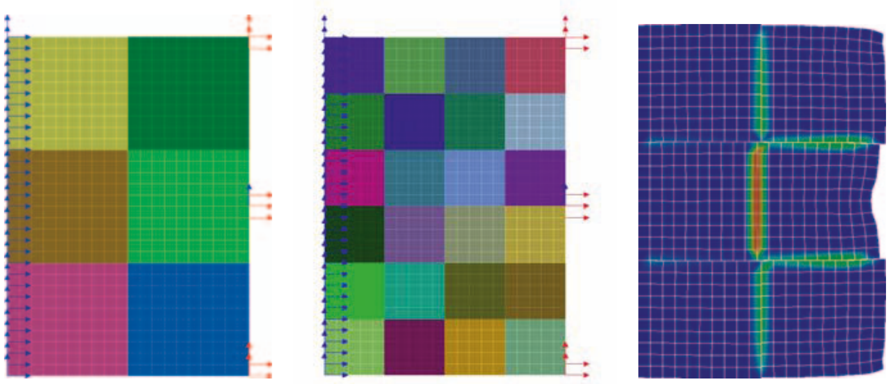


Fig. 1. 2D problem: decomposition in 6 subdomains (left), in 24 subdomains (middle), and computed stress distribution (right)

Table 1. Algorithms performance for 2D semicoercive problem with 6 and 24 subdomains.

	Outer iter.	Main iter.	subiter.	Primal plan.	Dual plan.
				(Exp. step)	(Proport.)
FETI-DPC	-	17/32	0/0	2/2	0/0
SMALBE	1/21	9/68	-	0/18	1/3

The second 3D model problem consists of two bricks in mutual contact. The bottom brick is fixed in all degrees of freedom while the upper one is fixed only in such a way, that only vertical rigid body movement is allowed. This situation is depicted in Figure 2 (left and middle). The forces are chosen so that not all constraints are active on the contact interface as in Figure 2 (right). We have analyzed two cases. The first one, with matching grid on the contact interface prescribes node-to-node contact conditions. The second one allows non-matching grids and the mortar elements were used for assembling of contact conditions. The resulting performance of algorithms is collected in the Table 2. Columns in this table have the same meaning as in 2D case.

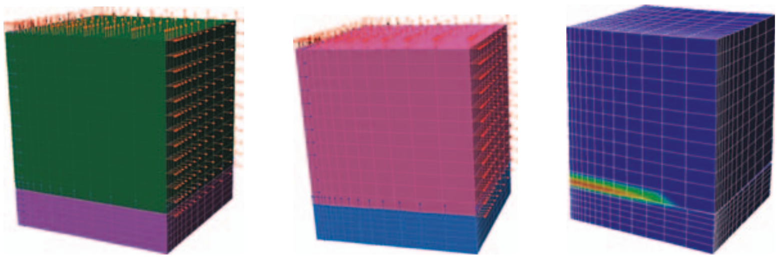


Fig. 2. 3D problem with matching grids (left), with non-matching grids (middle), and computed solution using non-matching grids (right)

Table 2. Algorithms performance for 3D problem with matching/non-matching grid on contact interface.

	Outer iter.	Main iter.	subiter.	Primal plan. (Exp. step)	Dual plan. (Proport.)
FETI-DPC	-	24/26	11/10	7/8	0/0
SMALBE	13/10	29/29	-	20/20	0/0

5 Comments and Conclusions

The FETI method turned out to be a powerful engine for the solution of contact problems of elasticity. Results of numerical experiments comply with recent theoretical results and indicate high efficiency of the methods presented here. Future research will include adaptation of the standard preconditioning strategies to the solution of inequality constraint problems, problems with friction (see e.g. [8]), and dynamic contact problems.

Acknowledgement. The research of the first three authors was supported by the grants 101/05/0423 and 101/04/1145 of the GA CR and by the projects AVO Z2076051, 1ET400300415, and ME641 of the Ministry of Education of the Czech Republic. The research of the last two authors was supported by the Sandia National Laboratories under Contract No. 31095.

References

[1] P. Avery, G. Rebel, M. Lesoinne, and C. Farhat. A numerically scalable dual-primal substructuring method for the solution of contact problems. I: the frictionless case. *Comput. Methods Appl. Mech. Engrg.*, 193(23-26):2403–2426, 2004.

[2] D. P. Bertsekas. *Nonlinear Optimization.* Athena Scientific - Nashua, 1999.

[3] Z. Dostál. Inexact semimonotonic augmented Lagrangians with optimal feasibility convergence for convex bound and equality constrained quadratic programming. *SIAM J. Numer. Anal.*, 43(1):96–115, 2005.

[4] Z. Dostál. An optimal algorithm for bound and equality constrained quadratic programming problems with bounded spectrum. *Computing*, 78:311–328, 2006.

[5] Z. Dostál and D. Horák. Scalability and FETI based algorithm for large discretized variational inequalities. *Math. Comput. Simulation*, 61:347–357, 2003.

[6] Z. Dostál and D. Horák. Theoretically supported scalable FETI for numerical solution of variational inequalities. *SIAM J. Numer. Anal.*, 2006. In press.

[7] Z. Dostál, D. Horák, and R. Kučera. Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE. *Comm. Numer. Methods Engrg.*, 2006. In press.

[8] Z. Dostál, D. Horák, R. Kučera, V. Vondrák, J. Haslinger, J. Dobiáš, and S. Pták. FETI based algorithms for contact problems: scalability, large displacements and 3d coulomb friction. *Comput. Methods Appl. Mech. Engrg.*, 194(2-5):395–409, 2005.

[9] Z. Dostál, D. Horák, and D. Stefanica. A scalable FETI-DP algorithm for a coercive variational inequality. *Appl. Numer. Math.*, 54(3-4):378–390, 2005.

[10] Z. Dostál, D. Horák, and D. Stefanica. A scalable FETI–DP algorithm for semi-coercive variational inequalities. *Comput. Methods Appl. Mech. Engrg.*, 2006. In press.

[11] Z. Dostál and J. Schöberl. Minimizing quadratic functions subject to bound constraints with the rate of convergence and finite termination. *Comput. Optim. Appl.*, 30:23–43, 2005.

[12] D. Dureisseix and C. Farhat. A numerically scalable domain decomposition method for solution of frictionless contact problems. *Internat. J. Numer. Methods Engrg.*, 50(12):2643–2666, 2001.

[13] C. Farhat, M. Lesoinne, P. LeTallec, K. Pierson, and D. Rixen. FETI-DP: A dual–prime unified FETI method. I: A faster alternative to the two–level FETI method. *Internat. J. Numer. Methods Engrg.*, 50(7):1523–1544, 2001.

[14] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115:365–387, 1994.

[15] C. Farhat and F.-X. Roux. An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems. *SIAM J. Sci. Statist. Comput.*, 13:379–396, 1992.

[16] R. Kornhuber. *Adaptive monotone multigrid methods for nonlinear variational problems.* Teubner - Verlag Stuttgart, 1997.

[17] J. Mandel and R. Tezaur. On the convergence of a Dual-Primal substructuring method. *Numer. Math.*, 88:543–558, 2001.

[18] A. Toselli and O. B. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Springer - Verlag Berlin Heidelberg, 2005.

Balancing Domain Decomposition Methods for Discontinuous Galerkin Discretization

Maksymilian Dryja[1]*, Juan Galvis[2], and Marcus Sarkis[2,3]

[1] Department of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland.

[2] Instituto Nacional de Matemática Pura e Aplicada, Estrada Dona Castorina 110, Rio de Janeiro 22460-320, Brazil.

[3] Department of Mathematical Sciences, Worcester Polytechnic Institute, Worcester, MA 01609, USA.

Summary. A discontinuous Galerkin (DG) discretization of a Dirichlet problem for second order elliptic equations with discontinuous coefficients in two dimensions is considered. The problem is considered in a polygonal region Ω which is a union of disjoint polygonal substructures Ω_i of size $O(H_i)$. Inside each substructure Ω_i, a triangulation $\mathcal{T}_{h_i}(\Omega_i)$ with a parameter h_i and a conforming finite element method are introduced. To handle nonmatching meshes across $\partial\Omega_i$, a DG method that uses symmetrized interior penalty terms on the boundaries $\partial\Omega_i$ is considered. In this paper we design and analyze Balancing Domain Decomposition (BDD) algorithms for solving the resulting discrete systems. Under certain assumptions on the coefficients and the mesh sizes across $\partial\Omega_i$, a condition number estimate $C(1 + \max_i \log^2 \frac{H_i}{h_i})$ is established with C independent of h_i, H_i and the jumps of the coefficients. The algorithm is well suited for parallel computations and can be straightforwardly extended to three-dimensional problems. Results of numerical tests are included which confirm the theoretical results and the imposed assumption.

1 Introduction

DG methods are becoming more and more popular for approximation of PDEs since they are well suited for dealing with complex geometries, discontinuous coefficients and local or patch refinements; see [2, 4] and the references therein. There are also several papers devoted to algorithms for solving DG discrete problems. In particular in connection with domain decomposition methods, we can mention [9, 10, 1] where overlapping Schwarz methods were proposed and analyzed for DG discretization of elliptic problems with continuous coefficients. In [4] a non optimal multilevel additive Schwarz method is designed and analyzed for the discontinuous coefficient case. In [3] a two-level ASM is proposed and analyzed for DG discretization of fourth order

* This work was supported in part by Polish Sciences Foundation under grant 2P03A00524.

problems. In those works, the coarse problems are based on polynomial coarse basis functions on a coarse triangulation. In addition, ideas of iterative substructuring methods and notions of discrete harmonic extensions are not explored, therefore, for the cases where the distribution of the coefficients ρ_i is not quasimonotonic, see [7], these methods when extended straightforwardly to 3-D problems have condition number estimates which might deteriorate as the jumps of the coefficients get more severe. To the best of our knowledge [5] is the only work in the literature that deals with iterative substructuring methods for DG discretizations with discontinuous coefficients, where we have successfully introduced and analyzed BDDC methods with different possible constraints on the edges. A goal of this paper is to design and analyze BDD algorithms, see [11, 8] and also [12], for DG discrete systems with discontinuous coefficients.

The paper is organized as follows. In Section 2, the differential problem and its DG discretization are formulated. In Section 3, the problem is reduced to a Schur complement problem with respect to the unknowns on $\partial\Omega_i$, and discrete harmonic functions defined in a special way are introduced. In Section 4, the BDD algorithm is designed and analyzed. The local problems are defined on $\partial\Omega_i$ and on faces of $\partial\Omega_j$ common to Ω_i, while the coarse space, restriction and prolongation operators are defined via a special partitioning of unity on the $\partial\Omega_i$. Sections 5 and 6 are devoted to numerical experiments and final remarks, respectively.

2 Differential and Discrete Problems

Consider the following problem: Find $u^* \in H_0^1(\Omega)$ such that

$$a(u^*, v) = f(v) \quad \text{for all } v \in H_0^1(\Omega) \tag{1}$$

where $a(u,v) = \sum_{i=1}^{N} \int_{\Omega_i} \rho_i \nabla u \nabla v dx$ and $f(v) = \int_\Omega f v dx$.

We assume that $\bar{\Omega} = \cup_{i=1}^N \bar{\Omega}_i$ and the substructures Ω_i are disjoint shape regular polygonal subregions of diameter $O(H_i)$ that form a geometrically conforming partition of Ω, i.e., for all $i \neq j$ the intersection $\partial\Omega_i \cap \partial\Omega_j$ is empty, or a common vertex or face of $\partial\Omega_i$ and $\partial\Omega_j$. We assume $f \in L^2(\Omega)$ and for simplicity of presentation let ρ_i be a positive constant, $i = 1, \ldots, N$.

Let us introduce a shape regular triangulation in each Ω_i with triangular elements and the mesh parameter h_i . The resulting triangulation on Ω is in general nonmatching across $\partial\Omega_i$. Let $X_i(\Omega_i)$ be a finite element (FE) space of piecewise linear continuous functions in Ω_i. Note that we do not assume that the functions in $X_i(\Omega_i)$ vanish on $\partial\Omega_i \cap \partial\Omega$. Define

$$X_h(\Omega) = X_1(\Omega_1) \times \cdots \times X_N(\Omega_N).$$

The discrete problem obtained by the DG method, see [2, 4], is of the form: Find $u_h^* \in X_h(\Omega)$ such that

$$a_h(u_h^*, v) = f(v) \quad \text{for all } v \in X_h(\Omega) \tag{2}$$

where

$$a_h(u,v) \equiv \sum_{i=1}^{N} b_i(u,v) \quad \text{and} \quad f(v) \equiv \sum_{i=1}^{N} \int_{\Omega_i} f v_i dx, \tag{3}$$

$$b_i(u,v) \equiv a_i(u,v) + s_i(u,v) + p_i(u,v), \tag{4}$$

$$a_i(u,v) \equiv \int_{\Omega_i} \rho_i \nabla u_i \nabla v_i dx, \tag{5}$$

$$s_i(u,v) \equiv \sum_{F_{ij} \subset \partial\Omega_i} \int_{F_{ij}} \frac{\rho_{ij}}{l_{ij}} \left(\frac{\partial u_i}{\partial n}(v_j - v_i) + \frac{\partial v_i}{\partial n}(u_j - u_i) \right) ds, \tag{6}$$

$$p_i(u,v) \equiv \sum_{F_{ij} \subset \partial\Omega_i} \int_{F_{ij}} \frac{\rho_{ij}}{l_{ij}} \frac{\delta}{h_{ij}} (u_j - u_i)(v_j - v_i) ds, \tag{7}$$

$$d_i(u,v) \equiv a_i(u,v) + p_i(u,v), \tag{8}$$

with $u = \{u_i\}_{i=1}^N \in X_h(\Omega)$ and $v = \{v_i\}_{i=1}^N \in X_h(\Omega)$. We set $l_{ij} = 2$ when $F_{ij} \equiv \partial\Omega_i \cap \partial\Omega_j$ is a common face of $\partial\Omega_i$ and $\partial\Omega_j$, and define $\rho_{ij} = 2\rho_i\rho_j/(\rho_i + \rho_j)$ as the harmonic average of ρ_i and ρ_j, and $h_{ij} = 2h_ih_j/(h_i + h_j)$. In order to simplify the notation we include the index $j = 0$ and set $l_{i0} = 1$ when $F_{i0} \equiv \partial\Omega_i \cap \partial\Omega$ has a positive measure, and set $u_0 = 0$ and $v_0 = 0$, and define $\rho_{i0} = \rho_i$ and $h_{i0} = h_i$. The outward normal derivative on $\partial\Omega_i$ is denoted by $\frac{\partial}{\partial n}$ and δ is the positive penalty parameter.

It is known that there exists a $\delta_0 = O(1) > 0$ such that for $\delta > \delta_0$, we obtain $2|s_i(u,u)| < d_i(u,u)$ and therefore, the problem (2) is elliptic and has a unique solution. An error bound of this method is given in [2] for continuous and in [4, 5] for discontinuous coefficients.

3 Schur Complement Problem

In this section we derive a Schur complement problem for the problem (2).

Define $\overset{o}{X}_i(\Omega_i)$ as the subspace of $X_i(\Omega_i)$ of functions that vanish on $\partial\Omega_i$. Let $u = \{u_i\}_{i=1}^N \in X_h(\Omega)$. For each $i = 1, \ldots, N$, the function $u_i \in X_i(\Omega)$ can be represented as

$$u_i = \hat{\mathcal{P}}_i u + \hat{\mathcal{H}}_i u, \tag{9}$$

where $\hat{\mathcal{P}}_i u$ is the projection of u into $\overset{o}{X}_i(\Omega_i)$ in the sense of $b_i(.,.)$. Note that since $\hat{\mathcal{P}}_i u$ and v_i belong to $\overset{o}{X}_i(\Omega_i)$, we have

$$a_i(\hat{\mathcal{P}}_i u, v_i) = b_i(\hat{\mathcal{P}}_i u, v_i) = a_h(u, v_i). \tag{10}$$

The $\hat{\mathcal{H}}_i u$ is the discrete harmonic part of u in the sense of $b_i(.,.)$, where $\hat{\mathcal{H}}_i u \in X_i(\Omega_i)$ is the solution of

$$b_i(\hat{\mathcal{H}}_i u, v_i) = 0 \qquad v_i \in \overset{o}{X}_i(\Omega_i), \tag{11}$$

with boundary data given by

$$u_i \text{ on } \partial\Omega_i \quad \text{and} \quad u_j \text{ on } F_{ji} = \partial\Omega_i \cap \partial\Omega_j. \tag{12}$$

We point out that for $v_i \in \overset{o}{X}_i(\Omega_i)$ we have

$$b_i(\hat{\mathcal{H}}_i u, v_i) = (\rho_i \nabla \hat{\mathcal{H}}_i u, \nabla v_i)_{L^2(\Omega_i)} + \sum_{F_{ij} \subset \partial\Omega_i} \frac{\rho_{ij}}{l_{ij}} (\frac{\partial v_i}{\partial n}, u_j - u_i)_{L^2(F_{ij})}. \tag{13}$$

Note that $\hat{\mathcal{H}}_i u$ is the classical discrete harmonic except at nodal points close to $\partial\Omega_i$. We will sometimes call $\hat{\mathcal{H}}_i u$ by discrete harmonic in a special sense, i.e., in the sense of $b_i(.,.)$ or $\hat{\mathcal{H}}_i$. Hence, $\hat{\mathcal{H}}u = \{\hat{\mathcal{H}}_i u\}_{i=1}^N$ and $\hat{\mathcal{P}}u = \{\hat{\mathcal{P}}_i u\}_{i=1}^N$ are orthogonal in the sense of $a_h(.,.)$. The discrete solution of (2) can be decomposed as $u_h^* = \hat{\mathcal{P}}u_h^* + \hat{\mathcal{H}}u_h^*$ where for all $v \in X_h(\Omega)$, $a_h(\hat{\mathcal{P}}u_h^*, \hat{\mathcal{P}}v) = f(\hat{\mathcal{P}}v)$ and

$$a_h(\hat{\mathcal{H}}u_h^*, \hat{\mathcal{H}}v) = f(\hat{\mathcal{H}}v). \tag{14}$$

Define $\Gamma \equiv (\cup_i \partial\Omega_{ih_i})$ where $\partial\Omega_{ih_i}$ is the set of nodal points of $\partial\Omega_i$. We note that the nodes on both side of $\cup_i\partial\Omega_i$ belong to Γ. We denote the space $V = V_h(\Gamma)$ as the set of all functions v_h in $X_h(\Omega)$ such that $\hat{\mathcal{P}}v_h = 0$, i.e., the space of discrete harmonic functions in the sense of $\hat{\mathcal{H}}_i$. The equation (14) is the Schur complement problem associated to (2).

4 Balancing Domain Decomposition

We design and analyze a BDD method [11, 12] for solving (14) and use the general framework of balancing domain decomposition methods; see [12]. For $i = 1, \ldots, N$, let V_i be auxiliary spaces and I_i prolongation operators from V_i to V, and define the operators $\tilde{T}_i : V \to V_i$ as

$$b_i(\tilde{T}_i u, v) = a_h(u, I_i v) \quad \text{for all } v \in V_i.$$

and set $T_i = I_i\tilde{T}_i$. The coarse problem is defined as

$$a_h(P_0 u, v) = a_h(u, v) \quad \text{for all } v \in V_0.$$

Then the BDD method is defined as

$$T = P_0 + (I - P_0) \left(\sum_{i=1}^N T_i \right) (I - P_0). \tag{15}$$

We next define the prolongation operators I_i and the local spaces V_i for $i = 1, ..., N$, and the coarse space V_0. The bilinear forms b_i and a_h are given by (4) and (3), respectively.

4.1 Local Problems

Let us denote by Γ_i the set of all nodes on $\partial\Omega_i$ and on neighboring faces $\bar{F}_{ji} \subset \partial\Omega_j$. We note that the nodes of ∂F_{ji} (which are vertices of Ω_j) are included in Γ_i. Define V_i as the vector space associated to the nodal values on Γ_i and extended via $\hat{\mathcal{H}}_i$ inside Ω_i. We say that $u \in V_i$ if it can be represented as $u := \{u_l^{(i)}\}_{l \in \#(i)}$, where $\#(i) = \{i \text{ and } \cup j : F_{ij} \subset \partial\Omega_i\}$. Here $u_i^{(i)}$ and $u_j^{(i)}$ stand for the nodal value of u on $\partial\Omega_i$ and $\bar{F}_{ji}$. We write $u = \{u_l^{(i)}\} \in V_i$ to refer to a function defined on Γ_i, and $u = \{u_i\} \in V$ to refer to a function defined on all Γ. Let us define the regular zero extension operator $\tilde{I}_i : V_i \to V$ as follows: Given $u \in V_i$, let $\tilde{I}_i u$ be equal to u on the nodes of Γ_i and zero on the nodes of $\Gamma \backslash \Gamma_i$. Then we associate with each $\Omega_k,\ k = 1, \cdots, N$, the discrete harmonic function u_k inside each Ω_k in the sense of $\hat{\mathcal{H}}_k$.

A face across Ω_i and Ω_j has two sides, the side inside $\bar{\Omega}_i$, denoted by F_{ij}, and the side inside $\bar{\Omega}_j$, denoted by F_{ji}. In addition, we assign to each face one master side $m(i,j) \in \{i,j\}$ and one slave side $s(i,j) \in \{i,j\}$. Then, using the *interface condition*, see below, we show that Theorem 1 holds, see below, with a constant C independent of the ρ_i, h_i and H_i.

The Interface Condition. We say that the coefficients $\{\rho_i\}$ and the local mesh sizes $\{h_i\}$ satisfy the *interface condition* if there exist constants C_0 and C_1, of order $O(1)$, such that for any face $F_{ij} = F_{ji}$ the following condition holds

$$h_{s(i,j)} \le C_0 h_{m(i,j)} \quad \text{and} \quad \rho_{s(i,j)} \le C_1 \rho_{m(i,j)}. \tag{16}$$

We associate with each $\Omega_i,\ i = 1, \cdots, N$, the weighting diagonal matrices $D^{(i)} = \{D_l^{(i)}\}_{l \in \#(i)}$ on Γ_i defined as follows:

- On $\partial\Omega_i \ \ (l = i)$

$$D_i^{(i)}(x) = \begin{cases} 1 \text{ if } & x \text{ is a vertex of } \partial\Omega_i, \\ 1 \text{ if } & x \text{ is an interior node of a master face } F_{ij} \\ 0 \text{ if } & x \text{ is an interior node of a slave face } F_{ij} \end{cases} \tag{17}$$

- On $\partial\Omega_j \ \ (l = j)$

$$D_j^{(i)}(x) = \begin{cases} 0 \text{ if } & x \text{ is an end point of } F_{ji}, \\ 1 \text{ if } & x \text{ is an interior node of a slave face } F_{ji} \\ 0 \text{ if } & x \text{ is an interior node of a master face } F_{ji} \end{cases} \tag{18}$$

- For $x \in F_{i0}$ we set $D_i^{(i)}(x) = 1$.

The prolongation operators $I_i : V_i \to V$, $i = 1, \ldots, N$, are defined as $I_i = \tilde{I}_i D^{(i)}$ and they form a partition of unity on Γ described as

$$\sum_{i=1}^{N} I_i \tilde{I}_i^T = I_\Gamma. \tag{19}$$

4.2 Coarse Problem

We define the coarse space $V_0 \subset V$ as

$$V_0 \equiv \text{Span}\{I_i \Phi^{(i)}, i = 1, ..., N\} \tag{20}$$

where $\Phi^{(i)} \in V_i$ denotes the function equal to one at every node of Γ_i.

Theorem 1. *If the interface condition (16) holds then there exists a positive constant C independent of h_i, H_i and the jumps of ρ_i such that*

$$a_h(u,u) \le a_h(Tu,u) \le C(1 + \log^2 \frac{H}{h}) a_h(u,u) \qquad \forall u \in V, \tag{21}$$

where T is defined in (15). Here $\log \frac{H}{h} = \max_i \ \log \frac{H_i}{h_i}$. *(See [6].)*

5 Numerical Experiments

In this section, we present numerical results for the preconditioner introduced in (15) and show that the bounds of Theorem 1 are reflected in the numerical tests. In particular we show that the interface condition (16) is necessary and sufficient.

We consider the domain $\Omega = (0,1)^2$ divided into $N = M \times M$ squares subdomains Ω_i and let $H = 1/M$. Inside each subdomain Ω_i we generate a structured triangulation with n_i subintervals in each coordinate direction and apply the discretization presented in Section 2 with $\delta = 4$. In the numerical experiments we use a red and black checkerboard type of subdomain partition. On the black subdomains we let $n_i = 2 * 2^{L_b}$ and on the red subdomains $n_i = 3 * 2^{L_r}$, where L_b and L_r are integers denoting the number of refinements inside each subdomain Ω_i. Hence, the mesh sizes are $h_b = \frac{2^{-L_b}}{2N}$ and $h_r = \frac{2^{-L_r}}{3N}$, respectively. We consider $-\mathrm{div}(\rho(x)\nabla u^*(x)) = 1$ in Ω with homogeneous Dirichlet boundary conditions. In the numerical experiments we run PCG until the l_2 initial residual is reduced by a factor of 10^6.

In the first test we consider the constant coefficient case $\rho = 1$. We consider different values of $M \times M$ coarse partitions and different values of local refinements $L_b = L_r$, therefore, keeping constant the mesh ratio $h_b/h_r = 3/2$. We place the master on the black subdomains. Table 1 lists the number of PCG iterations and in parenthesis the condition number estimate of the preconditioned system. We note that the interface condition (16) is satisfied. As expected from Theorem 1, the condition numbers appear to be independent of the number of subdomains and grow by a logarithmical factor when the size of the local problems increases. Note that in the case of continuous coefficients the Theorem 1 is valid without any assumption on h_b and h_r if the master sides are chosen on the larger meshes.

Table 1. PCG/BDD iterations count and condition numbers for different sizes of coarse and local problems and constant coefficients ρ_i.

M↓ L_r →	0	1	2	3	4	5
2	13 (6.86)	17 (8.97)	18 (12.12)	19 (16.82)	21 (22.23)	22 (28.25)
4	18 (8.39)	22 (11.30)	26 (14.74)	30 (19.98)	33 (26.64)	36 (34.19)
8	20 (8.89)	24 (11.57)	28 (14.82)	32 (20.03)	37 (26.64)	42 (34.04)
16	19 (9.02)	24 (11.63)	27 (14.83)	32 (20.05)	37 (26.67)	42 (34.06)

We now consider the discontinuous coefficient case where we set $\rho_i = 1$ on the black subdomains and $\rho_i = \mu$ on the red subdomains. The subdomains are kept fixed to 4×4. Table 2 lists the results on runs for different values of μ and for different levels of refinements on the red subdomains. On the black subdomains $n_i = 2$ is kept fixed. The masters are placed on the black subdomains. It is easy to see that the interface condition (16) holds if and only if μ is not large, which it is in agreement with the results in Table 2.

Table 2. PCG/BDD iterations count and condition numbers for different values of the coefficients and the local mesh sizes on the red subdomains only. The coefficients and the local mesh sizes on the black subdomains are kept fixed. The subdomains are also kept fixed to 4 × 4.

$\mu \downarrow L_r \rightarrow$	0	1	2	3	4
1000	90 (2556)	133 (3744)	184 (5362)	237 (7178)	303 (9102)
10	33 (29.16)	40 (42.31)	47 (58.20)	52 (75.55)	57 (94.59)
0.1	17 (8.28)	19 (8.70)	19 (9.21)	19 (9.50)	19 (9.65)
0.001	18 (8.83)	18 (8.95)	18 (9.46)	18 (9.83)	18 (10.08)

6 Final Remarks

We end this paper by mentioning extensions and alternative Neumann-Neumann methods for DG discretizations where the Theorem 1 holds: 1) The BDD algorithms can be straightforwardly extended to three-dimensional problems; 2) Additive Schwarz versions and inexact local Neumann solvers can be considered; see [6]; 3) On faces F_{ij} where h_i and h_j are of the same order, the values of (17) and (18) at interior nodes x of the faces F_{ij} and F_{ji} can be replaced by $\frac{\sqrt{\rho_i}}{\sqrt{\rho_i}+\sqrt{\rho_j}}$. 4) Similarly, on faces F_{ij} where ρ_i and ρ_j are of the same order, we can replace (17) and (18) at interior nodes x of the faces F_{ij} and F_{ji} by $\frac{h_i}{h_i+h_j}$. Finally, we remark the conditioning of the preconditioned systems deteriorates as we increase the penalty parameter δ to large values.

References

[1] P. F. Antonietti and B. Ayuso. Schwarz domain decomposition preconditioners for discontinuous Galerkin approximations of elliptic problems: non-overlapping case. Technical Report 20-VP, IMATI-CNR, June 2005. To appear in M2AN.

[2] D. N. Arnold, F. Brezzi, B. Cockburn, and D. Martin. Unified analysis of discontinuous Galerkin method for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2002.

[3] S. Brenner and K. Wang. Two-level additive Schwarz preconditioners for C^0 interior penalty methods. *Numer. Math.*, 102(2):231–255, 2005.

[4] M. Dryja. On discontinuous Galerkin methods for elliptic problems with discontinuous coefficients. *Comput. Methods Appl. Math.*, 3(1):76–85, 2003.

[5] M. Dryja, J. Galvis, and M. Sarkis. BDDC methods for discontinuous Galerkin discretization of elliptic problems. *J. Complexity*, 2007. To appear.

[6] M. Dryja, J. Galvis, and M. Sarkis. Neumann-Neumann methods for discontinuous Galerkin discretization of elliptic problems, 2007. In preparation.

[7] M. Dryja, M. Sarkis, and O. B. Widlund. Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.*, 72(3):313–348, 1996.

[8] M. Dryja and O. B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite elements problems. *Comm. Pure Appl. Math.*, 48(2):121–155, 1995.

[9] X. Feng and O. A. Karakashian. Two-level additive Schwarz methods for a discontinuous Galerkin approximation of second order elliptic problems. *SIAM J. Numer. Anal.*, 39(4):1343–1365, 2001.

[10] C. Lasser and A. Toselli. An overlapping domain decomposition preconditioners for a class of discontinuous Galerkin approximations of advection-diffusion problems. *Math. Comp.*, 72(243):1215–1238, 2003.

[11] J. Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9(3):233–241, 1993.

[12] A. Toselli and O.B. Widlund. *Domain Decomposition Methods—Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics*. Springer-Verlag, 2005.

Exact and Inexact FETI-DP Methods for Spectral Elements in Two Dimensions

Axel Klawonn[1], Oliver Rheinbach[1], and Luca F. Pavarino[2]

[1] Department of Mathematics, Universität Duisburg-Essen, 45117 Essen, Germany. {axel.klawonn,oliver.rheinbach}@uni-duisburg-essen.de

[2] Department of Mathematics, Università di Milano, Via Saldini 50, 20133 Milano, Italy. pavarino@mat.unimi.it

1 Introduction

High-order finite element methods based on spectral elements or *hp*-version finite elements improve the accuracy of the discrete solution by increasing the polynomial degree p of the basis functions as well as decreasing the element size h. The discrete systems generated by these high-order methods are much more ill-conditioned than the ones generated by standard low-order finite elements. In this paper, we will focus on spectral elements based on Gauss-Lobatto-Legendre (GLL) quadrature and construct nonoverlapping domain decomposition methods belonging to the family of Dual-Primal Finite Element Tearing and Interconnecting (FETI-DP) methods; see [4, 9, 7]. We will also consider inexact versions of the FETI-DP methods, i.e., irFETI-DP and iFETI-DP, see [8]. We will show that these methods are scalable and have a condition number depending only weakly on the polynomial degree.

2 Spectral Element Discretization of Second Order Elliptic Problems

Let T_{ref} be the reference square $(-1,1)^d$, $d = 2$, and let $Q_p(T_{\text{ref}})$ be the set of polynomials on T_{ref} of degree $p \geq 1$ in each variable. We assume that the domain Ω can be decomposed into N_e nonoverlapping finite elements T_k of characteristic diameter h, $\overline{\Omega} = \bigcup_{k=1}^{N_e} \overline{T}_k$, each of which is an affine image of the reference square or cube, $T_k = \phi_k(T_{\text{ref}})$, where ϕ_k is an affine mapping (more general maps could be considered as well). Later, we will group these elements into N nonoverlapping subdomains Ω_i of characteristic diameter H, forming themselves a coarse finite element partition of Ω, $\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i$, $\overline{\Omega}_i = \bigcup_{k=1}^{N_i} \overline{T}_k$. Hence, the fine element partition $\{T_k\}_{k=1}^{N_e}$ can be considered a refinement of the coarse subdomain partition $\{\Omega_i\}_{i=1}^{N}$, with matching finite element nodes on the boundaries of neighboring subdomains.

We consider linear, selfadjoint, elliptic problems on Ω, with zero Dirichlet boundary conditions on a part $\partial\Omega_D$ of the boundary $\partial\Omega$:

Find $u \in V = \{v \in H^1(\Omega) : \ v = 0 \ \text{ on } \ \partial\Omega_D\}$ such that

$$a(u,v) = \int_{\Omega} \rho(x) \nabla u \cdot \nabla v \, dx \quad = \quad \int_{\Omega} f v \, dx \qquad \forall \ v \in \ V. \tag{1}$$

Here, $\rho(x) > 0$ can be discontinuous, with very different values for different subdomains, but we assume this coefficient to vary only moderately within each subdomain Ω_i. In fact, without decreasing the generality of our results, we will only consider the piecewise constant case of $\rho(x) = \rho_i$, for $x \in \Omega_i$.

Conforming spectral element discretizations consist of continuous, piecewise polynomials of degree p in each element:

$$V^p = \{v \in V : v|_{T_i} \circ \phi_i \in Q_p(T_{\mathrm{ref}}), \ i = 1, \ldots, N_e\}.$$

A convenient tensor product basis for V^p is constructed using Gauss-Lobatto-Legendre (GLL) quadrature points; see Figure 1. Let $\{\xi_i\}_{i=0}^{p}$ denote the set of GLL points on $[-1,1]$ and σ_i the associated quadrature weights. Let $l_i(\cdot)$ be the Lagrange interpolating polynomial which vanishes at all the GLL nodes except ξ_i, where it equals one. The basis functions on the reference square are defined by a tensor product as $l_i(x_1) l_j(x_2)$, $\ 0 \le i, j \le p$. This basis is nodal, since every element of $Q_p(T_{\mathrm{ref}})$ can be written as $u(x_1, x_2) = \sum_{i=0}^{p} \sum_{j=0}^{p} u(\xi_i, \xi_j) l_i(x_1) l_j(x_2)$. Each integral of the continuous model (1) is replaced by GLL quadrature over each element

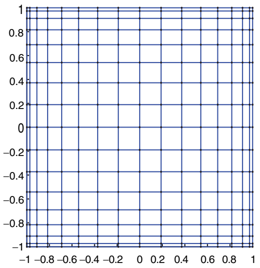


Fig. 1. Quadrilateral mesh defined by the Gauss-Lobatto-Legendre (GLL) quadrature points with $p = 16$ on one square element.

$$(u,v)_{p,\Omega} = \textstyle\sum_{k=1}^{N_e} \sum_{i,j=0}^{p} (u \circ \phi_k)(\xi_i, \xi_j)(v \circ \phi_k)(\xi_i, \xi_j) |J_k| \sigma_i \sigma_j, \tag{2}$$

where $|J_k|$ is the determinant of the Jacobian of ϕ_k. This inner product is uniformly equivalent to the standard L_2-inner product on $Q_p(T_{\mathrm{ref}})$. Applying these quadrature rules, we obtain the discrete elliptic problem:

$$\text{Find } u \in V^p \qquad \text{such that} \qquad a_p(u,v) = (f,v)_{p,\Omega} \quad \forall v \in V^p, \tag{3}$$

with discrete bilinear form $a_p(u,v) = \sum_{k=1}^{N_e} (\rho_k \nabla u, \nabla u)_{p,T_k}$ and each quadrature rule $(\cdot,\cdot)_{p,T_k}$ defined as in (2). Having chosen a basis for V^p, the discrete problem (3) is then turned into a linear system of algebraic equations $K_g u_g = f_g$, with K_g the globally assembled, symmetric, positive definite stiffness matrix; see [2] for more details.

3 The FETI-DP Algorithms

Let a domain $\Omega \subset \mathrm{I\!R}^2$ be decomposed into N nonoverlapping subdomains Ω_i of diameter H, each of which is the union of finite elements with matching finite element nodes on the boundaries of neighboring subdomains across the interface $\Gamma := \bigcup_{i \neq j} \partial\Omega_i \cap \partial\Omega_j$, where $\partial\Omega_i, \partial\Omega_j$ are the boundaries of Ω_i, Ω_j, respectively. The interface Γ is the union of edges and vertices. We regard edges in 2D as open sets shared by two subdomains, and vertices as endpoints of edges; see, e.g., [11, Chapter 4.2]. For a more detailed definition of faces, edges, and vertices in 2D and 3D; see [9, Section 3] and [7, Section 2].

For each subdomain Ω_i, $i = 1, \ldots, N$, we assemble the local stiffness matrices $K^{(i)}$ and load vectors $f^{(i)}$. We denote the unknowns on each subdomain by $u^{(i)}$. We then partition the unknowns $u^{(i)}$ into primal variables $u_\Pi^{(i)}$ and nonprimal variables $u_B^{(i)}$. As we only treat two dimensional problems here, the primal variables $u_\Pi^{(i)}$ will be associated with vertex unknowns whereas the nonprimal variables are interior $(u_I^{(i)})$ and dual $(u_\Delta^{(i)})$ unknowns. We will enforce the continuity of the solution in the primal unknowns $u_\Pi^{(i)}$ by global subassembly of the subdomain stiffness matrices $K^{(i)}$. For all other interface variables $u_\Delta^{(i)}$, we will introduce Lagrange multipliers to enforce continuity. We partition the stiffness matrices according to the different sets of unknowns,

$$K^{(i)} = \begin{bmatrix} K_{BB}^{(i)} & K_{\Pi B}^{(i)\,T} \\ K_{\Pi B}^{(i)} & K_{\Pi\Pi}^{(i)} \end{bmatrix}, \quad K_{BB}^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Delta I}^{(i)\,T} \\ K_{\Delta I}^{(i)} & K_{\Delta\Delta}^{(i)} \end{bmatrix},$$

$$\text{and} \qquad f^{(i)} = [f_B^{(i)} \; f_\Pi^{(i)}], \quad f_B^{(i)} = [f_I^{(i)} \; f_\Delta^{(i)}].$$

3.1 The Exact FETI-DP Algorithm

We define the block matrices

$$K_{BB} = \mathrm{diag}_{i=1}^N(K_{BB}^{(i)}), \quad K_{\Pi B} = \mathrm{diag}_{i=1}^N(K_{\Pi B}^{(i)}), \quad K_{\Pi\Pi} = \mathrm{diag}_{i=1}^N(K_{\Pi\Pi}^{(i)}),$$

and right hand sides $f_B^T = [f_B^{(1)\,T}, \ldots, f_B^{(N)\,T}]$, $f_\Pi^T = [f_\Pi^{(1)\,T}, \ldots, f_\Pi^{(N)\,T}]$.

By assembly of the local subdomain matrices in the primal variables using the operator $R_\Pi^T = [R_\Pi^{(1)\,T}, \ldots, R_\Pi^{(N)\,T}]$ with entries 0 or 1, we have the partially assembled global stiffness matrix $\widetilde{K}$ and right hand side $\tilde{f}$,

$$\widetilde{K} = \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix} = \begin{bmatrix} I_B & 0 \\ 0 & R_\Pi^T \end{bmatrix} \begin{bmatrix} K_{BB} & K_{\Pi B}^T \\ K_{\Pi B} & K_{\Pi\Pi} \end{bmatrix} \begin{bmatrix} I_B & 0 \\ 0 & R_\Pi \end{bmatrix},$$

$$\tilde{f} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \end{bmatrix} = \begin{bmatrix} I_B & 0 \\ 0 & R_\Pi^T \end{bmatrix} \begin{bmatrix} f_B \\ f_\Pi \end{bmatrix}.$$

Choosing a sufficient number of primal variables $u_\Pi^{(i)}$, i.e., all vertex unknowns, to constrain our solution, results in a symmetric, positive definite matrix $\widetilde{K}$.

To enforce continuity on the remaining interface variables $u_\Delta^{(i)}$ we introduce a jump operator B_B with entries $0, -1$ or 1 and Lagrange multipliers λ.

We can now formulate the FETI-DP saddle-point problem,

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & B_B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix} \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \lambda \end{bmatrix} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \end{bmatrix}. \tag{4}$$

By eliminating u_B and u_Π from the system (4), we obtain an equation system

$$F\lambda = d, \quad \text{where} \tag{5}$$

$F = B_B K_{BB}^{-1} B_B^T + B_B K_{BB}^{-1} \widetilde{K}_{B\Pi} \widetilde{S}_{\Pi\Pi}^{-1} \widetilde{K}_{\Pi B} K_{BB}^{-1} B_B^T$ and
$d = B_B K_{BB}^{-1} f_B - B_B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T \widetilde{S}_{\Pi\Pi}^{-1} (\tilde{f}_\Pi - \widetilde{K}_{\Pi B} K_{BB}^{-1} f_B)$. Let us define

$$K_{II} = \operatorname{diag}_{i=1}^N (K_{II}^{(i)}), \quad K_{\Delta I} = \operatorname{diag}_{i=1}^N (K_{\Delta I}^{(i)}), \quad K_{\Delta\Delta} = \operatorname{diag}_{i=1}^N (K_{\Delta\Delta}^{(i)}).$$

The theoretically almost optimal Dirichlet preconditioner M_{D} is then defined

$$\text{by} \quad M_{\mathrm{D}}^{-1} = B_{B,D} (R_\Delta^B)^T (K_{\Delta\Delta} - K_{\Delta I} K_{II}^{-1} K_{\Delta I}^T) R_\Delta^B B_{B,D}^T, \quad \text{where}$$

$R_\Delta^B = \operatorname{diag}_{i=1}^N (R_\Delta^{B\,(i)})$. The matrices $R_\Delta^{B\,(i)}$ are restriction operators with entries 0 or 1 which restrict the nonprimal degrees of freedom $u_B^{(i)}$ of a subdomain to the dual part $u_\Delta^{(i)}$. The matrices B_D are scaled variants of the jump operator B where the contribution from and to each interface node is scaled by the inverse of the multiplicity of the node. The multiplicity of a node is defined as the number of subdomains it belongs to. It is well known that for heterogeneous problems a more elaborate scaling is necessary, see, e.g., [9].

The original or standard, exact FETI-DP method is the method of conjugate gradients applied to the symmetric, positive definite system (5) using the preconditioner M_{D}^{-1}.

3.2 Inexact FETI-DP Algorithms

We will denote (4) as $\mathcal{A}x = \mathcal{F}$,

$$\text{where} \quad \mathcal{A} = \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & B_B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & 0 \\ B_B & 0 & 0 \end{bmatrix}, \; x = \begin{bmatrix} u_B \\ \tilde{u}_\Pi \\ \lambda \end{bmatrix}, \; \mathcal{F} = \begin{bmatrix} f_B \\ \tilde{f}_\Pi \\ 0 \end{bmatrix}.$$

We also write this equation

$$\begin{bmatrix} \widetilde{K} & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} u \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{f} \\ 0 \end{bmatrix}, \tag{6}$$

where $B = \begin{bmatrix} B_B & 0 \end{bmatrix}$, $u^T = [u_B^T \; \tilde{u}_\Pi^T]$, $\tilde{f}^T = [f_B^T \; \tilde{f}_\Pi^T]$. Eliminating u_B by one step of block elimination, we obtain the reduced system

$$\begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -\widetilde{K}_{\Pi B} K_{BB}^{-1} B_B^T \\ -B_B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T & -B_B K_{BB}^{-1} B_B^T \end{bmatrix} \begin{bmatrix} \tilde{u}_\Pi \\ \lambda \end{bmatrix} = \begin{bmatrix} \tilde{f}_\Pi - \widetilde{K}_{\Pi B} K_{BB}^{-1} f_B \\ -B_B K_{BB}^{-1} f_B \end{bmatrix}, \tag{7}$$

where $\widetilde{S}_{\Pi\Pi} = \widetilde{K}_{\Pi\Pi} - \widetilde{K}_{\Pi B} K_{BB}^{-1} \widetilde{K}_{\Pi B}^T$. For (7), we will also use the notation

$$\mathcal{A}_r x_r = \mathcal{F}_r, \quad \text{where} \quad x_r^T := [\tilde{u}_\Pi^T \;\; \lambda^T], \quad \text{and}$$

$$\mathcal{A}_r = \begin{bmatrix} \widetilde{S}_{\Pi\Pi} & -\widetilde{K}_{\Pi B} K_{BB}^{-1} B_B^T \\ -B_B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T & -B_B K_{BB}^{-1} B_B^T \end{bmatrix}, \quad \mathcal{F}_r := \begin{bmatrix} \tilde{f}_\Pi - \widetilde{K}_{\Pi B} K_{BB}^{-1} f_B \\ -B_B K_{BB}^{-1} f_B \end{bmatrix}.$$

The inexact FETI-DP methods are given by solving the saddle point problems (4) and (6) iteratively, using block triangular preconditioners and a suitable Krylov subspace method. For the saddle point problems (6) and (7), we introduce the block triangular preconditioners $\widehat{\mathcal{B}}_L$ and $\widehat{\mathcal{B}}_{r,L}$, respectively, as

$$\widehat{\mathcal{B}}_L^{-1} = \begin{bmatrix} \widehat{K}^{-1} & 0 \\ M^{-1} B \widehat{K}^{-1} & -M^{-1} \end{bmatrix}, \ \widehat{\mathcal{B}}_{r,L}^{-1} = \begin{bmatrix} \widehat{S}_{\Pi\Pi}^{-1} & 0 \\ -M^{-1} B_B K_{BB}^{-1} \widetilde{K}_{\Pi B}^T \widehat{S}_{\Pi\Pi}^{-1} & -M^{-1} \end{bmatrix},$$

where $\widehat{K}^{-1}$ and $\widehat{S}_{\Pi\Pi}^{-1}$ are assumed to be spectrally equivalent preconditioners for $\widetilde{K}$ and $\widetilde{S}_{\Pi\Pi}$, respectively, with bounds independent of the discretization parameters h, H. The matrix block M^{-1} is assumed to be a good preconditioner for the FETI-DP system matrix F and can be chosen as the Dirichlet preconditioner M_{D}^{-1} or any spectrally equivalent preconditioner. Our inexact FETI-DP methods are now given by using a Krylov space method for nonsymmetric systems, e.g., GMRES, to solve the preconditioned systems

$$\widehat{\mathcal{B}}_L^{-1} \mathcal{A} x = \widehat{\mathcal{B}}_L^{-1} \mathcal{F}, \quad \text{and} \quad \widehat{\mathcal{B}}_{r,L}^{-1} \mathcal{A}_r x_r = \widehat{\mathcal{B}}_{r,L}^{-1} \mathcal{F}_r,$$

respectively. The first will be denoted iFETI-DP and the latter irFETI-DP. Let us note that we can also use a positive definite reformulation of the two preconditioned systems, which allows the use of conjugate gradients, see [8] for further details.

4 Convergence Estimates

As shown in [11] for the two main families of overlapping Schwarz methods (Ch. 7.3) and iterative substructuring methods of wirebasket and Neumann-Neumann type (Ch. 7.4), the main domain decomposition results obtained for finite element discretizations of scalar elliptic problems can be transferred to the spectral element case; see [11, Ch. 7] for further details. The same tools can be used here to obtain the following estimate, see [10, 6] for further details.

Theorem 1. *The minimum eigenvalue of the FETI-DP operator is bounded from below by 1 and the maximum eigenvalue is bounded from above by* $C\left(1+\log\left(p\,\frac{H}{h}\right)\right)^2$, *with* $C > 0$ *independent of* p, h, H *and the values of the coefficients* ρ_i *of the elliptic operator.*

Similar convergence estimates hold for the inexact versions of FETI-DP, i.e., i(r)FETI-DP, if spectrally equivalent preconditioners are used instead of the direct solvers and GMRES instead of cg; see [8].

5 Numerical Results

We first investigate the growth of the condition number for an increasing number of subdomains. We expect to see the largest eigenvalue, and thus also the condition number, approaching a constant value, independent of coefficient jumps but dependent on the polynomial degree. We have used PETSc, the Portable Extensible Toolkit for Scientific Computing, see [1], for the parallel results in this section. In Table 1 we see the expected behavior for different polynomial degrees and fixed $H/h = 1$. From these results we choose to use a number of $N \geq 256$ subdomains in our experiments to study the asymptotic behavior of the condition number. In Table 2 we choose a sufficient number of subdomains and increase the polynomial degree from 2 to 32. We see that the condition number grows only slowly. In Table 2, we have also shown the CPU timings and iteration counts of irFETI-DP, additionally to the ones of FETI-DP. For irFETI-DP, we have used GMRES as Krylov subspace method and BoomerAMG [5] to precondition the FETI-DP coarse problem. BoomerAMG is a highly scalable distributed memory parallel algebraic multigrid solver and preconditioner; it is part of the high performance preconditioner library hypre [3]. From the table we see that also for spectral elements irFETI-DP compares very well with standard FETI-DP.

We report on the parallel scalability for 2 to 16 processors in Table 4 for FETI-DP and irFETI-DP. Both methods show basically the same performance and same scalability. Nevertheless, we expect irFETI-DP to be superior if coarse problems much larger than the ones here need to be solved. This will be the case for large numbers of subdomains, especially in 3D.

Table 1. One spectral element (p=2–32) per subdomain, N=4–576 subdomains, homogeneous problem and a problem with jumps, random right hand side, rtol=10^{-10}.

	FETI-DP											
	$\boldsymbol{\rho_{ij} = 1}$			$\boldsymbol{\rho_{ij} = 10^{(i-j)/4}}$			$\boldsymbol{\rho_{ij} = 1}$			$\boldsymbol{\rho_{ij} = 10^{(i-j)/4}}$		
N	It	λmax	λmin	It	λmax	λmin	It	λmax	λmin	It	λmax	λmin
	p=2			p=2			p=8			p=8		
4	**2**	1.05	1	**2**	1.05	1	**4**	1.89	1	**4**	1.89	1
16	**6**	1.45	1.0026	**6**	1.46	1.0018	**12**	4.38	1.0007	**12**	4.37	1.0004
64	**8**	1.61	1.0014	**8**	1.61	1.0013	**16**	4.86	1.0013	**18**	4.86	1.0009
256	**8**	1.64	1.0028	**8**	1.62	1.0013	**17**	5.00	1.0014	**19**	4.97	1.0008
576	**8**	1.66	1.0032	**8**	1.63	1.0016	**17**	5.01	1.0015	**20**	4.98	1.0009
	p=3			p=3			p=16			p=16		
4	**3**	1.21	1	**3**	1.21	1	**5**	2.57	1	**5**	2.57	1
16	**8**	2.10	1.0007	**8**	2.10	1.0004	**14**	6.65	1.0009	**15**	6.63	1.0008
64	**11**	2.32	1.0006	**11**	2.31	1.0006	**21**	7.42	1.0013	**21**	7.38	1.0008
256	**11**	2.37	1.0006	**12**	2.36	1.0004	**21**	7.58	1.0017	**25**	7.53	1.0009
576	**11**	2.38	1.0006	**13**	2.35	1.0006	**21**	7.62	1.0016	**26**	7.55	1.0006
	p=4			p=4			p=32			p=32		
4	**3**	1.37	1	**3**	1.37	1	**6**	3.42	1	**6**	3.42	1
16	**9**	2.65	1.0018	**10**	2.65	1.0008	**16**	9.48	1.0012	**17**	9.44	1.0009
64	**12**	2.95	1.0022	**13**	2.94	1.0011	**25**	10.58	1.0012	**25**	10.52	1.0009
256	**13**	3.01	1.0020	**14**	3.00	1.0013	**25**	10.81	1.0017	**31**	10.74	1.0008
576	**13**	3.03	1.0020	**15**	3.00	1.0005	**25**	10.86	1.0018	**33**	10.77	1.0005

Table 2. Homogeneous problem ($\rho = 1$). Increasing polynomial degree (p=2–32). Fixed subdomain sizes (H/h=1,2,4). FETI-DP and inexact reduced FETI-DP (irFETI-DP, GMRES). irFETI-DP uses one iteration of BoomerAMG with parallel Gauss-Seidel smoothing to precondition the coarse problem, rtol=10^{-7}.

			FETI-DP				irFETI-DP		
H/h	N	p	It	λmax	λmin	Time (16 Proc)	It	Time (16 Proc)	dof
1	4096	2	**7**	1.66	1.0074	2s	**7**	2s	16 129
		4	**10**	3.05	1.0217	4s	**9**	3s	65 025
		8	**13**	5.03	1.0067	6s	**11**	4s	261 121
		12	**15**	6.48	1.0260	11s	**13**	8s	588 289
		16	**16**	7.64	1.0121	23s	**14**	16s	1 046 529
		20	**17**	8.62	1.0114	53s	**14**	37s	1 635 841
		24	**18**	9.46	1.0138	94s	**16**	81s	2 356 225
		28	**18**	10.21	1.0183	155s	**16**	130s	3 207 681
		32	**19**	10.89	1.0227	256s	**17**	228s	4 190 209
2	1024	2	**9**	2.35	1.0020	1s	**8**	1s	16 129
		4	**12**	4.03	1.0146	2s	**11**	2s	65 025
		8	**15**	6.31	1.0232	4s	**12**	3s	261 121
		12	**17**	7.93	1.0177	10s	**15**	7s	588 289
		16	**18**	9.21	1.0133	23s	**17**	20s	1 046 529
		20	**19**	10.28	1.0186	43s	**17**	38s	1 635 841
		24	**20**	11.21	1.0247	83s	**18**	76s	2 356 225
		28	**21**	12.03	1.0294	164s	**18**	146s	3 207 681
		32	**22**	12.76	1.0230	276s	**18**	244s	4 190 209
4	256	2	**11**	3.18	1.0150	1s	**11**	1s	16 129
		4	**14**	5.14	1.0146	1s	**14**	1s	65 025
		8	**18**	7.70	1.0230	4s	**17**	4s	261 121
		12	**19**	9.49	1.0143	9s	**18**	9s	588 289
		16	**20**	10.89	1.0223	21s	**20**	20s	1 046 529
		20	**21**	12.05	1.0267	45s	**20**	42s	1 635 841
		24	**22**	13.05	1.0253	86s	**21**	84s	2 356 225
		28	**23**	13.94	1.0188	170s	**22**	164s	3 207 681
		32	**23**	14.73	1.0191	328s	**21**	280s	4 190 209

Table 3. Fixed polynomial degree (p=32), fixed subdomain sizes (H/h=1), increasing number of subdomains, $\rho = 1$, random right hand side, rtol=10^{-7}. Inexact FETI-DP for the block matrices using BoomerAMG and GMRES, local problem/coarse problem/Dirichlet preconditioner : (in)exact/(in)exact/(in)exact.

		iFETI-DP				FETI-DP		
p	N	It (i/i/i)	It (i/i/e)	It (i/e/e)	It (e/e/e)	It	$\lambda_{\min}$	$\lambda_{\max}$
32	4	13	13	13	6	6	3.42	1.0000
	16	22	21	20	16	17	9.48	1.0012
	64	30	30	29	24	25	10.57	1.0012
	100	30	30	30	24	24	10.69	1.0018
	144	30	29	30	24	25	10.75	1.0016

Table 4. Parallel scalability for p=20, N=256, H/h=4, rtol=10^{-7}.

	FETI-DP		irFETI-DP	
Proc	It	Time	It	Time
2	22	337s	20	309s
4	22	172s	20	156s
8	22	89s	20	82s
16	22	45s	20	42s

References

[1] S. Balay, K. Buschelman, V. Eijkhout, W.D. Gropp, D. Kaushik, M.G. Knepley, L.C. McInnes, B.F. Smith, and H. Zhang. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[2] C. Bernardi and Y. Maday. *Spectral Methods.* In Handbook of Numerical Analysis, Volume V: Techniques of Scientific Computing (Part 2), P. G. Ciarlet and J.-L. Lions, editors. North-Holland, 1997.

[3] R.D. Falgout, J.E. Jones, and U.M. Yang. The design and implementation of hypre, a library of parallel high performance preconditioners. In A.M. Bruaset, P. Bjorstad, and A. Tveito, editors, *Numerical solution of Partial Differential Equations on Parallel Computers, Lect. Notes Comput. Sci. Eng.*, volume 51, pages 267–294. Springer-Verlag, 2006.

[4] C. Farhat, M. Lesoinne, P. Le Tallec, K. Pierson, and D. Rixen. FETI-DP: a dual-primal unified FETI method – part I. A faster alternative to the two-level FETI method. *Internat. J. Numer. Methods Engrg.*, 50(7):1523–1544, 2001.

[5] V.E. Henson and U.M. Yang. Boomeramg: A parallel algebraic multigrid solver and preconditioner. *Appl. Numer. Math.*, 41:155–177, 2002.

[6] A. Klawonn, L.F. Pavarino, and O. Rheinbach. Spectral element FETI-DP and BDDC preconditioners with multi-element subdomains and inexact solvers in the plane. Technical report, February 2007.

[7] A. Klawonn and O. Rheinbach. A parallel implementation of Dual-Primal FETI methods for three dimensional linear elasticity using a transformation of basis. *SIAM J. Sci. Comput.*, 28:1886–1906, 2006.

[8] A. Klawonn and O. Rheinbach. Inexact FETI-DP methods. *Inter. J. Numer. Methods Engrg.*, 69:284–307, 2007.

[9] A. Klawonn and O.B. Widlund. Dual-Primal FETI Methods for Linear Elasticity. *Comm. Pure Appl. Math.*, 59:1523–1572, 2006.

[10] L.F. Pavarino. BDDC and FETI-DP preconditioners for spectral element discretizations. *Comput. Meth. Appl. Mech. Engrg.*, 196 (8):1380 – 1388, 2007.

[11] A. Toselli and O.B. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin Heidelberg New York, 2005.

On Multilevel BDDC

Jan Mandel[1]*, Bedřich Sousedík[1,2]†, and Clark R. Dohrmann[3]

[1] Department of Mathematical Sciences, University of Colorado at Denver and Health Sciences Center, Campus Box 170, Denver, CO 80217, USA. {jan.mandel,bedrich.sousedik}@cudenver.edu

[2] Department of Mathematics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7, 166 36 Prague 6, Czech Republic.

[3] Structural Dynamics Research Department, Sandia National Laboratories, Mail Stop 0847, Albuquerque NM 87185-0847, USA. crdohrm@sandia.gov

1 Introduction

The BDDC method [2] is the most advanced method from the BDD family [5]. Polylogarithmic condition number estimates for BDDC were obtained in [6, 7] and a proof that eigenvalues of BDDC and FETI-DP are same except for an eigenvalue equal to one was given in [7]. For important insights, alternative formulations of BDDC, and simplified proofs of these results, see [1] and [4].

In the case of many substructures, solving the coarse problem exactly is becoming a bottleneck. Since the coarse problem in BDDC has the same form as the original problem, the BDDC method can be applied recursively to solve the coarse problem approximately, leading to a multilevel form of BDDC in a straightforward manner [2]. Polylogarithmic condition number bounds for three-level BDDC (BDDC with two coarse levels) were proved in [10, 9]. This contribution is concerned with condition number estimates of BDDC with an arbitrary number of levels.

2 Abstract Multispace BDDC

All abstract spaces in this paper are finite dimensional. The dual space of a linear space U is denoted by U', and $\langle \cdot, \cdot \rangle$ is the duality pairing. We wish to solve the abstract linear problem

$$u \in U : a(u,v) = \langle f, v \rangle , \quad \forall v \in U, \tag{1}$$

for a given $f \in U'$, where a is a symmetric positive semidefinite bilinear form on some space $W \supset U$ and positive definite on U. The form $a(\cdot,\cdot)$ is called the energy inner product, the value of the quadratic form $a(u,u)$ is called the energy of u, and

* Partially supported by the NSF under grant CNS-0325314.

† Partially supported by the Grant Agency of the Czech Republic under grant 106/05/2731 and by the Program of the Information Society 1ET400300415.

the norm $\|u\|_a = a(u,u)^{1/2}$ is called the energy norm. The operator $A : U \mapsto U'$ associated with a is defined by

$$a(u,v) = \langle Au, v \rangle, \quad \forall u, v \in U.$$

Algorithm 1 (Abstract multispace BDDC) *Given spaces V_k and operators Q_k $(k = 1, \dots, M)$ such that*

$$U \subset V_1 + \cdots + V_M \subset W, \quad Q_k : V_k \to U,$$

define a preconditioner $B : r \in U' \longmapsto u \in U$ by

$$B : r \mapsto \sum_{k=1}^{M} Q_k v_k, \quad v_k \in V_k : \quad a(v_k, z_k) = \langle r, Q_k z_k \rangle, \quad \forall z_k \in V_k.$$

The following estimate can be proved from the abstract additive Schwarz theory [3]. The case when $M = 1$, which covers the existing two-level BDDC theory set in the spaces of discrete harmonic functions, was given in [8].

Lemma 1. *Assume that the subspaces V_k are energy orthogonal, the operators Q_k are projections, and*

$$\forall u \in U : u = \sum_{k=1}^{M} Q_k v_k \text{ if } u = \sum_{k=1}^{M} v_k, \; v_k \in V_k. \tag{2}$$

Then the abstract multispace BDDC preconditioner from Algorithm 1 satisfies

$$\kappa = \frac{\lambda_{\max}(BA)}{\lambda_{\min}(BA)} \leq \omega = \max_k \sup_{v_k \in V_k} \frac{\|Q_k v_k\|_a^2}{\|v_k\|_a^2}.$$

Note that (2) is a type of decomposition of unity property.

3 BDDC for a 2D Model Problem

Let $\Omega \subset \mathbb{R}^2$ be a bounded polygonal domain, decomposed into N nonoverlapping polygonal substructures Ω_i, $i = 1, ..., N$, which form a conforming triangulation. That is, if two substructures have a nonempty intersection, then the intersection is a vertex, or a whole edge. Let W_i be the space of Lagrangian $\mathbb{P}_1$ or $\mathbb{Q}_1$ finite element functions with characteristic mesh size h on Ω_i, and which are zero on the boundary $\partial\Omega$. Suppose that the nodes of the finite elements coincide on edges common to two substructures. Let

$$W = W_1 \times \cdots \times W_N,$$

$U \subset W$ be the subspace of functions that are continuous across the substructure interfaces, and

$$a(u,v) = \sum_{i=1}^{N} \int_{\Omega_i} \nabla u \nabla v, \quad u, v \in W.$$

We are interested in the solution of the problem (1).

Substructure vertices will be also called corners, and the values of functions from W on the corners are called *coarse degrees of freedom*. Let $\widetilde{W} \subset W$ be the space of all functions such that the values of any coarse degrees of freedom have a common value for all relevant substructures and vanish on $\partial\Omega$. Define $U_I \subset U \subset \widetilde{W}$ as the subspace of all functions that are zero on all substructure boundaries $\partial\Omega_i$, $\widetilde{W}_\Delta \subset \widetilde{W}$ as the subspace of all function such that their coarse degrees of freedom vanish, define $\widetilde{W}_\Pi$ as the subspace of all functions such that their coarse degrees of freedom between adjacent substructures coincide, and such that their energy is minimal. Then

$$\widetilde{W} = \widetilde{W}_\Delta \oplus \widetilde{W}_\Pi, \quad \widetilde{W}_\Delta \perp_a \widetilde{W}_\Pi. \tag{3}$$

Functions that are a-orthogonal to U_I are called discrete harmonic. In [7] and [8], the analysis was done in spaces of discrete harmonic functions after eliminating U_I; this is not the case here, so $\widetilde{W}$ does not consist of discrete harmonic functions only.

Let $E : \widetilde{W} \to U$ be the operator defined by taking the average over the substructure interfaces.

Algorithm 2 (Original BDDC) *Define the preconditioner $r \in U' \longmapsto u \in U$ as follows. Compute the interior pre-correction:*

$$u_I \in U_I : a\left(u_I, z_I\right) = \left\langle r, z_I \right\rangle, \quad \forall z_I \in U_I, \tag{4}$$

updated the residual:

$$r_B \in U', \quad \left\langle r_B, v \right\rangle = \left\langle r, v \right\rangle - a\left(u_I, v\right), \quad \forall v \in U$$

compute the substructure correction and the coarse correction:

$$\begin{aligned} u_\Delta &= E w_\Delta, \quad w_\Delta \in \widetilde{W}_\Delta : a\left(w_\Delta, z_\Delta\right) = \left\langle r_B, E z_\Delta \right\rangle, \quad \forall z_\Delta \in \widetilde{W}_\Delta \\ u_\Pi &= E w_\Pi, \quad w_\Pi \in \widetilde{W}_\Pi : a\left(w_\Pi, z_\Pi\right) = \left\langle r_B, E z_\Pi \right\rangle, \quad \forall z_\Pi \in \widetilde{W}_\Pi \end{aligned} \tag{5}$$

and the interior post-correction:

$$v_I \in U_I : a\left(v_I, z_I\right) = a\left(u_\Delta + u_\Pi, z_I\right), \quad \forall z_I \in U_I.$$

Apply the interior post-correction and add the interior pre-correction:

$$u = u_I + \left(u_\Delta + u_\Pi - v_I\right). \tag{6}$$

Denote by P the energy orthogonal projection from U to U_I. Then $I - P$ is known as the projection onto discrete harmonic functions.

Lemma 2. *The original BDDC preconditioner from Algorithm 2 is the abstract multispace BDDC from Algorithm 1 with $M = 3$ and*

$$\begin{aligned} &V_1 = U_I, \quad V_2 = (I-P)\widetilde{W}_\Delta, \quad V_3 = (I-P)\widetilde{W}_\Pi, \\ &Q_1 = I, \quad Q_2 = Q_3\left(I-P\right)E, \end{aligned}$$

and the assumptions of Lemma 1 are satisfied.

Because of (3) and and the fact that $\|I\|_a = 1$, we only need an estimate of $\|(I-P)\,Ew\|_a$ on $\widetilde{W}$, which is well known [6].

Theorem 1. *The condition number of the original BDDC algorithm satisfies $\kappa \le \omega$, where*

$$\omega = \sup_{w \in \widetilde{W}} \frac{\|(I-P)\,Ew\|_a^2}{\|w\|_a^2} \le C\left(1+\log\frac{H}{h}\right)^2 . \tag{7}$$

4 Multilevel BDDC and an Abstract Bound

The substructuring components from Section 3 will be denoted by an additional subscript $_1$, as Ω_1^i, $i = 1, \dots N_1$, etc., and called level 1. The spaces and operators involved can be written concisely as a part of a hierarchy of spaces and operators:

$$\left.\begin{array}{ccccccccc}
 & & U & & & & & & \\
 & & \parallel & & & & & & \\
U_{I1} & \overset{P_1}{\underset{\subset}{\leftarrow}} & U_1 & \overset{E_1}{\underset{\subset}{\leftarrow}} & \widetilde{W}_1 & \subset & W_1 & & \\
 & & & & \parallel & & & & \\
 & \widetilde{U}_2 & \overset{I_2}{\leftarrow} & \widetilde{W}_{\Pi 1} & \oplus & \widetilde{W}_{\Delta 1} & & & \\
 & & & \parallel & & & & & \\
 & U_{I2} & \overset{P_2}{\underset{\subset}{\leftarrow}} & U_2 & \overset{E_2}{\underset{\subset}{\leftarrow}} & \widetilde{W}_2 & \subset & W_2 & \\
 & & & & & \parallel & & & \\
 & & \widetilde{U}_3 & \overset{I_3}{\leftarrow} & \widetilde{W}_{\Pi 2} & \oplus & \widetilde{W}_{\Delta 2} & & \\
 & & & & \parallel & & & & \\
 & & & & \vdots & & & & \\
 & & & & \parallel & & & & \\
 & & U_{I,L-1} & \overset{P_{L-1}}{\underset{\subset}{\leftarrow}} & U_{L-1} & \overset{E_{L-1}}{\underset{\subset}{\leftarrow}} & \widetilde{W}_{L-1} & \subset & W_{L-1} \\
 & & & & & & \parallel & & \\
 & & & \widetilde{U}_L & \overset{I_L}{\leftarrow} & \widetilde{W}_{\Pi,L-1} & \oplus & \widetilde{W}_{\Delta,L-1} &
\end{array}\right\} \tag{8}$$

We will call the coarse problem (5) the level 2 problem. It has the same finite element structure as the original problem (1) on level 1, so we have $U_2 = \widetilde{W}_{\Pi 1}$. Level 1 substructures are level 2 elements, level 1 coarse degrees of freedom are level 2 degrees of freedom. The shape functions on level 2 are the coarse basis functions in $\widetilde{W}_{\Pi 1}$, which are given by the conditions that the value of exactly one coarse degree of freedom is one and others are zero, and that they are energy minimal in $\widetilde{W}_1$. Note that the resulting shape functions on level 2 are in general discontinuous between level 2 elements. Level 2 elements are then agglomerated into nonoverlapping level 2 substructures, etc. Level k elements are level $k-1$ substructures, and the level k substructures are agglomerates of level k elements. Level k substructures are denoted by Ω_k^i, and they are assumed to form a quasiuniform conforming triangulation with characteristic substructure size H_k. The degrees of freedom of level k elements are given by level $k-1$ coarse degrees of freedom, and shape functions on level k are determined by minimization of energy on each level $k-1$ substructure separately, so $U_k = \widetilde{W}_{\Pi,k-1}$. The mapping I_k is an interpolation from the level k degrees of freedom to functions in another space $\widetilde{U}_k$. For the model problem, $\widetilde{U}_k$ will consist of functions which are (bi)linear on each Ω_k^i. The averaging operators on level k, $E_k : \widetilde{W}_k \to U_k$, are defined by averaging of the values of level k degrees of freedom between level k substructures Ω_k^i. The space U_{Ik} consists of functions in U_k that are zero on the boundaries of all level k substructures, and $P_k : U_k \to U_{Ik}$ is the $a-$orthogonal projection in U_k onto U_{Ik}. For convenience, let Ω_0^i be the original finite elements, $H_0 = h$, and $I_1 = I$.

Algorithm 3 (Multilevel BDDC) *Given $r \in U_1'$, find $u \in U_1$ by (4)–(6), where the solution coarse problem (5) is replaced by the right hand side preconditioned by the same method, applied recursively. At the coarsest level, (5) is solved by a direct method.*

Lemma 3. *The multilevel BDDC preconditioner in Algorithm 3 is the abstract multispace BDDC preconditioner (Algorithm 1) with $M = 2L - 2$ and the spaces and operators*

$$V_1 = U_{I1}, \quad V_2 = (I - P_1)\widetilde{W}_{\Delta 1}, \quad V_3 = U_{I2}, \quad V_4 = (I - P_2)\widetilde{W}_{\Delta,2}, \ldots$$
$$V_{2L-4} = (I - P_{L-2})\widetilde{W}_{\Delta,L-2}, \quad V_{2L-3} = U_{I,L-1}, \quad V_{2L-2} = (I - P_{L-1})\widetilde{W}_{L-1},$$
$$Q_1 = I, \quad Q_2 = Q_3 = (I - P_1)\, E_1, \ldots$$
$$Q_{2L-4} = Q_{2L-3} = (I - P_1)\, E_1 \cdots (I - P_{L-2})\, E_{L-2},$$
$$Q_{2L-2} = (I - P_1)\, E_1 \cdots (I - P_{L-1})\, E_{L-1},$$

satisfying the assumptions of Lemma 1.

The following bound follows from writing of multilevel BDDC as multispace BDDC in Lemma 3 and the estimate for multispace BDDC in Lemma 1.

Lemma 4. *If for some $\omega_k \geq 1$,*

$$\|(I - P_k)E_k w_k\|_a^2 \leq \omega_k \|w_k\|_a^2, \quad \forall w_k \in \widetilde{W}_k, \quad k = 1, \ldots, L-1, \tag{9}$$

then the multilevel BDDC preconditioner satisfies $\kappa \leq \prod_{k=1}^{L-1} \omega_k$.

5 Multilevel BDDC Bound for the 2D Model Problem

To apply Lemma 4, we need to generalize the estimate (7) to coarse levels. From (7), it follows that for some $\widetilde{C}_k$ and all $w_k \in U_k$, $k = 1, \ldots, L-1$,

$$\min_{u_{Ik} \in U_{Ik}} \|I_k E_k w_k - I_k u_{Ik}\|_a^2 \leq \widetilde{C}_k \left(1 + \log \frac{H_k}{H_{k-1}}\right)^2 \|I_k w_k\|_a^2 . \tag{10}$$

Denote $|w|_{a,\Omega_k^i} = \left(\int_{\Omega_k^i} \nabla w \nabla w\right)^{1/2}$.

Lemma 5. *For all $k = 0, \ldots, L-1$, $i = 1, \ldots, N_k$,*

$$c_{k,1} \left|I_{k+1} w\right|_{a,\Omega_k^i}^2 \leq |w|_{a,\Omega_k^i}^2 \leq c_{k,2} \left|I_{k+1} w\right|_{a,\Omega_k^i}^2 , \quad \forall w \in \widetilde{W}_{\Pi k}, \ \forall\, \Omega_k^i, \tag{11}$$

with $c_{k,2}/c_{k,1} \leq \overline{C}_k$, independently of $H_0, \ldots, H_{k+1}$.

Proof. For $k = 0$, (11) holds because $I_1 = I$. Suppose that (11) holds for some $k < L - 2$ and let $w \in \widetilde{W}_{\Pi,k+1}$. From the definition of $\widetilde{W}_{\Pi,k+1}$ by energy minimization,

$$|w|_{a,\Omega^i_{k+1}} = \min_{w_\Delta \in \widetilde{W}_{\Delta,k+1}} |w + w_\Delta|_{a,\Omega^i_{k+1}} . \tag{12}$$

From (12) and the induction assumption, it follows that

$$\begin{aligned} c_{k,1} & \min_{w_\Delta \in \widetilde{W}_{\Delta,k+1}} |I_{k+1}w + I_{k+1}w_\Delta|^2_{a,\Omega^i_{k+1}} \\ & \leq \min_{w_\Delta \in \widetilde{W}_{\Delta,k+1}} |w + w_\Delta|^2_{a,\Omega^i_k} \leq c_{k,2} \min_{w_\Delta \in \widetilde{W}_{\Delta,k+1}} |I_{k+1}w + I_{k+1}w_\Delta|^2_{a,\Omega^i_k} \end{aligned} \tag{13}$$

Now from [10, Lemma 4.2], applied to the piecewise linear functions of the form $I_{k+1}w$ on Ω^i_{k+1},

$$c_1 |I_{k+2}w|^2_{a,\Omega^i_{k+1}} \leq \min_{w_\Delta \in \widetilde{W}_{\Delta,k+1}} |I_{k+1}w + I_{k+1}w_\Delta|^2_{a,\Omega^i_{k+1}} \leq c_2 |I_{k+2}w|^2_{a,\Omega^i_{k+1}} \tag{14}$$

with c_2/c_1, bounded independently of $H_0, \dots, H_{k+1}$. Then (12), (13) and (14) imply (11) with $\overline{C}_k = \overline{C}_{k-1}c_2/c_1$.

Theorem 2. *The multilevel BDDC with for the model problem with corner corner coarse degrees of freedom satisfies the condition number estimate*

$$\kappa \leq \textstyle\prod_{k=1}^{L-1} C_k \left(1 + \log \dfrac{H_k}{H_{k-1}}\right)^2 .$$

Proof. By summation of (11), we have

$$c_{k,1} \|I_k w\|^2_a \leq \|w\|^2_a \leq c_{k,2} \|I_k w\|^2_a , \quad \forall w \in U_k,$$

with $c_{k,2}/c_{k,1} \leq \overline{C}_k$, so from (10),

$$\|(I - P_k)E_k w_k\|^2_a \leq C_k \left(1 + \log \frac{H_k}{H_{k-1}}\right)^2 \|w_k\|^2_a , \quad \forall w_k \in \widetilde{W}_k,$$

with $C_k = \overline{C}_k \widetilde{C}_k$. It remains to use Lemma 4.

For $L = 3$ we recover the estimate by [10]. In the case of uniform coarsening, i.e. with $H_k/H_{k-1} = H/h$ and the same geometry of decomposition on all levels $k = 1, \dots L - 1$, we get

$$\kappa \leq C^{L-1} (1 + \log H/h)^{2(L-1)} . \tag{15}$$

6 Numerical Examples and Conclusion

A multilevel BDDC preconditioner was implemented in Matlab for the 2D Laplace equation on a square domain with periodic boundary conditions. For these boundary conditions, all subdomains at each level are identical and it is possible to solve very large problems on a single processor. The periodic boundary conditions result in a

Table 1. 2D Laplace equation results for $H/h = 2$. The number of levels is Nlev (Nlev $= 2$ for the standard approach), the number iterations is iter, the condition number estimate is κ, and the total number of degrees of freedom is ndof.

Nlev	corners only		corners and faces		ndof
	iter	κ	iter	κ	
2	2	1.5625	1	1	16
3	8	1.8002	5	1.1433	64
4	11	2.4046	7	1.2703	256
5	14	3.4234	8	1.3949	1,024
6	17	4.9657	9	1.5199	4,096
7	20	7.2428	9	1.6435	16,384
8	25	10.5886	10	1.7696	65,536

Table 2. 2D Laplace equation results for $H/h = 4$.

Nlev	corners only		corners and faces		ndof
	iter	κ	iter	κ	
2	9	2.1997	6	1.1431	256
3	14	4.0220	8	1.5114	4,096
4	21	7.7736	10	1.8971	65,536
5	30	15.1699	12	2.2721	1,048,576

Table 3. 2D Laplace equation results for $H/h = 8$.

Nlev	corners only		corners and faces		ndof
	iter	κ	iter	κ	
2	14	3.1348	7	1.3235	4,096
3	23	7.8439	10	2.0174	262,144
4	36	19.9648	13	2.7450	16,777,216

stiffness matrix with a single zero eigenvalue, but this situation can be accommodated in preconditioned conjugate gradients by removing the mean from the right hand side of $Ax = b$. The coarse grid correction at each level is replaced by the BDDC preconditioned coarse residual.

Numerical results are in Tables 1-3. As predicted by Theorem 2, the condition number grows slowly in the ratios of mesh sizes for a fixed number of levels L. However, for fixed H_i/H_{i-1} the growth of the condition number is seen to be exponential in L. With additional constraints by side averages, the condition number is seen to grow linearly. Our explanation is that a bound similar to Theorem 2 still applies, though possibly with (much) smaller constants, so the exponential growth of the condition number is no longer apparent.

References

[1] S.C. Brenner and L.-Y. Sung. BDDC and FETI-DP without matrices or vectors. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1429–1435, 2007.

[2] C.R. Dohrmann. A preconditioner for substructuring based on constrained energy minimization. *SIAM J. Sci. Comput.*, 25(1):246–258, 2003.

[3] M. Dryja and O.B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.*, 48(2):121–155, 1995.

[4] J. Li and O.B. Widlund. FETI-DP, BDDC, and block Cholesky methods. *Internat. J. Numer. Methods Engrg.*, 66(2):250–271, 2006.

[5] J. Mandel. Balancing domain decomposition. *Comm. Numer. Methods Engrg.*, 9(3):233–241, 1993.

[6] J. Mandel and C.R. Dohrmann. Convergence of a balancing domain decomposition by constraints and energy minimization. *Numer. Linear Algebra Appl.*, 10(7):639–659, 2003.

[7] J. Mandel, C.R. Dohrmann, and R. Tezaur. An algebraic theory for primal and dual substructuring methods by constraints. *Appl. Numer. Math.*, 54(2):167–193, 2005.

[8] J. Mandel and B. Sousedík. Adaptive selection of face coarse degrees of freedom in the BDDC and the FETI-DP iterative substructuring methods. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1389–1399, 2007.

[9] X. Tu. Three-level BDDC in three dimensions. To appear in *SIAM J. Sci. Comput.*, 2007.

[10] X. Tu. Three-level BDDC in two dimensions. *Internat. J. Numer. Methods Engrg.*, 69(1):33–59, 2007.

The All-floating BETI Method: Numerical Results

Günther Of

Institute of Computational Mathematics, Graz University of Technology,
Steyrergasse 30, A-8010 Graz, Austria, of@tugraz.at

Summary. The all-floating BETI method considers all subdomains as floating subdomains and improves the overall asymptotic complexity of the BETI method. This effect and the scalability of the method are shown in numerical examples.

1 Introduction

The boundary element tearing and interconnecting (BETI) method has been derived by [9] as the boundary element counterpart of the well-known FETI methods introduced by [4]. In the standard BETI method, floating and non-floating subdomains have to be treated differently. This is rather easy for the potential equation, but in linear elastostatics it gets more involved since the number of rigid body motions which have to be considered may vary from one subdomain to another. The FETI–DP methods, see [3], introduce some global primal variables to guarantee the invertibility of all local Steklov-Poincaré operators. The choice of these primal variables is important for the performance and gets more involved in linear elastostatics; see [7]. The all-floating BETI method overcomes these difficulties and improves the overall asymptotic complexity. The idea is to consider all subdomains as floating subdomains by tearing off the Dirichlet boundary conditions. This gives a unified treatment for all subdomains and an “optimal” preconditioning of the Steklov-Poincaré operators. At the DD17 Conference a similar approach was presented for the FETI method, called TotalFETI; see [2].

2 Boundary Element Tearing and Interconnecting Method

As model problem the Dirichlet boundary value problem

$$\begin{aligned} -\mathrm{div}[\alpha(x)\nabla u(x)] &= 0 \qquad && \text{for } x \in \Omega \\ u(x) &= g(x) \qquad && \text{for } x \in \Gamma = \partial\Omega \end{aligned}$$

of the potential equation is considered. $\Omega \subset \mathbb{R}^3$ is a bounded Lipschitz domain decomposed into p non-overlapping subdomains Ω_i with Lipschitz boundaries $\Gamma_i =$

$\partial\Omega_i$. Further, the coefficient function $\alpha(\cdot)$ is piecewise constant such that $\alpha(x) = \alpha_i > 0$ for $x \in \Omega_i,\ i = 1, \ldots, p$. Instead of the global boundary value problem, the corresponding local boundary value problems may be considered for local functions u_i with transmission conditions

$$u_i(x) = u_j(x) \qquad \text{and} \qquad \alpha_i \frac{\partial}{\partial n_i} u_i(x) + \alpha_j \frac{\partial}{\partial n_j} u_j(x) = 0$$

for $x \in \Gamma_{ij} = \Gamma_i \cap \Gamma_j$ and $i \leq j$. n_i denotes the outer normal vector of the subdomain Ω_i. The Dirichlet domain decomposition method is based on a strong coupling of the Dirichlet data across the coupling interfaces by introducing a global function $u \in H^{1/2}(\Gamma_s)$ on the skeleton $\Gamma_S := \bigcup_{i=1}^{p} \Gamma_i$. A weak coupling of the Neumann data is applied using a variational formulation. After the discretization, the global system of linear equations

$$\widetilde{S}_h \widetilde{\mathbf{u}} = \sum_{i=1}^{p} A_i^\top \widetilde{S}_{i,h} A_i \widetilde{\mathbf{u}} = \sum_{i=1}^{p} A_i^\top \mathbf{f}_i \tag{1}$$

has to be solved. The connectivity matrices $A_i \in \mathbb{R}^{M_i \times M}$ map the global nodes to the local nodes and the global vector $\widetilde{\mathbf{u}}$ to the local vectors $\widetilde{\mathbf{u}}_i = A_i \widetilde{\mathbf{u}}$. The coefficients of the vectors $\mathbf{f}_i$ of the right-hand side are given by

$$\mathbf{f}_i[k] = -\int_{\Gamma_i} (S_i \widetilde{g})(x) \varphi_k^i(x) ds_x.$$

Here, the potential $u = \widetilde{u} + \widetilde{g}$ is split into an extension $\widetilde{g}$ of the given Dirichlet data g and into the unknown part $\widetilde{u}$. A matching discretization of the boundaries Γ_i into plane triangles is used. The potentials u_i are approximated by piecewise linear and continuous basis functions $\{\psi_n^i\}_{n=1}^{N_i}$ on each subdomain. Piecewise constant basis functions $\{\varphi_k^i\}_{k=1}^{M_i}$ are used for the approximation of the local fluxes t_i. The matrices

$$\widetilde{S}_{i,h} = D_{i,h} + (\frac{1}{2} M_{i,h}^\top + K_{i,h}^\top) V_{i,h}^{-1} (\frac{1}{2} M_{i,h} + K_{i,h})$$

are discrete approximations of the local Steklov-Poincaré operators S_i, the so-called Dirichlet to Neumann maps. The boundary element matrices

$$\begin{aligned} V_{i,h}[\ell,k] &= \langle V_i \varphi_k^i, \varphi_\ell^i \rangle_{\Gamma_i}, & K_{i,h}[\ell,n] &= \langle K_i \psi_n^i, \varphi_\ell^i \rangle_{\Gamma_i}, \\ D_{i,h}[m,n] &= \langle D_i \psi_n^i, \psi_m^i \rangle_{\Gamma_i}, & M_{i,h}[\ell,n] &= \langle \psi_n^i, \varphi_\ell^i \rangle_{\Gamma_i} \end{aligned}$$

are realized by the fast multipole method, see [5], using integration by parts for the matrix $D_{i,h}$ of the hypersingular operator; see [12]. The use of the fast multipole method reduces the quadratic effort for a matrix times vector multiplication and the quadratic memory requirements of a standard boundary element method to almost linear ones up to some polylogarithmic factor. The involved boundary integral operators are the single layer potential V_i, the double layer potential K_i and the hypersingular operator D_i defined by

$$\begin{aligned} (V_i t_i)(x) &= \int_{\Gamma_i} U^*(x,y) t_i(y) ds_y, \\ (K_i u_i)(x) &= \int_{\Gamma_i} \frac{\partial}{\partial n_{i,y}} U^*(x,y) u_i(y) ds_y, \end{aligned}$$

$$(D_i u_i)(x) = -\frac{\partial}{\partial n_{i,x}} \int_{\Gamma_i} \frac{\partial}{\partial n_{i,y}} U^*(x,y) u_i(y) ds_y.$$

The global system of linear equations (1) is preconditioned by

$$C_{\widetilde{S}}^{-1} = \sum_{i=1}^{p} A_i^\top V_{i,lin,h} A_i$$

in the conjugate gradient method. This preconditioner provides a good preconditioning of the local Steklov-Poincaré operators S_i; see [14].

2.1 Standard BETI Method

Instead of the global system (1), the equivalent minimization problem

$$F(\widetilde{\mathbf{u}}) = \min_{\widetilde{\mathbf{v}} \in \mathbb{R}^M} \sum_{i=1}^{p} \left[\frac{1}{2} (\widetilde{S}_{i,h} A_i \widetilde{\mathbf{v}}, A_i \widetilde{\mathbf{v}}) - (\mathbf{f}_i, A_i \widetilde{\mathbf{v}}) \right] \tag{2}$$

is considered in the BETI method. Introducing local vectors $\widetilde{\mathbf{v}}_i := A_i \widetilde{\mathbf{v}}$ tears off the local potentials $\widetilde{v}_i$ at the coupling interfaces. Therefore, only local minimization problems have to be considered. The interconnection is done by introducing the constraints

$$\sum_{i=1}^{p} B_i \mathbf{v}_i = \mathbf{0} \tag{3}$$

to reinforce the continuity of the potentials across the coupling interfaces. Each line of a matrix B_i has at most one non-zero entry. This entry is either 1 or -1. At a global node with r adjacent subdomains $r-1$ constraints are used to guarantee that the corresponding local coefficients of these subdomains are equal. So non redundant Lagrange multipliers are used. It remains to solve p local minimization problems with the constraints (3). Introducing Lagrangian multipliers $\boldsymbol{\lambda}$ gives the system of linear equations

$$\begin{pmatrix} \widetilde{S}_{1,h} & & & -B_1^\top \\ & \ddots & & \vdots \\ & & \widetilde{S}_{p,h} & -B_p^\top \\ B_1 & \dots & B_p & 0 \end{pmatrix} \begin{pmatrix} \widetilde{\mathbf{u}}_1 \\ \vdots \\ \widetilde{\mathbf{u}}_p \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_p \\ \mathbf{0} \end{pmatrix}. \tag{4}$$

The local Steklov-Poincaré operators of the subdomains which share a piece with the Dirichlet boundary Γ are invertible, i.e., $\widetilde{\mathbf{u}}_i = \widetilde{S}_{i,h}^{-1}(\mathbf{f}_i + B_i^\top \boldsymbol{\lambda})$. The local Steklov-Poincaré operators of the other subdomains, called floating subdomains, are singular. For suitable compatibility and normalization conditions, the local solutions can be given by $\widetilde{\mathbf{u}}_i = \widehat{S}_{i,h}^{-1}(\mathbf{f}_i + B_i^\top \boldsymbol{\lambda}) + \gamma_i$ with some arbitrary constants γ_i and modified local Steklov-Poincaré operators defined by

$$\langle \widehat{S}_i u, v \rangle := \langle S_i u, v \rangle + \beta_i \langle u, 1 \rangle_{\Gamma_i} \langle v, 1 \rangle_{\Gamma_i}.$$

The parameters β_i of this stabilization can be chosen suitably; see [13]. If the first q subdomains are floating subdomains and the remaining are non-floating, the system (4) can be written as the Schur complement system

$$\left[\sum_{i=1}^{q} B_i \widehat{S}_{i,h}^{-1} B_i^\top + \sum_{i=q+1}^{p} B_i S_{i,h}^{-1} B_i^\top\right] \boldsymbol{\lambda} + G\boldsymbol{\gamma} = \sum_{i=1}^{q} B_i \widehat{S}_{i,h}^{-1} \mathbf{f}_i + \sum_{i=q+1}^{p} B_i S_{i,h}^{-1} \mathbf{f}_i \qquad (5)$$

or in the compact form

$$F\boldsymbol{\lambda} + G\boldsymbol{\gamma} = \mathbf{d} \qquad \text{with} \qquad G^\top \boldsymbol{\lambda} = ((\mathbf{f}_i, \mathbf{1}))_{i=1:q}$$

and $G = (B_1\mathbf{1}, \ldots, B_q\mathbf{1})$. As [6], the Lagrangian multipliers $\boldsymbol{\lambda}$ and the constants $\boldsymbol{\gamma}$ are determined by

$$P^\top F\boldsymbol{\lambda} = P^\top \mathbf{d} \qquad \text{and} \qquad \boldsymbol{\gamma} = (G^\top QG)^{-1} G^\top Q(\mathbf{d} - F\boldsymbol{\lambda})$$

with the orthogonal projection $P = I - QG(G^\top QG)^{-1}G^\top$. Using the scaled hypersingular BETI preconditioner, see [9],

$$C^{-1} = (BC_\alpha^{-1}B^\top)^{-1} BC_\alpha^{-1} D_h C_\alpha^{-1} B^\top (BC_\alpha^{-1}B^\top)^{-1}$$

with appropriate scaling matrices C_α, see [1, 6], the condition number of the preconditioned BETI system can be estimated by

$$\kappa(C^{-1}F) \le c\,(1 + \log H/h)^2$$

independent of jumps in the coefficients α_i; see [9].

2.2 All-floating BETI Method

A disadvantage of the BETI formulation (4) is that the condition number for the inversion of the local Steklov-Poincaré operator of non-floating subdomains is increasing logarithmically for the used preconditioning by the single layer potential as a boundary integral operator of opposite order; see [10]. The all-floating BETI method considers all subdomains as floating subdomains by tearing off the Dirichlet boundary conditions. This gives a simple unified treatment for all subdomains and a "optimal" preconditioning of the local Steklov-Poincaré operators.

As in the case of the standard BETI method, the global minimization problem (2) is split into local minimization problems

$$F(\widetilde{\mathbf{u}}_i) = \min_{\widetilde{\mathbf{v}}_i} \frac{1}{2}(\widetilde{S}_{i,h}\widetilde{\mathbf{v}}_i, \widetilde{\mathbf{v}}_i) + (\widetilde{S}_{i,h}\widetilde{\mathbf{g}}_i, \widetilde{\mathbf{v}}_i)$$

by introducing the local vectors $\widetilde{\mathbf{v}}_i = A_i\widetilde{\mathbf{v}}$. Now, the unknown part $\widetilde{\mathbf{v}}_i$ of the local Dirichlet datum and the known part $\widetilde{\mathbf{g}}_i$ given by the Dirichlet boundary conditions are reunited in the function $v_i = \widetilde{v}_i + \widetilde{g}_i$. The coefficients of these local functions can be determined by equivalent local minimization problems

$$\widetilde{F}(\mathbf{u}_i) = \min_{\mathbf{v}_i \in \mathbb{R}^{M_i}} \frac{1}{2}(\widetilde{S}_{i,h}\mathbf{v}_i, \mathbf{v}_i).$$

Additional local constraints are used to guarantee that the Dirichlet boundary conditions are satisfied, i.e., $\sum_{i=1}^p \widetilde{B}_i\mathbf{v}_i = \mathbf{b}$. These constraints include the constraints of the standard BETI method, for which the entries of the right hand side $\mathbf{b}$ are zero. The additional local constraints are of the type $\mathbf{v}_i[k] = g(x_k)$ where k is the local index of a Dirichlet node x_k. Then the corresponding line of the matrix $\widetilde{B}_i$ has one non-zero entry equal to one and the entry of the right hand side $\mathbf{b}$ is given by $g(x_k)$. Again, Lagrangian multipliers $\boldsymbol{\lambda}$ are introduced to get the system of linear equations

$$\begin{pmatrix} \widetilde{S}_{1,h} & & & -\widetilde{B}_1^\top \\ & \ddots & & \vdots \\ & & \widetilde{S}_{p,h} & -\widetilde{B}_p^\top \\ \widetilde{B}_1 & \dots & \widetilde{B}_p & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_1 \\ \vdots \\ \mathbf{u}_p \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \vdots \\ \mathbf{0} \\ \mathbf{b} \end{pmatrix}.$$

The corresponding Schur complement system has the simpler structure

$$\sum_{i=1}^{p} \widetilde{B}_i \widehat{S}_{i,h}^{-1} \widetilde{B}_i^\top \boldsymbol{\lambda} + G\boldsymbol{\gamma} = \mathbf{b}.$$

This system can be solved as described for the standard BETI method, but now all subdomains are floating subdomains. The all-floating approach can be extended to mixed boundary value problems and linear elastostatics; see [11].

The BETI system and the all-floating system are solved as two-fold saddle point problems, which are derived by reintroducing the local fluxes. In these formulations, no interior inversions of the local Steklov-Poincaré operators and of the single layer potentials are needed. Therefore, this is the fastest way to solve the BETI systems; see [8]. If the used algebraic multigrid preconditioners of the local single layer potentials are optimal and the two-fold saddle point formulations are used, the number of iterations of the conjugate gradient method for the standard BETI method is bounded by $\mathcal{O}((1+\log(H/h))^2)$ and the total complexity is of order $\mathcal{O}(N_i \log^4 N_i)$, since the fast multipole method has a complexity of order $\mathcal{O}(N_i \log^2 N_i)$. In this notation, h is the global mesh-size and H denotes the diameter of the subdomains. For the all-floating BETI method, the number of iterations is reduced to the order of $\mathcal{O}(1+\log(H/h))$ and the number of arithmetic operations is of order $\mathcal{O}(N_i \log^3 N_i)$ correspondingly. This will be proven in an upcoming paper for linear elastostatics.

3 Numerical Results

As an academic test example, the domain decomposition is given by a cube subdivided into eight smaller cubes with boundaries of 24 triangles each. The robustness of the preconditioner with respect to jumping coefficients has been shown in a previous paper; see [8]. Here, a constant coefficient $\alpha = 1$ for all subdomains is considered. In Table 1, the computational times t_1 and t_2 for setting up the system and for solving in seconds and the numbers of iterations It of the conjugate gradient method with a relative accuracy of 10^{-8} are compared for the standard Dirichlet domain decomposition method (1) and for the twofold saddle point formulations of the standard and of the all-floating BETI methods are compared for six uniform refinement steps. Note that the problem sizes of the local subproblems with N_i boundary elements are large.

On the first refinement levels, there is an overhead of the iteration numbers and the times for solving the system of the all-floating method in comparison to the standard BETI method. This effect is due to the larger number of degrees of freedom of the all-floating method. The improved asymptotic complexity of the all-floating formulation pays off for the last three refinement levels, as the numbers of iterations

Table 1. Comparison of the BETI methods

		DDD (1)			BETI			all-floating		
L	N_i	t_1	t_2	It.	t_1	t_2	It.	t_1	t_2	It.
0	24	1	0	4(1)	2	1	11	2	2	28
1	96	1	1	14(8)	3	2	37	3	3	40
2	384	5	4	20(10)	4	6	40	8	7	40
3	1536	14	25	22(11)	15	38	45	22	38	43
4	6144	77	167	24(11)	76	243	50	93	227	46
5	24576	333	1626	26(11)	333	2036	56	342	1798	48
6	98304	1443	9262	29(12)	1445	10130	62	1477	8693	51

are reduced. Finally, the all-floating method is faster than the standard domain decomposition method. The Dirichlet problem is the most challenging problem for the all-floating method, since the Dirichlet problem gives more additional constraints than a mixed boundary value problem. Therefore, the speedup by the all-floating method is better for mixed boundary value problems; see [11].

Finally, a test of the scalability of the all-floating method is presented. The results of a domain decomposition of the cube into 64 subcubes are compared to the results of a domain decomposition into eight subcubes. The triangulation of the surfaces of the eight cubes is finer with 96 instead of 24 triangles per subcube on the coarsest level, such that the triangulations of the whole cube are the same for both decompositions. Therefore, the decompositions are comparable. The results for five refinement steps are given in Table 2.

Table 2. Scalability of the all-floating BETI method

	8 finer subdomains						64 subdomains					
L	N_i	#duals	t_1	t_2	It.	D-error	N_i	#duals	t_1	t_2	It.	D-error
0	96	221	4	3	29	$1.35e-2$	24	613	5	6	32	$1.10e-2$
1	384	753	8	7	36	$3.88e-3$	96	1865	7	5	37	$3.45e-3$
2	1536	2777	21	34	41	$9.91e-4$	384	6481	11	10	47	$9.09e-4$
3	6144	10665	82	194	46	$2.28e-4$	1536	24161	23	48	52	$2.22e-4$
4	24576	41801	287	1811	53	$6.13e-5$	6144	93313	90	307	60	$4.67e-5$
5	98304	165513	1358	10485	62	$1.61e-5$	24576	366785	312	2658	70	$1.40e-5$

The iteration numbers of the decomposition into 64 subdomains are slightly increased compared to the eight subdomains, but this may be caused by the more complex coupling with up to 26 neighbors instead of seven. Except for the first three levels, the computational times for the 64 subdomains are about four times faster than for the eight subdomains. This is the most one can expect, since the additional coupling interfaces double the numbers of local degrees of freedom. Due to these additional local degrees of freedom, the more accurate approximations of the local Steklov-Poincaré operators give a reduced L_2 error for the approximation of the potential. The total numbers of degrees of freedom are 1345177 for the decomposition into eight subdomains and 2726209 for the decomposition into 64 subdomains.

4 Conclusion

The all-floating BETI method simplifies the treatment of floating and non-floating subdomains and improves the asymptotic behavior of the BETI method. In combination with a fast multipole boundary element method, it provides an almost optimal complexity with respect to the number of iterations, the arithmetical complexity and the memory requirements. The all-floating BETI method has already been extended to mixed boundary value problems and linear elastostatics.

Acknowledgement. This work has been supported by the German Research Foundation 'Deutsche Forschungsgemeinschaft (DFG)' under the grant SFB 404 'Multifield Problems in Continuum Mechanics', University of Stuttgart.

References

[1] S. C. Brenner. An additive Schwarz preconditioner for the FETI method. *Numer. Math.*, 94(1):1–31, 2003.

[2] Z. Dostál, D. Horák, and R. Kučera. Total FETI - an easier implementable variant of the FETI method for numerical solution of elliptic PDE. *Comm. Numer. Methods Engrg.*, 22(12):1155–1162, 2006.

[3] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.*, 7(7-8):687–714, 2000.

[4] C. Farhat and F.-X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Int. J. Numer. Methods Eng.*, 32(6):1205–1227, 1991.

[5] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[6] A. Klawonn and O. B. Widlund. FETI and Neumann-Neumann iterative substructuring methods: Connections and new results. *Commun. Pure Appl. Math.*, 54(1):57–90, 2001.

[7] A. Klawonn and O. B. Widlund. Dual-Primal FETI methods for linear elasticity. *Comm. Pure Appl. Math.*, 59:1523–1572, 2006.

[8] U. Langer, G. Of, O. Steinbach, and W. Zulehner. Inexact data–sparse boundary element tearing and interconnecting methods. *SIAM J. Sci. Comput.*, 29(1):290–314, 2007.

[9] U. Langer and O. Steinbach. Boundary element tearing and interconnecting methods. *Computing*, 71(3):205–228, 2003.

[10] W. McLean and O. Steinbach. Boundary element preconditioners for a hypersingular integral equation on an interval. *Adv. Comput. Math.*, 11(4):271–286, 1999.

[11] G. Of. *BETI–Gebietszerlegungsmethoden mit schnellen Randelementverfahren und Anwendungen.* PhD thesis, Universität Stuttgart, 2006.

[12] G. Of, O. Steinbach, and W. L. Wendland. The fast multipole method for the symmetric boundary integral formulation. *IMA J. Numer. Anal.*, 26:272–296, 2006.

[13] O. Steinbach. *Numerische Näherungsverfahren für elliptische Randwertprobleme. Finite Elemente und Randelemente.* B.G. Teubner, Stuttgart, Leipzig, Wiesbaden, 2003.

[14] O. Steinbach and W. L. Wendland. The construction of some efficient preconditioners in the boundary element method. *Adv. Comput. Math.*, 9(1-2):191–216, 1998.

MINISYMPOSIUM 6: Multiphysics Problems

Organizers: Ronald H.W. Hoppe[1] and Ralf Kornhuber[2]

[1] University of Augsburg, Institute of Mathematics, Germany.
hoppe@math.uni-augsburg.de

[2] Freie Universität Berlin, Institut für Mathematik II, Germany.
kornhuber@math.fu-berlin.de

Coupled heterogeneous phenomena are not an exception but the rule in advanced numerical simulations of fluid dynamics, microelectronics, hydrodynamics, haemodynamics, electrodynamics or acoustics. Mathematical understanding and efficient numerical solvers become more and more important. The aim of this minisymposium is to bring together scientists working in this field to report about recent developments.

Modeling and Simulation of Piezoelectrically Agitated Acoustic Streaming on Microfluidic Biochips

Harbir Antil[1], Andreas Gantner[2], Ronald H.W. Hoppe[1,2], Daniel Köster[2], Kunibert Siebert[2], and Achim Wixforth[3]

[1] University of Houston, Department of Mathematics
(http://www.math.uh.edu/~rohop/)
[2] University of Augsburg, Institute for Mathematics
(http://scicomp.math.uni-augsburg.de)
[3] University of Augsburg, Institute of Physics
(http://www.physik.uni-augsburg.de/exp1)

Summary. Biochips, of the microarray type, are fast becoming the default tool for combinatorial chemical and biological analysis in environmental and medical studies. Programmable biochips are miniaturized biochemical labs that are physically and/or electronically controllable. The technology combines digital photolithography, microfluidics and chemistry. The precise positioning of the samples (e.g., DNA solutes or proteins) on the surface of the chip in pico liter to nano liter volumes can be done either by means of external forces (active devices) or by specific geometric patterns (passive devices). The active devices which will be considered here are nano liter fluidic biochips where the core of the technology are nano pumps featuring surface acoustic waves generated by electric pulses of high frequency. These waves propagate like a miniaturized earthquake, enter the fluid filled channels on top of the chip and cause an acoustic streaming in the fluid which provides the transport of the samples. The mathematical model represents a multiphysics problem consisting of the piezoelectric equations coupled with multiscale compressible Navier-Stokes equations that have to be treated by an appropriate homogenization approach. We discuss the modeling approach and present algorithmic tools for numerical simulations as well as visualizations of simulation results.

1 Introduction

Microfluidic biochips and microarrays are used in pharmaceutical, medical and forensic applications as well as in academic research and development for high throughput screening, genotyping and sequencing by hybridization in genomics, protein profiling in proteomics, and cytometry in cell analysis (see [7]). Traditional technologies such as fluorescent dyes, radioactive markers, or nanoscale gold-beads only allow a relatively small number of DNA probes per assay, and they do not provide information about the kinetics of the processes. With the need for better sensitivity, flexibility, cost-effectiveness and a significant speed-up of the hybridization, the current technological trend is to integrate the microfluidics on the chips itself. A new type of nanotechnological devices are Surface Acoustic Wave (SAW) driven microfluidic biochips (cf. [4, 9]).

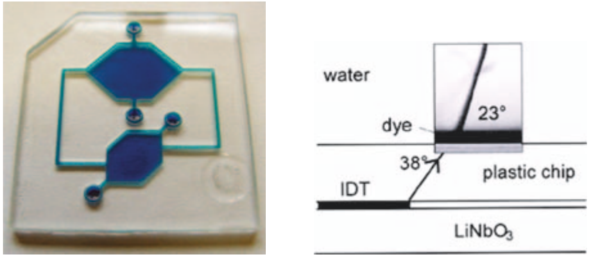


Fig. 1. Microfluidic biochip with network of microchannels (left), and sharp jet created by surface acoustic waves (right)

The experimental technique is based on piezoelectrically actuated Surface Acoustic Waves (SAW) on the surface of a chip which transport the droplet containing probe along a lithographically produced network of microchannels to marker molecules placed at prespecified surface locations (cf. Fig. 1 (left)). These microfluidic biochips allow the in-situ investigation of the dynamics of hybridization processes with extremely high time resolution.

The SAWs are excited by interdigital transducers and are diffracted into the device where they propagate through the base and enter the fluid filled microchannel creating a sharp jet on a time-scale of nanoseconds (cf. Fig. 1 (right)). The acoustic waves undergo a significant damping along the microchannel resulting in an acoustic streaming on a time-scale of milliseconds. The induced fluid flow transports the probes to reservoirs within the network where a chemical analysis is performed.

2 Modeling of SAW Driven Microfluidic Biochips

Mathematical models for SAW biochips are based on the linearized equations of piezoelectricity in $Q_1 := (0, T_1) \times \Omega_1$

$$\rho_1 \frac{\partial^2 u_i}{\partial t^2} - \frac{\partial}{\partial x_j} c_{ijkl} \frac{\partial u_k}{\partial x_l} - \frac{\partial}{\partial x_j} e_{kij} \frac{\partial \Phi}{\partial x_k} = 0 \ , \tag{1a}$$

$$\frac{\partial}{\partial x_j} e_{jkl} \frac{\partial u_k}{\partial x_l} - \frac{\partial}{\partial x_j} \epsilon_{jk} \frac{\partial \Phi}{\partial x_k} = 0 \tag{1b}$$

with appropriate initial conditions at $t = 0$ and boundary conditions on $\Gamma_1 := \partial\Omega_1$. Here, ρ_1 and $\mathbf{u} = (u_1, u_2, u_3)^T$ denote the density of the piezoelectric material and the mechanical displacement vector. Moreover, $\boldsymbol{\epsilon} = (\epsilon_{ij})$ stands for the permittivity tensor and Φ for the electric potential. The tensors $\mathbf{c} = (c_{ijkl})$ and $\mathbf{e} = (e_{ikl})$ refer to the forth order elasticity tensor and third-order piezoelectric tensor, respectively. The modeling of the micro-fluidic flow is based on the compressible Navier-Stokes equations in $Q_2 := (0, T_2) \times \Omega_2$

$$\begin{aligned} \rho_2 \, \big(\frac{\partial \mathbf{v}}{\partial t} + (\mathbf{v} \cdot \nabla)\mathbf{v}\big) \ &= \ - \nabla p + \eta \ \Delta \mathbf{v} + \big(\zeta + \frac{\eta}{3}\big) \ \nabla(\nabla \cdot \mathbf{v}) \ , && \text{(2a)} \\ \frac{\partial \rho_2}{\partial t} + \nabla \cdot (\rho_2 \ \mathbf{v}) \ &= 0 \ , && \text{(2b)} \\ \mathbf{v}(x + \mathbf{u}(x,t), t) \ &= \ \frac{\partial \mathbf{u}}{\partial t}(x,t) \quad \text{on } (0, T_2) \times \Gamma_2 && \text{(2c)} \end{aligned}$$

with suitable initial conditions at $t = 0$ (see, e.g., [1, 2]). In the model, compressible and non-linear effects are the driving force of the resulting flow. Here, $\rho_2, \mathbf{v} = (v_1, v_2, v_3)^T$ and p are the density of the fluid, the velocity, and the pressure. η and ζ refer to the shear and the bulk viscosity. The boundary conditions include the time derivative $\partial \mathbf{u}/\partial t$ of the displacement of the walls $\Gamma_2 = \partial \Omega_2$ of the microchannels caused by the surface acoustic waves. Therefore, the coupling of the piezoelectric and the Navier-Stokes equations is only in one direction. No back-coupling of the fluid onto the SAWs is considered. It has to be emphasized that the induced fluid flow involves extremely different time scales. The damping of the sharp jets created by the SAWs represents a process with a time scale of nanoseconds, whereas the resulting acoustic streaming reaches an equilibrium on a time scale of milliseconds.

SAWs are usually excited by interdigital transducers located at $\Gamma_{1,D} \subset \Gamma_1$ operating at a frequency $f \approx 100$ MHz with wavelength $\lambda \approx 40$ μm. The time-harmonic ansatz leads to the saddle point problem:

Find $(\mathbf{u}, \Phi) \in \mathbf{V} \times W$, where $\mathbf{V} \subset H^1(\Omega_1)^3, W \subset H^1(\Omega_1)$, such that for all $(\mathbf{w}, \Psi) \in \mathbf{V} \times W$

$$\begin{aligned} &\int\limits_{\Omega_1} c_{ijkl} \varepsilon_{kl}(\mathbf{u}) \varepsilon_{ij}(\bar{\mathbf{w}}) dx - \omega^2 \int\limits_{\Omega_1} \rho_1 u_i \bar{w}_i dx + \int\limits_{\Omega_1} e_{kij} \frac{\partial \Phi}{\partial x_k} \varepsilon_{ij}(\bar{\mathbf{w}}) dx = <\boldsymbol{\sigma}_{\mathbf{n}}, \mathbf{w}>, \\ &\int\limits_{\Omega_1} e_{ijk} \ \varepsilon_{ij}(\mathbf{u}) \ \frac{\partial \bar{\Psi}}{\partial x_k} \ dx \qquad - \qquad \int\limits_{\Omega_1} \epsilon_{ij} \ \frac{\partial \Phi}{\partial x_i} \ \frac{\partial \bar{\Psi}}{\partial x_j} \ dx \quad = < D_n, \Psi > . \end{aligned}$$

Here, $(\varepsilon_{ij}(\mathbf{u}))$ stands for the linearized strain tensor and $\omega = 2\pi f$. The elastic and electric Neumann boundary data are supposed to satisfy $\boldsymbol{\sigma}_n \in H^{-\frac{1}{2}}(\Gamma_\sigma)^2$ and $D_n \in H^{-\frac{1}{2}}(\Gamma_D)$ with $< \cdot, \cdot >$ in the above system denoting the respective dual products. Then, the following results holds true (cf. [3]):

Theorem 1. *For the above saddle point problem, the Fredholm alternative holds true. In particular, if ω^2 is not an eigenvalue of the associated eigenvalue problem, there exists a unique solution $(\mathbf{u}, \Phi) \in \mathbf{V} \times W$.*

In order to cope with the two time-scales character of the fluid flow in the microchannels (penetration of the SAWs within nanoseconds and induced acoustic streaming within milliseconds), we perform a separation of the time-scales by homogenization. In particular, we consider an expansion of the velocity $\mathbf{v}$ in a scale parameter $\varepsilon > 0$ representing the maxímal displacement of the walls

$$\mathbf{v} \;=\; \mathbf{v}_0 \;+\; \varepsilon\,\mathbf{v}' \;+\; \varepsilon^2\,\mathbf{v}'' \;+\; O(\varepsilon^3)$$

and analogous expansions of the pressure p and the density ρ_2. We set $\mathbf{v}_1 := \varepsilon\mathbf{v}', \mathbf{v}_2 := \varepsilon^2\mathbf{v}''$ and define $p_i, \rho_{2,i}, 1 \leq i \leq 2$, analogously. Collecting all terms of order $O(\varepsilon)$ results in the linear system

$$\rho_{2,0}\frac{\partial \mathbf{v}_1}{\partial t} \;-\; \eta\;\Delta\mathbf{v}_1 \;-\; \Big(\zeta + \frac{\eta}{3}\Big)\;\nabla(\nabla\cdot\mathbf{v}_1) \;+\; \nabla p_1 \;= \mathbf{0} \qquad \text{in } Q_2\ , \tag{3a}$$

$$\frac{\partial \rho_{2,1}}{\partial t} \;+\; \rho_{2,0}\;\nabla\cdot\mathbf{v}_1 = \mathbf{0} \qquad \text{in } Q_2\ , \tag{3b}$$

$$p_1 \;=\; c_0^2\;\rho_{2,1} \quad \text{in } Q_2 \quad , \quad \mathbf{v}_1 \;=\; \frac{\partial \mathbf{u}}{\partial t} \quad \text{on } \Gamma_2\ , \tag{3c}$$

where c_0 represents the small signal sound speed in the fluid. The system describes the propagation of damped acoustic waves.
Collecting all terms of order $O(\varepsilon^2)$ and performing the time-averaging $\langle w\rangle := T^{-1}\int_{t_0}^{t_0+T} w\,dt$, where $T := 2\pi/\omega$, we arrive at the Stokes system in Ω_2

$$-\eta\;\Delta\mathbf{v}_2 - \Big(\zeta + \frac{\eta}{3}\Big)\,\nabla(\nabla\cdot\mathbf{v}_2) + \nabla p_2 \;= \langle -\rho_{2,1}\frac{\partial \mathbf{v}_1}{\partial t} - \rho_{2,0}[\nabla\mathbf{v}_1]\mathbf{v}_1\rangle\ , \tag{4a}$$

$$\rho_{2,0}\nabla\cdot\mathbf{v}_2 \;= \langle -\nabla\cdot(\rho_{2,1}\mathbf{v}_1)\rangle\ , \tag{4b}$$

$$\mathbf{v}_2 \;=\; -\;\langle[\nabla\mathbf{v}_1]\mathbf{u}\rangle \quad \text{on } \Gamma_2\ . \tag{4c}$$

The Stokes system describes the stationary flow pattern, called acoustic streaming, resulting after the relaxation of the high frequency surface acoustic waves.
As far as analytical results for the Navier-Stokes equations (3a)-(3c) are concerned, one can show existence and uniqueness of a weak periodic solution assuming the forcing term to be a periodic function. Moreover, under some extra regularity assumption it can be shown that the periodically extended solution converges to an oscillating equilibrium state (see [5]).

Theorem 2. *Assume that the forcing term is a periodic function of period T. Then, the linear Navier-Stokes equations* (3a)-(3c) *have a unique weak periodic solution* $(\mathbf{v}_{per}, p_{per}) \in H^1((0,T); H^{-1}(\Omega)^3 \times L_0^2(\Omega))$.
Moreover, if $(\tilde{\mathbf{v}}, \tilde{p})$ resp. $(\tilde{\mathbf{v}}_{per}, \tilde{p}_{per})$ are extensions of the solution resp. the periodic solution of the Navier-Stokes equation with periodic forcing term to arbitrary large $\tau > 0$ and if $(\partial_{tt}^2\tilde{\mathbf{v}}, \partial_{tt}^2\tilde{p}), (\partial_{tt}^2\tilde{\mathbf{v}}_{per}, \partial_{tt}^2\tilde{p}_{per}) \in L^2((0,\tau);\mathbf{H})$, where $\mathbf{H} := L^2(\Omega)^3 \times L_0^2(\Omega)$, then there holds

$$\|(\tilde{\mathbf{v}}(t), \tilde{p}(t)) - (\tilde{\mathbf{v}}_{per}(t), \tilde{p}_{per}(t))\|_{\mathbf{H}} \;=\; O(t^{-1/2})\ .$$

3 Simulation of the SAWs and the Microfluidic Flows

The time-harmonic acoustic problem is solved in the frequency domain by using $\mathbb{P}_1$ conforming finite elements with respect to a hierarchy of simplicial triangulations. This leads to an algebraic saddle point problem

$$\begin{pmatrix} A_\omega & B \\ B^T & -C \end{pmatrix}\begin{pmatrix} U \\ \Phi \end{pmatrix} \;=\; \begin{pmatrix} f \\ g \end{pmatrix}\ . \tag{5}$$

For the numerical solution of (5) we use multilevel preconditioned CG for the associated Schur complementbased on a block-diagonal preconditioner

$$P^{-1} = \begin{pmatrix} \tilde{A}_\omega^{-1} & 0 \\ 0 & \tilde{C}^{-1} \end{pmatrix} \tag{6}$$

with BPX- or hierarchical-type preconditioners for the stiffness matrices associated with the mechanical displacement and electric potential, respectively (see [3] for details).

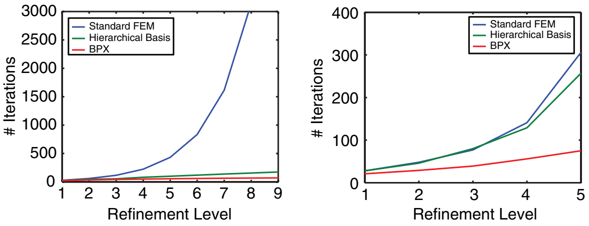


Fig. 2. Performance of the multilevel preconditioned CG compared with standard CG in 2D (left) and 3D (right)

Fig. 2 displays the performance of the multilevel preconditioners compared to the standard single-grid iterative solver both in 2D (on the left) and in 3D (on the right). For the BPX-preconditioner we observe the expected asymptotic independence on the refinement level (cf., e.g., [6]).

The Navier Stokes equations (3a)-(3c) with a periodic excitation on the boundary Γ_2 are discretized in time by the Θ-scheme (cf., e.g., [8]), until a specific condition for periodicity of the pressure is met. The discretization in space is taken care of by Taylor-Hood elements with respect to a hierarchy of simplicial triangulations. On the other hand, for the discretization of the time-averaged Stokes system we use the same techniques as for the time-harmonic acoustic problem (see [5] for details).

4 Numerical Simulation Results

The following simulation results are based on 2D computations that have been carried out for $LiNbO_3$ as the piezoelectric material and assuming the fluid in the microchannels to be water at 20°C. For a precise specification of the geometrical and material data we refer to [3] and [5]. Fig. 3 shows the amplitudes of the electric potential at an operating frequency of 100 MHz (left) and the polarized Rayleigh waves by means of the displacement vectors (right). The amplitudes of the displacement waves are in the region of nanometers. The SAWs are strictly confined to the surface of the substrate. Their penetration depth into the piezoelectric material is in the range of one wavelength. Rayleigh surface waves characteristically show an elliptical displacement, i.e., the displacements in the x_1- and x_2-direction are 90^o out of phase with one another. Additionally, the amplitude of the surface displacement in the x_2-direction is larger than that along the SAW propagation axis x_1.

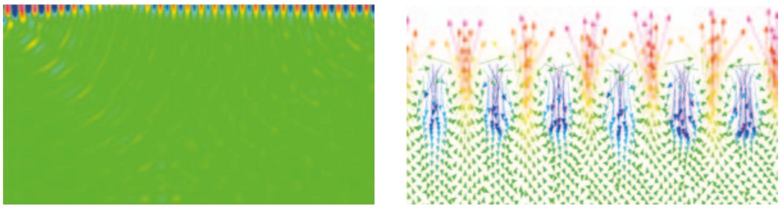


Fig. 3. Electric potential wave (100 MHz) (left) and mechanical displacement vectors (right)

Fig. 4 (left) displays the effective force creating the sharp jet in the fluid (cf. Fig. 1 (right)) which can be easily computed by means of $\mathbf{F} := \rho_{2,0}\langle(\mathbf{v}_1\cdot\nabla)\mathbf{v}_1 + \mathbf{v}_1(\nabla\cdot\mathbf{v}_1)\rangle$, where the brackets stand for the time average removing the fast oscillations of the sound wave. Fig. 4 (right) contains a visualization of the associated velocity field.

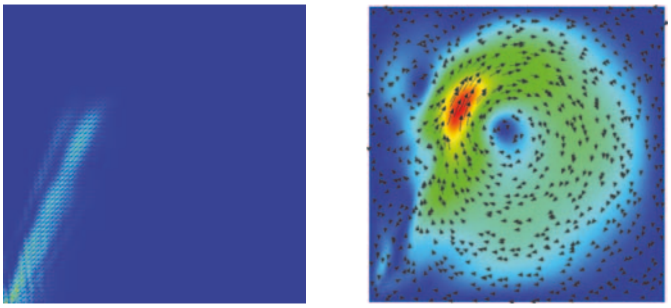


Fig. 4. Effective force and associated velocity field

Fig. 5 shows the strong damping of the acoustic waves in the fluid where excitation occurs through an SAW running from left to right along the lower edge at a frequency of 100 MHz.

We have performed a model validation by a comparison of experimental data with numerical simulation results. Fig. 6 (left) displays the measured streaming pattern visualized by a fluorescence video microscope for an experimental layout consisting of a typical biochip with an IDT placed on top of a standard YXl 128$\hat{o}$ substrate (the IDT is visible at the bottom right of Fig. 6 (left)). Fig. 6 (right) shows the result of a

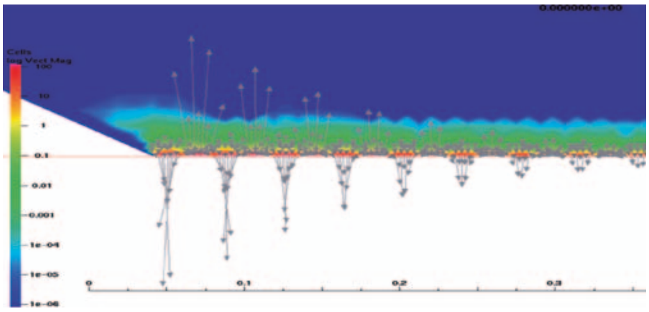


Fig. 5. Strong damping of the SAWs after penetration into the fluid

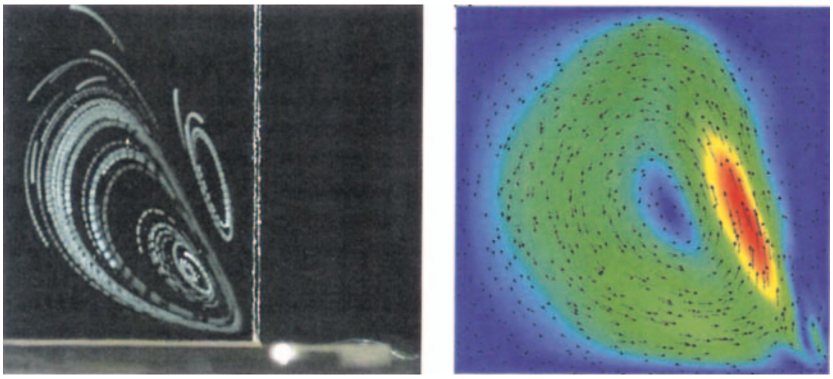


Fig. 6. Model validation: experimental measurements (left) and numerical simulation results (right)

simulation run based on the data of the experimental setting. A similar qualitative behavior can be observed. More importantly, for the resulting acoustic streaming the simulation results are quantitatively in good agreement with the experimental data.

Acknowledgement. The authors acknowledge support by the German National Science Foundation within the Priority program 'Optimization with Partial Differential Equations'. The work of the first author is further partially supported by the NSF under Grant-No. DMS-0511611.

References

[1] C.E. Bradley. Acoustic streaming field structure: The influence of the radiator. *J. Acoust. Soc. Am.*, 100:1399–1408, 1996.

[2] B. Desjardins and C.-K. Lin. A survey of the compressible Navier-Stokes equations. *Taiwanese Journal of Mathematics*, 3:123–137, 1999.

[3] A. Gantner, R.H.W. Hoppe, D. Köster, K. Siebert, and A. Wixforth. Numerical simulation of piezoelectrically agitated surface acoustic waves on microfluidic biochips. *Comput. Vis. Sci.*, 10(3):145–161, 2007.

[4] Z. Guttenberg, H. Müller, H. Habermüller, A. Geisbauer, J. Pipper, J. Felbel, M. Kilepinski, J. Scriba, and A. Wixforth. Planar chip device for pcr and hybridization with surface acoustic wave pump. *Lab Chip*, 5:308–317, 2005.

[5] D. Köster. Numerical simulation of acoustic streaming on saw-driven biochips. *Dissertation, Inst. of Math., Univ. Augsburg*, 2006.

[6] P. Oswald. *Multilevel Finite Element Approximations: Theory and Applications.* Teubner, Stuttgart, 1994.

[7] J. Pollard and B. Castrodale. Outlook for dna microarrays: emerging applications and insights on optimizing microarray studies. *Report. Cambridge Health Institute, Cambridge*, 2003.

[8] J.W. Thomas. *Numerical Partial Differential Equations: Finite Difference Methods.* Springer, Berlin-Heidelberg-New York, 1995.

[9] A. Wixforth, C. Strobl, C. Gauer, A. Toegl, J. Scriba, and Z. Guttenberg. Acoustic manipulation of small droplets. *Anal. Bioanal. Chem.*, 379:982–991, 2004.

Numerical Approximation of a Steady MHD Problem

Marco Discacciati

Johann Radon Institute for Computational and Applied Mathematics (RICAM),
Altengerberstraße 69, A-4040 Linz, Austria. marco.discacciati@oeaw.ac.at

Summary. We consider a magnetohydrodynamic (MHD) problem which models the steady flow of a conductive incompressible fluid confined in a bounded region and subject to the Lorentz force exerted by the interaction of electric currents and magnetic fields. We present an iterative method inspired to operator splitting to solve this nonlinear coupled problem, and a discretization based on conforming finite elements.

1 Introduction

MHD studies the interaction of electrically conductive fluids and electromagnetic fields. One of the most interesting aspects of this interaction is the possibility to generate the so-called Lorentz's force, which permits to influence the motion of the fluid in a completely contactless way. With this respect, an important application of MHD occurs in the production of metals.

The mathematical modeling of the processes taking place in such industrial plants is very involved since it requires to take into account many phenomena (multiphase and free-surface flows, electromagnetic fields, temperature effects, chemical reactions, etc.). However, the core model describing the interaction between the liquid metal and the magnetic fields is a nonlinear system formed by Navier-Stokes' and Maxwell's equations coupled by Ohm's law and Lorentz's force. The literature concerning both the mathematical analysis and the finite element approximation of this coupled problem is broad (see, e.g., [5, 7, 8, 9, 10] and references therein).

In this paper, we consider a formulation of a steady MHD problem as a nonlinear coupled system in five unknowns, namely, magnetic field, velocity and pressure of the fluid, electric currents and potential, which presents a "nested" saddle-point structure. Moreover, following the common numerical approach in electromagnetism, we express the magnetic field as the solution of a curl-curl problem, instead of using the Biot-Savart law (see, e.g., [10]).

After briefly discussing the well-posedness of this problem (Sect. 2), we propose and analyze an iterative solution method based on operator-splitting techniques (Sect. 3). In Sect. 4, we present a conforming finite element approximation, and we discuss the algebraic form of the iterative schemes. Finally, we present some numerical results (Sect. 5).

2 Setting and Well-Posedness of the Problem

We consider a bounded domain $\Omega \subset \mathbb{R}^3$ of class $\mathscr{C}^{1,1}$ (see, e.g., [1]), which contains a bounded Lipschitz subdomain $\Omega_f \subset \Omega$ filled by an electrically conductive fluid. An external conductor Ω_s is attached to a part of the boundary $\Gamma_s \subset \partial\Omega_f$, in order to inject an electric current into Ω_f. Finally, let Ω_e be an external device which possibly generates a magnetic field $\mathbf{B}_e$. A schematic representation of the domain is shown in Fig. 1.

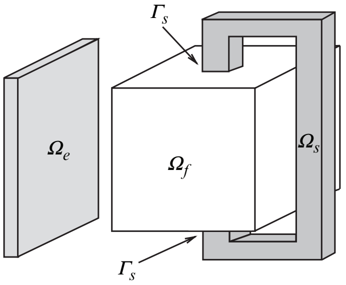


Fig. 1. Schematic representation of the computational setting

In Ω_s we assign an electric current $\mathbf{J}_s$ which originates a magnetic field, say $\bar{\mathbf{B}}_s$. This current is such that div $\mathbf{J}_s = 0$ in Ω_s, $\mathbf{J}_s \cdot \mathbf{n} = 0$ on $\partial\Omega_s \setminus \Gamma_s$, and $\mathbf{J}_s \cdot \mathbf{n} = j_s$ on Γ_s, where $\mathbf{n}$ denotes the unit normal vector directed outward of $\partial\Omega_f$. The (known) function $j_s \in L^2(\Gamma_s)$ fulfills the compatibility condition $\int_{\Gamma_s} j_s = 0$. We suppose that the contact interface Γ_s between Ω_f and Ω_s is perfectly conductive, i.e.

$$\mathbf{J}_s \cdot \mathbf{n} = j_s = \mathbf{J}_f \cdot \mathbf{n} \quad \text{on } \Gamma_s \,. \tag{1}$$

In the fluid domain Ω_f we have a current $\mathbf{J}_f$ which generates a magnetic field $\bar{\mathbf{B}}_f$. The global magnetic field $\mathbf{B}$ is thus due to the superposition of three components: $\mathbf{B} = \mathbf{B}_e + \bar{\mathbf{B}}_s + \bar{\mathbf{B}}_f$.

The motion of the incompressible conductive fluid in Ω_f is described by the steady Navier-Stokes' equations:

$$-\eta\triangle\mathbf{u} + \rho(\mathbf{u}\cdot\nabla)\mathbf{u} + \nabla p - \mathbf{J}_f \times \mathbf{B}|_{\Omega_f} = \mathbf{0}, \qquad \text{div}\,\mathbf{u} = 0 \quad \text{in } \Omega_f, \tag{2}$$

where $\mathbf{u}$ and p are the velocity and the pressure of the fluid, respectively, while $\eta, \rho > 0$ are the fluid viscosity and density. We supplement (2) with the Dirichlet boundary condition $\mathbf{u} = \mathbf{g}$ on $\partial\Omega_f$, $\mathbf{g}$ being an assigned velocity field such that $\int_{\partial\Omega_f} \mathbf{g}\cdot\mathbf{n} = 0$. $\mathbf{J}_f \times \mathbf{B}|_{\Omega_f}$ is the Lorentz force exerted on the fluid by the interaction of the magnetic field $\mathbf{B}$ and the electric current $\mathbf{J}_f$.

Finally, the electric current $\mathbf{J}_f$ satisfies

$$\sigma^{-1}\mathbf{J}_f + \nabla\phi - \mathbf{u}\times\mathbf{B}|_{\Omega_f} = \mathbf{0}, \qquad \operatorname{div}\mathbf{J}_f = 0 \quad \text{in } \Omega_f, \tag{3}$$

where ϕ is the electric potential and $\sigma > 0$ is the electric conductivity of the fluid. We impose the boundary condition (1) on Γ_s, while we set $\mathbf{J}_f \cdot \mathbf{n} = 0$ on $\partial\Omega_f \setminus \Gamma_s$.

In order to give a more useful representation of the magnetic field $\mathbf{B}$, we consider the divergence-free extension $\mathbf{E}_s j_s$ of j_s. $\mathbf{E}_s$ is a continuous extension operator $\mathbf{E}_s : L^2(\Gamma_s) \to \boldsymbol{H}(\text{div};\Omega_f)$, such that $\mathbf{E}_s j_s \cdot \mathbf{n} = 0$ on $\partial\Omega_f \setminus \Gamma_s$, $\mathbf{E}_s j_s \cdot \mathbf{n} = j_s$ on Γ_s, and $\operatorname{div}(\mathbf{E}_s j_s) = 0$ in Ω_f (see [2]). Then, we decompose $\mathbf{J}_f = \mathbf{J}_0 + \mathbf{E}_s j_s$, with $\mathbf{J}_0 \in \boldsymbol{H}(\text{div};\Omega_f)$, $\mathbf{J}_0 \cdot \mathbf{n} = 0$ on $\partial\Omega_f$.

Now, let us consider the currents

$$\overline{\mathbf{J}}_s = \begin{cases} \mathbf{E}_s j_s & \text{in } \Omega_f, \\ \mathbf{J}_s & \text{in } \Omega_s, \\ \mathbf{0} & \text{in } \Omega \setminus (\overline{\Omega}_f \cup \overline{\Omega}_s), \end{cases} \qquad \text{and} \quad \overline{\mathbf{J}}_0 = \begin{cases} \mathbf{J}_0 & \text{in } \Omega_f, \\ \mathbf{0} & \text{in } \Omega \setminus \overline{\Omega}_f, \end{cases} \tag{4}$$

where $\overline{\mathbf{J}}_0$ satisfies $\operatorname{div}\overline{\mathbf{J}}_0 = 0$ in Ω and $\overline{\mathbf{J}}_0 \cdot \mathbf{n} = 0$ on $\partial\Omega$. Then, the magnetic fields $\mathbf{B}_s$ and $\mathbf{B}_0$ generated by $\overline{\mathbf{J}}_s$ and $\overline{\mathbf{J}}_0$, respectively, can be represented as the solution of the problems:

$$\begin{aligned} \mathbf{curl}\,(\mu^{-1}\mathbf{B}_s) &= \overline{\mathbf{J}}_s \ \text{ in } \Omega\,, \\ \operatorname{div}\mathbf{B}_s &= 0 \quad \text{in } \Omega\,, \\ \mathbf{B}_s \cdot \mathbf{n} &= 0 \quad \text{on } \partial\Omega\,, \end{aligned} \qquad \text{and} \qquad \begin{aligned} \mathbf{curl}\,(\mu^{-1}\mathbf{B}_0) &= \overline{\mathbf{J}}_0 \ \text{ in } \Omega\,, \\ \operatorname{div}\mathbf{B}_0 &= 0 \quad \text{in } \Omega\,, \\ \mathbf{B}_0 \cdot \mathbf{n} &= 0 \quad \text{on } \partial\Omega\,. \end{aligned} \tag{5}$$

$\mu > 0$ is the magnetic permeability that we assume to be constant in Ω.

The usual approach to compute $\mathbf{B}_0$ is to introduce a vector potential $\mathbf{A}$ such that $\mathbf{curl}\,\mathbf{A} = \mathbf{B}_0$, and to reformulate the corresponding problem (5) as

$$\begin{aligned} \mathbf{curl}\,(\mu^{-1}\mathbf{curl}\,\mathbf{A}) + \varepsilon\mathbf{A} &= \overline{\mathbf{J}}_0 \quad \text{in } \Omega\,, \\ \mathbf{A}\times\mathbf{n} &= \mathbf{0} \quad \text{on } \partial\Omega\,. \end{aligned} \tag{6}$$

Remark that the boundary condition $\mathbf{A}\times\mathbf{n} = \mathbf{0}$ on $\partial\Omega$ implies $\mathbf{B}_0 \cdot \mathbf{n} = 0$ on $\partial\Omega$, and that the perturbation term of order $O(\varepsilon)$, with $0 < \varepsilon \ll 1$, has been added to guarantee the uniqueness of the solution $\mathbf{A}$, which otherwise would be defined only up to gradients of arbitrary scalar functions.

Using these notations, we can rewrite the magnetic field $\mathbf{B}$ as $\mathbf{B} = \mathbf{B}_e + \mathbf{B}_s + \mathbf{curl}\,\mathbf{A}$, $\mathbf{A}$ being the only unknown component which depends on the unknown current $\mathbf{J}_0$.

Notice that the MHD problem (2), (3) and (6) would be nonlinear even if we considered instead of (2) the Stokes equations:

$$-\eta\triangle\mathbf{u} + \nabla p - \mathbf{J}_f \times \mathbf{B}|_{\Omega_f} = \mathbf{0}, \qquad \operatorname{div}\mathbf{u} = 0 \quad \text{in } \Omega_f\,. \tag{7}$$

Indeed, there is an intrinsic nonlinearity due to the coupling terms $\mathbf{J}_f \times \mathbf{B}|_{\Omega_f}$ and $\mathbf{u}\times\mathbf{B}|_{\Omega_f}$. For simplicity, we illustrate our solution method avoiding for the moment the nonlinearity due to the convection term $(\mathbf{u}\cdot\nabla)\mathbf{u}$ in (2). Thus, from now on, we regard (7), (3), (6) as our MHD problem.

The well-posedness of the MHD system can be proved using the Banach Contraction Theorem. In particular, we can state the following result (see [2]).

Proposition 1. *Assume that $\mathbf{B}_{e|\Omega_f} \in (L^3(\Omega_f))^3$. If the physical parameters μ^{-1}, η, σ^{-1} are sufficiently large, whereas the boundary data $\mathbf{g} \in (H^{1/2}(\partial\Omega_f))^3$, $j_s \in L^2(\Gamma_s)$ and the assigned magnetic field $\mathbf{B}_e$ are small enough, the MHD problem has a unique solution $\mathbf{A} \in \boldsymbol{H}(\mathbf{curl};\Omega)$, $\mathbf{u} \in (H^1(\Omega_f))^3$, $p \in L^2_0(\Omega_f)$, $\mathbf{J}_f \in \boldsymbol{H}(\mathrm{div};\Omega_f)$, $\phi \in L^2_0(\Omega_f)$.*

The conditions imposed on the physical parameters are required to ensure the existence of a solution and its uniqueness. From the proof in [2], we can see that the larger the viscosity η of the fluid and the smaller its conductivity σ, the larger the boundary data $\mathbf{g}$ and j_s may become.

3 Iterative Solution Methods

In this section, we consider possible methods to compute the solution of the MHD problem by independently solving its fluid and magnetic-field subproblems. In particular, we first compute **curl A** in Ω, and then we linearize (7) and (3) in Ω_f using the magnetic field just obtained. The latter problems can be solved separately or in a coupled fashion. Ad-hoc solution techniques known in literature may be used to deal with each subproblem, and "reusage" of existing specific codes may be envisaged as well.

Precisely, we propose the following algorithm.

Consider an initial guess $\mathbf{J}_0^{(0)}$ for the electric current in Ω_f, and set $\overline{\mathbf{J}}_0^{(0)}$ as in (4). For $k \geq 0$, until convergence,

1. solve (6) with $\overline{\mathbf{J}}_0^{(k)}$ as right-hand side to compute $\mathbf{A}^{(k)}$.
2. Then, solve the problem in Ω_f using one of the following strategies:
   2a. *Coupled approach:* solve in Ω_f the system

$$-\eta\triangle\mathbf{u}^{(k+1)} + \nabla p^{(k+1)} - (\mathbf{J}_0^{(k+1)} + \mathbf{E}_s j_s) \times (\mathbf{B}_e + \mathbf{B}_s + \mathbf{curl}\,\mathbf{A}^{(k)}) = \mathbf{0}, \tag{8}$$

$$\mathrm{div}\,\mathbf{u}^{(k+1)} = 0, \tag{9}$$

$$\sigma^{-1}(\mathbf{J}_0^{(k+1)} + \mathbf{E}_s j_s) + \nabla\phi^{(k+1)} - \mathbf{u}^{(k+1)} \times (\mathbf{B}_e + \mathbf{B}_s + \mathbf{curl}\,\mathbf{A}^{(k)}) = \mathbf{0}, \tag{10}$$

$$\mathrm{div}\,(\mathbf{J}_0^{(k+1)} + \mathbf{E}_s j_s) = 0. \tag{11}$$

   2b. *Split approach:* solve first the Stokes problem (8)-(9) in Ω_f, taking $\mathbf{J}_0^{(k)}$ instead of $\mathbf{J}_0^{(k+1)}$. Then, using the velocity field $\mathbf{u}^{(k+1)}$ just computed, solve (10)-(11).
3. In both cases 2a/b, define the electric current at the successive step possibly considering a relaxation: $\mathbf{J}_0^{(k+1)} \leftarrow \theta\, \mathbf{J}_0^{(k+1)} + (1-\theta)\mathbf{J}_0^{(k)}$, $0 < \theta \leq 1$.

Under the same hypotheses of Proposition 1, we can show that there exists a positive radius $\rho_J > 0$ such that $\mathbf{J}_0^{(k)}$ converges with respect to the L^2-norm in the ball $B_J = \{\mathbf{J} \in \boldsymbol{H}(\mathrm{div};\Omega_f) : \|\mathbf{J}\|_{L^2(\Omega_f)} \leq \rho_J\}$ (see [2]).

4 Conforming Finite Element Approximation

Let $\mathcal{T}_h$ be a regular triangulation of $\overline{\Omega}$ made up of tetrahedra, such that the triangulations induced on $\Omega \setminus \overline{\Omega}_f$ and Ω_f are compatible on $\partial\Omega_f$.

We discretize the MHD system considering the $\boldsymbol{H}(\mathbf{curl};\Omega)$-conforming Nédélec elements to approximate the vector potential $\mathbf{A}$, the $\boldsymbol{H}(\mathrm{div};\Omega_f)$-conforming Raviart-Thomas elements for the electric current and potential, and the Taylor-Hood elements for the Stokes problem (see, e.g., [11] for a presentation of these spaces). Thanks to the inf-sup stability enjoyed by the Raviart-Thomas and the Taylor-Hood elements, it can be proved that also this compound finite element approximation is inf-sup stable, without any further compatibility requirement (see [2]).

Finally, remark that using Raviart-Thomas and Nédélec elements we can deal in a more natural way also with non-convex polyhedral domains, where the magnetic field is in general not in H^1, and it would be erroneously represented by elements of Lagrangian type.

Let us now briefly consider the algebraic form of step 2 in the algorithm of Sect. 3.

After computing the discrete field $\mathbf{A}_h^{(k)}$ (h denotes finite element approximations), we assemble the following matrix which realizes at the discrete level the coupling between the fluid and the electric-current problems in Ω_f:

$$\mathrm{C}_{ij}^{(k)} = -\int_{\Omega_f} [\mathbf{J}_h^i \times (\mathbf{B}_e^h + \mathbf{B}_s^h + \mathbf{curl}\,\mathbf{A}_h^{(k)})] \cdot \mathbf{v}_h^j \,.$$

($\mathbf{J}_h^i$ and $\mathbf{v}_h^j$ denote basis functions for the discrete spaces of the electric currents and fluid velocity, respectively.)

Then, in the coupled approach 2a, one has to solve the 4×4 block linear system:

$$\begin{pmatrix} \mathrm{A} & \mathrm{B}^T & (\mathrm{C}^{(k)})^T & 0 \\ \mathrm{B} & 0 & 0 & 0 \\ -\mathrm{C}^{(k)} & 0 & \mathrm{D} & \mathrm{E}^T \\ 0 & 0 & \mathrm{E} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_h^{(k+1)} \\ p_h^{(k+1)} \\ \mathbf{J}_{0,h}^{(k+1)} \\ \phi_h^{(k+1)} \end{pmatrix} = \mathbf{f} \,, \tag{12}$$

whose matrix presents a “nested” saddle-point structure.

On the other hand, the decoupled strategy 2b requires to solve first the linear system (which corresponds to the Stokes problem):

$$\begin{pmatrix} \mathrm{A} & \mathrm{B}^T \\ \mathrm{B} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_h^{(k+1)} \\ p_h^{(k+1)} \end{pmatrix} = \begin{pmatrix} -(\mathrm{C}^{(k)})^T \mathbf{J}_{0,h}^{(k)} \\ \mathbf{0} \end{pmatrix} + \text{boundary terms}, \tag{13}$$

and then the system associated to (3):

$$\begin{pmatrix} \mathrm{D} & \mathrm{E}^T \\ \mathrm{E} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{J}_{0,h}^{(k+1)} \\ \phi_h^{(k+1)} \end{pmatrix} = \begin{pmatrix} \mathrm{C}^{(k)} \mathbf{u}_h^{(k+1)} \\ \mathbf{0} \end{pmatrix} + \text{boundary terms.} \tag{14}$$

Thus, an iteration (13)-(14) corresponds to a Gauss-Seidel step for (12).

5 Numerical Results

We consider a rectangular domain Ω_f between two parallel conductive wires at 0.25 m from its lateral walls and bottom surface (Fig. 2, left). An electric current in the wires originates a magnetic field $\mathbf{B}_e$ (Fig. 2, right). We impose $\mathbf{u} = \mathbf{0}$ on $\partial\Omega_f$, $\mathbf{J}_f \cdot \mathbf{n} = 0$ on the lateral boundary, while we assign $\mathbf{J}_f \cdot \mathbf{n} = 20$ A on the top and bottom surfaces. The physical parameters are chosen to represent a melted metal.

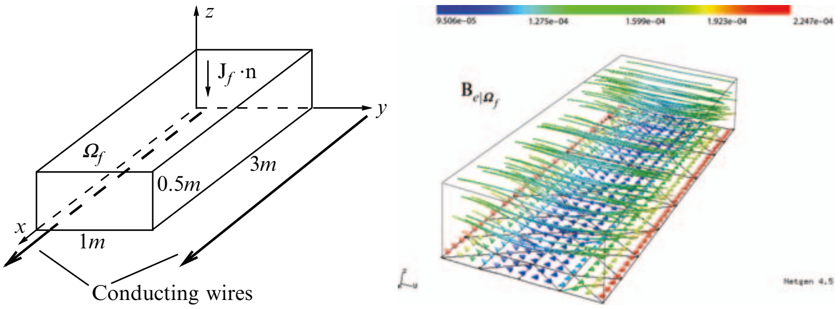


Fig. 2. Schematic representation of the setting (left), and restriction of the field lines of the magnetic field $\mathbf{B}_e$ to Ω_f (right)

We consider two uniform computational meshes made of tetrahedra, and few possible choices of the degree of the polynomials used for the finite element approximation. We apply the decoupled algorithm 2b, using a relaxation with $\theta = 0.4$. The stopping criterion is based on the relative increment of the unknown $\mathbf{J}_f$, with tolerance 10^{-5}. The implementation has been done within NGSolve (http://www.hpfem.jku.at/ngsolve/), while the linear systems have been solved using the direct solver PARDISO (http://www.computational.unibas.ch/cs/scicomp/software/pardiso/).

As shown in Fig. 3, the interaction between the magnetic fields and the electric current $\mathbf{J}_f$ gives rise to a double symmetric rotational movement of the fluid with larger velocity towards the bottom of Ω_f. The iterations required to convergence are reported in table 1. We observe that they remain bounded and essentially independent of the number of unknowns.

6 Conclusions and Perspectives

The preliminary numerical results obtained using the decoupled iterative scheme seem quite promising. However, a more thorough analysis of the convergence rate is in order, especially concerning the dependence on the mesh size h and on the degree of the polynomials used. Moreover, we are investigating effective techniques for the saddle-point problem (12) (see [4]), with particular interest in applications to optimal control, where problems with a similar structure are quite often encountered when imposing the optimality (Karush-Kuhn-Tucker) conditions (see [3, 6]).

Acknowledgement. The support of the Austrian Academy of Science is gratefully acknowledged.

References

[1] R.A. Adams. *Sobolev Spaces*. Academic Press, New York, 1975.

Table 1. Convergence results for two computational grids and different choices of the degrees of the finite elements

order of Nédélec FE		1	2	1
order of Taylor-Hood FE		2	2	2
order of Raviart-Thomas FE		1	1	2
Grid 1	dofs	6071	9018	11543
(344 tetrahedra)	iterates	22	22	22
Grid 2	dofs	42602	64224	85130
(2752 tetrahedra)	iterates	22	22	22

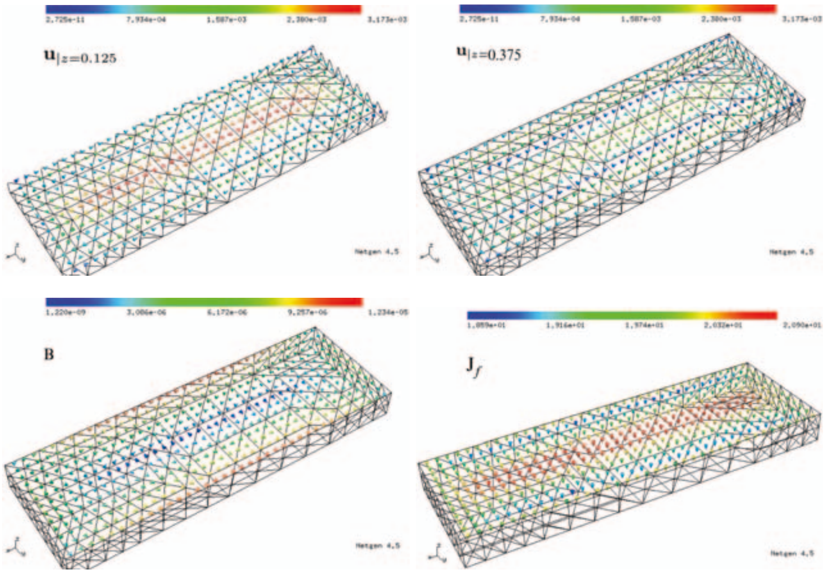


Fig. 3. Computed velocity field across the plane $z = 0.125$ (top left) and $z = 0.375$ (top right), total magnetic field $\mathbf{B}$ (bottom left), and electric current $\mathbf{J}_f$ (bottom right)

[2] M. Discacciati. Mathematical and numerical analysis of a stready magnetohydrodynamic problem. Technical Report 38, RICAM, Linz, 2006.

[3] M. Discacciati and R. Griesse. Finite element solution of a stationary optimal control problem in magnetohydrodynamics: Stokes case. Technical report, RICAM, Linz, 2007. In preparation.

[4] M. Discacciati and W. Zulehner. Preconditioning techniques for a saddle-point problem arising in magnetohydrodynamics. Technical report, RICAM, Linz, 2007. In preparation.

[5] J.F. Gérbeau, C. Le Bris, and T. Lelièvre. *Mathematical Methods for the Magnetohydrodynamics of Liquid Metals.* Oxford Science Publications, New York, 2006.

[6] R. Griesse and K. Kunisch. Optimal control for a stationary MHD system in velocity-current formulation. *SIAM J. Control Optim.*, 45(5):1822–1845, 2006.

[7] J.L. Guermond and P.D. Minev. Mixed finite element approximation of an MHD problem involving conducting and insulating regions: the 3D case. *Numer. Methods Partial Differential Equations*, 19(6):709–731, 2003.

[8] M.D. Gunzburger, A.J. Meir, and J.S. Peterson. On the existence, uniqueness, and finite element approximation of solutions of the equations of stationary, incompressible magnetohydrodynamics. *Math. Comp.*, 56(194):523–563, 1991.

[9] W.J. Layton, A.J. Meir, and P.G. Schmidt. A two-level discretization method for the stationary MHD equations. *Electr. Trans. Numer. Anal.*, 6:198–210, 1997.

[10] A.J. Meir and P.G. Schmidt. Analysis and numerical approximation of a stationary MHD flow problem with nonideal boundary. *SIAM J. Numer. Anal.*, 36(4):1304–1332, 1999.

[11] P. Monk. *Finite Element Methods for Maxwell's Equations.* Oxford Science Publications, New York, 2003.

Mortar and Discontinuous Galerkin Methods Based on Weighted Interior Penalties

Paolo Zunino

MOX, Department of Mathematics, Politecnico di Milano
paolo.zunino@polimi.it

1 Introduction

A wide class of discontinuous Galerkin (DG) methods, the so called interior penalty methods, arise from the idea that inter-element continuity could be attained by mimicking the techniques previously developed for weakly enforcing suitable boundary conditions for PDE's, see [7]. Although the DG methods are usually defined by means of the so called numerical fluxes between neighboring mesh cells, see [1], for most of the interior penalty methods for second order elliptic problems it is possible to correlate the expression of the numerical fluxes with a corresponding set of local interface conditions that are weakly enforced on each inter-element boundary. Such conditions are suitable to couple elliptic PDE's with smooth coefficients and it seems that a little attention is paid to the case of problems with discontinuous data or to the limit case where the viscosity vanishes in some parts of the computational domain.

In this paper, we discuss the derivation of a DG method arising from a set of generalized interface conditions, considered in [4], which are adapted to couple both elliptic and hyperbolic problems. In order to obtain such method, it is necessary to modify the definition of the numerical fluxes, replacing the standard arithmetic mean with suitably weighted averages where the weights depend on the coefficients of the problem. Even though the underlying ideas could be equivalently applied both to mortars and DG methods, we privilege here the discussion of the latter case, since the former has already been considered in [2].

In the framework of mortar finite-element methods, different authors have highlighted the possibility of using an average with weights that differ from one half, see [8, 5]. These works present several mortaring techniques to match conforming finite elements on possibly non conforming computational meshes. However, these works do not consider any connection between the averaging weights and the coefficients of the problem. More recently, Burman and Zunino [2] have introduced this dependence for an advection-diffusion-reaction problem with discontinuous viscosity, and they have shown that the application of the harmonic mean on the edges where the viscosity is discontinuous improves the stability of the numerical scheme. In this work, we aim to generalize the definition of such method and to apply it to the DG case. After introducing the model problem and some notation, particular attention will be devoted here to illustrate how the definition of the scheme and the corresponding numerical fluxes obey to the requirement of obtaining a method which is well posed and robust not only in the elliptic regimen, but also in the presence of a locally vanishing viscosity. A complete a-priori error analysis is not addressed here, but we illustrate the behavior of the method by means of some numerical tests.

2 Derivation of the Numerical Method

We aim to find u, the solution of the following boundary value problem,

$$\begin{cases} \nabla\cdot(-\epsilon\nabla u + \beta u) + \mu u = f & \text{in } \Omega \subset \mathbb{R}^d,\ d = 2, 3, \\ \left[\frac{1}{2}\big(|\beta \cdot n| - \beta \cdot n\big) + \chi_{\partial\Omega}(\epsilon)\right]u = 0 & \text{on } \partial\Omega, \end{cases} \tag{1}$$

where Ω is a polygonal domain, n is the outer normal unit vector with respect to $\partial\Omega$ and $\chi_{\partial\Omega}(\epsilon) \geq 0$, satisfying $\chi_{\partial\Omega}(0) = 0$, will be made precise later. Here $\mu \in L^\infty(\Omega)$ is a positive function and $\beta \in [W^{1,\infty}(\Omega)]^d$ is a vector function such that $\mu + \frac{1}{2}(\nabla \cdot \beta) \geq \mu_0 > 0$, $f \in L^2(\Omega)$ and ϵ is a nonnegative function in $L^\infty(\Omega)$. The well posedness of problem (1), with $\epsilon \in W^1_\infty(\Omega)$, is addressed in [6] and references therein.

For the numerical approximation of problem (1) we consider a shape regular triangulation T_h of the domain Ω, we denote with K an element in T_h and with $n_{\partial K}$ its outward unit normal. We define a totally discontinuous approximation space,

$$V_h := \{v_h \in L^2(\Omega);\ \forall K \in T_h, v_h|_K \in \mathbb{P}^k\},\ \text{with } k > 0.$$

Let Γ_e be the set of the element edges $e \subset \partial K$ such that $e \cap \partial\Omega = \emptyset$ and let n_e be the unit normal vector associated to e. Nothing is said hereafter depends on the arbitrariness on the sign of n_e. We denote with $\Gamma_{\partial\Omega}$ the collection of the edges on $\partial\Omega$. For all $e \in \Gamma_e \cup \Gamma_{\partial\Omega}$ let h_e be the size of the edge. For any $v_h \in V_h$ we define,

$$v_h^{\mp}(x) := \lim_{\delta\to 0^+} v_h(x \mp \delta n_e) \text{ for a.e. } x \in e,\ \text{with } e \in \Gamma_e.$$

When not otherwise indicated, the v_h^- value is implied. Similar definitions apply to all fields that are two-valued on the internal interfaces. The jump over interfaces is defined as $[\![v_h(x)]\!] := v_h^-(x) - v_h^+(x)$. We denote the arithmetic mean with $\{v_h(x)\} := \frac{1}{2}(v_h^-(x) + v_h^+(x))$. We also introduce the weighted averages for any $e \in \Gamma_e$ and a.e. $x \in e$,

$$\begin{aligned} \{v_h(x)\}_w &:= w_e^-(x)v_h^-(x) + w_e^+(x)v_h^+(x), \\ \{v_h(x)\}^w &:= w_e^+(x)v_h^-(x) + w_e^-(x)v_h^+(x), \end{aligned}$$

where the weights necessarily satisfy $w_e^-(x) + w_e^+(x) = 1$. We say that these averages are conjugate, because they satisfy the following identity,

$$[\![v_h w_h]\!] = \{v_h\}_w[\![w_h]\!] + \{w_h\}^w[\![v_h]\!],\ \forall v_h, w_h \in V_h. \tag{2}$$

The role of $\{\cdot\}_w$ and $\{\cdot\}^w$ can also be interchanged, but for symmetry this choice does not affect the final setting of the method. Finally, there is no need to extend the definitions of jumps and averages on the boundary $\partial\Omega$, because the contributions of Γ_e and $\Gamma_{\partial\Omega}$ will be always treated separately.

To set up a numerical approximation scheme for problem (1), we assume for simplicity that ϵ is piecewise constant on T_h. We define $\sigma_h(v_h) := -\epsilon\nabla v_h + \beta v_h$ or simply σ_h if the flux is applied to the primal unknown u_h, and we consider the Galerkin discretization method in V_h, which originates from the following expression,

$$\int_\Omega f v_h = \int_\Omega \Big(\nabla\cdot\sigma_h v_h + \mu u_h v_h\Big) = \sum_{K\in T_h}\int_K \Big(\nabla\cdot\sigma_h v_h + \mu u_h v_h\Big)$$
$$= \sum_{K\in T_h}\Big[\int_K \Big(-\sigma_h\cdot\nabla v_h + \mu u_h v_h\Big) + \int_{\partial K}\sigma_h\cdot n_{\partial K} v_h\Big],\ \forall v_h\in V_h. \quad (3)$$

Then, considering the identity,

$$\sum_{K\in T_h}\int_{\partial K}\sigma_h\cdot n_{\partial K}v_h = \sum_{e\in\Gamma_e}\int_e [\![\sigma_h v_h]\!]\cdot n_e + \sum_{e\in\Gamma_{\partial\Omega}}\int_e(\sigma_h v_h)\cdot n,$$

and replacing it into (3), owing to (2) we obtain,

$$\sum_{e\in\Gamma_e}\int_e\Big(\{\sigma_h\}_w\cdot n_e[\![v_h]\!] + [\![\sigma_h]\!]\cdot n_e\{v_h\}^w\Big) + \sum_{e\in\Gamma_{\partial\Omega}}\int_e \sigma_h\cdot n v_h$$
$$+\sum_{K\in T_h}\int_K\Big(-\sigma_h\cdot\nabla v_h + \mu u_h v_h\Big) = \int_\Omega f v_h,\ \forall v_h\in V_h. \quad (4)$$

We need now to apply suitable conditions on each inter-element interface and on the boundary of the domain. To this aim, we define $\gamma_e(\epsilon,\beta) := \frac{1}{2}\big(|\beta\cdot n_e| - \varphi_e(\epsilon)\beta\cdot n_e\big) + \chi_e(\epsilon)h_e^{-1}$, where $\chi_e(\epsilon)\geq 0$ such that $\chi_e(0)=0$ and $|\varphi_e(\epsilon)|\leq 1$ will be defined later, and we set $[\![\sigma_h]\!]\cdot n_e = 0$, $\gamma_e(\epsilon,\beta)[\![u_h]\!]=0$ on any $e\in\Gamma_e$. We also set $\gamma_{\partial\Omega}(\epsilon,\beta) := \frac{1}{2}\big(|\beta\cdot n| - \beta\cdot n\big) + \chi_{\partial\Omega}(\epsilon)h_e^{-1}$. Introducing the boundary and local interface conditions into (4) we obtain,

$$\sum_{K\in T_h}\int_K\Big(-\sigma_h\cdot\nabla v_h + \mu u_h v_h\Big) + \sum_{e\in\Gamma_e}\int_e\{\sigma_h\}_w\cdot n_e[\![v_h]\!] + \sum_{e\in\Gamma_{\partial\Omega}}\int_e\sigma_h\cdot n v_h$$
$$+\sum_{e\in\Gamma_e}\int_e\gamma_e(\epsilon,\beta)[\![u_h]\!][\![v_h]\!] + \sum_{e\in\Gamma_{\partial\Omega}}\int_e\gamma_{\partial\Omega}(\epsilon,\beta)u_h v_h = \int_\Omega f v_h, \forall v_h\in V_h. \quad (5)$$

The left hand side of equation (5) can be split in two parts. The former corresponds to the symmetric terms and it reads as follows,

$$a_h^s(u_h,v_h) := \sum_{K\in T_h}\int_K\Big[\epsilon\nabla u_h\cdot\nabla v_h + \big(\mu + \tfrac{1}{2}\nabla\cdot\beta\big)u_h v_h$$
$$+\sum_{e\in\Gamma_e}\int_e\Big[-\{\epsilon\nabla u_h\}_w\cdot n_e[\![v_h]\!] - \{\epsilon\nabla v_h\}_w\cdot n_e[\![u_h]\!] + \big(\tfrac{1}{2}|\beta\cdot n_e| + \chi_e(\epsilon)h_e^{-1}\big)[\![u_h]\!][\![v_h]\!]\Big]$$
$$+\sum_{e\subset\Gamma_{\partial\Omega}}\int_e\Big[-\epsilon\nabla u_h\cdot n v_h - \epsilon\nabla v_h\cdot n u_h + \big(\tfrac{1}{2}|\beta\cdot n| + \chi_{\partial\Omega}(\epsilon)h_e^{-1}\big)u_h v_h\Big],$$

where we have added the new terms $\{\epsilon \nabla v_h\}_w \cdot n_e [\![u_h]\!]$ on Γ_e and $\epsilon \nabla v_h \cdot n u_h$ on $\Gamma_{\partial\Omega}$ to preserve symmetry. The remaining part of the bilinear form is,

$$
\begin{aligned}
a_h^r(u_h, v_h) := &- \sum_{K \in \mathcal{T}_h} \int_K \Big[(\beta u_h) \cdot \nabla v_h + \tfrac{1}{2} (\nabla \cdot \beta) u_h v_h \Big] \\
&+ \sum_{e \in \Gamma_e} \int_e \Big[\{\beta u_h\}_w \cdot n_e [\![v_h]\!] - \tfrac{1}{2} \varphi_e(\epsilon) \beta \cdot n_e [\![u_h]\!][\![v_h]\!] \Big] + \sum_{e \in \Gamma_{\partial\Omega}} \int_e \tfrac{1}{2} \beta \cdot n u_h v_h.
\end{aligned}
$$

Then, setting $a_h(u_h, v_h) := a_h^s(u_h, v_h) + a_h^r(u_h, v_h)$ and $F(v_h) := \int_\Omega f v_h$ our prototype of method reads as follows: find $u_h \in V_h$ such that,

$$
a_h(u_h, v_h) = F(v_h), \ \forall v_h \in V_h. \tag{6}
$$

Before proceeding, we choose the weights w_e^-, w_e^+ on each edge such that $w_e^- \epsilon^- = w_e^+ \epsilon^+$, and accordingly we define, $\omega_e(\epsilon) := \frac{1}{2} \{\epsilon\}_w = w_e^- \epsilon^- = w_e^+ \epsilon^+$. Together with $w_e^+ + w_e^- = 1$ this leads to the expressions,

$$
\begin{aligned}
w_e^- &= \frac{\epsilon^+}{\epsilon^- + \epsilon^+}, \quad w_e^+ = \frac{\epsilon^-}{\epsilon^- + \epsilon^+}, \quad \text{if } \epsilon^- + \epsilon^+ > 0, \\
\text{or } w_e^- &= w_e^+ = \tfrac{1}{2}, \quad \text{if } \epsilon^- = \epsilon^+ = 0.
\end{aligned} \tag{7}
$$

Replacing (7) into $\{\epsilon\}_w$, we observe that it is equivalent to the harmonic mean of the coefficient ϵ across the edges. In what follows, we will see how this is related to the behavior of the method. For any admissible value of $\chi_e(\epsilon)$ and $\varphi_e(\epsilon)$ we observe that method (6) is by construction consistent with respect to the weak formulation of problem (1). Now, the definition of $\chi_e(\epsilon)$ and $\varphi_e(\epsilon)$ has to be made precise in order to enforce that $a_h(\cdot, \cdot)$ is coercive in the following norm,

$$
\begin{aligned}
|||v_h|||^2 := &\|\epsilon^{\frac{1}{2}} \nabla v_h\|_{0,\mathcal{T}_h}^2 + \|\mu_0^{\frac{1}{2}} v_h\|_{0,\mathcal{T}_h}^2 \\
&+ \|(\tfrac{1}{2}|\beta \cdot n_e| + \tfrac{1}{2}\{\epsilon\}_w h_e^{-1})^{\frac{1}{2}} [\![v_h]\!]\|_{0,\Gamma_e}^2 + \|(\tfrac{1}{2}|\beta \cdot n| + \epsilon h_e^{-1})^{\frac{1}{2}} v_h\|_{0,\Gamma_{\partial\Omega}}^2,
\end{aligned}
$$

where we have introduced the notation $\|v_h\|_{0,\mathcal{T}_h}^2 := \sum_{K \in \mathcal{T}_h} \|v_h\|_{0,K}^2$, and $\|v_h\|_{0,\Gamma_e}^2 := \sum_{e \in \Gamma_e} \|v_h\|_{0,e}^2$, being $\|\cdot\|_{0,K}$ and $\|\cdot\|_{0,e}$ the L^2-norms on K and e respectively. Then, we consider the bilinear form $a_h^r(\cdot, \cdot)$ that can be manipulated as follows,

$$
\begin{aligned}
a_h^r(u_h, u_h) &= \sum_{e \in \Gamma_e} \int_e \Big[\beta \cdot n_e \{u_h\}_w [\![u_h]\!] - \beta \cdot n_e \{u_h\} [\![u_h]\!] - \tfrac{1}{2} \beta \cdot n_e \varphi_e(\epsilon) [\![u_h]\!]^2 \Big] \\
&= \tfrac{1}{2} \sum_{e \in \Gamma_e} \int_e (2w_e^- - \varphi_e(\epsilon) - 1) \beta \cdot n_e [\![u_h]\!]^2 = 0,
\end{aligned} \tag{8}
$$

provided that we set $\varphi_e(\epsilon) := (2w_e^- - 1)$ or equivalently, owing to (7),

$$
\varphi_e(\epsilon) = \frac{2\epsilon^+}{\epsilon^+ + \epsilon^-} - 1 = -\frac{[\![\epsilon]\!]}{2\{\epsilon\}}, \quad \text{if } \{\epsilon\} > 0, \tag{9}
$$

and $\varphi_e(\epsilon) = 0$ if $\{\epsilon\} = 0$. Definition (9) satisfies $|\varphi_e(\epsilon)| \leq 1$ and the expression $\frac{1}{2}\big(|\beta \cdot n_e| - \varphi_e(\epsilon) \beta \cdot n_e\big)$ represents a natural generalization of the standard upwind scheme. As a consequence of (8), the coercivity only depends on the properties of $a_h^s(\cdot, \cdot)$. First of all, it is straightforward to verify that,

$$\begin{aligned}
&\sum_{K\in T_h}\int_K \Big[\epsilon(\nabla u_h)^2 + \big(\mu + \tfrac{1}{2}\nabla\cdot\beta\big)u_h^2\Big] \\
&+ \sum_{e\in\Gamma_e}\int_e \big(\tfrac{1}{2}|\beta\cdot n_e| + \chi_e(\epsilon)h_e^{-1}\big)[\![u_h]\!]^2 + \sum_{e\in\Gamma_{\partial\Omega}}\int_e \big(\tfrac{1}{2}|\beta\cdot n| + \chi_{\partial\Omega}(\epsilon)h_e^{-1}\big)u_h^2 \\
&\geq \|\epsilon^{\frac{1}{2}}\nabla u_h\|_{0,T_h}^2 + \|\mu_0^{\frac{1}{2}}u_h\|_{0,T_h}^2 + \|\big(\tfrac{1}{2}|\beta\cdot n_e| + \chi_e(\epsilon)h_e^{-1}\big)^{\frac{1}{2}}[\![u_h]\!]\|_{0,\Gamma_e}^2 \\
&+ \|\big(\tfrac{1}{2}|\beta\cdot n| + \chi_{\partial\Omega}(\epsilon)h_e^{-1}\big)^{\frac{1}{2}}u_h\|_{0,\Gamma_{\partial\Omega}}^2. \qquad (10)
\end{aligned}$$

To treat the remaining terms of $a_h^s(u_h, u_h)$, as usual for DG methods, we make use of the following trace/inverse inequality,

$$h_e\|\nabla v_h\cdot n_e\|_{0,e}^2 \leq C_I\|\nabla v_h\|_{0,K}^2, \ \forall K\in T_h \text{ and } \forall e\in\partial K,$$

where $C_I > 0$ does not depend on h_e. Then, we obtain the following bounds,

$$\begin{aligned}
&2\sum_{e\in\Gamma_e}\int_e \{\epsilon\nabla u_h\}_w\cdot n_e[\![u_h]\!] + 2\sum_{e\in\Gamma_{\partial\Omega}}\int_e \epsilon\nabla u_h\cdot n u_h \\
=&2\sum_{e\in\Gamma_e}\int_e \omega_e(\nabla u_h^- + \nabla u_h^+)\cdot n_e[\![u_h]\!] + 2\sum_{e\in\Gamma_{\partial\Omega}}\int_e \epsilon\nabla u_h\cdot n u_h \\
\leq&\sum_{e\in\Gamma_e}\Big[\alpha h_e\Big(\|(\epsilon^-)^{\frac{1}{2}}\nabla u_h^-\cdot n_e\|_{0,e}^2 + \|(\epsilon^+)^{\frac{1}{2}}\nabla u_h^+\cdot n_e\|_{0,e}^2\Big) + \frac{1}{\alpha h_e}\|\omega_e^{\frac{1}{2}}[\![u_h]\!]\|_{0,e}^2\Big] \\
&+ \sum_{e\in\Gamma_{\partial\Omega}}\Big[\alpha h_e\|\epsilon^{\frac{1}{2}}\nabla u_h\cdot n\|_{0,e}^2 + \frac{1}{\alpha h_e}\|\epsilon^{\frac{1}{2}}u_h\|_{0,e}^2\Big] \\
\leq&6\alpha C_I\|\epsilon^{\frac{1}{2}}\nabla u_h\|_{0,T_h}^2 + \frac{1}{\alpha}\|(\tfrac{1}{2}\{\epsilon\}_w)^{\frac{1}{2}}h_e^{-\frac{1}{2}}[\![u_h]\!]\|_{0,\Gamma_e}^2 + \frac{1}{\alpha}\|\epsilon^{\frac{1}{2}}h_e^{-\frac{1}{2}}u_h\|_{0,\Gamma_{\partial\Omega}}^2. \qquad (11)
\end{aligned}$$

The coercivity of $a_h(\cdot,\cdot)$ in the norm $|||\cdot|||$ directly follows from the combination of (8), (10) and (11) provided α is such that $6\alpha C_I < 1$ and,

$$\chi_e(\epsilon) := \tfrac{1}{2}\zeta\{\epsilon\}_w, \quad \chi_{\partial\Omega}(\epsilon) := \zeta\epsilon, \qquad (12)$$

where ζ is a suitable constant such that $\zeta > \frac{1}{\alpha}$. Due to (9) and (12) the method (6) is completely determined.

By virtue of the second Strang lemma and owing to the continuity (not addressed here), the consistency and the coercivity of the bilinear form $a_h(\cdot,\cdot)$, it is possible to prove optimal a-priori error estimates in the norm $|||\cdot|||$ for problem (6). This analysis has been fully addressed in [3] in the case of a similar method applied to anisotropic diffusivity.

3 Numerical Results and Conclusions

In order to pursue a quantitative comparison between our scheme and the standard interior penalty method, we aim to build up a test problem featuring discontinuous coefficients which allows us to analytically compute the exact solution. To this aim, we consider the following test case, already proposed in [2]. We split the domain Ω into two subregions, $\Omega_1 = (x_0, x_{\frac{1}{2}})\times(y_0, y_1)$, $\Omega_2 = (x_{\frac{1}{2}}, x_1)\times(y_0, y_1)$ and we choose for simplicity $x_0 = 0$, $x_{\frac{1}{2}} = 1$, $x_1 = 2$ while $y_0 = 0$, $y_1 = \frac{1}{2}$. The viscosity $\epsilon(x, y)$ is a discontinuous function across the interface $x = x_{\frac{1}{2}}$, for any $y \in (y_0, y_1)$. Precisely, we consider a constant $\epsilon(x, y)$ in each subregion with several values for ϵ_1 in Ω_1 and a fixed $\epsilon_2 = 1.0$ in Ω_2. In the case $\beta = [1, 0]$, $\mu = 0$, $f = 0$ and the boundary conditions $u_1(x_0, y) = 1$, $u_2(x_1, y) = 0$ for simplicity, the exact solution of the problem on each subregion Ω_1, Ω_2 can be expressed as an exponential function with respect to x independently from y. The global solution $u(x, y)$ is then provided by choosing the value at the interface, $u(x_{\frac{1}{2}}, y)$, in order to ensure the continuity of both $u(x, y)$ and the normal fluxes with respect to the interface, namely $-\epsilon(x, y)\partial_x u(x, y)$. For the corresponding explicit expressions of $u\Big(x_{\frac{1}{2}}, y\Big)$ and $u_1(x, y)$, $u_2(x, y)$, we remand to [2].

In the following numerical simulations, our reference standard interior penalty method (IP) is obtained by replacing the weights $w_e^- = w_e^+ = \frac{1}{2}$ into (6). To compare the method proposed here (WIP) with IP we consider a uniform triangulation T_h with $h = 0.05$ and we apply piecewise linear elements. We perform a quantitative comparison based on the energy norm of the error $|||u - u_h|||$ and on the following indicator, $\Delta_{extr} := \max(|\max_\Omega(u_h) - \max_\Omega(u)|, |\min_\Omega(u_h) - \min_\Omega(u)|)$ which quantifies to which extent the numerical solution exceeds the extrema of the exact one. The results reported in table 1 and in figure 1 put into evidence that the WIP scheme performs better than the standard IP method, particularly in those cases where the solution is non smooth and at the same time the computational mesh is not completely adequate to capture the singularities. This happens in particular for the smallest value of ϵ_1, precisely $\epsilon_1 = 5\ 10^{-3}$, while in the other cases the two methods are equivalent. In the case $\epsilon_1 = 5\ 10^{-3}$ the weighted interior penalties turn out to be very effective, since they allow the scheme to approximate the very steep boundary layer at the interface $x = x_{\frac{1}{2}}$ with a jump. Conversely, the standard interior penalty scheme computes a solution that is almost continuous. As can be observed in figure 1, this behavior promotes the instability of the approximate solution in the neighborhood of the boundary layer, because the computational mesh is not adequate to smoothly approximate the very high gradients across the interface. The quantity Δ_{extr} shows that the the spurious oscillations generated in this case reach the 40% of the maximum of the exact solution. The different behavior of the two methods can also be interpreted observing that, disregarding the advective terms, in the case of the standard IP scheme the satisfaction of the inter-element continuity is proportional to $\{\epsilon\}$, as the neighboring elements of each edge were ideally connected by two adjacent springs of stiffness ϵ^- and ϵ^+. Conversely, in the WIP case the mortar between elements is proportional to $\{\epsilon\}_w$, which is the harmonic mean of the values ϵ^-, ϵ^+ and corresponds to the stiffness of two sequential springs of stiffness ϵ^- and ϵ^+ respectively. The latter case seems to be more natural for problems with discontinuous coefficients.

References

[1] D.N. Arnold, F. Brezzi, B. Cockburn, and L.D. Marini. Unified analysis of discontinuous Galerkin methods for elliptic problems. *SIAM J. Numer. Anal.*, 39(5):1749–1779, 2001/02.

Table 1. Quantitative comparison between WIP and a standard IP method.

	$\|\|\|u - u_h\|\|\|$			Δ_{extr}		
ϵ_1	$5\ 10^{-1}$	$5\ 10^{-2}$	$5\ 10^{-3}$	$5\ 10^{-1}$	$5\ 10^{-2}$	$5\ 10^{-3}$
WIP	8.151e-03	5.629e-02	1.858e-01	1.069e-04	1.016e-04	7.302e-02
IP	8.137e-03	5.779e-02	3.208e-01	1.069e-04	1.016e-04	4.412e-01

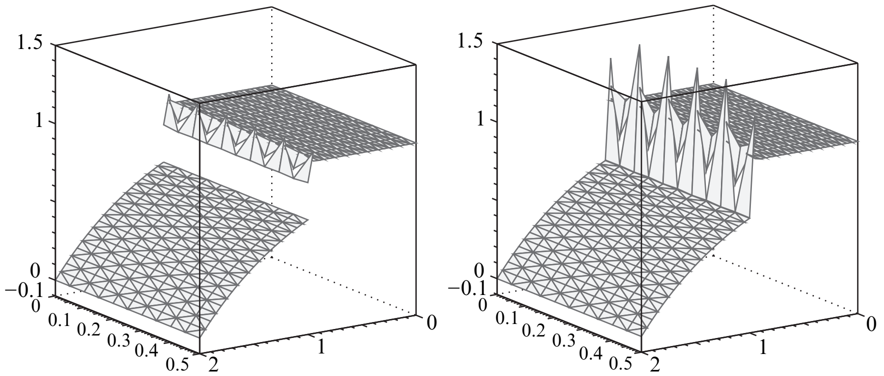


Fig. 1. The solutions computed by WIP (left) and IP (right) for $\epsilon_1 = 5\ 10^{-3}$.

[2] E. Burman and P. Zunino. A domain decomposition method based on weighted interior penalties for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 44(4):1612–1638, 2006.

[3] A. Ern, A. Stephansen, and P. Zunino. A Discontinuous Galerkin method with weighted averages for advection-diffusion equations with locally vanishing and anisotropic diffusivity. Technical Report 103, MOX, Department of Mathematics, Politecnico di Milano, 2007. Submitted.

[4] F. Gastaldi and A. Quarteroni. On the coupling of hyperbolic and parabolic systems: analytical and numerical approach. *Appl. Numer. Math.*, 6(1-2):3–31, 1989/90. Spectral multi-domain methods (Paris, 1988).

[5] B. Heinrich and K. Pönitz. Nitsche type mortaring for singularly perturbed reaction-diffusion problems. *Computing*, 75(4):257–279, 2005.

[6] P. Houston, Ch. Schwab, and E. Süli. Discontinuous *hp*-finite element methods for advection-diffusion-reaction problems. *SIAM J. Numer. Anal.*, 39(6):2133–2163, 2002.

[7] J. Nitsche. On Dirichlet problems using subspaces with nearly zero boundary conditions. In *The mathematical foundations of the finite element method with applications to partial differential equations (Proc. Sympos., Univ. Maryland, Baltimore, Md., 1972)*, pages 603–627. Academic Press, New York, 1972.

[8] R. Stenberg. Mortaring by a method of J. A. Nitsche. In *Computational mechanics (Buenos Aires, 1998)*. Centro Internac. Métodos Numér. Ing., Barcelona, 1998.

MINISYMPOSIUM 7: Optimized Schwarz Methods: Promises and Challenges

Lahcen Laayouni

School of Science and Engineering, Al Akhawayn University, B.P. 1890, Avenue Hassan II, Ifrane 53000- Morocco. L.Laayouni@aui.ma

In the last two decades, many investigations have been devoted to improve the performance of the classical Schwarz methods. Optimized Schwarz methods (OSM) are one of the competitive candidates among other modern domain decomposition methods. OSM had significantly enhance the performance of the classical Schwarz methods. Those improvements are based essentially on using different type of transmission conditions between subdomains. The key idea is to exchange more information between subdomains which corresponds to communicate solutions and theirs derivatives instead of exchanging solutions only. Rigorous Fourier analysis for different decompositions and different differential equations had shown the efficiency and robustness of OSM as promised. Now, the big challenges are to extend the performances of optimized Schwarz methods to consider high dimensional differential equations and systems of PDE's with complicated geometries. In this minisymposium we give some answers to different aspects of those challenges.

A New Domain Decomposition Method for the Compressible Euler Equations Using Smith Factorization

Victorita Dolean[1] and Frédéric Nataf[2]

[1] Univ. de Nice Sophia-Antipolis, Laboratoire J.-A. Dieudonné, Nice, France. dolean@math.unice.fr
[2] Université Paris VI, Laboratoire Jacques-Louis Lions, Paris, France, nataf@ann.jussieu.fr

Summary. In this work we design a new domain decomposition method for the Euler equations in 2 dimensions. The starting point is the equivalence with a third order scalar equation to whom we can apply an algorithm inspired from the Robin-Robin preconditioner for the convection-diffusion equation [1]. Afterwards we translate it into an algorithm for the initial system and prove that at the continuous level and for a decomposition into 2 sub-domains, it converges in 2 iterations. This property cannot be conserved strictly at discrete level and for arbitrary domain decompositions but we still have numerical results which confirm a very good stability with respect to the various parameters of the problem (mesh size, Mach number, ...).

1 Introduction

The need of using domain decomposition methods when solving partial differential equations is nowadays more and more obvious. The challenge is now the acceleration of these methods. Different possibilities were studied such as the use of optimized interface conditions on the artificial boundaries between subdomains or the preconditioning of a substructured system defined at the interface. The former were widely studied and analyzed for scalar problems. The preconditioning methods have also known a wide development in the last decade. The Neumann-Neumann algorithms for symmetric second order problems have been the subject of numerous works. An extension of these algorithms to non-symmetric scalar problems (the so called Robin-Robin algorithms) has been done in [1] for advection-diffusion problems. As far as optimized interface conditions are concerned, when dealing with supersonic flows, whatever the space dimension is, imposing the appropriate characteristic variables as interface conditions leads to a convergence of the algorithm which is optimal with regards to the number of subdomains. This property is generally lost for subsonic flows except for the case of one-dimensional problems, when the optimality is expressed by the fact that the number of iterations is equal to the number of subdomains (see [2] and [7] for more details). In the subsonic case and in two or three dimensions, we can find a formulation with classical (natural) transmission conditions in [7, 8] or with more general interface conditions in [3] and optimized transmission conditions in [5]. The analysis of such algorithms applied to systems proved to be very different from the scalar case, see [4]. The generalization of the above domain decomposition methods to the system of the Euler equations is difficult in the subsonic case in dimensions equal or higher to two.

In this work, we consider a preconditioning technique for the system of the compressible Euler equations in the subsonic case. The paper is organized as follows: in Section 2 we will first show the equivalence between the 2D Euler equations and a third order scalar problem, which is quite natural by considering a Smith factorization of this system, see [9]. In Section 3 we define an optimal algorithm for the third order scalar equation. It is inspired from the idea of the Robin-Robin algorithm [1] applied to a convection-diffusion problem. Afterwards in Section 4 we back-transform it and define the corresponding algorithm applied to the Euler system. All the previous results have been obtained at the continuous level and for a decomposition into 2 unbounded subdomains. In the Section 5, numerical results confirm the very good stability of the algorithm with respect to the various parameters of the problem (mesh size, Mach number, ...).

2 A Third Order Scalar Problem

In this section we will show the equivalence between the linearized and time discretized Euler system and a third order scalar equation. The motivation for this transformation is that a new algorithm is easier to design for a scalar equation than for a system of partial differential equations. The starting point of our analysis is given by the linearized form of the Euler equations written in primitive variables (p, u, v, S). In the following we suppose that the flow is isentropic, which allows us to drop the equation of the entropy (which is totally decoupled from the others). We denote by $W = (P, U, V)^T$ the vector of unknowns and by A and B the jacobian matrices of the fluxes $F_i(w)$ to whom we already applied the variable change from conservative to primitive variables. In the following, we shall denote by $\bar{c}$ the speed of the sound and we consider the linearized form (we will mark by the bar symbol, the state around which we linearize) of the Euler equations:

$$\mathcal{P}W \equiv (\beta I + A\partial_x + B\partial_y)\, W = f \tag{1}$$

where $\beta = \frac{1}{\Delta t} > 0$, characterized by the following jacobian matrices:

$$A = \begin{pmatrix} \bar{u} & \bar{\rho}\bar{c}^2 & 0 \\ 1/\bar{\rho} & \bar{u} & 0 \\ 0 & 0 & \bar{u} \end{pmatrix} \qquad B = \begin{pmatrix} \bar{v} & 0 & \bar{\rho}\bar{c}^2 \\ 0 & \bar{v} & 0 \\ 1/\bar{\rho} & 0 & \bar{v} \end{pmatrix}. \tag{2}$$

In Computational Fluid Dynamics, problems of the form (1) have to be solved repeatedly. We shall design a new domain decomposition method for this purpose. We build and analyze our method for the constant coefficient case ($\bar{c}$, $\bar{u}$, $\bar{v}$ and $\bar{\rho}$ are constants) and for only two subdomains. But the resulting algorithm can be applied to the general case.

We first recall the Smith factorization of a matrix with polynomial entries ([9], Theorem 1.4):

Theorem 1. *Let n be an positive integer and A a $n \times n$ matrix with polynomial entries with respect to the variable λ: $A = (a_{ij}(\lambda))_{1\leq i,j\leq n}$. Then, there exist matrices E, D and F with polynomial entries satisfying the following properties:*

- *$det(E), det(F)$ are constants,*
- *D is a diagonal matrix uniquely determined up to a multiplicative constant,*
- *$A = EDF$.*

We first take formally the Fourier transform of the system (1) with respect to y (the dual variable is ξ). We keep the partial derivatives in x since in the sequel we shall consider a domain decomposition with an interface whose normal is in the x direction. We note

$$\hat{\mathcal{P}} = \begin{pmatrix} \beta + \bar{u}\partial_x + i\xi\mathbf{b} & \bar{\rho}\bar{c}^2\partial_x & i\bar{\rho}\bar{c}^2\xi \\ \frac{1}{\bar{\rho}}\partial_x & \beta + \bar{u}\partial_x + i\xi\mathbf{b} & 0 \\ \frac{i\xi}{\bar{\rho}} & 0 & \beta + \bar{u}\partial_x + i\mathbf{b}\xi \end{pmatrix} \tag{3}$$

We can perform a Smith factorization of $\hat{\mathcal{P}}$ by considering it as a matrix with polynomials in ∂_x entries. We have $\hat{\mathcal{P}} = EDF$ where

$$D = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \hat{\mathcal{L}}\hat{\mathcal{G}} \end{pmatrix}, \; E = \begin{pmatrix} i\bar{\rho}\bar{c}^2\xi & 0 & 0 \\ 0 & \bar{u} & 0 \\ \beta + \bar{u}\partial_x + i\mathbf{b}\xi & E_2 & \dfrac{\bar{c}^2 - \bar{u}^2}{i\xi\bar{\rho}\bar{c}^2} \end{pmatrix} \tag{4}$$

and

$$F = \begin{pmatrix} \dfrac{\beta + \bar{u}\partial_x + i\xi\mathbf{b}}{i\xi\bar{\rho}\bar{c}^2} & \dfrac{\partial_x}{i\xi} & 1 \\ \dfrac{\partial_x}{\bar{\rho}\bar{u}} & \dfrac{\beta + \bar{u}\partial_x + i\xi\mathbf{b}}{\bar{u}} & 0 \\ \dfrac{1}{(\beta + i\xi\mathbf{b})(\bar{u}^2 - \bar{c}^2)} & \dfrac{\bar{\rho}\bar{u}}{(\beta + i\xi\mathbf{b})(\bar{u}^2 - \bar{c}^2)} & 0 \end{pmatrix} \tag{5}$$

where

$$E_2 = \bar{u}\frac{(-\bar{u}\bar{c}^2 + \bar{u}^3)\partial_{xx} + (2\bar{u}^2 - \bar{c}^2)(\beta + i\xi\mathbf{b})\partial_x + \bar{u}((\beta + i\xi\mathbf{b})^2 + \xi^2\bar{c}^2)}{\bar{c}^2(i\beta + i\xi\mathbf{b})},$$

and

$$\begin{aligned} \hat{\mathcal{L}} &= \beta^2 + 2i\xi\bar{u}\mathbf{b}\partial_x + 2\beta(\bar{u}\partial_x + i\xi\mathbf{b}) + (\bar{c}^2 - \mathbf{b}^2)\xi^2 - (\bar{c}^2 - \bar{u}^2)\partial_{xx} \\ \hat{\mathcal{G}} &= \beta + \bar{u}\partial_x + i\xi\mathbf{b} \end{aligned} \tag{6}$$

Equation (4) suggests that the derivation of a domain decomposition method (DDM) for the third order operator $\mathcal{L}\mathcal{G}$ is a key ingredient for a DDM for the compressible Euler equations.

3 A New Algorithm Applied to a Scalar Third Order Problem

In this section we will describe a new algorithm applied to the third order operator found in the section 2. We want to solve $\mathcal{L}\mathcal{G}(Q) = g$ where Q is scalar unknown function and g is a given right hand side. The algorithm will be based on the Robin-Robin algorithm [1] for the convection-diffusion problem. Then we will prove its convergence in 2 iterations. Without loss of generality we assume in the sequel that the flow is subsonic and that $\bar{u} > 0$ and thus we have $0 < \bar{u} < \bar{c}$.

3.1 The Algorithm for a Two-domain Decomposition

We consider now a decomposition of the plane $\mathbb{R}^2$ into two non-overlapping subdomains $\Omega_1 = (-\infty, 0) \times \mathbb{R}$ and $\Omega_2 = (0, \infty) \times \mathbb{R}$. The interface is $\Gamma = \{x = 0\}$. The outward normal to domain Ω_i is denoted $\mathbf{n_i}$, $i = 1, 2$. Let $Q^{i,k}$, $i = 1, 2$ represent the approximation to the solution in subdomain i at the iteration k of the algorithm. We define the following algorithm:

ALGORITHM 1 *We choose the initial values $Q^{1,0}$ and $Q^{2,0}$ such that $\mathcal{G}Q^{1,0} = \mathcal{G}Q^{2,0}$. We compute $(Q^{i,k+1})_{i=1,2}$ from $(Q^{i,k})_{i=1,2}$ by the following iterative procedure:*
Correction step *We compute the corrections $\tilde{Q}^{1,k}$ and $\tilde{Q}^{2,k}$ as solution of the homogeneous local problems:*

$$\begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{1,k} = 0 \ \ in\ \Omega_1, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}\tilde{Q}^{1,k} \cdot \mathbf{n_1} = \gamma^k, \ \ on\ \Gamma, \end{cases} \begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{2,k} = 0 \ \ in\ \Omega_2, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}\tilde{Q}^{2,k} \cdot \mathbf{n_2} = \gamma^k, \ \ on\ \Gamma, \\ \tilde{Q}^{2,k} = 0, \ \ on\ \Gamma, \end{cases} \tag{7}$$

where $\gamma^k = -\frac{1}{2}\left[A\nabla\mathcal{G}Q^{1,k} \cdot \mathbf{n_1} + A\nabla\mathcal{G}Q^{2,k} \cdot \mathbf{n_2}\right]$.
Update step. *We update $Q^{1,k+1}$ and $Q^{2,k+1}$ by solving the local problems:*

$$\begin{cases} \mathcal{L}\mathcal{G}Q^{1,k+1} = g, \ \ in\ \Omega_1, \\ \mathcal{G}Q^{1,k+1} = \mathcal{G}Q^{1,k} + \delta^k, \ \ on\ \Gamma, \end{cases} \begin{cases} \mathcal{L}\mathcal{G}\tilde{Q}^{2,k+1} = g, \ \ in\ \Omega_2, \\ \mathcal{G}Q^{2,k+1} = \mathcal{G}Q^{2,k} + \delta^k, \ \ on\ \Gamma, \\ Q^{2,k+1} = Q^{1,k} + \tilde{Q}^{1,k}, \ \ on\ \Gamma, \end{cases} \tag{8}$$

where $\delta^k = \frac{1}{2}\left[\mathcal{G}\tilde{Q}^{1,k} + \mathcal{G}\tilde{Q}^{2,k}\right]$.

Proposition 1. *Algorithm 1 converges in* 2 *iterations.*

See [6] for the details of the proof.

4 A New Algorithm Applied to the Euler System

After having found an optimal algorithm which converges in two iterations for the third order model problem, we focus on the Euler system by translating this algorithm into an algorithm for the Euler system. It suffices to replace the operator $\mathcal{L}\mathcal{G}$ by the Euler system and Q by the last component $F(W)_3$ of $F(W)$ in the boundary conditions. The algorithm reads:

ALGORITHM 2 *We choose the initial values $W^{1,0}$ and $W^{2,0}$ such that $\mathcal{G}F(W^{1,0})_3 = \mathcal{G}F(W^{2,0})_3$ and we compute $(W^{i,k+1})_{i=1,2}$ from $(W^{i,k})_{i=1,2}$ by the following iterative procedure:*
Correction step *We compute the corrections $\tilde{W}^{1,k}$ and $\tilde{W}^{2,k}$ as solution of the homogeneous local problems:*

$$\begin{cases} \mathcal{P}\tilde{W}^{1,k} = 0 \ \ in\ \Omega_1, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}F(\tilde{W}^{1,k})_3 \cdot \mathbf{n_1} = \gamma^k \ \ on\ \Gamma, \end{cases} \begin{cases} \mathcal{P}\tilde{W}^{2,k} = 0 \ \ in\ \Omega_2, \\ (A\nabla - \frac{1}{2}\mathbf{a})\mathcal{G}F(\tilde{W}^{2,k})_3 \cdot \mathbf{n_2} = \gamma^k \ \ on\ \Gamma, \\ \tilde{F}(W^{2,k})_3 = 0, \Gamma, \end{cases} \tag{9}$$

where $\gamma^k = -\frac{1}{2}\left[A\nabla\mathcal{G}F(W^{1,k})_3 \cdot \mathbf{n_1} + A\nabla\mathcal{G}F(W^{2,k})_3 \cdot \mathbf{n_2}\right]$.
Update step. *We update* $W^{1,k+1}$ *and* $W^{2,k+1}$ *by solving the local problems:*

$$\begin{cases} \mathcal{P}W^{1,k+1} = f, \ \mathit{in}\ \Omega_1, \\ \mathcal{G}F(W^{1,k+1})_3 = \mathcal{G}F(W^{1,k})_3 + \delta^k \ \mathit{on}\ \Gamma, \end{cases} \begin{cases} \mathcal{P}\tilde{W}^{2,k+1} &= f, \ \mathit{in}\ \Omega_2, \\ \mathcal{G}F(W^{2,k+1})_3 &= \mathcal{G}F(W^{2,k})_3 + \delta^k \mathit{on}\ \Gamma, \\ F(W^{2,k+1})_3 &= F(W^{1,k})_3 \\ &\quad + F(\tilde{W}^{1,k})_3 \ \mathit{on}\ \Gamma, \end{cases} \tag{10}$$

where $\delta^k = \frac{1}{2}\left[\mathcal{G}F(\tilde{W}^{1,k})_3 + \mathcal{G}F(\tilde{W}^{2,k})_3\right]$.

This algorithm is quite complex since it involves second order derivatives of the unknowns in the boundary conditions on $\mathcal{G}F(W)_3$. It is possible to simplify it. We write it for a decomposition in two subdomains with an outflow velocity at the interface of domain Ω_1 but with an interface not necessarily rectilinear. In this way, it is possible to figure out how to use for a general domain decomposition. In the sequel, $\mathbf{n} = (n_x, n_y)$ denotes the outward normal to domain Ω_1, $\partial_n = \nabla \cdot \mathbf{n} = (\partial_x, \partial_y) \cdot \mathbf{n}$ the normal derivative at the interface, $\partial_\tau = (-\partial_y, \partial_x) \cdot \mathbf{n}$ the tangential derivative, $U_n = Un_x + Vn_y$ and $U_\tau = -Un_y + Vn_x$ are respectively the normal and tangential velocity at the interface between the subdomains. Similarly, we denote $\bar{u}_n$ (resp. $\bar{u}_\tau$) the normal (resp. tangential) component of the velocity around which we have linearized the equations.

ALGORITHM 3 *We choose the initial values* $W^{i,0} = (P^{i,0}, U^{i,0}, V^{i,0})$, $i = 1, 2$ *such that* $P^{1,0} = P^{2,0}$ *and we compute* $W^{i,k+1}$ *from* $W^{i,k}$ *by the iterative procedure with two steps:*
Correction step *We compute the corrections* $\tilde{W}^{1,k}$ *and* $\tilde{W}^{2,k}$ *as solution of the homogeneous local problems:*

$$\begin{cases} \mathcal{P}\tilde{W}^{1,k} = 0, \ \mathit{in}\ \Omega_1, \\ -(\beta + \bar{u}_\tau\partial_\tau)\tilde{U}_n^{1,n} + \bar{u}_n\partial_\tau\tilde{U}_\tau^{1,k} = \gamma^k \ \mathit{on}\ \Gamma, \end{cases} \begin{cases} \mathcal{P}\tilde{W}^{2,k} = 0, \ \mathit{in}\ \Omega_2, \\ (\beta + \bar{u}_\tau\partial_\tau)\tilde{U}_n^{2,k} - \bar{u}_n\partial_\tau\tilde{U}_\tau^{2,k} = \gamma^k, \Gamma \\ \tilde{P}^{2,k} + \bar{\rho}\bar{u}_n\tilde{U}_n^{2,k} = 0 \ \mathit{on}\ \Gamma, \end{cases} \tag{11}$$

where $\gamma^k = -\frac{1}{2}\left[(\beta + \bar{u}_\tau\partial_\tau)(U_n^{2,k} - U_n^{1,k}) + \bar{u}_n\partial_\tau(\tilde{U}_\tau^{1,k} - \tilde{U}_\tau^{2,k})\right]$.
Update step. *We compute the update of the solution* $W^{1,k+1}$ *and* $W^{2,k+1}$ *as solution of the local problems:*

$$\begin{cases} \mathcal{P}W^{1,k+1} = f_1, \ \mathit{in}\ \Omega_1, \\ P^{1,k+1} = P^{1,k} + \delta^k \ \mathit{on}\ \Gamma, \end{cases} \begin{cases} \mathcal{P}W^{2,k+1} &= f_2, \ \mathit{in}\ \Omega_2, \\ P^{2,k+1} = P^{2,k} + \delta^k \ \mathit{on}\ \Gamma, & \\ (P + \bar{\rho}\bar{u}_n U_n)^{2,k+1} &= (P + \bar{\rho}\bar{u}_n U_n)^{1,k} \\ &\quad + (\tilde{P} + \bar{\rho}\bar{u}_n\tilde{U}_n)^{1,k} \ \mathit{on}\ \Gamma, \end{cases} \tag{12}$$

where $\delta^k = \frac{1}{2}\left[\tilde{P}^{1,k} + \tilde{P}^{2,k}\right]$.

Proposition 2. *For a domain* $\Omega = \mathbb{R}^2$ *divided into two non overlapping half planes, algorithms 2 and 3 are equivalent and both converge in two iterations.*

See [6] for the details of the proof.

5 Numerical Results

We present here a set of results of numerical experiments on a model problem. We compare the method proposed and the classical method defined in [4]. We considered a decomposition into different number of subdomains and for a linearization around a constant or non-constant flow. The computational domain is given by the rectangle $[0, 4] \times [0, 1]$ with a uniform discretization using 80×20 points. The numerical investigation is limited to the resolution of the linear system resulting from the first implicit time step using a Courant number CFL=100. In the following, for the new algorithm, each iteration counts for 2 as we need to solve twice as much local problems than the classical one. For an easier comparison of the algorithms, the figures shown in the tables are the number of subdomains solves. We also used substructuring (solving a system with interface variables only) and the iteration number necessary to achieve convergence by means of a GMRES method is also presented. We are solving the homogeneous equations verified by the error vector at the first time step.

The first set of tests concerns a stripwise decomposition into 3 subdomains. The same kind of tests are carried out as in the 2 subdomain case. Table 1 summarizes the number of Schwarz iterations required to reduce the initial linear residual by a factor 10^{-6} for different values of the reference Mach number for the new and the classical algorithm (the tangential velocity is given by the expression $M_t(y) = 0.1(1 + \cos(\pi y))$). For a linearization around a variable state for a general flow at the interface where the tangential Mach number is given by $M_t = 0.1(1 + \cos(\pi y))$, and the initial normal velocity is given by the expression $M_n(y) = 0.5(0.2{+}0.04 \tanh(y/0.2)))$, the same conclusion yield as in the two-domain case. As of intermediate conclusion we can state that the iteration number is only slightly increasing when going from 2 to 3 subdomains.

Table 1. Iteration count for different values of M_n

M_n	Classical (iterative)	Classical (GMRES)	New DDM (iter)	New DDM (GMRES)
0.001	32	26	20	16
0.01	31	26	20	16
0.1	29	21	18	16
0.2	25	19	18	16
0.3	23	16	18	16
0.4	21	15	16	16
0.5	19	13	16	14
0.6	16	12	16	14
0.7	14	11	16	14
0.8	13	11	16	14

The next set of tests concerns a decomposition into 4 subdomains using a 2×2 decomposition of a $40 \times 40 = 1600$ point mesh. No special treatment of the cross points is done or coarse space added. This could be a reason why the iterative version of the algorithm does not converge. Nevertheless, the accelerated algorithm by a GMRES method converges as showed in Table 2 which summarizes the number if iterations for different values of the reference Mach number for both algorithms (the tangential velocity is given by the expression $M_t(y) = 0.1(1 + \cos(\pi y))$ and the normal Mach number is constant at the interface). We can see the the new algorithm behaves similarly as the classical one for low Mach numbers.

Table 2. Iteration count for different values of M_n

M_n	Classical(iter)	Classical (GMRES)	New DDM (GMRES)
0.001	101	28	28
0.01	86	28	28
0.1	54	26	26
0.2	38	23	30
0.3	35	23	32

The latest results show clearly the need of a coarse space as this is done for the FETI-DP methods, in order to improve the performance of the method which has already shown promising results in the case of the stripwise decompositions.

6 Conclusion

In this paper we designed a new domain decomposition for the Euler equations inspired by the idea of the Robin-Robin preconditioner applied to the advection-diffusion equation. We used the same principle after reducing the system to scalar equations via a Smith factorization. The resulting algorithm behaves very well for the low Mach numbers, where usually the classical algorithm does not give very good results. We can reduce the number of iteration by almost a factor 4 both for linearization around a constant and variable state.

References

[1] Y. Achdou, P. Le Tallec, F. Nataf, and M. Vidrascu. A domain decomposition preconditioner for an advection-diffusion problem. *Comput. Methods Appl. Mech. Engrg.*, 184:145–170, 2000.

[2] M. Bjørhus. A note on the convergence of discretized dynamic iteration. *BIT*, 35:291–296, 1995.

[3] S. Clerc. Non-overlapping Schwarz method for systems of first order equations. *Cont. Math*, 218:408–416, 1998.

[4] V. Dolean, S. Lanteri, and F. Nataf. Convergence analysis of a Schwarz type domain decomposition method for the solution of the Euler equations. *Appl. Numer. Math.*, 49:153–186, 2004.

[5] V. Dolean and F. Nataf. An optimized Schwarz algorithm for the compressible Euler equations. Technical Report 556, CMAP - Ecole Polytechnique, 2004.

[6] V. Dolean and F. Nataf. A new domain decomposition method for the compressible Euler equations. *ESAIM-M2AN (Modelisation Mathematique et Analyse Numerique)*, 40:689–703, 2006.

[7] A. Quarteroni. Domain decomposition methods for systems of conservation laws: spectral collocation approximation. *SIAM J. Sci. Stat. Comput.*, 11:1029–1052, 1990.

[8] A. Quarteroni and L. Stolcis. Homogeneous and heterogeneous domain decomposition methods for compressible flow at high Reynolds numbers. Technical Report 33, CRS4, 1996.

[9] J.T. Wloka, B. Rowley, and B. Lawruk. *Boundary Value Problems for Elliptic Systems.* Cambridge University Press, Cambridge, 1995.

Optimized Domain Decomposition Methods for Three-dimensional Partial Differential Equations

Lahcen Laayouni

School of Science and Engineering, Al Akhawayn University, B.P. 1890, Avenue Hassan II, Ifrane 53000- Morocco. L.Laayouni@aui.ma

Summary. Optimized Schwarz methods (OSM) have shown to be an efficient iterative solver and preconditioner in solving partial differential equations. Different investigations have been devoted to study optimized Schwarz methods and many applications have shown their great performance compared to the classical Schwarz methods. By simply making slight modifications of transmission conditions between subdomains, and without changing the size of the matrix, we obtain a fast and a robust family of methods. In this paper we give an extension of optimized Schwarz methods to cover three-dimensional partial differential equations. We present the asymptotic behaviors of optimal and optimized Schwarz methods and compare it to the performance of the classical Schwarz methods. We confirm the obtained theoretical results with numerical experiments.

1 Introduction

The classical Schwarz algorithm has a long history. In 1869, Jacob Schwarz introduced an alternating procedure to prove existence and uniqueness of solutions to Laplace's equation on irregular domains. More than a century later the Schwarz method was used as a computational method in [9]. The advent of computers with parallel architecture give a wide popularity to this method. Recently, [6, 7] gives a mathematical analysis of the Schwarz alternating method at the continuous level and presented different versions of the method, including the extension to many subdomains decomposition. The method was investigated as a preconditioner for discretized problems in [2]. The convergence properties of the classical Schwarz methods are well understood for a wide variety of problems, see e.g., [12, 11]. Recently a new class of Schwarz methods know as optimized Schwarz methods have been introduced to enhance the convergence properties of the classical Schwarz methods. They converge uniformly faster than the classical Schwarz methods due to the exchange of solution and its derivatives between subdomains. Many studies have been devoted to OSM more specifically in $1d$ and $2d$ spaces, see e.g., [5, 3]. A convergence analysis of OSM was done in [4], where a uniform convergence independently of the mesh parameter h has been proved. Those methods have been investigated for problems with discontinuity and anisotropy, see e.g., [8], they were also analyzed for systems of PDE's see [1]. Some industrial applications of OSM in the domain of weather predictions are shown in [10]. For a comparison of OSM with modern DDM like direct Schur methods, FETI and their variants see, e.g. [3, 8]. In this paper we give an extension of OSM to three-dimensional partial differential equations.

2 The Classical Schwarz Method

Throughout this paper we consider the following model problem

$$L(u) = (\eta - \varDelta)(u) = f, \qquad \text{in} \quad \varOmega = \mathbb{R}^3, \quad \eta > 0, \tag{1}$$

where we require the solution to be bounded at the infinity. We decompose $\varOmega$ into $\varOmega_1 = (-\infty, \ell) \times \mathbb{R}^2$, and $\varOmega_2 = (0, \infty) \times \mathbb{R}^2$, where $\ell \geq 0$ is the size of the overlap. The Jacobi Schwarz method on this decomposition is given by

$$\begin{array}{ll} Lu_1^n = f, \text{ in } \varOmega_1, & u_1^n(\ell, y, z) = u_2^{n-1}(\ell, y, z), \\ Lu_2^n = f, \text{ in } \varOmega_2, & u_2^n(0, y, z) = u_1^{n-1}(0, y, z). \end{array} \tag{2}$$

By linearity we consider only the case $f = 0$ and analyze convergence to the zero solution. Taking a Fourier transform of the Schwarz algorithm (2) in y and z directions, we obtain

$$\begin{array}{ll} (\eta + k^2 + m^2 - \partial_{xx})\hat{u}_1^n = 0, \;\; x < \ell, \; k \in \mathbb{R}, \; m \in \mathbb{R}, & \hat{u}_1^n(\ell, k, m) = \hat{u}_2^{n-1}(\ell, k, m), \\ (\eta + k^2 + m^2 - \partial_{xx})\hat{u}_2^n = 0, \;\; x > 0, \; k \in \mathbb{R}, \; m \in \mathbb{R}, & \hat{u}_2^n(0, k, m) = \hat{u}_1^{n-1}(0, k, m), \end{array}$$

where k and m are the frequencies in y and z directions, respectively. Therefore the solutions in the Fourier domain take the form

$$\hat{u}_j^n(x, k, m) = A_j(k, m)e^{\lambda_1(k,m)x} + B_j(k, m)e^{\lambda_2(k,m)x}, \qquad j = 1, 2, \tag{3}$$

where $\lambda_1(k, m) = \kappa$ and $\lambda_2(k, m) = -\kappa$, with $\kappa = \sqrt{\eta + k^2 + m^2}$. Due to the condition on the iterates at the infinity and using transmission conditions, we find that

$$\hat{u}_1^{2n}(0, k, m) = e^{-2\ell\kappa}\hat{u}_1^0(0, k, m) \quad \text{and} \quad \hat{u}_2^{2n}(\ell, k, m) = e^{-2\ell\kappa}\hat{u}_2^0(\ell, k, m). \tag{4}$$

Thus the convergence factor of the classical Schwarz method is given by

$$\rho_{cla} = \rho_{cla}(\eta, k, m, \ell) := e^{-2\ell\kappa} \leq 1, \quad \forall k \in \mathbb{R}, \quad \forall m \in \mathbb{R}. \tag{5}$$

The convergence factor depends on the problem parameter η, the size of the overlap ℓ and on k and m. Figure 1 on the left shows the dependence of the convergence factor on k and m for an overlap $\ell = \frac{1}{100}$ and $\eta = 1$. This shows that the classical Schwarz method damp efficiently high frequencies, whereas for low frequencies the algorithm is very slow.

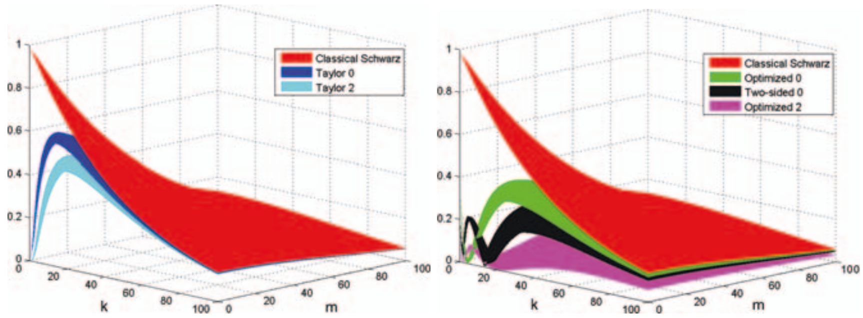


Fig. 1. Left: The convergence factor ρ_{cla} compared to ρ_{T0} and ρ_{T2}. **Right:** The convergence factor ρ_{cla} compared to ρ_{OO0} and ρ_{OO2} and to the convergence factor of two-sided optimized Robin method.

3 The Optimal Schwarz Method

We introduce the following modified algorithm

$$\begin{aligned} L(u_1^n) &= f, \text{ in } \Omega_1, \quad (S_1+\partial_x)(u_1^n)(\ell,.,.) = (S_1+\partial_x)(u_2^{n-1})(\ell,.,.),\\ L(u_2^n) &= f, \text{ in } \Omega_2, \quad (S_2+\partial_x)(u_2^n)(0,.,.) = (S_2+\partial_x)(u_1^{n-1})(0,.,.), \end{aligned} \tag{6}$$

where S_j, $j=1,2$, are linear operators along the interface that depend on y and z. As for the classical Schwarz method it suffices by linearity to consider the case $f=0$. Taking a Fourier transform of the new algorithm (6), we obtain

$$\begin{aligned} (\eta+k^2+m^2-\partial_{xx})\hat{u}_1^n &= 0, \quad x<\ell,\ k\in\mathbb{R},\ m\in\mathbb{R},\\ (\sigma_1(k,m)+\partial_x)(\hat{u}_1^n)(\ell,k,m) &= (\sigma_1(k,m)+\partial_x)(\hat{u}_2^{n-1})(\ell,k,m), \end{aligned} \tag{7}$$

$$\begin{aligned} (\eta+k^2+m^2-\partial_{xx})\hat{u}_2^n &= 0, \quad x>0,\ k\in\mathbb{R},\ m\in\mathbb{R},\\ (\sigma_2(k,m)+\partial_x)(\hat{u}_2^n)(0,k,m) &= (\sigma_2(k,m)+\partial_x)(\hat{u}_1^{n-1})(0,k,m), \end{aligned} \tag{8}$$

where $\sigma_j(k,m)$ is the symbol of the operator $S_j(y,z)$. We proceed as in the case of the classical Schwarz method and using transmission conditions, we obtain

$$\hat{u}_1^{2n}(0,k,m) = \frac{\sigma_1(k,m)-\kappa}{\sigma_1(k,m)+\kappa}.\frac{\sigma_2(k,m)+\kappa}{\sigma_2(k,m)-\kappa}e^{-2\ell\kappa}\hat{u}_1^0(0,k,m). \tag{9}$$

Defining the new convergence factor ρ_{opt} by

$$\rho_{opt} = \rho_{opt}(\eta,k,m,\ell,\sigma_1,\sigma_2) := \frac{\sigma_1(k,m)-\kappa}{\sigma_1(k,m)+\kappa}.\frac{\sigma_2(k,m)+\kappa}{\sigma_2(k,m)-\kappa}e^{-2\ell\kappa}. \tag{10}$$

We compare the convergence factor $\rho_{opt}(\eta,k,m,\ell,\sigma_1,\sigma_2)$ with the one of the classical Schwarz method given in (5), and one can see that they differ only by the factor in front of the exponential term. Choosing for the symbols

$$\sigma_1(k,m) := \kappa \qquad \text{and} \quad \sigma_2(k,m) := -\kappa, \tag{11}$$

the new convergence factor vanishes identically, $\rho_{opt}\equiv 0$, and the algorithm converges in two iterations, independently of the initial guess, the overlap size ℓ and the problem parameter η. This is an optimal result since convergence in less than two iterations is impossible, due to the exchange information necessity between the subdomains. Furthermore, with this choice of σ_j the exponential factor in the convergence factor becomes irrelevant and one can have Schwarz methods without overlap. In practice we need to back transform the transmission conditions with σ_1 and σ_2 from the Fourier domain to the physical domain to obtain S_1 and S_2. The fact that σ_j contain a square-root, the optimal operators S_j are non-local operators. In the next section we will approximate σ_j by polynomials in ik and im, so S_j would consist of derivatives in y and z and thus be local operators.

4 Optimized Schwarz Methods

We approximate the symbols $\sigma_j(k,m)$ found in (11) as follows

$$\sigma_1^{app}(k,m) = p_1 + q_1(k^2+m^2) \quad \text{and} \quad \sigma_2^{app}(k,m) = -p_2 - q_2(k^2+m^2). \tag{12}$$

Hence the convergence factor (10) of the optimized Schwarz methods becomes

$$\rho = \rho(\eta,k,m,\ell,p_1,p_2,q_1,q_2) := \frac{\kappa - p_1 - q_1(k^2+m^2)}{\kappa + p_1 + q_1(k^2+m^2)} \cdot \frac{\kappa - p_2 - q_2(k^2+m^2)}{\kappa + p_2 + q_2(k^2+m^2)} e^{-2\ell\kappa}. \tag{13}$$

Theorem 1. *The optimized Schwarz method (6) with transmission conditions defined by the symbols (12) converges for $p_j > 0$, $q_j \geq 0$, $j = 1,2$, faster than the classical Schwarz method (2), $|\rho| < |\rho_{cla}|$ for all k and m.*

Proof. The absolute value of the term in front of the exponential in the convergence factor (13) of the optimized Schwarz method is strictly smaller than 1 provided $p_j > 0$, and $q_j \geq 0$ which shows that $|\rho| < |\rho_{cla}|$ for all k and m.

Now, we introduce a low frequency approximations using a Taylor expansions about zero. Expanding the symbols $\sigma_j(k,m)$, $j = 1,2$, we obtain

$$\begin{aligned} \sigma_1(k,m) &= \sqrt{\eta} + \tfrac{1}{2\sqrt{\eta}}(k^2+m^2) + \mathcal{O}_1(k^4,m^4), \\ \sigma_2(k,m) &= -\sqrt{\eta} - \tfrac{1}{2\sqrt{\eta}}(k^2+m^2) + \mathcal{O}_2(k^4,m^4), \end{aligned} \tag{14}$$

where $\mathcal{O}_1(k^4,m^4)$ and $\mathcal{O}_2(k^4,m^4)$ contain high order terms in m and k. The convergence factor ρ_{T0} of the zeroth order Taylor approximation is defined by

$$\rho_{T0}(\eta,k,m,\ell) = \left(\frac{\kappa - \sqrt{\eta}}{\kappa + \sqrt{\eta}}\right)^2 e^{-2\ell\kappa}, \tag{15}$$

and the convergence factor ρ_{T2} of the second order Taylor approximation would have the form

$$\rho_{T2}(\eta,k,m,\ell) = \left(\frac{\kappa - \sqrt{\eta} - \frac{1}{2\sqrt{\eta}}(k^2+m^2)}{\kappa + \sqrt{\eta} + \frac{1}{2\sqrt{\eta}}(k^2+m^2)}\right)^2 e^{-2\ell\kappa}. \tag{16}$$

Figure 1 on the left shows the convergence factors obtained with this choice of transmission conditions compared to the convergence factor ρ_{cla}. One can clearly see that OSM are uniformly better than the classical Schwarz method, in particular the low frequency behavior is greatly improved. Note that OSM converge even without overlap. In particular, we have the following theorem.

Theorem 2. *The optimized Schwarz methods with Taylor transmission conditions and overlap ℓ have an asymptotically superior performance than the classical Schwarz method with the same overlap. As ℓ goes to zero, we have*

$$\max_{|k|\leq\frac{\pi}{\ell},|m|\leq\frac{\pi}{\ell}} |\rho_{cla}(\eta,k,m,\ell)| = 1-2\sqrt{\eta}\ell+\mathcal{O}(\ell^2),$$

$$\max_{|k|\leq\frac{\pi}{\ell},|m|\leq\frac{\pi}{\ell}} |\rho_{T0}(\eta,k,m,\ell)| = 1-4\sqrt{2}\eta^{1/4}\sqrt{\ell}+\mathcal{O}(\ell),$$

$$\max_{|k|\leq\frac{\pi}{\ell},|m|\leq\frac{\pi}{\ell}} |\rho_{T2}(\eta,k,m,\ell)| = 1-8\eta^{1/4}\sqrt{\ell}+\mathcal{O}(\ell).$$

Without overlap, the optimized Schwarz methods with Taylor transmission conditions are asymptotically comparable to the classical Schwarz method with overlap ℓ. As ℓ goes to zero, we have

$$\max_{|k|\leq\frac{\pi}{\ell},|m|\leq\frac{\pi}{\ell}} |\rho_{T0}(\eta,k,m,0)| = 1-4\frac{\sqrt{\eta}}{\pi}\ell+\mathcal{O}(\ell^2),$$

$$\max_{|k|\leq\frac{\pi}{\ell},|m|\leq\frac{\pi}{\ell}} |\rho_{T2}(\eta,k,m,0)| = 1-8\frac{\sqrt{\eta}}{\pi}\ell+\mathcal{O}(\ell^2).$$

Proof. The proof is based on a Taylor expansion of the convergence factors, where we estimate the maximum frequency by π/ℓ.

Zeroth Order Optimized Transmission Conditions

Using the same zeroth order transmission conditions on both sides of the interface, $p_1 = p_2 = p$ and $q_1 = q_2 = 0$, the convergence factor in (13) becomes

$$\rho_{OO0}(\eta,k,m,\ell,p) := \left(\frac{\kappa-p}{\kappa+p}\right)^2 e^{-2\kappa\ell}. \tag{17}$$

To find the optimal parameter p^* of the associated Schwarz method, known as Optimized of Order 0 ($OO0$), we need to solve the following min-max problem

$$\min_{p\geq 0}(\max_{k,m}|\rho_{OO0}(\eta,k,m,\ell,p)|) = \min_{p\geq 0}\left(\max_{k,m}\left(\frac{\kappa-p}{\kappa+p}\right)^2 e^{-2\kappa\ell}\right). \tag{18}$$

We introduce the minimum and the maximum frequencies $f_{\min}$ and $f_{\max}$ of all the frequencies k and m. The asymptotic performance of the Optimized zeroth order Schwarz method is given by the next theorem, where we omit the proof due to the restriction on the present paper.

Theorem 3. *(Robin asymptotic)*
The asymptotic performance of the Schwarz method with optimized Robin transmission conditions and overlap ℓ, as ℓ goes to zero, is given by

$$\max_{\substack{k,m \\ f_{\min}\leq\sqrt{k^2+m^2}\leq\frac{\pi}{\ell}}} |\rho_{OO0}(\eta,k,m,\ell,p^*)| = 1-4.2^{1/6}(f_{\min}^2+\eta)^{1/6}\ell^{1/3}+\mathcal{O}(\ell^{2/3}). \tag{19}$$

The asymptotic performance of $OO0$ without overlap is asymptotically equivalent to the classical Schwarz method with overlap ℓ, as ℓ goes to zero, we have

$$\max_{\substack{k,m \\ f_{\min}\leq\sqrt{k^2+m^2}\leq\frac{\pi}{\ell}}} |\rho_{OO0}(\eta,k,m,0,p^*)| = 1-4\frac{(f_{\min}^2+\eta)^{1/4}}{\sqrt{\pi}}\sqrt{\ell}+\mathcal{O}(\ell). \tag{20}$$

Proof. The idea of the proof in the case of overlapping subdomains is based on the ansatz $p^* = C\ell^{\alpha}$, where $\alpha < 0$ and Taylor expansion of the convergence factor with $p = p^*$. A computation shows that $p^* = \frac{(4(f_{min}^2+\eta))^{1/3}}{2}\ell^{-1/3}$.

Second Order Optimized Transmission Conditions

Using the same second order transmission conditions on both sides of the interface, $p_1 = p_2 = p$ and $q_1 = q_2 = q$, the expression (13) of the convergence factor simplifies to

$$\rho_{OO2}(\eta,k,m,\ell,p,q) = \left(\frac{\kappa-p-q(k^2+m^2)}{\kappa+p+q(k^2+m^2)}\right)^2 e^{-2\kappa\ell}. \tag{21}$$

To determine the optimal parameters p^* and q^* for OSM of Order 2 ($OO2$), we need to solve the min-max problem

$$\min_{p,q\geq 0}\left(\max_{k,m}|\rho_{OO2}(\eta,k,m,\ell,p,q)|\right) = \min_{p,q\geq 0}\left(\max_{k,m}\left(\frac{\kappa-p-q(k^2+m^2)}{\kappa+p+q(k^2+m^2)}\right)^2 e^{-2\kappa\ell}\right). \tag{22}$$

We have the following.

Theorem 4. *(Second order)*
The asymptotic performance of the Schwarz method with optimized second order transmission conditions and overlap ℓ, as ℓ goes to zero, is given by

$$\max_{\substack{k,m \\ f_{\min}\leq\sqrt{k^2+m^2}\leq f_{\max}}} |\rho_{OO2}(\eta,k,m,\ell,p^*,q^*)| = 1-4.2^{3/5}(f_{\min}^2+\eta)^{1/10}\ell^{1/5}+\mathcal{O}(\ell^{2/5}). \tag{23}$$

The asymptotic performance of $OO2$ without overlap is equivalent to the classical Schwarz with overlap ℓ. As ℓ approaches zero, we obtain

$$\max_{\substack{k,m \\ f_{\min}\leq\sqrt{k^2+m^2}\leq f_{\max}}} |\rho_{OO2}(\eta,k,m,0,p^*,q^*)| = 1-4\frac{\sqrt{2}(f_{\min}^2+\eta)^{1/8}}{\pi^{1/4}}\ell^{1/4}+\mathcal{O}(\ell^{1/2}). \tag{24}$$

Proof. We do a Taylor expansion of the convergence factor with $p^* = C_1\ell^{\alpha}$ and $q^* = C_2\ell^{\beta}$, where $\alpha < 0$ and $\beta > 0$, we show that $p^* = 2^{-3/5}(f_{min}^2+\eta)^{2/5}\ell^{-1/5}$ and $q^* = 2^{-1/5}(f_{min}^2+\eta)^{-1/5}\ell^{3/5}$.

Figure 1 on the right shows a comparison of the convergence factors of the optimized Schwarz methods with the classical Schwarz method. We also compare the convergence factor of the classical Schwarz method with the convergence factor of the two-sided optimized Schwarz method, where we use different Robin transmission conditions between the two subdomains. As one can see the optimized Schwarz methods have a great performance compared to the classical Schwarz method.

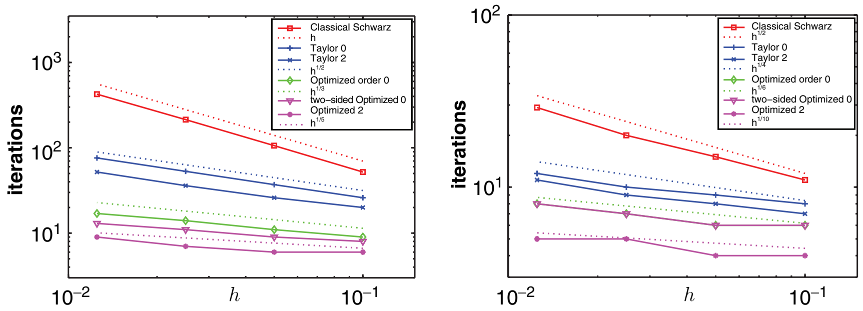


Fig. 2. Number of iterations required by the classical and the optimized Schwarz methods, with overlap $\ell = h$. On the left the methods are used as iterative solvers, and on the right as preconditioners for a Krylov method.

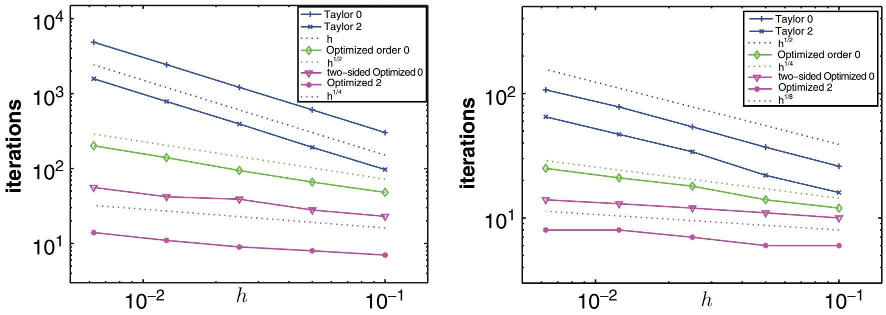


Fig. 3. Number of iterations required by the optimized Schwarz methods without overlap between subdomains. On the left the methods are used as iterative solvers, and on the right as preconditioners for a Krylov method.

5 Numerical Experiments

We perform numerical experiments for our model problem (1) on the unit cube, $\Omega = (0, 1)^3$. We decompose the unit cube Ω into two subdomains $\Omega_1 = (0, b) \times (0, 1)^2$ and $\Omega_2 = (a, 1) \times (0, 1)^2$, where $0 < a \leq b < 1$, so that the overlap is $\ell = b - a$. We use a finite difference discretization with the classical seven-point discretization and a uniform mesh parameter h. In practice, we usually use a small overlap between subdomains, in our experiments we chose the overlap ℓ to be exactly the mesh parameter h, i.e., $\ell = h$. Figure 2 on the left shows the number of iterations versus the mesh parameter h in the case of an overlap, for all the methods used as an iterative solvers, on the right the methods are used as preconditioners for a Krylov method. In figure 3 we show the number of iterations in the case of non-overlapping subdomains. On the left the methods are used as iterative solvers, whilst on the right the methods are used as preconditioners for a Krylov method. For both decompositions the numerical results show the asymptotic behavior predicted by the analysis.

6 Conclusion

In this paper we presented an extension of the optimal and optimized Schwarz methods to cover three-dimensional partial differential equations. We showed the impact of transmission conditions on the convergence factor of Classical Schwarz method. We also showed theoretically and numerically that the optimized Schwarz methods are fast and have a great improved performance compared to the classical Schwarz method.

Acknowledgement. The author acknowledges the support of the Canadian Foundation for Climate and Atmospheric Sciences (CFCAS). The authors would like to thank the referee for the valuable comments to improve the present paper.

References

[1] V. Dolean and F. Nataf. A new domain decomposition method for the compressible Euler equations. *M2AN Math. Model. Numer. Anal.*, 40(4):689–703, 2006.

[2] M. Dryja and O.B. Widlund. Some domain decomposition algorithms for elliptic problems. In *Iterative Methods for Large Linear Systems (Austin, TX, 1988)*, pages 273–291. Academic Press, Boston, MA, 1990.

[3] M.J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.

[4] M.J. Gander and G.H. Golub. A non-overlapping optimized Schwarz method which converges with arbitrarily weak dependence on *h*. In *Domain Decomposition Methods in Science and Engineering*, pages 281–288. Natl. Auton. Univ. Mex., México, 2003.

[5] M.J. Gander, L. Halpern, and F. Nataf. Optimized Schwarz methods. In *Domain Decomposition Methods in Sciences and Engineering (Chiba, 1999)*, pages 15–27. DDM.org, Augsburg, 2001.

[6] P.-L. Lions. On the Schwarz alternating method. I. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987)*, pages 1–42. SIAM, Philadelphia, PA, 1988.

[7] P.-L. Lions. On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 202–223. SIAM, Philadelphia, PA, 1990.

[8] Y. Maday and F. Magoulès. Optimized Schwarz methods without overlap for highly heterogeneous media. *Comput. Methods Appl. Mech. Engrg.*, 196(8):1541–1553, 2007.

[9] K. Miller. Numerical analogs to the Schwarz alternating procedure. *Numer. Math.*, 7:91–103, 1965.

[10] A.L. Qaddouri, L. Loisel, J. S. Côté, and M.J. Gander. Optimized Schwarz methods with an overset grid system for the shallow-water equations. *Appl. Numer. Math.*, 2007. In press.

[11] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* The Clarendon Press Oxford University Press, New York, 1999.

[12] B.F. Smith, P.E. Bjørstad, and W.D. Gropp. *Domain Decomposition.* Cambridge University Press, Cambridge, 1996.

Optimized Schwarz Methods with the Yin-Yang Grid for Shallow Water Equations

Abdessamad Qaddouri

Recherche en prévision numérique, Atmospheric Science and Technology
Directorate, Environment Canada, Dorval, Québec, Canada.
abdessamad.qaddouri@ec.gc.ca

Summary. An efficient implementation of the Schwarz method is used to solve the 2D linear system of shallow-water equations (SWEs) on the sphere with an overset grid system named Yin-Yang. In this paper the convergence of optimized Schwarz method for solving an elliptic problem is increased by substructuring the algorithm in terms of interface unknowns. In this work we show, by numerical tests, that the use of the Yin-Yang grid avoids the "poles problems" resulting from the global latitude-longitude mesh convergence in the polar regions.

1 Introduction

The Yin-Yang overset grid (see Fig. 1) was suggested in [4] as a quasi-uniform grid free from the polar singularity. This grid system is constructed by overlapping two perpendicularly oriented identical parts of latitude-longitude grid. In [5] it was shown that the Schwarz domain decomposition method can be successfully applied to linear SWEs. In each subdomain the same system of equations is discretized by an implicit time method using the same time step where at each time step an elliptic problem is solved. In [5] this method was used to solve the 1D SWEs on the circle and the elliptic problem was solved by the iterative optimized Schwarz method (see [2] and references therein). In this paper we apply the same method to the 2D linear SWEs on the Yin-Yang grid. We show that it is possible to increase the convergence of the optimized Schwarz method for solving the elliptic problem by using the substructuring formulation. Because the two subgrids of the Yin-Yang grid do not match, the variable update at the subdomain interfaces is done by cubic Lagrange interpolation. The complete 2D linear SWE solutions on the Yin-Yang grid are compared to integrations using the spectral model on a Gaussian grid and a finite-difference model on a latitude-longitude grid. The paper is organized as follows: in Section 2, we present the 2D linear SWEs with their discretized form and solution method, in Section 3 the 2D positive definite Helmhotz problem is solved on the Yin-Yang grid, the solver is either an iterative formulation or an substructuring formulation of the optimized Schwarz method, in Section 4 numerical results are shown and finally concluding remarks are given in Section 5.

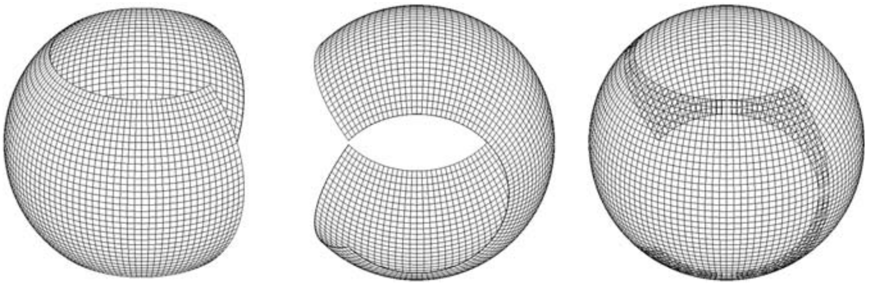


Fig. 1. Yin-Yang grid system. The Yang grid is shown on the left, the Yin grid in the middle, and their composition on the right

2 The 2D Linear Shallow Water Equations

In this section, we develop an implicit global linear shallow-water model on the sphere by using a domain decomposition method on Yin-Yang grid. The major goal of this work is the demonstration that this type of grid system removes the poles problems. Each of two component of Yin-Yang overset grid spans the Subregion S_l, $l = 1, 2$, defined by

$$S_{(l)} = \{(\lambda, \theta); \quad |\theta| \leq \frac{\pi}{4} + \delta \quad |\lambda| \leq \frac{3\pi}{4} + \delta\}, \tag{1}$$

where (λ, θ) are the longitude and the latitude with respect to the local cartesian referential (see [1]) and δ is the minimum overlap. The relationship between the Yin coordinate and Yang coordinate is denoted in cartesian coordinates by

$$(x_{(l)}, y_{(l)}, z_{(l)}) = (-x_{(3-l)}, z_{(3-l)}, y_{(3-l)}) \qquad l = 1, 2. \tag{2}$$

The governing equations for each subdomain l, $l = 1, 2$ are the linear SWEs on a non rotating sphere of radius a

$$\left[\frac{\partial U^{(l)}}{\partial t} + \frac{1}{a^2}\frac{\partial \phi^{(l)}}{\partial \lambda}\right] = 0, \tag{3}$$

$$\left[\frac{\partial V^{(l)}}{\partial t} + \frac{1}{a^2}\cos\theta\frac{\partial \phi^{(l)}}{\partial \theta}\right] = 0, \tag{4}$$

$$\left[\frac{\partial \phi^{(l)}}{\partial t} + \phi^* D^{(l)}\right] = 0, \tag{5}$$

where a is the Earth radius, $(U,\ V)$ are the wind images [wind multiplied by $\frac{\cos\theta}{a}$], ϕ the perturbation geopotential from the reference geopotential ϕ^* and D is the divergence defined by

$$D = \frac{1}{\cos^2\theta}(\frac{\partial U}{\partial \lambda} + \cos\theta\frac{\partial V}{\partial \theta}). \tag{6}$$

The ϕ field gradient components and divergence D give rise to high frequency gravity waves. They are always integrated implicitly in time, enabling the use of a long time step Δt. A time discretization of equations (3)-(5) is

$$\left[\frac{U^{(l)}}{\tau}+\frac{1}{a^2}\frac{\partial\phi^{(l)}}{\partial\lambda}\right](\lambda,\theta,t)=R_U^{(l)}=\left[\frac{U^{(l)}}{\tau}-\frac{1}{a^2}\frac{\partial\phi^{(l)}}{\partial\lambda}\right](\lambda,\theta,t-\Delta t)\,, \tag{7}$$

$$\left[\frac{V^{(l)}}{\tau}+\frac{1}{a^2}\cos\theta\frac{\partial\phi^{(l)}}{\partial\theta}\right](\lambda,\theta,t)=R_V^{(l)}=\left[\frac{V^{(l)}}{\tau}-\frac{1}{a^2}cos\theta\frac{\partial\phi^{(l)}}{\partial\theta}\right](\lambda,\theta,t-\Delta t)\,, \tag{8}$$

$$\left[\frac{\phi^{(l)}}{\tau}+\phi^*D^{(l)}\right](\lambda,\theta,t)=R_{\phi^{(l)}}=\left[\frac{\phi^{(l)}}{\tau}-\phi^*D^{(l)}\right](\lambda,\theta,t-\Delta t)\,, \tag{9}$$

where $\tau=\Delta t/2$. As for the 1D SWEs in [5], the second order finite difference spatial discretization is done on a staggered Arakawa C grid [1] where the variables U, V and ϕ are carried at alternate points in space on (U-grid), ($v-$grid) and (ϕ-grid), respectively. We use (N,M) grid points for the scalar (ϕ-grid), (N-1,M) grid points for (U-grid) and (N,M-1) grid points for (V-grid). The discretized equations on each subdomain l are

$$\left[\frac{U_{i,j}^{(l)}}{\tau}+\frac{1}{a^2}\frac{\phi_{i+1,j}^{(l)}-\phi_{i,j}^{(l)}}{h_\lambda}\right]=R_U^{(l)},\quad i=1,N-1;j=1,M \tag{10}$$

$$\left[\frac{V_{i,j}^{(l)}}{\tau}+\frac{1}{a^2}\cos\theta_j\frac{\phi_{i,j+1}^{(l)}-\phi_{i,j}^{(l)}}{h_\theta}\right]=R_V^{(l)}\quad i=1,N;j=1,M-1 \tag{11}$$

$$\left[\frac{\phi_{i,j}^{(l)}}{\tau}+\phi^*D_{i,j}^{(l)}\right]=R_\phi^{(l)}\quad i=2,N-1;j=2,M-1 \tag{12}$$

with

$$D_{i,j}^{(l)}=\frac{1}{\cos^2\theta_j}\left(\frac{U_{i,j}^{(l)}-U_{i-1,j}^{(l)}}{h_\lambda}+\cos\theta_j\frac{V_{i,j}^{(l)}-V_{i,j-1}^{(l)}}{h_\theta}\right), \tag{13}$$

where h_λ, h_θ are the grid spacing along the longitudinal and latitudinal directions respectively. The resulting equations (10)-(12) are combined to obtain a single discretized elliptic equation (14) for ϕ which is solved by the optimized Schwarz method discussed in the following sub-sections, and the wind is updated by equations (3)-(4). Then given fields $U^{(l)}$, $V^{(l)}$ and $\phi^{(l)}$ at the previous time step, the solution method is summarized as follows:

- The right-hand sides $R_U^{(l)}$, $R_V^{(l)}$ and $R_\phi^{(l)}$ are calculated in parallel on the two subdomains.
- The elliptic problem is solved and the geopotential $\phi^{(l)}$ is updated on the two subdomains. Interpolation and communication are required to obtain values at subdomain interfaces.
- The wind vector fields ($U^{(l)}$,$V^{(l)}$) are updated in parallel on the two subdomains.

2.1 Iterative Formulation of the Optimized Schwarz Method

Similar to the classical Schwarz method, we solve the discretized problems iteratively in each subdomain

$$\begin{aligned}&-\frac{2+h_\theta\tan\theta_j}{2h_\theta^2}\phi_{i,j-1}^{(l),k}-\frac{1}{\cos^2\theta_jh_\lambda^2}\phi_{i-1,j}^{(l),k}+(\eta+\frac{2}{\cos^2\theta_jh_\lambda^2}+\frac{2}{h_\theta^2})\phi_{i,j}^{(l),k}\\&-\frac{1}{\cos^2\theta_jh_\lambda^2}\phi_{i+1,j}^{(l),k}-\frac{2-h_\theta\tan\theta_j}{2h_\theta^2}\phi_{i,j+1}^{(l),k}=R_{i,j}^{(l),k},\quad i=1,\dots,N;\ j=1,\dots,M,\end{aligned} \tag{14}$$

where $\eta = \frac{a}{\phi^* \Delta t^2}$ is a positive and constant parameter and R is the corresponding right-hand-side function. Following the ideas of [2], we use the following discretizations of the higher order transmission conditions on each interface $\Gamma_d^{(l)}(d = 1, \cdots 4)$

$$\frac{\partial \phi^{(l),k}}{\partial \nu_l} + \beta_d^{(l)} \phi^{(l),k} + \alpha_d^{(l)} \frac{\partial^2 \phi^{(l),k}}{\partial \tau_l^2} = \frac{\partial \phi^{(3-l),k-1}}{\partial \nu_l} + \beta_d^{(l)} \phi^{(3-l),k-1} + \alpha_d^{(l)} \frac{\partial^2 \phi^{(3-l),k-1}}{\partial \tau_l^2}. \tag{15}$$

The symbol $\frac{\partial}{\partial \nu_l}$ stands for the normal derivative of each subdomain and $\frac{\partial}{\partial \tau_l}$ is the corresponding tangential derivative. The $\alpha_d^{(l)}, \beta_d^{(l)}$ are real parameters introduced to optimize the performance of the method. We obtain these coefficients numerically, assuming the coefficients $\alpha_d^{(l)}$ and $\beta_d^{(l)}$ are independent of the boundary d and are antisymmetric, i.e.

$$\alpha_d^{(1)} = -\alpha_d^{(2)}, \quad \beta_d^{(1)} = -\beta_d^{(2)}. \tag{16}$$

In Fig. 2 we represent the performance for the solution of the elliptic problem where each subdomain consists of 90×30 grid points, the overlap δ is one grid point spacing and the Helmhotz coefficient η is equal to one. For the approximate optimal values of the parameters $\alpha_d^{(l)}$ and $\beta_d^{(l)}$, the corresponding methods (see Fig. 2) converge in a small number of iterations and the convergence is much better than the convergence of the classical Schwarz method. This gain more than compensates for the extra cost of computing the needed additional derivatives.

2.2 Substructuring Formulation of the Optimized Schwarz Method

It is possible to both increase the robustness of the optimized Schwarz method and its convergence speed by replacing the above fixed point iterative solver by a Krylov-type method (see [2] and references therein). This is made possible by substructuring the algorithm in terms of interface unknowns, that we denote here by $T_d^{(l)}$ for each subdomain l and interface d. We consider the substructuring formulation of the elliptic problem in equations (14)-(15) where the unknowns $T_d^{(l)}$ are equal to the right-hand side of equation (15), and we rewrite the problem to be solved on each subdomain as

$$\begin{aligned} A^{(l)} \phi^{(l)} &= R^{(l)} \\ B_d^{(l)} \phi^{(l)} &= B_d^{(l)} \phi^{(3-l)} = T_d^{(l)}, \end{aligned} \tag{17}$$

where $B_d^{(l)}$ is the transmission operator which is the identity in the case of Dirichlet conditions. In the previous iterative Schwarz method we solve iteratively, with iteration number $k = 1, \ldots, k_{max}$, in each subdomain the system of equations

$$A^{(l)} \phi^{(l),k} = R^{(l)} + B_d^{(l)} \phi^{(3-l),k-1} = R^{(l)} + T_d^{(l),k-1}, \tag{18}$$

where the matrix $A^{(l)}$ now includes the corrections from the transmission conditions. The Schwarz method corresponds then to the solution for the interface problem unknowns $T_d^{(l)}$ by the Jacobi algorithm

$$T_d^{(l),k} = B_d^{(l)} (A^{(3-l)})^{-1} T_d^{(3-l),k-1} + B_d^{(l)} (A^{(3-l)})^{-1} R^{(3-l)}. \tag{19}$$

We can improve the convergence by considering GMRES, or any other Krylov method, in order to solve the interface system of equations

$$\begin{bmatrix} I & -B_d^{(l)}(A^{(3-l)})^{-1} \\ -B_d^{(3-l)}(A^{(l)})^{-1} & I \end{bmatrix} \begin{bmatrix} T_d^{(l)} \\ T_d^{(3-l)} \end{bmatrix} = \begin{bmatrix} B_d^{(l)}(A^{(3-l)})^{-1}R^{(3-l)} \\ B_d^{(3-l)}(A^{(l)})^{-1}R^{(l)} \end{bmatrix}. \qquad (20)$$

The spectral radius of the left-hand-side matrix in equation (20) depends on the choice of the interface conditions $B_d^{(l)}$. Once the interfaces functions $T_d^{(l)}$ are known, the subproblem solutions are updated in parallel. In Table 1 we consider the two interface equation solvers, Jacobi and GMRES, and we give the number of iterations so that the maximum error is smaller than 10^{-6}. We see that the optimized Robin or second-order transmission conditions give a significant improvement when using either solver.

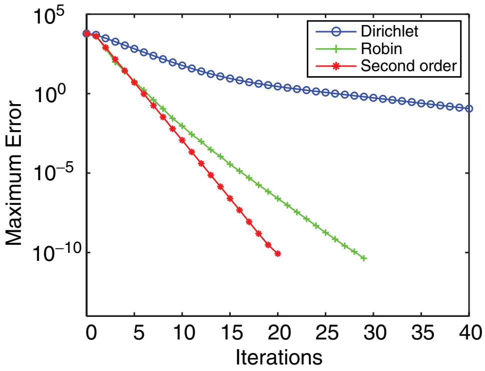


Fig. 2. Convergence behavior for the iterative classical Schwarz and the iterative optimized Schwarz methods, overlap $= 1h$ and $\eta = 1$

Table 1. Number of iterations for 3 interface conditions and 2 solvers. Each panel is 90 × 30, overlap= $1h$ and $\eta = 1$.

Boundary Cond.	ASM	GMRES
Dirichlet	116	26
Robin	20	12
Second order	16	9

3 Numerical Results

In order to assess the capability of the Yin-Yang grid to alleviate the pole problem, we consider the solutions for the 2D linear SWEs on both a global latitude-longitude grid and the Yin-Yang grid and we compare them with a spectral model solution which is free of the pole problem. We start from an initial state at rest (U and $V = 0$) with the initial geopotential perturbation given by:

$$\phi(\lambda, \theta, t = 0) = 2\ \mathit{\Omega} a u_0\ \sin^3(\theta)\cos(\theta)\sin(\lambda) \tag{21}$$

where $\mathit{\Omega} = 0.00007292 s^{-1}$, and $u_0 = 20 m/s$. If we expand ϕ in terms of truncated series of spherical harmonics, its spectrum shows that the non-zero contributions are only from total wavenumbers 2 and 4. We take a global lat-lon grid with the resolution 80 ×40, Yin-Yang grid with resolution 60×20 in each panel, and we compare with an equivalent spectral model with a truncation at wavenumber 39. There must be no interaction between the time tendencies of different spectral coefficients. No new spectral mode can appear during the evolution from the initial state. The spectrum (not plotted here) of the perturbation of the geopotential given by the 3 models after 4 hours shows that for the spectral model the only non zeros are the contributions from wavenumbers 2 and 4 and the initial energy, initially potential, is now divided into kinetic and potential parts. In the spectrum of the perturbation of the geopotential given by using the lat-lon grid and the Yin-Yang grid, there are non zero contributions from wavenumbers other than 2 and 4, however the energy at these wavenumbers is much smaller when using the Yin-Yang grid. We show in Fig. 3 the difference between the perturbation of the geopotential after 4 hours given by using the lat-lon and Yin-Yang gridpoint models and by the spectral model. For lat-lon we can see that the biggest difference is near the poles. The difference between the solution on the Yin-Yang grid and the spectral model is the same everywhere on the globe. The maximum difference after 24 hours (not shown here) between the Yin-Yang gridpoint model and the spectral model solutions corresponds to only 1 m height difference. We can conclude that the use of the quasi-uniform Yin-Yang grid eliminates the pole problem and gives much more accurate solutions than with the standard lat-lon grid.

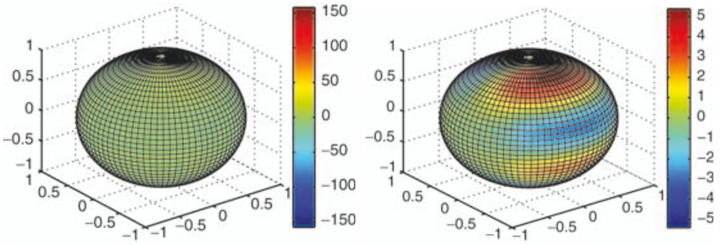


Fig. 3. Geopotential differences between spectral and Lat/lon (left), Yin-Yang and spectral (right). The two subfigures have not the same scale

4 Conclusion

In this paper we have shown that the Schwarz domain decomposition methods are practical for obtaining solutions of the linear SWEs on the sphere with the overset grid system Yin-Yang. In a previous study [5] the convergence analysis of the 1D iterative solution of the elliptic problem on the circle yielded an analytical formula for the optimized coefficients of the transmission conditions. In this paper the optimized coefficients for the 2D case on the Yin-Yang grid were found numerically and it was possible to both increase the robustness of the optimized Schwarz method and its convergence speed by using the substructuring formulation. We have demonstrated that the use of the quasi-uniform Yin-Yang grid effectively eliminates the pole problem and gives more accurate solutions than when using the latitude-longitude grid.

We have not yet validated the full 2D SWEs model on the Yin-Yang grid, but 2D passive semi-Lagrangian advection has been thoroughly tested for this grid, see [5, 6], and the results were comparable to the value in [3]. In future work we will complete the validation of the SWEs with the Yin-Yang grid using the test set of Williamson et al. [7], and finally we will consider the Yin-Yang grid system with more than one regular subdomain in each panel.

Acknowledgement. This research was partly supported by the Canadian Foundation for Climate and Atmospheric Sciences through a grant to the QPF network and by the Office of Science (BER), U.S. Department of Energy, Grant No. DE-FG02-01ER63199. I would like to thank Jean Côté and Martin Gander for reviewing the manuscript.

References

[1] A. Arakawa and V. Lamb. Computational design of the basic dynamical processes of the ucla general circulation model. In *Methods in Computational Physics*, volume 17, pages 174–267. Academic Press, 1977.

[2] M. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.

[3] R. Jacob-Chien, J.J. Hack, and D.L. Williamson. Spectral transform solutions to shallow water test set. *J. Comput. Phys.*, 119:164–187, 1995.

[4] A. Kageyama and T. Sato. Yin-yang grid: An overset grid in spherical geometry. *Geochem. Geophys. Geosyst.*, 5(9), 2004.

[5] A. Qaddouri, J. Côté, M. Gander, and S. Loisel. Optimized Schwarz methods with an overset grid system for the shallow-water equations: Preliminary results. *Appl. Numer. Math.*, 2007. In press.

[6] A. Qaddouri, L. Laayouni, J. Côté, and M. Gander. Optimized Schwarz methods with an overset grid system for the shallow-water equations. *Research Activities in Atmospheric and Oceanic Modelling WMO/TD-1276*, 35(3):21–22, 2005.

[7] D.L. Williamson, J.B. Drake, J.J. Hack, R. Jacob-Chien, and P.N. Swarztrauber. A standard test set for numerical approximations to the shallow water equations in spherical geometry. *J. Comput. Phys.*, 102:211–224, 1992.

MINISYMPOSIUM 8: Robust Methods for Multiscale PDE Problems

Organizers: Ivan G. Graham and Rob Scheichl

University of Bath, Department of Mathematical Sciences, United Kingdom.
I.G.Graham@bath.ac.uk, R.Scheichl@maths.bath.ac.uk

This minisymposium focused on recent developments in analysis and implementation of preconditioners for elliptic problems with highly variable multiscale coefficients, including cases where the coefficient variation cannot be effectively resolved by a practical coarser mesh. Many examples arise, for example in deterministic and stochastic models in hydrogeology, and in oil reservoir modeling. Standard coarsening techniques based on polynomial interpolation do not work well for such problems and in this minisymposium we will focus on recently proposed better techniques, such as multiscale finite element coarsening, optimized interface preconditioners, deflation, and algebraic approaches which are designed to accommodate coefficient behavior.

The talks in the minisymposium covered various topics, including algebraic coarsening methods for non-overlapping domain decompositions; a general theory of robustness for multilevel methods; application of multiscale finite element methods to coarsening; a new theory of aggregation methods for problems with highly variable coefficients and application of multiscale methods in diffusion and absorption in chloroplasts.

Mixed-Precision Preconditioners in Parallel Domain Decomposition Solvers

Luc Giraud[1], Azzam Haidar[2], and Layne T. Watson[3]

[1] ENSEEIHT-IRIT, 2 Rue Camichel 31071 Toulouse Cedex, France. giraud@n7.fr
[2] CERFACS, 42 Av. Coriolis, 31057 Toulouse Cedex, France. haidar@cerfacs.fr
[3] Departments of Computer Science and Mathematics, Virginia Polytechnic Institute & State University, Blacksburg, Virginia, USA. ltw@cs.vt.edu

1 Introduction

Motivated by accuracy reasons, many large-scale scientific applications and industrial numerical simulation codes are fully implemented in 64-bit floating-point arithmetic. On the other hand, many recent processor architectures exhibit 32-bit computational power that is significantly higher than for 64-bit. One recent and significant example is the IBM CELL multiprocessor that is projected to have a peak performance near 256 Gflops in 32-bit and "only" 26 GFlops in 64-bit computation. We might legitimately ask whether all the calculation should be performed in 64-bit or if some pieces could be carried out in 32-bit. This leads to the design of mixed-precision algorithms. However, the switch from 64-bit operations into 32-bit operations increases rounding error. Thus we have to be careful when choosing 32-bit arithmetic so that the introduced rounding error or the accumulation of these rounding errors does not produce a meaningless solution. For the solution of linear systems, mixed-precision algorithms (single/double, double/quadruple) have been studied in dense and sparse linear algebra mainly in the framework of direct methods (see [5, 4, 8, 9]). For such approaches, the factorization is performed in low precision, and, for not too ill-conditioned matrices, a few steps of iterative refinement in high precision arithmetic is enough to recover a solution to full 64-bit accuracy (see [4]). For nonlinear systems, though, mixed-precision arithmetic is the essence of algorithms such as inexact Newton.

For linear iterative methods, we might wonder if such mixed-precision algorithms can be designed. The most natural way, in Krylov subspace methods, is to implement all but the preconditioning steps in high precision. The preconditioner is expected to "approximatively" solve the original problem, so introducing a slight perturbation by performing this step in low precision might not affect dramatically the convergence rate of the iterative scheme. In this paper, we investigate the use of mixed-precision preconditioners in parallel domain decomposition, where the 32-bit calculations are expected to significantly reduce not only the elapsed time of a simulation but also the memory required to implement the preconditioner.

The paper is organized as follows. In Section 2 we motivate using 32-bit rather than 64-bit from a speed perspective. Section 3 is devoted to a brief exposition of the non-overlapping domain decomposition technique we consider for the parallel numerical experiments discussed in Section 4.

2 Mixed-Precision Algorithms

Counter to the 64-bit RISC trend, for some recent architectures, a 64-bit operation is more expensive than a 32-bit one. In particular, those that possess a SSE (streaming SIMD extension) execution unit can perform either two 64-bit instruction or four 32-bit instructions in the same time. This class of chip includes for instance the IBM PowerPC, the Power MAC G5, the AMD Opteron, the CELL, and the Intel Pentium. Table 1 reports the performance of basic dense kernels involved in numerical linear algebra: the _GEMV BLAS-2 matrix-vector product and the _POTRF/_POTRS LAPACK Cholesky factorization and backward/forward substitution. It can be seen that 32-bit calculation generally outperforms 64-bit. For a more exhaustive set of experiments on various computing platforms, refer to [8, 9]. The source of time reduction is not only the processing units that perform more operations per clock-cycle, but also a better usage of the complex memory hierarchy that provides ultra-fast memory transactions by reducing the stream of data block traffic across the internal bus and bringing larger blocks of computing data into the cache. This provides a speed up of two in 32-bit compared to 64-bit computation for BLAS-3 operations in most LAPACK routines.

Table 1. Elapsed time (sec) to perform BLAS-2 and LAPACK routines on various platforms when the size m of the matrices is varied.

CRAY XD1 AMD Opteron processor

n	DGEMV	SGEMV	Ratio	DPOTRF	SPOTRF	Ratio	DPOTRS	SPOTRS	Ratio
2000	0.012	0.005	2.18	0.823	0.462	1.78	0.010	0.004	2.22
7000	0.121	0.056	2.16	29.41	16.04	1.83	0.116	0.056	2.07

MAC Power PC G5 processor VMX/AltiVec extensions

n	DGEMV	SGEMV	Ratio	DPOTRF	SPOTRF	Ratio	DPOTRS	SPOTRS	Ratio
2000	0.028	0.008	3.51	0.828	0.453	1.82	0.032	0.022	1.45
7000	0.354	0.122	2.90	23.71	13.27	1.78	0.372	0.355	1.05

Another advantage of 32-bit floating point arithmetic is that data storage is reduced by half, providing an increase in data throughput. Similarly, in a distributed memory environment, the message sizes are halved.

3 Exploiting 32-bit Calculation in Domain Decomposition

Consider the following second order self-adjoint elliptic problem in the unit cube $\Omega = (0,1)^3 \subset \mathbb{R}^3$:

$$\begin{cases} -\nabla(a(x,y,z)\nabla u) = f(x,y) & \text{in} \quad \Omega, \\ \qquad\qquad\qquad\quad u = 0 & \text{on} \quad \partial\Omega, \end{cases} \tag{1}$$

where $a(x,y,z) \in \mathbb{R}^3$ is a positive definite symmetric matrix function. We assume that the domain Ω is partitioned into N non-overlapping subdomains Ω_1,..., Ω_N and boundary $\Gamma = \cup\ \Gamma_i$, where $\Gamma_i = \partial\Omega_i \backslash \partial\Omega$. We discretize (1) by using a finite element method resulting in a symmetric positive definite linear system, $A_h u_h = f_h$. Let I denote the union of the interior points in the subdomains, and let B denote the interface points separating the subdomains. Then grouping the unknowns corresponding to I in the vector u_I and the unknowns corresponding to B in the vector u_B, we obtain the following reordering of the fine grid problem:

$$\begin{pmatrix} A_{II} & A_{IB} \\ A_{IB}^T & A_{BB} \end{pmatrix} \begin{pmatrix} u_I \\ u_B \end{pmatrix} = \begin{pmatrix} f_I \\ f_B \end{pmatrix}. \tag{2}$$

Eliminating u_I in the second block row leads to the following reduced equation for u_B:

$$Su_B = f_B - A_{IB}^T A_{II}^{-1} f_I \quad \text{with} \quad S = A_{BB} - A_{IB}^T A_{II}^{-1} A_{IB}, \tag{3}$$

where S is referred to as the Schur complement matrix that is symmetric positive definite if A_h is symmetric positive definite. Let $\mathcal{R}_{\Gamma_i} : \Gamma \rightarrow \Gamma_i$ be the canonical point-wise restriction that maps full vectors defined on Γ into vectors defined on Γ_i. The Schur complement matrix (3) can be written as the sum of elementary matrices, $S = \sum_{i=1}^{N} \mathcal{R}_{\Gamma_i}^T S^{(i)} \mathcal{R}_{\Gamma_i}$, where $S^{(i)} = A_{\Gamma_i \Gamma_i} - A_{\Gamma_i I_i} A_{I_i I_i}^{-1} A_{I_i \Gamma_i}$ is a local Schur complement. We define the assembled local Schur complement, $\bar{S}^{(i)} = \mathcal{R}_{\Gamma_i} S \mathcal{R}_{\Gamma_i}^T$, that corresponds to the restriction of the Schur complement to the interface Γ_i. The assembled local Schur complement can be computed in a parallel environment from the local Schur complement via a few neighbor to neighbor communications. We define the additive Schwarz preconditioner by $M_{AS} = \sum_{i=1}^{N} \mathcal{R}_{\Gamma_i}^T \left(\bar{S}^{(i)}\right)^{-1} \mathcal{R}_{\Gamma_i}$, (see [3]).

We propose to take advantage of the 32-bit speed and memory benefit and build some part of the code in 32-bit. Our goal is to use costly 64-bit arithmetic only where necessary to preserve accuracy. We consider here the simple approach of performing all the steps of a Krylov subspace method except the preconditioning in 64-bit [10]. In this respect, it is important to note that the preconditioner only attempts to approximate the inverse of the matrix S. Since our matrices are symmetric positive definite, the Krylov subspace method of choice is conjugate gradient (CG). In our mixed-precision implementation only the preconditioned residual is computed in 32-bit. The import of this strategy is that the Gaussian elimination (factorization) of the local assembled Schur complement (used as preconditioner), and the forward and the back substitutions to compute the preconditioned residual, are performed in 32-bit while the rest of the algorithm is implemented in 64-bit.

Since the local assembled Schur complement is dense, cutting the size of this matrix in half has a considerable effect in terms of memory space. Another benefit is in the total amount of communication that is required to assemble the preconditioner. As for the memory required to store the preconditioner, the size of the exchanged messages is also half that for 64-bit. Consequently, if the network latency is neglected, the overall time to build the preconditioner for the 32-bit implementation should be half that for the 64-bit implementation. These improvements are illustrated by detailed numerical experiments with the mixed-precision implementation reported in Section 4.

4 Numerical Results

The target computer is the Terascale computer system X located at Virginia Tech's. The system Xserve is a 1,100 dual G5 processor nodes ran at 2.3GHz; each node has 4GB of main memory. The G5 cluster operates at 20.24 64-bit Teraflops peak, and the networking consists both of standard Ethernet for "non-computational" tasks, and special Mellanox Cougar InfiniBand 4x HCA networks for high-bandwidth and low-latency communications. In a parallel distributed memory environment, the domain decomposition strategy is followed to assign each local PDE problem (subdomain) to one processor that works independently of other processors and exchange data using MVAPICH (MPI for InfiniBand on VAPI Layer). The code is written in Fortran 90 and compiled with the IBM compiler.

In what follows we start by taking a brief look at our parallel implementation that relies on a unique feature of the multifrontal sparse direct solver Mumps (see [1, 2]); that offers the possibility to compute the Schur complement matrices $S^{(i)}$ at an affordable memory and computational cost thanks to its multifrontal approach. Those local Schur complement matrices computed explicitly on each processor are then assembled using neighbor to neighbor communication, which is independent of the number of processors. Then they are factorized using the dense linear Lapack kernel, to construct the additive Schwarz preconditioners. Finally, we note that the solution of this reduced linear system associated with the Schur complement is typically performed by a distributed preconditioned conjugate gradient solver.

In this section we compare the performance of a fully 64-bit with a mixed-precision implementation. For all the parallel experiments, we solve either the two-dimensional or the three-dimensional elliptic PDE defined respectively in the unit square or cube, using a uniform domain decomposition into equal sized squares or cubes. Since the goal is to study the numerical efficiency of the preconditioner, we only perform scaled experiments where the matrix size for the subdomains is kept constant (i.e., constant $\frac{H}{h}$ where H is the diameter of the subdomains, and h is the mesh size) when the number of subdomains is increased. In the table, we refer to the fully 64-bit and mixed-precision experiments as M_d and M_m, respectively.

In order to illustrate the effect on the convergence rate, we report in Table 2 the number of conjugate gradient iterations to reduce the scaled residual $\frac{\|r_k\|}{\|b\|}$ below 10^{-8}, where b is the right-hand side of the Schur complement system. We consider the smooth and not too ill-conditioned problems associated with the Poisson equation and a heterogeneous diffusion problem with coefficient jumps from 1 to 10^3 in various places of the unit square or cube. This latter example gives rise to more ill-conditioned linear systems to solve. Finally, while keeping constant the size of the subdomains, we vary their numbers and consider two different subdomain sizes.

In Table 2, we report result observed on the two dimensional model. Because only local preconditioner are considered, it can be seen that the number of iterations grows with the number of subdomains. In terms of iterations, it can be seen that for the two different problems M_m behaves closely to M_d. With this choice of the mixed-precision, it can be expected a reduction of the global elapsed time as described below. For the three dimensional case, the behavior of the preconditioners is depicted in Table 2. The first observation, that we do not further develop, is that the preconditioner, which does not implement any coarse space component to account for the global coupling of the PDEs, does not scale too badly when the number of subdomains is increased. Its scalability with respect to the size of the subdomains is also acceptable as only a slight increase is observed when we go from subdomains with about 15,000 degrees of freedom (dof) to subdomains with about 43,000 dof. On the accuracy effect of the mixed-precision usage, it can be observed once again that it only moderately increases the number of iterations, and the increase does not depend much either on the number of subdomains or on the size of the subdomains. As expected, the growth is also slightly larger on the ill-conditioned heterogeneous problem as for the Poisson problem.

Table 2. Number of conjugate gradient iterations when the number of subdomains and the subdomain grid is varied: $2D$ and $3D$ case.

2D experiments		*Poisson Problem*				*Discontinuous Problem*			
subdomain grid		25	64	144	256	25	64	144	256
35 × 35	M_d	15	24	33	40	23	33	55	61
	M_m	15	26	33	41	23	34	55	62
1000 × 1000	M_d	20	34	45	55	37	47	76	91
	M_m	21	35	47	57	39	48	78	93

3D experiments		*Poisson Problem*				*Discontinuous Problem*			
subdomain grid		27	64	125	216	27	64	125	216
25 × 25 × 25	M_d	17	24	26	31	23	33	36	44
	M_m	19	26	28	33	24	34	39	45
35 × 35 × 35	M_d	19	26	30	33	25	35	40	47
	M_m	21	29	30	35	25	37	42	49

In Table 3 we report on three-dimensional numerical experiments related to the construction of the preconditioner for different problem sizes, varying the number of processors from 27 up to 216 (i.e., varying the decomposition of the cube from 3×3×3 up to 6×6×6). We depict the preconditioner setup time for both M_d and M_m. The row entitled "init" corresponds to the calculation of the local Schur complement using the Mumps package. The construction of the local Schur complements that are involved in the matrix-vector product in the conjugate gradient algorithm is performed in both cases in 64-bit arithmetic, so the cost is the same for the two variants. The "setup precond" row is the time required to assemble and factorize, using Lapack, the assembled local Schur complement. As might be expected, these results show that the 32-bit preconditioner setup time is significantly smaller than that for 64-bit. Note that the 32-bit arithmetic cut in half the time for assembling the local Schur matrix, due to halving the amount of communication. Also the $\mathcal{O}(n^3)$ floating-point operations of the LL^T factorization $[S/D]POTRF$ are about a factor of 1.8 faster in 32-bit.

Table 3. Parallel performance for various steps of the preconditioned conjugate gradient implementations (35 × 35 × 35 subdomain grid).

# proc	27		64		125		216	
	M_d	M_m	M_d	M_m	M_d	M_m	M_d	M_m
init	26.8	26.8	26.8	26.8	26.8	26.8	26.8	26.8
setup precond	21.2	12.3	21.2	12.3	21.3	12.3	21.4	12.4
time per iter	0.73	0.68	0.73	0.69	0.76	0.71	0.76	0.72
total	66.3	56.1	73.5	64.6	78.5	68.9	83.9	74.5
# iter	25	25	35	37	40	42	47	49

From a memory viewpoint, using M_m saves 150 MB of memory per processor for the example with about 43,000 dof per subdomain. From a computational perspective, memory and CPU time, the saving is clear. It can be seen that the time per iteration is almost constant and does not depend much on the number of processors for both preconditioners. In terms of the overall computing time, the row entitled "total" in Table 3 displays the overall elapsed time to solve the heterogeneous diffusion problem with 43,000 dof per subdomain when the number of domains is varied. These results show that on the most difficult problem the time saved by the use of mixed-precision arithmetic still compensates for a slight increase in the number of iterations, and that M_m outperforms M_d.

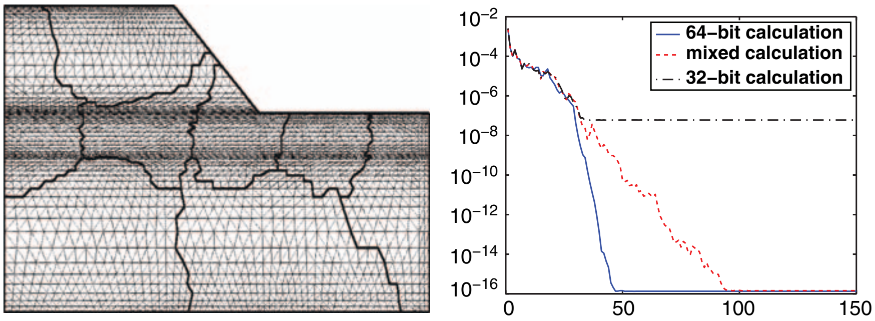


Fig. 1. Convergence history of $\|r_k\|/\|b\|$ (right) on a 15 000 dof problem in mixed finite element device modeling simulation on a mosfet (left).

In Figure 1, we report the convergence history of PCG using entire 64/32-bits and mixed arithmetic calculation on an unstructured 2D problem arising from mixed finite element discretization in device modeling simulations (150 000 dof and 16 subdomains) [7]. This real life example exhibits similar numerical behavior as the ones observed on the academic examples of Table 2. Namely, the pure 32-bit calculation has a limiting accuracy much larger than the mixed and the full 64-bit computation.

5 Concluding Remarks

In a linear iterative parallel domain decomposition solver, the use of 32-bit arithmetic was limited to the preconditioning step. The main advantage of using mixed-precision is that it reduces the data storage, the computational time, and the communication overhead while only marginally degrading the convergence rate without preventing to reach similar accuracy as full 64-bit calculation. This work is just a first study of mixed arithmetic implementation, and other variants will be considered in future work. While the current work is purely experimental, some theoretical studies deserve to be undertaken following possibly some techniques presented in [11]. Finally, we mention that the one-level preconditioner presented here can be considered in a two-level scheme, we refer to [6] for more details on that aspect.

References

[1] P.R. Amestoy, I.S. Duff, J. Koster, and J.-Y. L'Excellent. A fully asynchronous multifrontal solver using distributed dynamic scheduling. *SIAM J. Matrix Anal. Appl.*, 23(1):15–41, 2001.

[2] P.R. Amestoy, A. Guermouche, J.-Y. L'Excellent, and S. Pralet. Hybrid scheduling for the parallel solution of linear systems. *Parallel Comput.*, 32(2):136–156, 2006.

[3] L.M. Carvalho, L. Giraud, and G. Meurant. Local preconditioners for two-level non-overlapping domain decomposition methods. *Numer. Linear Algebra Appl.*, 8(4):207–227, 2001.

[4] J. Demmel, Y. Hida, W. Kahan, S.X. Li, S. Mukherjee, and E.J. Riedy. Error bounds from extra precise iterative refinement. Technical Report UCB/CSD-04-1344, LBNL-56965, University of California in Berkeley, 2006. Short version appeared in ACM Trans. Math. Software, vol. 32, no. 2, pp 325-351, June 2006.

[5] J. W. Demmel, S. C. Eisenstat, J. R. Gilbert, S. X. Li, and J. W. H. Liu. A supernodal approach to sparse partial pivoting. *SIAM J. Matrix Anal. Appl.*, 20(3):720–755, 1999.

[6] L. Giraud, A. Haidar, and L.T. Watson. Parallel scalability study of three dimensional additive Schwarz preconditioners in non-overlapping domain decomposition. Technical Report TR/PA/07/05, CERFACS, Toulouse, France, 2007.

[7] L. Giraud, A. Marrocco, and J.-C. Rioual. Iterative versus direct parallel substructuring methods in semiconductor device modelling. *Numer. Linear Algebra Appl.*, 12(1):33–53, 2005.

[8] J. Kurzak and J. Dongarra. Implementation of the mixed-precision high performance LINPACK benchmark on the CELL processor. Technical Report LAPACK Working Note #177 UT-CS-06-580, University of Tennessee Computer Science, September 2006.

[9] J. Langou, J. Langou, P. Luszczek, J. Kurzak, A. Buttari, and J. Dongarra. Exploiting the performance of 32 bit floating point arithmetic in obtaining 64 bit accuracy. Technical Report LAPACK Working Note #175 UT-CS-06-574, University of Tennessee Computer Science, April 2006.

[10] S. Lanteri. Private communication, 2006.

[11] G. Meurant. *The Lanczos and Conjugate Gradient Algorithms: From Theory to Finite Precision Computations*. Software, Environments, and Tools 19. SIAM, 2006.

Coefficient-explicit Condition Number Bounds for Overlapping Additive Schwarz

Ivan G. Graham and Rob Scheichl

Department of Mathematical Sciences, University of Bath, Bath BA2 7AY, United Kingdom. {I.G.Graham,R.Scheichl}@bath.ac.uk

1 Introduction

In this paper we discuss new domain decomposition preconditioners for piecewise linear finite element discretizations of boundary-value problems for the model elliptic problem

$$-\nabla \cdot (\mathcal{A}\nabla u) \;=\; f \;, \tag{1}$$

in a bounded polygonal or polyhedral domain $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3 with suitable boundary data on the boundary $\partial\Omega$. The tensor $\mathcal{A}(x)$ is assumed isotropic and symmetric positive definite, but may vary with many orders of magnitude in an unstructured way on Ω. Many examples arise in groundwater flow and oil reservoir modeling.

Let $\mathcal{T}^h$ be a conforming shape-regular simplicial mesh on Ω and let $\mathcal{S}^h(\Omega)$ denote the space of continuous piecewise linear finite elements on $\mathcal{T}^h$. The finite element discretization of (1) in $\mathcal{V}^h$ (the n-dimensional subspace of functions in $\mathcal{S}^h(\Omega)$ which vanish on essential boundaries), yields the linear system:

$$A\mathbf{u} = \mathbf{f} \;, \tag{2}$$

and it is well-known that the conditioning of A worsens when $\mathcal{T}^h$ is refined or when the heterogeneity (characterized by the range of $\mathcal{A}$) becomes large. It is of interest to find solvers for (2) which are robust to changes in the mesh width h as well as to the heterogeneity.

While there are many papers which solve (2) for "layered media" in which discontinuities in $\mathcal{A}$ are simple interfaces that can be resolved by a coarse mesh (see e.g. [4, 12] and the references therein), until recently there was no rigorously justified method for general heterogeneous media. We present here a summary of some recent papers [6, 7, 10, 11] where a new analysis of domain decomposition methods for (2) (which have inherent robustness with respect to h) was presented. This analysis indicates explicitly how subdomains and coarse spaces should be designed in order to achieve robustness also with respect to heterogeneities. More precisely this analysis introduces new "robustness indicators" (which depend on the choice of subdomains and coarse space and in particular depend on the energy of the coarse space basis functions) and proves that, if these indicators are controlled, then the preconditioner will be robust. Papers [6, 7] then go on to consider the use of multiscale finite elements to build coarse spaces for domain decomposition and prove a number of results which indicate their robustness in cases where standard coarsening methods fail to be robust. Papers [10, 11] consider aggregation-based coarsening (as introduced e.g. in [13, 2]) and prove similar results as in the multiscale finite element case.

The coarse spaces proposed in [6] yield coefficient-dependent prolongation operators, similar to those which have been tested empirically in the context of (Schur complement based) domain decomposition methods in [3, 5]. The concept of energy-minimizing coarse spaces also appears in several papers on the construction of algebraic multigrid methods [14, 9, 15], but their behavior in the presence of heterogeneity is not analyzed. The use of multiscale finite elements as coarseners was also proposed in [1], but again this was in the Schur-complement context and the analysis depended on classical periodic homogenization theory. The analysis in [6] does not require periodicity and does not appeal to homogenization theory. We are also not aware of any theoretical results which make explicit the dependency of the condition number on heterogeneities in $\mathcal{A}$ in the case of the aggregation-based coarse spaces proposed in [10].

Given a finite overlapping open covering of subdomains $\{\Omega_i : i = 1, \ldots, s\}$ of Ω, with each $\overline{\Omega}_i$ assumed to consist of a union of elements from $\mathcal{T}^h$, and a coarse basis $\{\Phi_j : j = 1, \ldots, N\} \subset \mathcal{V}^h$, we study two-level additive Schwarz preconditioners

$$M_{AS}^{-1} \;=\; \sum_{i=0}^{s} R_i A_i^{-1} R_i^T \,. \tag{3}$$

Here, for $i = 1, \ldots, s$, R_i denotes the restriction matrix from freedoms in Ω to freedoms in Ω_i and $(R_0)_{j,p} = \Phi_j(x_p)$, where x_p, $p = 1, \ldots, n$, are the interior nodes of the fine mesh $\mathcal{T}^h$. The matrices A_i are then defined via the Galerkin product $A_i := R_i A R_i^T$.

For the purposes of exposition we will only describe the theory for scalar $\mathcal{A}$ in (1), i.e. $\mathcal{A} = \alpha I$, and restrict to the case of homogeneous Dirichlet boundary conditions. For theoretical purposes, we shall also assume that $\alpha \geq 1$. This is no loss of generality, since problem (2) can be scaled by $(\min_x \alpha(x))^{-1}$ without changing its conditioning. Throughout the paper, the notation $C \lesssim D$ (for two quantities C, D) means that C/D is bounded above independently of the *mesh parameters* and of the *coefficient function* α.

2 Coefficient-explicit Schwarz Theory I

The assumptions on the coarse space and on the overlapping subdomains made in the papers [6, 7] and [10, 11] are different. We start in this section with the theory presented in [10, 11]. Although there we only applied the theory to aggregation-based coarsening, we will show here that it can also be applied in the case of the multiscale coarsening introduced in [6, 7], leading to a slightly different condition number bound than the one in [6, 7]. The key assumption in this section is that the support of each coarse space basis function is fully contained in at least one subdomain. For details and proofs see [10].

We start with a linearly independent set $\{\Phi_j : j = 1, \ldots, N_H\} \subset \mathcal{S}^h(\Omega)$. This set contains the functions defined above, but also some functions which do not vanish on the boundary, so $N_H > N$. We set $\omega_j = \text{interior}\,(\text{supp}\{\Phi_j\})$ with diameter H_j. For theoretical purposes we assume that the $\{\omega_j\}$ form a shape-regular overlapping cover of Ω and that the overlap between any support ω_j and its neighbors is uniformly of size δ_j. In addition we make the following assumptions:

(C1) For all $j = 1, \ldots, N_H$ there is an $i_j \in \{1, \ldots, s\}$ such that $\omega_j \subset \Omega_{i_j}$.
(C2) $\sum_{j=1}^{N_H} \Phi_j(x) = 1$, for all $x \in \bar{\Omega}$.
(C3) $\|\Phi_j\|_{L_\infty(\Omega)} \lesssim 1$.

We assume that the functions Φ_j are numbered in such a way that $\Phi_j \in \mathcal{V}^h$ for all $j \leq N$ and $\Phi_j \notin \mathcal{V}^h$ for all $j > N$. Thus we can denote the coarse space by $\mathcal{V}_0 = \text{span}\{\Phi_j : j = 1, \ldots, N\}$ and we have $\mathcal{V}_0 \subset \mathcal{V}^h$.

Note that although we have not directly made any assumptions on the overlap of the subdomains, (C1) implies that the overlap of a subdomain Ω_i and its neighbors is always bounded from below by $\min_{\{j\,:\,\omega_j \subset \Omega_i\}} \delta_j$.

It is known (see e.g. [12]) that in order to bound $\kappa(M_{AS}^{-1}A)$, we need to assume some upper bounds on $|\Phi_j|^2_{H^1(\Omega)}$. We take a novel approach here and introduce a quantity which also reflects how the coarse space handles the coefficient heterogeneity:

Definition 1. (Coarse space robustness indicator I).

$$\gamma_\infty(\alpha) \;=\; \max_{j=1}^{N_H} \left\{\delta_j^2 \, \|\alpha|\nabla\Phi_j|^2\|_{L_\infty(\Omega)}\right\} \;.$$

The quantity $\gamma_\infty(\alpha)$ appears in our estimates for the two–level preconditioner below. Note that, roughly speaking, this robustness indicator is well-behaved if the Φ_j have small gradient wherever α is large. The weight δ_i^2 is chosen to make $\gamma_\infty(\alpha) \lesssim 1$ when $\alpha = 1$.

We can now state one of the main results from [10] (Theorem 3.8):

Theorem 1. *Assume that (C1)-(C3) hold true. Then*

$$\kappa\left(M_{AS}^{-1}A\right) \;\lesssim\; \gamma_\infty(\alpha) \left(1 + \max_{j=1}^{N_H} \frac{H_j}{\delta_j}\right) \;.$$

Example 1 (Linear Finite Element Coarsening). In the classical case, i.e. when $\{\Phi_j\}$ is the standard nodal basis for the continuous piecewise linear functions on a coarse simplicial mesh $\mathcal{T}^H$, we have $\delta_j \sim H_j$ and $\gamma_\infty(\alpha) \lesssim \max_{x\in\Omega} \alpha(x)$, and so $\gamma_\infty(\alpha) \lesssim 1$ when $\alpha \sim 1$. When $\alpha(x) \to \infty$ for some $x \in \Omega$ then Theorem 1 suggests that linear coarsening may not be robust anymore. The numerical results in Table 1 (left) show that this is indeed the case and that $\gamma_\infty(\alpha)$ is a good indicator for the loss of robustness. The results in Table 1 are for $\Omega = [0,1]^2$ and $\alpha(x) = \hat{\alpha}$ on an “island” in the interior of each coarse element $K \in \mathcal{T}^H$ a distance $O(H)$ away from ∂K, with $\alpha(x) = 1$ otherwise (for a precise description of α see [6, Example 5.1]). Also, there is exactly one subdomain Ω_{i_j} per coarse node x_j^H with $\omega_j \subset \Omega_{i_j}$ (to ensure (C1)).

However, our framework leaves open the possibility of choosing the Φ_j to depend on α in such a way that $\gamma_\infty(\alpha)$ is still well-behaved. The next two examples give two possible ways of constructing such Φ_j.

Table 1. Standard additive Schwarz with linear coarsening (left) and with multiscale coarsening (right) for $h = 512^{-1}$ and $H = 8h$.

$\hat{\alpha}$	$\kappa(M_{AS}^{-1}A)$	$\gamma_\infty(\alpha)$
10^0	5.2	2
10^1	9.1	20
10^2	58.1	200
10^3	471	2000
10^4	1821	20000

$\hat{\alpha}$	$\kappa(M_{AS}^{-1}A)$	$\gamma_\infty(\alpha)$
10^0	5.2	2
10^1	5.2	9.5
10^2	5.2	14.2
10^3	5.2	14.9
10^4	5.2	15.0

Example 2 (Aggregation-based Coarsening). Let $\mathcal{N} = \{x_1, \ldots, x_n\}$ be the degrees of freedom on the fine mesh, and let $\{W_j : j = 1, \ldots, N_H\}$ be a non-overlapping partition of $\mathcal{N}$ (i.e. $\cup\{W_j : j = 1, \ldots, N_H\} = \mathcal{N}$ and $W_j \cap W_{j'} = \emptyset \ \ \forall j \neq j'$). For each j, we define a coefficient vector $\boldsymbol{\Phi}^j \in \mathbb{R}^n$ such that $\Phi_p^j = 1$, if node $x_p \in W_j$, and $\Phi_p^j = 0$ otherwise. Let $\Phi_j \in S^h(\Omega)$ be the linear finite element function with nodal values $\boldsymbol{\Phi}^j$. Note that although the aggregates W_j are non-overlapping, the supports ω_j of the functions Φ_j are. The overlap essentially consists of one layer of fine grid elements and so for quasi-uniform $\mathcal{T}^h$ we have $\delta_j \sim h$. In [10] (see also [2]) we go on to smooth these functions by using a simple damped Jacobi smoother. This increases the overlap. However, here we will only consider the simplest case of no smoothing.

It follows immediately from the above construction that the Φ_j are linearly independent and satisfy (C2) and (C3). Therefore, if the covering $\{\Omega_i\}$ is chosen such that (C1) is satisfied, then Theorem 1 implies

$$\kappa\left(M_{AS}^{-1}A\right) \lesssim \gamma_\infty(\alpha) \max_{j=1}^{N_H} \frac{H_j}{h}. \tag{4}$$

The Φ_j have nonzero gradient only in the overlap of ω_j and so, provided α is well-behaved in the overlap, $\gamma_\infty(\alpha)$ can be bounded independent of $\max_{x\in\Omega} \alpha(x)$. In [10] we present an algorithm to choose aggregates W_j which can be proved to satisfy this for certain choices of "binary" coefficient functions α by using the idea of strong connections in A from algebraic multigrid (AMG). Given an aggregation "radius" $r \in \mathbb{N}$ and a threshold for strong connections, roughly speaking each of the aggregates W_j is calculated by finding the strongly–connected graph r-neighborhood of a suitably chosen seed node $x_j^H \in \mathcal{N}$. The aggregation procedure in [10] uses an advancing front in the graph induced by A to choose good seed nodes. We refer to [10] for details and numerical results with binary and random media.

Example 3 (Multiscale Finite Element Coarsening I). Let $\mathcal{T}^H$ be a shape-regular mesh of coarse simplices on Ω with a typical element of $\mathcal{T}^H$ being the (closed) set K, which we assume to consist of the union of a set of fine grid elements $\tau \in \mathcal{T}^h$. Also, let $\{x_j^H : j = 1, \ldots, N_H\}$ be the set of nodes of $\mathcal{T}^H$ and let $\mathcal{F}^H$ denote the set of all (closed) faces of elements in $\mathcal{T}^H$. (In the 2D case "faces" should be interpreted to mean "edges".) Finally, introduce also the *skeleton* $\Gamma = \cup\{f : f \in \mathcal{F}^H\}$, i.e. the set of all faces of the mesh, including those belonging to the outer boundary $\partial\Omega$.

Here, each of the coarse space basis functions Φ_j is associated with a node x_j^H of $\mathcal{T}^H$. They are obtained by extending (via a discrete harmonic extension with respect to the original elliptic operator (1)) predetermined boundary data on the faces which contain x_j^H, into the interior of each element K. To introduce boundary data for each $j = 1, \dots, N_H$, we introduce functions $\psi_j : \Gamma \to \mathbb{R}$ which are required to be piecewise linear (with respect to the mesh $\mathcal{T}^h$ on Γ) and are required also to satisfy the assumptions:

(M1) $\psi_j(x_{j'}^H) \;=\; \delta_{j,j'}\;,\quad j, j' = 1, \dots, N_H,$

(M2) $0 \;\leq\; \psi_j(x) \;\leq\; 1\;,$ and $\sum_{j=1}^{N_H} \psi_j(x) \;=\; 1\;,$ for all $x \in \Gamma\;,$

(M3) $\psi_j \;\equiv\; 0$ on all faces $f \in \mathcal{F}^H$ such that $x_j^H \not\in f\;.$

Using ψ_j as boundary data, for each $j = 1, \dots, N_H$, the basis functions $\Phi_j \in \mathcal{S}^h(\Omega)$, are then defined by discrete $\alpha-$harmonic extension of ψ_j into the interior of each $K \in \mathcal{T}^H$. That is, for each $K \in \mathcal{T}^H$, $\Phi_j|_K \in \{v_h \in \mathcal{S}^h(K) : v_h|_{\partial K} = \psi_j|_{\partial K}\}$ is such that

$$\textstyle\int_K \alpha \nabla(\Phi_j|_K) \cdot \nabla v_h \;=\; 0 \quad \text{for all} \quad v_h \in \mathcal{S}_0^h(K) \tag{5}$$

where $\mathcal{S}^h(K)$ and $\mathcal{S}_0^h(K)$ are the continuous piecewise linear finite element spaces with respect to $\mathcal{T}^h$ restricted to K.

The obvious example of boundary data ψ_j satisfying (M1)–(M3) are the standard hat functions on $\mathcal{T}^H$ restricted to the faces (edges) of the tetrahedron (triangle) K. However, these are not so appropriate if α varies strongly near the boundary ∂K. The *oscillatory* boundary conditions suggested in [8] are more useful in this case (see [6] for details).

This recipe specifies $\Phi_j \in S^h(\Omega)$ which can immediately be seen to be linearly independent and to satisfy the assumptions (C2) and (C3) (see [6]). Moreover, we have $\delta_j \sim H_j$. Therefore, if the covering $\{\Omega_i\}$ is chosen such that (C1) is satisfied, then Theorem 1 implies

$$\kappa\left(M_{AS}^{-1}A\right) \;\lesssim\; \gamma_\infty(\alpha). \tag{6}$$

The numerical results in Table 1 (right), obtained for the test problem introduced in Example 1, show that additive Schwarz with multiscale coarsening can indeed be robust even when the coarse mesh does not resolve discontinuities in α and that our theory accurately predicts this (cf. (6)). For more numerical results with multiscale coarsening see [6, 7].

3 Coefficient-explicit Schwarz Theory II

In practice, Assumption (C1) may be too restrictive, as it may require quite generous overlap of the subdomains (e.g. in the case of multiscale coarsening). The theory in [6, 7] does not require (C1). However, it requires an underlying coarse mesh and is therefore not as easily applicable to other more general coarse spaces such as the aggregation-based ones in Example 2. For details and proofs on this section see [6].

Let $\mathcal{T}^H$ be a shape-regular coarse mesh as defined in Example 3, and for every $K \in \mathcal{T}^H$ let $H_K = \mathrm{diam}(K)$. We will now replace Assumption (C1) by

(C1') $\Phi_j(x_{j'}^H) = \delta_{j,j'}\;,\; j, j' = 1, \dots, N_H,$ and $\mathrm{supp}(\Phi_j) \subset \cup\{K : x_j^H \in K\}.$

This implies that the Φ_j are linearly independent and that $\mathcal{V}_0 = \mathrm{span}\{\Phi_j : j = 1, \dots, N\} \subset \mathcal{V}^h$. However, even though we no longer need Assumption (C1) we do still need a mild assumption on the relative size of the subdomains and the coarse mesh. For shape-regular subdomains Ω_i we can write this as

(C4) $H_K \lesssim \mathrm{diam}(\Omega_i)\,, \quad$ for all $K \in \{K : K \cap \bar{\Omega}_i \neq \emptyset\}$ and $i = 1, \ldots, s$,

although we note that in [6] this requirement is generalized to allow highly anisotropic subdomains such as may arise in the application of mesh partitioning software. Note that (C4) does not impose any direct structural relation between coarse mesh and subdomains.

The condition number estimate in this section separates robustness with respect to the coarse space from robustness with respect to the overlapping covering. We therefore introduce two robustness indicators. Analogous to $\gamma_\infty(\alpha)$ we first introduce a quantity which reflects how the coarse space handles the coefficient heterogeneity. However, here we measure the "energy" of the coarse space basis functions in the L_2-norm instead of the L_∞-norm.

Definition 2. (Coarse space robustness indicator II).

$$\gamma_2(\alpha) \;=\; \max_{j=1}^{N_H} \left\{ H_j^{2-d}\, |\Phi_j|^2_{H^1(\Omega),\alpha} \right\} \quad \textit{where} \quad H_j \;=\; \mathrm{diam}(\omega_j)\ .$$

The second quantity which we introduce measures in a certain sense the ability of the subdomains Ω_i to handle the coefficient heterogeneity.

Definition 3. (Partitioning robustness indicator).

$$\pi(\alpha) \;=\; \inf_{\{\chi_i\} \,\in\, \Pi(\{\Omega_i\})} \left(\max_{i=1}^{s} \left\{ \delta_i^2 \; \left\| \alpha |\nabla \chi_i|^2 \right\|_{L_\infty(\Omega)} \right\} \right)$$

where δ_i is here the overlap for subdomain Ω_i and $\Pi(\{\Omega_i\})$ denotes the set of all partitions of unity $\{\chi_i\} \subset W^1_\infty(\Omega)$ subordinate to the cover $\{\Omega_i\}$.

Roughly speaking, $\pi(\alpha)$ is well-behaved if there is a partition of unity whose members have small gradient wherever α is large. The weight δ_i^2 is chosen to make $\pi(\alpha) \lesssim 1$ when $\alpha = 1$.

Using these two robustness indicators and under the assumptions made in this section we can now state one of the main results from [6] (Theorem 3.9):

Theorem 2. *Assume that (C1') and (C2)-(C4) hold true. Then*

$$\kappa\left(M_{AS}^{-1} A\right) \;\lesssim\; \pi(\alpha)\, \gamma_2(1) \left(1 + \max_{i=1}^{s} \frac{H(\Omega_i)}{\delta_i} \right) \;+\; \gamma_2(\alpha)\ .$$

where $H(\Omega_i) = \max_{\{K\,:\,K \cap \bar{\Omega}_i \neq \emptyset\}} H_K$ is the local coarse mesh diameter.

Note that, if in addition we assume (C1), this bound does not reduce to the bound in Theorem 1. The results of Theorems 1 and 2 and the ways in which they are proved are genuinely different. Since in either case a slightly different set of robustness indicators is involved they provide two separate tools by which to establish the robustness of a particular coarse space. We will discuss this in more detail below.

Example 4 (Multiscale Finite Element Coarsening II). By definition the multiscale basis functions Φ_j constructed in Example 3 also satisfy (C1'). Therefore, if the covering $\{\Omega_i\}$ is chosen such that (C4) is satisfied, then Theorem 2 applies. As in [10] in the case of aggregation-based coarsening, it is shown in [6] (under some technical assumptions) that $\gamma_2(\alpha)$ can be bounded independently of $\max_{x \in \Omega} \alpha(x)$. Moreover, the numerical experiments in [6] show that these bounds are sharp and that the new preconditioner has greatly improved performance over standard preconditioners even in the random coefficient case.

Finally, to compare the bounds in Theorems 1 and 2 in the case of multiscale coarsening, let $\Omega_j = \omega_j$, for $j = 1, \ldots, N_H$. This implies that $\delta_j \sim H(\Omega_j)$ and so

$$\kappa\left(M_{AS}^{-1}A\right) \lesssim \pi(\alpha)\,\gamma_2(1) + \gamma_2(\alpha). \tag{7}$$

However, in this case (C1) also holds true and we can apply Theorem 2 to obtain $\kappa\left(M_{AS}^{-1}A\right) \lesssim \gamma_\infty(\alpha)$ (cf. (6)). It is not clear which of the two bounds in (6) and in (7) is sharper. Since the Φ_j form a partition of unity subordinate to the covering $\{\Omega_j\}$, we could bound $\pi(\alpha)$ by $\gamma_\infty(\alpha)$ and apply a trivial bound to $\gamma_2(\alpha)$ to obtain

$$\kappa\left(M_{AS}^{-1}A\right) \lesssim \gamma_2(1)\,\gamma_\infty(\alpha). \tag{8}$$

This would suggest that the bound in (6) is sharper than the one in (7). However, the inequalities which we used to obtain (8) from (7) are known to be not sharp in general, leaving open the possibility that (7) may be sharper for a particular choice of α.

Linear algebra aspects of multiscale coarsening which also reveal a link to iterative substructuring are considered in [7]. Extensions of the methods and the theory to multiplicative, hybrid and deflation variants are also in [6, 7].

References

[1] J. Aarnes and T.Y. Hou. Multiscale domain decomposition methods for elliptic problems with high aspect ratios. *Acta Mathematica Applicatae Sinica (English Ser.)*, 18:63–76, 2002.

[2] M. Brezina and P. Vanek. A black-box iterative solver based on a two-level Schwarz method. *Computing*, 63:233–263, 1999.

[3] L.M. Carvalho, L. Giraud, and P. Le Tallec. Algebraic two-level preconditioners for the Schur complement method. *SIAM J. Sci. Comput.*, 22:1987–2005, 2001.

[4] T.F. Chan and T.P. Mathew. Domain decomposition algorithms. In *Acta Numerica, 1994*, pages 61–143. Cambridge Univ. Press, Cambridge, 1994.

[5] L. Giraud, F. Guevara Vasquez, and R.S. Tuminaro. Grid transfer operators for highly-variable coefficient problems in two-level non-overlapping domain decomposition methods. *Numer. Linear Algebra Appl.*, 10:467–484, 2003.

[6] I.G. Graham, P.O. Lechner, and R. Scheichl. Domain decomposition for multiscale PDEs. Technical Report 11/06, BICS, 2006. Submitted. Available electronically at http://www.bath.ac.uk/math-sci/BICS/.

[7] I.G. Graham and R. Scheichl. Robust domain decomposition algorithms for multiscale PDEs. Technical Report 14/06, BICS, 2006. Submitted. Available electronically at http://www.bath.ac.uk/math-sci/BICS/.

[8] T.Y. Hou and X.-H. Wu. A multiscale finite element method for elliptic problems in composite materials and porous media. *J. Comput. Phys.*, 134:169–189, 1997.

[9] J.E. Jones and P.S. Vassilevski. AMGe based on element agglomeration. *SIAM J. Sci. Comput.*, 23:109–133, 2001.

[10] R. Scheichl and E. Vainikko. Additive Schwarz and aggregation-based coarsening for elliptic problems with highly variable coefficients. Technical Report 9/06, BICS, 2006. Submitted. Available electronically at http://www.bath.ac.uk/math-sci/BICS/.

[11] R. Scheichl and E. Vainikko. Robust aggregation-based coarsening for additive Schwarz in the case of highly variable coefficients. In J. Périaux P. Wesseling, E. Onate, editor, *Proc. European Conference on Computational Fluid Dynamics, ECCOMAS CFD 2006*, 2006.

[12] A. Toselli and O.B. Widlund. *Domain Decomposition Methods–Algorithms and Theory*. Springer-Verlag, Berlin, 2005.

[13] P. Vanek, J. Mandel, and M. Brezina. Algebraic multigrid by smoothed aggregation for 2nd and 4th order elliptic problems. *Computing*, 56:179–196, 1996.

[14] W.L. Wan, T.F. Chan, and B. Smith. An energy-minimizing interpolation for robust multigrid methods. *SIAM J. Sci. Comput.*, 21:1632–1649, 2000.

[15] J. Xu and L. Zikatanov. On an energy minimizing basis for algebraic multigrid methods. *Comput. Visual. Sci.*, 7:121–127, 2004.

Robust Norm Equivalencies and Preconditioning

Karl Scherer

Institut für Angewandte Mathematik, University of Bonn, Wegelerstr. 6, 53115 Bonn, Germany

Summary. In this contribution we report on work done in continuation of [1, 2] where *additive multilevel methods* for the construction of preconditioners for the stiffness matrix of the Ritz- Galerkin procedure were considered with emphasis on the model problem $-\nabla\omega\nabla u = f$ with a scalar weight ω.

We present an new approach leading to a preconditioner based on a modification of the construction in [4] using weighted scalar products thereby improving that one in [2]. Further we prove an upper bound in the underlying norm equivalencies which is up to a fixed level completely independent of the weight ω, whereas the lower bound involves an assumption about the local variation the coefficient function which is still weaker than in [1]. More details will be presented in a forthcoming paper.

1 Preliminaries

1.1 Ritz -Galerkin-Method

Let Ω be a bounded domain in $\mathbb{R}^2$ and $H_0^1(\Omega) = Y$ be the Hilbert space defined as the closure of $C_0^\infty(\Omega)$ with respect to the usual Sobolev norm. Further let A be an elliptic operator defined on $H_0^1(\Omega)$ with an associated coercive and symmetric bilinear form $a(u,v)$. The *Lax-Milgram Theorem* guarantees then a unique solution $u \in Y$ of

$$a(u,v) = (f,v) := \int_\Omega f \cdot v \; dx, \qquad \forall v \in Y,$$

for any $f \in L_2(\Omega)$. Define the *Ritz -Galerkin approximation* $u_h \in \mathcal{V}_h \subset Y$ by

$$a(u_h,v) = (f,v), \qquad \forall v \in \mathcal{V}_h.$$

If $\psi_1, \cdots, \psi_N$ is a basis of $\mathcal{V}_h$, u_h is obtained by the equations:

$$\sum_{i=1}^N \alpha_i \; a(\psi_i,\psi_k) = (f,\psi_k), \quad 1 \le k \le N, \qquad u_h := \sum_{i=1}^N \alpha_i\psi_i.$$

These equations are solved *iteratively* in the form

$$u^{(\nu+1)} = u^{(\nu)} - \omega\ \mathcal{C}r^{(\nu)}, \quad \nu = 0, 1, 2, \cdots \tag{1}$$

where $r^{(\nu)} := \mathcal{A}u^{(\nu)} - b$ with *stiffness matrix* $\mathcal{A} \equiv \mathcal{A}_\psi := \Big(a(\psi_i, \psi_k)\Big)_{i,k}$ and $b := \{(f, \psi_k)\}$. Further ω denotes a relaxation factor and $\mathcal{C}$ a *preconditioner matrix.* The goal is to achieve $\kappa(\mathcal{CA}) \ll \kappa(\mathcal{A})$ or at least of order $\mathcal{O}(1)$ independent of N.

A basic fact is: If $\mathcal{C}$ is the matrix associated to operator $C : \mathcal{V} \to \mathcal{V}$ satisfying

$$\gamma\ (u\ ,\ C^{-1}\ u)\ \leq a(u,u)\ \leq\ \Gamma\ (u\ ,\ C^{-1}\ u), \quad u \in \mathcal{V}, \tag{2}$$

then $\kappa(\mathcal{CA}) \leq \Gamma/\gamma$. Thus $\mathcal{C}$ can be taken as a discrete analogue of C or an approximative inverse of $B = C^{-1}$.

In the theory of *Additive Multi-level-Methods* an approach to construct the bilinear form with associated B is to assume a hierarchical sequence of subspaces

$$\mathcal{V}_0 \subset \mathcal{V}_1 \subset \cdots \subset \mathcal{V}_J := \mathcal{V}_h\ \subset Y\ \subset X := L_2(\Omega), \tag{3}$$

and construct *bounded linear projections* $Q_j : \mathcal{V} \longrightarrow \mathcal{V}_j$ with

$$\beta_0\ a(u,u) \leq \sum_{j=0}^{J} d_j^2\ ||Q_j u - Q_{j-1}u||_X^2 \leq \beta_1\ a(u,u), \tag{4}$$

with $Q_0 u := 0$ and suitable coefficients $\{d_j\}$. β_0, β_1 are constants not depending on the d_j, $u \in \mathcal{V}_h$ or J.
Then define the positive definite operator $B = C^{-1}$ by

$$(u\ ,\ B\ u) := \sum_{j=1}^{J} d_j^2\ ||Q_j u - Q_{j-1}u||_X^2, \qquad u \in \mathcal{V}. \tag{5}$$

2 A Diffusion Problem as a Model Problem

2.1 Spectral Equivalencies

Let $\mathcal{T}_0$ be an initial coarse triangulation of a region $\Omega \subset \mathbb{R}^2$. Regular refinement of triangles leads to triangulations $\mathcal{T}_0 \subset \mathcal{T}_1 \subset \cdots \subset \mathcal{T}_J = \mathcal{T}$.

Each triangle in $\mathcal{T}_k$ is geometrically similar to a triangle of $\mathcal{T}_0$. We define then the $\{\mathcal{V}_j\}_{j=1}^J$ in (3) as spaces of piecewise linear functions with respect to these triangulations. Also its elements have to satisfy Dirichlet boundary conditions. In particular there exists a nodal basis $\psi_k^{(j)}$ for $\mathcal{V}_j = \text{span}\{\psi_k\}$.

In the following we consider the model problem

$$a(u,v) := \int_\Omega \omega\ (\nabla u, \nabla v) \tag{6}$$

which correspond s to the differential operator $A = \nabla \cdot \omega \nabla$.
Observe that in case $u, v \in \mathcal{V}_j,\ j = 0, 1, \cdots, J$ the bilinear form $a(u,v)$ reduces in view of $\nabla v = \text{const.}$ on $T \in \mathcal{T}_j$ to

$$a(u,v) = a_j(u,v) := \sum_{T\in\mathcal{T}_j} \omega_T \int_T (\nabla u, \nabla v) \tag{7}$$

with average weights

$$\omega_T := \frac{1}{\mu(T)} \int_T \omega dx, \qquad \mu(T) \ = \ \text{area of } T.$$

This leads to *weighted norms*

$$\|v\|_{j,\omega}^2 := \sum_{T\in\mathcal{T}_j} \omega_T \int_T |v|^2, \qquad \|v\|_\omega := \|v\|_{J,\omega} \,. \tag{8}$$

Instead of the usual orthogonal projections $Q_j : \mathcal{V} \to \mathcal{V}_j$ we define now in contrast to [1] operators $Q_j^\omega : \mathcal{V} \to \mathcal{V}_j$ with *level-depending weights* by

$$(Q_j^\omega u, v)_{j,\omega} = (u,v), \qquad \forall v \in \mathcal{V}_j \tag{9}$$

and $A_j^\omega : \mathcal{V}_j \to \mathcal{V}_j$ for $u \in \mathcal{V}_j$ by

$$(A_j^\omega u, v)_{j,\omega} = a(u,v), \qquad \forall v \in \mathcal{V}_j. \tag{10}$$

Then the following modification of a well-known result in the theory of multilevel methods (cf. surveys [3, 5] of J. Xu and H. Yserentant) can be proved.

Theorem 1. *Suppose that there exists a decomposition $u = \sum_{k=0}^J u_k$ for $u \in \mathcal{V}$ with $u_k \in \mathcal{V}_k$ and positive definite operators $B_k^\omega : \mathcal{V}_k \to \mathcal{V}_k$ satisfying*

$$\sum_{k=0}^J (B_k^\omega u_k, u_k)_{k,\omega} \le K_1 \ a(u,u), \tag{11}$$

then $C^\omega := \sum_{k=0}^J (B_k^\omega)^{-1} Q_k^\omega$ satisfies $\lambda_{\min}\ (C^\omega A) \ge 1/K_1$.
If the operators B_k^ω further satisfy

$$a(\sum_{k=0}^J w_k, \sum_{l=0}^J w_l) \ \le \ K_2 \ \sum_{k=0}^J (B_k^\omega w_k, w_k)_{k,\omega}, \qquad w_k := (B_k^\omega)^{-1} Q_k^\omega A u \tag{12}$$

then $\lambda_{\max}\ (C^\omega A) \le K_2$, i.e. the operator C^ω is spectrally equivalent to A.

The proof will be given in a forthcoming paper by M. Griebel and M.A. Schweitzer.

For the diffusion problem (6) we can choose the operator B_k^ω now simply as $B_k u_k := 4^k u_k$ for $u_k \in \mathcal{V}_k$, hence

$$C\ u := \sum_{k=0}^J 4^{-k} Q_k^\omega u. \tag{13}$$

This has several advantages over the approach in [1] which uses direct norm equivalencies like in (4). For spectral equivalence of C with A one needs to prove the upper inequality (11) in the form

$$\sum_{k=0}^{J} 4^k (u_k, u_k)_{k,\omega} \le K_1 \; a(u,u), \tag{14}$$

only for *some* decomposition $u = \sum_{k=0}^{J} u_k$. However (12) has to be verified in the form

$$a(\sum_{k=0}^{J} w_k, \sum_{l=0}^{J} w_l) \;\le\; K_2 \; \sum_{k=0}^{J} 4^k (w_k, w_k)_{k,\omega}, \tag{15}$$

for *any* decomposition $v = \sum_{k=0}^{J} w_k, w_k \in \mathcal{V}_k$.
These *weighted Jackson- and Bernstein inequalities* will be verified in the next sections in a *robust form*, i.e. the constants depend only weakly from the diffusion coefficient ω.

Another advantage of the above theorem is that (13) leads to a practical form for the preconditioning matrix $\mathcal{C}$ in (1), namely one shows that the operator C above is spectrally equivalent to the operator

$$\tilde{C} := \sum_{k=0}^{J} 4^{-k} M_k^\omega, \qquad M_k^\omega v := \sum_{i \in \mathcal{N}_k} \frac{(v, \psi_i^{(k)})}{(1, \psi_i^{(k)})_{k,\omega}} \psi_i^{(k)}$$

where $\{\psi_i^{(k)}\}_{i=1}^{N_k}$ denotes the nodal basis of $\mathcal{V}_k$ for $k \ge 1$. Thus the operators $M_k^\omega u$ replace the operators Q_k^ω defined as in (9). The reason for this is that one can show (up to an absolute constant)

$$(Q_k^\omega u, u) \approx \sum_{i \in \mathcal{N}_k} \frac{(u, \psi_i^{(k)})^2}{(1, \psi_i^{(k)})_{k,\omega}} = \left(u, \sum_{i \in \mathcal{N}_k} \frac{(u, \psi_i^{(k)}) \psi_i^{(k)}}{(1, \psi_i^{(k)})_{k,\omega}} \right) = (M_k^\omega u, u).$$

Details as well as the realization of this conditioner in optimal complexity will be presented in the forthcoming paper by M. Griebel and M.A. Schweitzer.

We remark that it can be modified still further to obtain a preconditioner

$$\hat{C} u := \sum_{k=0}^{J} \sum_{i \in \mathcal{N}_k} \frac{(u, \psi_i^{(k)})}{a(\psi_i^{(k)}, \psi_i^{(k)})} \psi_i^{(k)}.$$

2.2 A Weighted Bernstein-Type Inequality

According to (15) we consider here arbitrary decompositions

$$u = \sum_{k=0}^{J} w_k, \qquad w_k \in \mathcal{V}_k \tag{16}$$

of an element $u \in \mathcal{V}_J$. In the following we employ the $a-$orthogonal operators $Q_j^a : \mathcal{V}_J \to \mathcal{V}_j$ defined by

$$a(Q_j^a u, v) = a(u, v), \qquad u \in \mathcal{V}_J, v \in \mathcal{V}_j,$$

so that the elements $v_j := Q_j^a u - Q_{j-1}^a u, v_0 := Q_0^a u$ satisfy

$$u = \sum_{=0}^{J} v_j, \qquad a(v_k, v_j) = \delta_{j,k}, \qquad a(u,u) = \sum_{=0}^{J} a(v_j, v_j). \tag{17}$$

We introduce then the following assumption on the weight ω : there exists a constant C_ω independent of j and a number $\rho < 2$ such that for all $T \in \mathcal{T}_j$

$$\sup\{\omega_U/\omega_T : U \in \mathcal{T}_k, U \subset T\} \;\leq\; C_\omega \; \rho^{k-j}, \qquad j \leq k. \tag{18}$$

Lemma 1. *Under the above assumption on the weight ω there holds the "hybrid" Bernstein type inequality*

$$\|v_j\|_a \leq \; 6\sqrt{6}\; C_1\sqrt{C_2 C_\omega}\; (2/\rho)^{j/2} \sum_{k=j}^{J} (2\rho)^{k/2} \; \|w_k\|_{k,\omega} \tag{19}$$

where $C_1 := \max_{T\in\mathcal{T}_0} \operatorname{diam}(T) \geq \max_{T\in\mathcal{T}_0} \sqrt{\mu(T)}$, and
$C_2 := \max_{T\in\mathcal{T}_0} \operatorname{diam}(T)/\sqrt{\mu(T)}$ are constants which depend on the initial triangulation $\mathcal{T}_0$ only.

Proof. In view of the representation $u = \sum_{k=0}^{j-1} w_k$ we have by (17)

$$a(v_j, v_j) = a(v_j, u) = \sum_{k=j}^{J} a(v_j, w_k)\,. \tag{20}$$

By integration by parts we obtain, keeping in mind that $\nabla w_k, \nabla v_j$ are constant on $U \in \mathcal{T}_k$ and $T \in \mathcal{T}_j$, respectively,

$$\begin{aligned} a(v_j, w_k) &= \sum_{U\in\mathcal{T}_k} \omega_U \int_U (\nabla v_j, \nabla w_k) = \sum_{U\in\mathcal{T}_k} \omega_U \int_{\partial U} w_k(\nabla v_j, n_{\partial U}) \\ &= \sum_{T\in\mathcal{T}_j} \sum_{U\subset T} \omega_U \int_{\partial U} w_k(\nabla v_j, n_{\partial U}) = \sum_{T\in\mathcal{T}_j} \sum_{U\in S_k(T)} \omega_U \int_{\partial U} w_k(\nabla v_j, n_{\partial U}), \end{aligned}$$

where $S_k(T)$ denotes the boundary strip along ∂T consisting of triangles $U \in \mathcal{T}_k, U \subset T$. Applying the Cauchy-Schwarz inequality gives

$$a(v_j, w_k) \leq \Big(\sum_{T\in\mathcal{T}_j} \sum_{U\in S_k(T)} \omega_U \int_{\partial U} |w_k|^2 \Big)^{1/2} \Big(\sum_{T\in\mathcal{T}_j} \sum_{U\in S_k(T)} \omega_U \int_{\partial U} \|\nabla v_j\|^2 \Big)^{1/2}. \tag{21}$$

Concerning the first double sum we note that

$$\int_{\partial U} |w_k|^2 \;\leq\; \operatorname{diam}(U)[b_1^2 + b_2^2 + b_3^2] \leq 12\; C_2 C_1\; 2^k \int_U |w_k|^2,$$

where we have used $\operatorname{diam}(U) \leq C_2 C_1\; 2^k\, \mu(U)$ and the formula

$$\int_U |w_k|^2 = \frac{\mu(U)}{12}[b_1^2 + b_2^2 + b_3^2 + (b_1 + b_2 + b_3)^2]$$

for linear functions v on U with vertices b_1, b_2, and b_3. It follows that

$$\sum_{T\in\mathcal{T}_j} \omega_T \int_{\partial T} |w_k|^2 \le \sum_{T\in\mathcal{T}_j} \omega_T \sum_{U\in S_k(T)} \int_{\partial U} |w_k|^2 \le 12C_2C_1\; 2^k \|w_k\|^2_{k,\omega}. \tag{22}$$

For the second factor in (21) note that by assumption (19) and by the fact that $\mu(S_k(T))/\mu(T) \le 6\cdot 2^{j-k}$ (cf. [5])

$$\begin{aligned}\sum_{T\in\mathcal{T}_j}\sum_{U\in S_k(T)} \omega_U \int_{\partial U} \|\nabla v_j\|^2 &\le C_\omega\; \rho^{k-j} \sum_{T\in\mathcal{T}_j}\sum_{U\in S_k(T)} \omega_T \int_{\partial U} \|\nabla v_j\|^2 \\ &\le 3C_1 2^k\; C_\omega\; \rho^{k-j} \sum_{T\in\mathcal{T}_j}\sum_{U\in S_k(T)} \omega_T \int_U \|\nabla v_j\|^2 \\ &\le 18C_1 2^k\; C_\omega\; (\rho/2)^{k-j} \sum_{T\in\mathcal{T}_j} \omega_T \int_T \|\nabla v_j\|^2\end{aligned}$$

Inserting this and (22) into (21) inequality (19) follows by (20).

With the help of this lemma the Bernstein-type inequality (15) can be established. It improves the corresponding ones in [1, 2] in that assumption (18) is weaker and at the same time more simple than those made there.

Theorem 2. *Consider a sequence of uniformly refined triangulations $\mathcal{T}_j$ and the respective sequence of nested spaces $\mathcal{V}_j$ of linear finite elements. Then, under assumption (18) on the weight ω with $\rho < 2$ in (6) there holds the upper bound*

$$a(u,u) \;\le\; 432C_1^2C_2C_\omega\; \frac{2}{(\sqrt{2}-\sqrt{\rho})^2} \sum_{j=0}^{J} 2^{2j}\; \|w_j\|^2_{j,\omega} \tag{23}$$

for w_j given in (16).

Proof. By summing the estimate (18) according to (17) we get

$$a(u,u) = \sum_{j=0}^{J} \|v_j\|_a^2 \;\le\; 216C_1^2\; C_2C_\omega\; (2/\rho)^j \Big(\sum_{k=j}^{J} (2\rho)^{k/2}\; \|w_k\|_{k,\omega}\Big)^2. \tag{24}$$

From this an upper bound for $a(u,u)$ follows by application of a Hardy inequality to the latter double sum. If quantities s_j, c_j are defined by

$$s_j := \sum_{k=j}^{J} a_k, \quad s_{-1} := 0, \qquad c_j := \sum_{l=0}^{j} b^l, \quad c_{J+1} := 0$$

with $a_k \ge 0$ and $b > 1$ such an inequality reads

$$\Big(\sum_{j=0}^{J} b^j\; s_j^2\Big)^{1/2} \le \frac{\sqrt{b}}{\sqrt{b}-1} \Big(\sum_{j=0}^{J} b^j\; a_j^2\Big)^{1/2}.$$

Application of this with $a_k := (2\rho)^{k/2}\|w_k\|_{k,\omega}$ and $b = 2/\rho$ to yields

$$\sum_{j=0}^{J} (2/\rho)^j \Big(\sum_{k=j}^{J} (2\rho)^{k/2}\; \|w_k\|_{k,\omega}\Big)^2 \le \frac{2}{(\sqrt{2}-\sqrt{\rho})^2} \sum_{j=0}^{J} 2^{2j}\; \|w_j\|^2_{j,\omega}$$

and after insertion into (24) the bound (23) for $a(u,u)$.

2.3 A Weighted Jackson-Type Inequality

The goal here is to establish inequality (11), i.e. to prove

$$\sum_{k=0}^{J} 4^k \|v_k\|_{k,\omega} \leq K_1 \; a(u,u), \qquad u \in \mathcal{V}_J. \tag{25}$$

By Theorem 2.1 we can employ a particular decomposition of u. We choose

$$u = \sum_{j=0}^{J} v_k, \qquad v_k := Q_j^a u - Q_{j-1}^a u \;\text{ as in }\; (17). \tag{26}$$

The *basic idea* is as in [1] to prove a local estimate for $\|v_j\|_{j,\omega}$ on subdomains $U \subset \Omega$ by modifying the duality technique of Aubin-Nitsche. The following result gives an estimate which improves the corresponding one in [1] in that the constant does not depend on the weight ω.

Lemma 2. *Let* $U = \operatorname{supp} \psi_l^{(j-1)}$ *be the support of a nodal function in* $\mathcal{V}_{j-1}$. *There holds*

$$\|v_j\|_{j,\omega,U} \;\leq\; \operatorname{diam}(U) \, \Big(\|\nabla v_j\|_{j,\omega,U} \;+\; 18 C_R \; \|v_j\|_{j,\omega,U} \Big), \tag{27}$$

where C_R *is an absolute constant.*

Proof: We give only a rough idea of it. For triangles $S \in \mathcal{T}_j$ with $T \subset U$ consider the Dirichlet problems

$$-\Delta \phi_S = v_j \qquad \text{on } S, \qquad \phi_S|_{\partial S} = \psi_l^{(j-1)}|_{\partial S}.$$

Then $|v_j|^2 = -v_j \cdot \Delta\phi_S$ on U. Partial integration on each $S \subset U$ gives

$$\|v_j\|_{j,\omega,U}^2 = \left| \sum_{S \subset U} \omega_S \int_S (\nabla\phi_S, \nabla v_j) - \sum_{S \subset U} \omega_S \int_{\partial S} v_j (\nabla \phi_S, n_{\partial S}) \right|.$$

The rest of the proof consists in a careful estimate of both terms on the right hand side. Concerning details we refer again to the forthcoming paper with by M. Griebel and M.A. Schweitzer. We mention only that the constant C_R above arises from the well-known regularity result

$$\|\phi_S\|_{2,2,S} \;\leq\; C_R \; \|v_j\|_{0,S}^2.$$

□

Now by the assumption made on the triangulations there holds $\operatorname{diam}(U) \leq C_0 2^{-j}$ with a constant C_0 depending only on the initial triangulation. Then choose j_0 as the smallest integer with $2^{j_0} = 27\sqrt{3} C_R C_0$ and the second term on the right hand side in (27) is $\leq (2/3)\|v_j\|_{j,\omega,U}$ for all $j \geq j_0$.
If we insert this, square and multiply the resulting inequality with the factor 4^j, the summation with respect to U and $j \geq j_0$ yields

Theorem 3. *There holds the Jackson-type inequality*

$$\sum_{j=j_0}^{J} 4^j \|v_j\|_{j,\omega}^2 \leq 9C_0^2 \sum_{j=j_0}^{J} \sum_{U} a(v_j, v_j)_U \leq \; 9C_0^2 \sum_{j=j_0}^{J} a(v_j, v_j) \tag{28}$$

for $j_0 = \log_2 \; (27\sqrt{3} C_R C_0)$.

If one solves at first the Ritz-Galerkin equations up to level $j_0 - 1$ the preconditioning to the levels $j \geq j_0$ would be robust under condition (18) on the weight ω.

Another possibility would be to establish a bound of the remaining sum on the left hand side up to level $j_0 - 1$. Here one has to use a different argument at the expense of a dependence of the corresponding constant on ω. However one can achieve this under a condition which is weaker than (18).

Corollary 1. *Under the condition (18) on the weight* ω *the discretized version of the operator* $\tilde{C}$ *in (13) yields a robust preconditioning in (1) for the diffusion problem.*

3 Concluding Remarks

The results represented here are concerned with the classical additive multi-level method for solving Ritz-Galerkin equations with piecewise linear elements by preconditioning. The proofs given or indicated here for the necessary norm equivalencies simplify and improve those in [1]. They show that for the diffusion problem a simple modification (13) of the classical preconditioner makes it robust for a large class of diffusion coefficients ω. It covers all piecewise constant functions independent of the location of jumps, their number or their frequency. In particular we do not require the jumps to be aligned with the mesh on any level , i.e. no mesh must resolve the jumps.

However the constants in the Jackson- and Bernstein type inequalities involve the height of the maximal jump. If we assume that $m_\omega := \min_{x \in \Omega} \omega(x) = 1, M_\omega := \max_{x \in \Omega} \omega(x) = \epsilon^{-1}$ a bound for the constant C_ω in assumption (13) is given by ϵ^{-1}. For most practical purposes it is therefore necessary to assume that M_ω is not too big. By the form of (13) one sees that even singularities of maximal height ρ^J and exponential growth limited by ρ are admissible.

References

[1] M. Griebel, K. Scherer, and M.A. Schweitzer. Robust norm-equivalencies for diffusion problems. *Math. Comp.*, 76:1141–1161, 2007.

[2] K. Scherer. Weighted norm-equivalences for preconditioning. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lect. Notes Comput. Sci. Eng.*, pages 405–413. Springer, Berlin, 2005.

[3] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Rev.*, 34(4):581–613, 1992.

[4] H. Yserentant. Two preconditioners based on the multi-level splitting of finite element spaces. *Numer. Math.*, 58(2):163–184, 1990.

[5] H. Yserentant. Old and new convergence proofs for multigrid methods. In *Acta Numerica, 1993*, pages 285–326. Cambridge Univ. Press, Cambridge, 1993.

MINISYMPOSIUM 9: Subspace Correction Methods

Organizers: Ralf Hiptmair[1], Ralf Kornhuber[2], and Jinchao Xu[3]

[1] ETH Zürich, Switzerland. hiptmair@sam.math.ethz.ch
[2] Free University of Berlin, Germany. kornhuber@math.fu-berlin.de
[3] Pennsylvania State University, USA. xu@math.psu.edu

Subspace correction methods are well established as a unifying framework for multigrid and domain decomposition. It also serves as a powerful tool both for the construction and the analysis of efficient iterative solvers for discretized partial differential equations. This minisymposium provides an overview on recent results in this field with special emphasis on linear and nonlinear systems.

Fast and Reliable Pricing of American Options with Local Volatility

Ralf Forster[1]*, Ralf Kornhuber[1], Karin Mautner[2], and Oliver Sander[1]

[1] FU Berlin, Institut für Mathematik, Arnimallee 6, 14195 Berlin, Germany
[2] HU Berlin, Institut für Mathematik, Unter den Linden 6, 10099 Berlin, Germany

Summary. We present globally convergent multigrid methods for the nonsymmetric obstacle problems as arising from the discretization of Black–Scholes models of American options with local volatilities and discrete data. No tuning or regularization parameters occur. Our approach relies on symmetrization by transformation and data recovery by superconvergence.

1 Introduction

Since Black and Scholes published their seminal paper [3] in 1973, the pricing of options by means of deterministic partial differential equations or inequalities has become standard practice in computational finance. An option gives the right (but not the obligation) to buy (call option) or sell (put option) a share for a certain value (exercise price K) at a certain time T (exercise date). On the exercise day T, the value of an option is given by its pay–off function $\varphi(S) = \max(K-S, 0) =: (K-S)_+$ for put options and $\varphi(S) = (S-K)_+$ for call options. In contrast to European options which can only be exercised at the expiration date T, American options can be exercised at any time until expiration. As a consequence, the pay–off function $\varphi(S)$ constitutes an a priori lower bound for the value V of American options which leads to an obstacle problem for V. While Black and Scholes started off with a constant risk–less interest rate and volatility, existence, uniqueness, and discretization is now well understood even for stochastic volatility [1]. On the other hand, an explosive growth of different kinds of equity derivatives on global markets has led to a great variety of well–tuned local volatility models, where the volatility is assumed to be a deterministic (and sometimes even smooth) function of time and space [5, 6]. As such kind of models are used for thousands and thousands of simulations each day, highly efficient and reliable solvers are an ongoing issue in banking practice. Particular difficulties arise from the spatial obstacle problems resulting from implicit time

* This work was supported in part by DFG as Project E7 of Matheon. We gratefully acknowledge the stimulating collaboration with Prof. Dr. P. Deuflhard (ZIB/FU Berlin) and with our project partners Dr. M. Overhaus and Dr. A. Ferraris (DBQuant, Deutsche Bank London).

discretization. The multigrid solver by Brandt and Cryer [4, 15, 16] lacks reliability in terms of a convergence proof and might fail in actual computations. Globally convergent multigrid methods with mesh–independent convergence rates [2, 12] are available for symmetric problems. Such algorithms were applied in [11] after symmetrization of the underlying bilinear form by suitable transformation. However, only constant coefficients were considered there.

In this paper, we present globally convergent multigrid methods for local volatility models with real–life data. To this end, we extend the above 'symmetrization by transformation' approach to variable coefficients. No continuous functions but only discrete market observations are available in banking practice. Therefore, we develop a novel recovery technique based on superconvergence in order to provide sufficiently accurate approximations of the coefficient functions and their derivatives. Finally, we present some numerical computations for an American put option with discrete dividends on a single share.

2 Continuous Problem and Semi–discretization in Time

The Black–Scholes model for the value $V(S,t)$ of an American put option at asset price $S \in \Omega_\infty = [0,\infty)$ and time $t \in [0,T)$ can be written as the following degenerate parabolic complementary problem [1, 5]

$$\begin{aligned} -\tfrac{\partial V}{\partial t} - \tfrac{\sigma^2}{2} S^2 \tfrac{\partial^2 V}{\partial S^2} - \mu S \tfrac{\partial V}{\partial S} + rV \geq 0 \,, \qquad V - \varphi \geq 0 \,, \\ \Big(- \tfrac{\partial V}{\partial t} - \tfrac{\sigma^2}{2} S^2 \tfrac{\partial^2 V}{\partial S^2} - \mu S \tfrac{\partial V}{\partial S} + rV \Big) \Big(V - \varphi \Big) = 0 \,, \end{aligned} \tag{1}$$

in backward time t with stopping condition $V(\cdot, T) = \varphi$ and the pay–off function $\varphi(S) = (K - S)_+$ with exercise price K. The risk–less interest rate $r(t)$, the strictly positive volatility surface $\sigma(S,t)$, and $\mu(t) = r(t) - d(t)$ with continuous dividend yield $d(t)$ are given functions.

Numerical computations require bounded approximations of the unbounded interval Ω_∞. Additional problems result from the degeneracy at $S = 0$. Hence, Ω_∞ is replaced by the bounded interval $\Omega = [S_{\min}, S_{\max}] \subset \Omega_\infty$, $0 < S_{\min} < S_{\max} < \infty$. Appropriate boundary conditions will now be discussed for the example of a put option. Recall that a put option is the right to sell an asset for a fixed price K. If the price of the asset S tends to infinity, the option becomes worthless, because the holder would not like to lose an increasing amount of money by exercising it. Note that $\varphi(S_{\max}) = 0$ for sufficiently large $S_{\max}$. On the other hand, if the asset price tends to zero, then the holder would like to exercise the option almost surely to obtain almost maximal pay–off $\approx K \approx \varphi(S_{\min})$. Hence, we consider the *truncation* of (1) with $S \in \Omega$ and boundary conditions

$$V(S_{\min}) = \varphi(S_{\min}) \,, \quad V(S_{\max}) = \varphi(S_{\max}) \,. \tag{2}$$

Note that the boundary conditions are consistent with the stopping condition $V(T, \cdot) = \varphi$. As $S_{\min} \to 0$, $S_{\max} \to \infty$, the solutions of the resulting truncated problem converge to the solution of the original problem [1].

As usual, we replace backward time t by forward time $\tau = T - t$ to obtain an initial boundary value problem. We now apply a semidiscretization in time by the implicit Euler scheme using the given grid $0 = \tau_0 < \tau_1 < \cdots < \tau_N = T$ with time steps $h_j := \tau_j - \tau_{j-1}$. We introduce the abbreviations $V_j = V(\cdot, \tau_j)$, $\sigma_j = \sigma(\cdot, \tau_j)$, $\mu_j := \mu(\tau_j)$, and $r_j := r(\tau_j)$. Starting with the initial condition $V_0 = \varphi$, the approximation V_j on time level $j = 1, \dots, N$ is obtained from the complementary problem

$$\begin{aligned} -\tfrac{\sigma_j^2}{2} S^2 V_j'' - \mu_j S V_j' + (h_j^{-1} + r_j) V_j - h_j^{-1} V_{j-1} \geq 0 \;, \quad V_j - \varphi \geq 0 \;, \\ \Big(-\tfrac{\sigma_j^2}{2} S^2 V_j'' - \mu_j S V_j' + (h_j^{-1} + r_j) V_j - h_j^{-1} V_{j-1} \Big) \Big(V_j - \varphi \Big) = 0 \;, \end{aligned} \tag{3}$$

on Ω with boundary conditions taken from (2). For convergence results see [1].

3 Symmetrization and Spatial Discretization

We now derive a reformulation of the spatial problem (3) involving a non–degenerate differential operator in divergence form. To this end, we introduce the transformed volatilities and the transformed variables

$$\alpha(x) = \sigma_j(S(x)) \;, \quad u(x) = e^{-\beta(x)} V_j(S(x)) \;, \qquad S(x) = e^x \;, \quad x \in \overline{X} \;, \tag{4}$$

on the interval $X = (x_{\min}, x_{\max})$ with $x_{\min} = \log(S_{\min})$, $x_{\max} = \log(S_{\max})$, utilizing the function

$$\beta(x) = \tfrac{1}{2} x + \log\big(\alpha(x)\big) - \log\big(\alpha(0)\big) - \mu_j \int_0^x \frac{ds}{\alpha^2(s)} \;. \tag{5}$$

Observe that α, β usually vary in each time step.

Theorem 1. *Assume $\sigma_j \in C^2(\overline{\Omega})$ and $\sigma_j(S) \geq c > 0$ for all $S \in \Omega$. Then the linear complementary problem*

$$-(au')' + bu - f \geq 0 \;, \quad u - \psi \geq 0 \;, \quad \big(-(au')' + bu - f \big)\big(u - \psi \big) = 0 \tag{6}$$

with coefficients

$$a = \tfrac{\alpha^2}{2} \;, \qquad b = h_j^{-1} + r_j + \tfrac{1}{8\alpha^2}\big(\alpha^2 - 2\mu_j\big)^2 - \tfrac{\alpha''\alpha^2 + 2\mu_j \alpha'}{2\alpha} \;, \tag{7}$$

right hand side $f = h_j^{-1} e^{-\beta} V_{j-1}(S(\cdot))$, obstacle $\psi = e^{-\beta}\varphi(S(\cdot))$, and boundary conditions $u(x_{\min}) = \psi(x_{\min})$, $u(x_{\max}) = \psi(x_{\max})$ is equivalent to (3) *in the sense that u defined in* (4) *solves* (6), *if and only if V_j solves* (3).

The proof follows from basic calculus. Observe that b might become negative for strongly varying $\alpha(x) = \sigma_j(S(x))$ due to the last term in the definition of b, which could even lead to a stability constraint on the time step h_j. We never encountered such difficulties for realistic data.

For a given spatial grid $x_{\min} = x_0 < x_1 \cdots < x_M = x_{\max}$ the finite element discretization of (6) can be written as the discrete convex minimization problem

$$U = \operatorname*{argmin}_{v \in \mathcal{K}} \int_X \tfrac{1}{2}\big(a(v')^2 + bv^2\big) - fv \; dx \tag{8}$$

with $\mathcal{K}$ denoting the discrete, closed, convex set

$$\mathcal{K} = \{v \in C(X) \mid v|_{[x_{i-1},x_i]} \text{ is linear }, \ v(x_i) \geq \psi(x_i) \ \forall i = 1, \ldots, M, \\ v(x_0) = \psi(x_0), \ v(x_M) = \psi(x_M)\} \ .$$

The fast and reliable solution of (8) can be performed, e.g., by globally convergent multigrid methods [2, 12].

4 Data Recovery

In reality, $r(t)$, $\mu(t)$, and $\sigma(S,t)$ are not available as given functions but have to be calibrated from discrete market observations. To this end, it is common practice in computational finance to introduce sufficient smoothness, e.g., by cubic spline approximation of the local volatility [7, 14] which would suggest even C^4–regularity of $\sigma(S,t)$. We refer to [1] for further information. From now on we assume that the data are given in vectors or matrices of point values of sufficiently smooth functions. The grid points usually have nothing to do with the computational grid.

Intermediate function values can be approximated to second order by piecewise linear interpolation. As our transformation technique also requires $\frac{\partial \sigma}{\partial S}$ and $\frac{\partial^2 \sigma}{\partial S^2}$, we now derive an algorithm for the approximation of higher derivatives by successive linear interpolation in suitable superconvergence points. Note that superconvergence has a long history in finite elements (cf., e.g., [13] and the literature cited therein). For two–dimensional functions such as $\sigma(S,t)$, this recovery technique can be applied separately in both variables.

From now on, let $w_k = w(s_k)$ denote given function values at given grid points $s_0 < s_1 < \cdots < s_K$ with mesh size $h = \max_{k=1,\ldots,K}(s_k - s_{k-1})$. Starting with $s_k^{(0)} = s_k$, we introduce a hierarchy of pivotal points

$$s_k^{(n)} = \frac{s_k + \cdots + s_{k-n}}{n+1} \ , \qquad k = n, \ldots, K \ , \quad n \leq K \ . \tag{9}$$

Note that $s_n^{(n)} < s_{n+1}^{(n)} < \cdots < s_K^{(n)}$ with $s_k^{(n)} \in (s_{k-1}^{(n-1)}, s_k^{(n-1)})$ and

$$0 \leq \max_{k=n+1,\ldots,K} (s_k^{(n)} - s_{k-1}^{(n)}) \leq h \ . \tag{10}$$

In the case of equidistant grids the pivotal points either coincide with grid points (n even) or with midpoints (n uneven). Let

$$L_{k-1}^{(n)}(s) = \frac{s_k^{(n)} - s}{s_k^{(n)} - s_{k-1}^{(n)}} \ , \qquad L_k^{(n)}(s) = \frac{s - s_{k-1}^{(n)}}{s_k^{(n)} - s_{k-1}^{(n)}}$$

denote the linear Lagrange polynomials on the interval $[s_{k-1}^{(n)}, s_k^{(n)}]$. We now introduce piecewise linear approximations p_n of $w^{(n)}$ by successive piecewise interpolation. More precisely, we set

$$p_0(s) = \sum_{j=k-1}^{k} w(s_j) L_j^{(0)}(s) \ , \qquad p_n(s) = \sum_{j=k-1}^{k} p'_{n-1}(s_j^{(n)}) L_j^{(n)}(s) \tag{11}$$

for $s \in [s_{k-1}, s_k]$, $k = 1, \ldots, K$, and $s \in [s_{k-1}^{(n)}, s_k^{(n)}]$, $k = n+1, \ldots, K$, respectively. The approximation p_n can be regarded as the piecewise linear interpolation of divided differences.

Lemma 1. *The derivative p'_{n-1} has the representation*

$$p'_{n-1}(s_k^{(n)}) = n!\; w[s_{k-n},\ldots,s_k]\;, \qquad k = n,\ldots,K\;, \tag{12}$$

where $w[s_{k-n},\ldots,s_k]$denotes the divided differences of w with respect to $s_{k-n},\ldots,s_k$.

Proof. Recall that $s_k^{(n)} \in (s_{k-1}^{(n-1)}, s_k^{(n-1)})$. Using the definitions (9), (11), we immediately get

$$p'_{n-1}(s_k^{(n)}) = \frac{p_{n-1}(s_k^{(n-1)}) - p_{n-1}(s_{k-1}^{(n-1)})}{s_k^{(n-1)} - s_{k-1}^{(n-1)}} = \frac{n\left(p'_{n-2}(s_k^{(n-1)}) - p'_{n-2}(s_{k-1}^{(n-1)})\right)}{s_k - s_{k-n}}$$

so that the assertion follows by straightforward induction.

We are now ready to state the main result of this section.

Theorem 2. *Assume that $w \in C^{n+2}[s_0, s_K]$ and let p_n be defined by* (11)*. Then*

$$\max_{s\in[s_n^{(n)}, s_K^{(n)}]} |w^{(n)}(s) - p_n(s)| \le (n + \tfrac{1}{2})\|w^{(n+2)}\|_\infty\, h^2$$

holds with $\|w^{(n+2)}\|_\infty = \max_{x\in[s_0,s_K]} |w^{(n+2)}(x)|$.

Proof. Let $s \in [s_{k-1}^{(n)}, s_k^{(n)}]$ and denote $\varepsilon_n(s) = w^{(n)}(s) - p'_{n-1}(s)$. Exploiting the linearity of interpolation and a well–known interpolation error estimate (cf., e.g., [8, Theorem 7.16]), we obtain

$$w^{(n)}(s) - p_n(s) = \tfrac{w^{(n+2)}(\zeta)}{2}(s - s_{k-1}^{(n)})(s - s_k^{(n)}) + L_{k-1}^{(n)}(s)\varepsilon_n(s_{k-1}^{(n)}) + L_k^{(n)}(s)\varepsilon_n(s_k^{(n)})$$

with some $\zeta \in (s_{k-1}^{(n)}, s_k^{(n)})$. In the light of (10), it is sufficient to show that $|\varepsilon_n(s_{k-1}^{(n)})| + |\varepsilon_n(s_k^{(n)})| \le n\|w^{(n+2)}\|_\infty h^2$. Utilizing (9) and Lemma 1, we get

$$\varepsilon_n(s_k^{(n)}) = w^{(n)}\left(\frac{1}{n+1}\sum_{i=k-n}^{k} s_i\right) - n!\; w[s_{k-n},\ldots,s_k] =: A - B\;.$$

The Hermite–Genocchi formula (cf., e.g., [8, Theorem 7.12]) yields

$$B = n! \int_{\Sigma^n} w^{(n)}\left(\sum_{i=k-n}^{k} x_i s_i\right)\, dx\;,$$

where Σ^n denotes the n–dimensional unit simplex

$$\Sigma^n = \{x \in \mathbb{R}^{n+1} | \textstyle\sum_{i=0}^n x_i = 1 \text{ and } x_i \ge 0\}\;.$$

As $|\Sigma^n| = 1/n!$, the value A is just the centroid formula for the quadrature of the integral B [9]. It is obtained by simply replacing the integrand by its barycentric value. Using a well–known error estimate [10], we obtain

$$|\varepsilon_n(s_k^{(n)})| \le \frac{\|w^{(n+2)}\|_\infty}{2(n+1)(n+2)} \sum_{i=k-n}^{k} |s_i - s_k^{(n)}|^2\;.$$

Now the assertion follows from the straightforward estimate $|s_i - s_k^{(n)}| \le nh$.

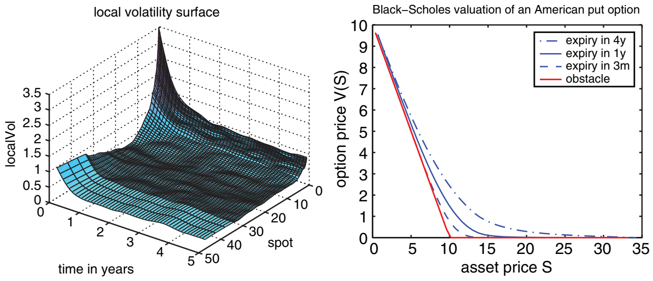


Fig. 1. Local volatility σ and computed values V at time $t = 0$.

In the remaining boundary regions $s \in [s_0, s_n^{(n)}]$ and $s \in [s_K^{(n)}, s_K]$, the function p_n can still be defined according to (11) once a hierarchy of additional pivotal points $s_k^{(n)}$ for $k = 0, \ldots, n-1$ and $k = K+1, \ldots, K+n$ has been selected. However, the approximation in such regions then reduces to first order, unless additional boundary conditions of u at s_0 and s_K are incorporated.

5 Numerical Results

For confidentiality reasons, we consider an American put option on an artificial single share with EURIBOR interest rates, strike price $K = 10$€, and an artificial but typical volatility surface σ as depicted in the left picture of Figure 1 (see also [5, 6]) for the different expiry dates $T = 3/12$, 1, and 4 years. Discrete dividends of $\delta_i = 0.3$ € are paid after $t_i = 4/12, 16/12, 28/12, 40/12$ years. In order to incorporate discrete dividend payments into our model (1), $V(S)$ is replaced by $\tilde{V}(\tilde{S})$, φ, σ are replaced by the shifted functions $\tilde{\varphi}(\tilde{S}) = \varphi(\tilde{S} + D)$, $\tilde{\sigma}(\tilde{S}, \cdot) = \sigma(\tilde{S} + D, \cdot)$ and we set $d = 0$. Here, $D(t)$ is the present value of all dividends yet to be paid until maturity [5, p. 7f.]. We set $\tilde{S}_{\min} = e^{-1}$ and $\tilde{S}_{\max} = e^{3.5}$. Finally, $V(S) = \tilde{V}(S - D)$ is the desired value of the option.

Local volatility data are given on a grid $S_0 = 0.36 < S_1 < \cdots < S_K = 100$. The transformed grid points $x_k = \log(S_k)$ are equidistant for $S_k < 4$, $S_k > 30$ while the original grid points S_k are equidistant for $4 < S_k < 30$ thus reflecting nicely the slope of the volatility surface for small S. To approximate α', α'' occurring in

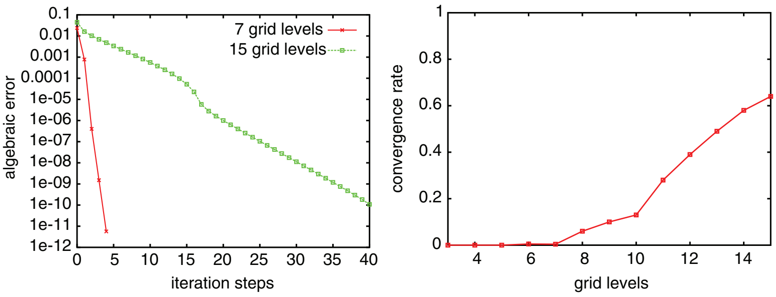


Fig. 2. Iteration history and averaged convergence rates

Theorem 1, we use the recovery procedure (11) with respect to an extension of the hierarchy $S_k^{(2)}$ as defined in (9), though second order accuracy is only guaranteed for $s \in [S_2^{(2)}, S_K^{(2)}]$ (cf. Theorem 2). For the actual data set, the coefficient b is positive and thus the transformed problem (6) is uniquely solvable, if the time steps satisfy $h_j < 0.35$ years. Note that much smaller time steps are required for accuracy reasons.

The transformed interval $\overline{X} = [-1, 3.5]$ is discretized by an equidistant grid with mesh size $H = 1/128 = 2^{-7}$ and we use the uniform time step $h = T/100$ years, for simplicity. Such step sizes are typical as to desired accuracies. The solutions at time $t = 0$ for the different expiry dates are depicted in the right picture of Figure 1. Note that only the options with the long maturity of 1 or 4 years are influenced by dividend payments until expiry date. The spatial problems of the form (6) were solved by truncated monotone multigrid [12] with respect to $J = 7$ grid levels as obtained by uniform coarsening. The initial iterates on time level j were taken from the preceding time level for $j > 1$ and from the obstacle function ψ for $j = 1$. We found that two or three $V(1,1)$ sweeps were sufficient to reduce the algebraic error $\|u_j - u_j^\nu\|$ in the energy norm below 10^{-10}. The corresponding iteration history on the initial time level is shown in the left picture of Figure 2. The iteration history for $H = 1/32768 = 2^{-15}$ and the averaged convergence rates as depicted in the right picture illustrate the convergence behavior for decreasing mesh size.

References

[1] Y. Achdou and O. Pironneau. *Computational Methods for Option Pricing.* SIAM, Philadelphia, 2005.

[2] L. Badea. Convergence rate of a Schwarz multilevel method for the constrained minimization of nonquadratic functionals. *SIAM J. Numer. Anal.*, 44:449–477, 2006.

[3] F. Black and M. Scholes. The pricing of options and corporate liabilities. *J. Polit. Econ.*, 81:637–659, 1973.

[4] A. Brandt and C.W. Cryer. Multigrid algorithms for the solution of linear complementary problems arising from free boundary problems. *SIAM J. Sci. Stat. Comput.*, 4:655–684, 1983.

[5] O. Brockhaus, M. Farkas, A. Ferraris, D. Long, and M. Overhaus. *Equity Derivatives and Market Risk Models.* Risk Books, 2000.

[6] O. Brockhaus, A. Ferraris, C. Gallus, D. Long, R. Martin, and M. Overhaus. *Modelling and Hedging Equity Derivatives.* Risk Books, 1999.

[7] T.F. Coleman, Y. Li, and A. Verma. Reconstructing the unknown local volatility function. *J. Comp. Finance*, 2:77–102, 1999.

[8] P. Deuflhard and A. Hohmann. *Numerical Analysis in Modern Scientific Computing: An Introduction.* Springer, 2003.

[9] I.J. Good and R.A. Gaskins. The centroid method of numerical integration. *Numer. Math.*, 16:343–359, 1971.

[10] A. Guessab and G. Schmeißer. Convexity results and sharp error estimates in approximate multivariate integration. *Math. Comp.*, 73:1365–1384, 2004.

[11] M. Holtz and A. Kunoth. B-spline based monotone multigrid methods. *SIAM J. Numer. Anal.*, to appear.

[12] R. Kornhuber. Monotone multigrid methods for elliptic variational inequalities I. *Numer. Math.*, 69:167 – 184, 1994.

[13] M. Křížek and P. Neittaanmäki. On superconvergence techniques. *Acta Appl. Math.*, 9:175–198, 1987.

[14] R. Lagnado and S. Osher. A technique for calibrating derivative security pricing models: Numerical solution of an inverse problem. *J. Comp. Finance*, 1:13–25, 1997.

[15] C.W. Oosterlee. On multigrid for linear complementarity problems with application to American-style options. *ETNA*, 15:165–185, 2003.

[16] C. Reisinger and G. Wittum. On multigrid for anisotropic equations and variational inequalities "pricing multi–dimensional European and American options". *Comput. Vis. Sci.*, 7:189–197, 2004.

A New Kind of Multilevel Solver for Second Order Steklov-Poincaré Operators

Qiya Hu

LSEC and Institute of Computational Mathematics and Scientific/Engineering Computing, Academy of Mathematics and Systems Science, Chinese Academy of Sciences, Beijing, P.R. China. hqy@lsec.cc.ac.cn

Summary. In this paper we are concerned with the construction of a preconditioner for the Steklov-Poincaré operator arising from a non-overlapping domain decomposition method for second-order elliptic problems in three-dimensional domains. We first propose a new kind of multilevel decomposition of the finite element space on the interface associated with a general quasi-uniform triangulation. Then, we construct a multilevel preconditioner for the underlying Steklov-Poincaré operator. The new multilevel preconditioner enjoys optimal computational complexity, and almost optimal convergence rate.

1 Introduction

The construction of domain decomposition preconditioners has been investigated in various ways in the literature, see, for example, [7]. This kind of preconditioner involves a set of local solvers (Steklov-Poincaré or Poincaré-Steklov operators), which result in dense stiffness matrices. It seems difficult to design cheap inexact solvers (preconditioners) for Steklov-Poincaré operators, unless the underlying triangulation has some particular structures (refer to [8]).

In the present paper, we propose a new kind of multilevel technique for preconditioning Steklov-Poincaré operators. The two main ingredients of this technique are the introduction of a multilevel *domain* decomposition for each local interface, and the construction of a series of coarse solvers associated with such decomposition. One of the main differences between the new method and the traditional multilevel one is that a series of refined grids is unnecessary for the new method (compare [5, 6] and [9]). It will be shown that the new multilevel method has almost optimal convergence and optimal computational complexity.

The new idea advanced in this paper can be extended to some other non-overlapping domain decomposition methods. For example, we can use the new technique to develop a class of substructuring methods with inexact solvers (refer to [4]).

2 Preliminaries

Let Ω be a bounded polyhedron in $\mathbb{R}^3$. Consider the model problem

$$\begin{cases} -div(a\nabla u) = f, \quad in \ \ \Omega, \\ \ \ u = 0, \quad on \ \ \partial\Omega, \end{cases} \tag{1}$$

where the coefficient $a \in L^\infty(\Omega)$ is a positive function.

Let $\mathcal{T}_h = \{\tau_i\}$ be a regular and quasi-uniform triangulation of Ω with $\tau_i's$ being non-overlapping simplexes of size h. The set of nodes of $\mathcal{T}_h$ is denoted by $\mathcal{N}_h$. Then, let $V_h(\Omega)$ be the piecewise linear finite element subspace of $H_0^1(\Omega)$ associated with $\mathcal{T}_h$:

$$V_h(\Omega) = \{v \in H_0^1(\Omega) : \ v|_\tau \in \mathbb{P}_1 \ \ \forall \tau \in \mathcal{T}_h\},$$

where $\mathbb{P}_1$ is the space of linear polynomials. The finite element approximation of (1) is: find $u_h \in V_h(\Omega)$ such that

$$(a\nabla u_h, \ \nabla v_h) = (f, v_h), \quad \forall v_h \in V_h(\Omega). \tag{2}$$

We will apply a non-overlapping domain decomposition method to solve (2). For the ease of notation, we consider only the case with two subdomains (see [4] for the general case).

Let Ω be decomposed into the union of two polyhedrons Ω_1 and Ω_2, which can be written as the union of some elements in $\mathcal{T}_h$, and satisfy $\Omega_1 \cap \Omega_2 = \emptyset$. Without loss of generality, we assume that the coefficient $a(p)$ is a piecewise constant function, and that each subdomain Ω_k is chosen such that $a(p)$ is equal to a constant a_k in Ω_k $(k = 1, 2)$. Set

$$V_h(\Omega_k) = \{v|_{\Omega_k} : \ \forall v \in V_h(\Omega)\} \quad (k = 1, 2).$$

We denote by Γ the common face of Ω_1 and Ω_2 (i.e., $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$), and we define

$$V_h(\Gamma) = \{v|_\Gamma : \ \forall v \in V_h(\Omega)\}.$$

Let $\varphi_h = u_h|_\Gamma$ denote the Dirichlet interface unknown. After eliminating the interior variables from (2), one gets the interface equation (see [7] for the details)

$$S_h\varphi_h = g_h. \tag{3}$$

In the case of two subdomains, the operator S_h is the discrete Steklov-Poincaré operator. It is easy to see that S_h results in a dense stiffness matrix.

In the following, we propose a new technique for preconditioning S_h based on a multilevel domain decomposition for Γ.

3 Multilevel Decompositions for $V_h(\Gamma)$

This section is devoted to establishing a stable multilevel decomposition of $V_h(\Gamma)$ based on a multilevel domain decomposition of Γ.

3.1 Multilevel Decomposition for Γ

The sketch of the multilevel decomposition can be described as follows. We first decompose Γ into the union of several non-overlapping polygons, and then further decompose each resulting polygon into the union of several smaller non-overlapping polygons. We can repeat this process such that each polygon generated by the final decomposition contains only a few nodes.

For convenience, a set of *closed* polygons on the same plane is called *non-overlapping* if the intersection of two neighboring polygons of this set is a common edge or vertex of the two polygons. Let J and m_k $(k = 1, \cdots, J)$ be given positive integers, and set $M_k = m_1 \cdots m_k$, for $k = 1, \cdots, J$.

The first-level decomposition. Decompose Γ into the union of non-overlapping closed polygons $\Gamma_1^{(1)}, \ldots, \Gamma_{m_1}^{(1)}$ in the standard way. We assume that all the polygons $\Gamma_r^{(1)}$ have almost the same "size" d_1.

Successively continuing this procedure, we get a hierarchical decompositions of Γ.

The second-level decomposition. Let each $\Gamma_r^{(1)}$ be further decomposed into the union of m_2 non-overlapping closed sub-polygons of $\Gamma_r^{(1)}$.

The k-level decomposition for $2 \leq k \leq J$. After generating Γ_r^{k-1} at the $(k-1)$-level decomposition, we decompose each $\Gamma_r^{(k-1)}$ into the union of m_k non-overlapping sub-polygons.

Finally, we get the multilevel decomposition for Γ

$$\Gamma = \bigcup_{r=1}^{m_1} \Gamma_r^{(1)} = \bigcup_{r=1}^{M_2} \Gamma_r^{(2)} = \cdots = \bigcup_{r=1}^{M_J} \Gamma_r^{(J)}.$$

For a fixed k, the closed sub-polygons $\Gamma_r^{(k)}$ $(r = 1, \cdots, M_k)$ satisfy the following conditions:

(a) each $\Gamma_r^{(k)}$ has size d_k for some $d_k \in (h, 1)$;

(b) the union of all $\Gamma_r^{(k)}$ $(r = 1, \cdots, M_k)$ constitutes a non-overlapping decomposition for Γ.

Remark 1. Each $\Gamma_r^{(k)}$ may not be the union of some elements of Γ, so the multilevel decomposition described above can be constructed in a simple manner. Note that there is no extra restriction on the triangulation on Γ (in fact the subdivision of the interface Γ does not relate to the triangulation).

3.2 Multilevel Decomposition for $V_h(\Gamma)$

The desired multilevel decomposition involves a set of small local subspaces and a series of coarse subspaces.

Small local subspaces. Let φ_Γ^p denote the nodal basis function of $V_h(\Gamma)$ associated with the node p on Γ. Set

$$V_h(\Gamma_r^{(J)}) = span\{\varphi_\Gamma^p : \ p \in \Gamma_r^{(J)}\} \ \ (r = 1, \cdots, M_J).$$

Coarse subspaces. For convenience, define $M_0 = 1$ and $\Gamma_1^{(0)} = \Gamma$. For $k < J$, let $\mathcal{F}_{\Gamma_r^{(k)}}$, $\mathcal{E}_{\Gamma_r^{(k)}}$ and $\mathcal{V}_{\Gamma_r^{(k)}}$ denote respectively the set of the m_{k+1} sub-polygons, the set of the edges and the set of vertices generated by the $(k+1)$-th level decomposition

$$\Gamma_r^{(k)} = \bigcup_{l=1}^{m_{k+1}} \Gamma^{(k+1)}_{m_{k+1}(r-1)+l} \quad (r = 1, \cdots, \textsc{m}_k).$$

For a sub-polygon $\mathrm{F} \in \mathcal{F}_{\Gamma_r^{(k)}}$, set $\mathrm{F}^{in} = \mathrm{F} \backslash \partial \mathrm{F}$ and define the sub-polygon basis $\varphi_{\mathrm{F}} \in V_h(\Gamma)$ by [1]

$$\varphi_{\mathrm{F}}(p) = \begin{cases} 1, & \text{if } \ p \in \mathrm{F}^{in} \cap \mathcal{N}_h, \\ 0, & \text{if } \ p \in (\Gamma \backslash \mathrm{F}^{in}) \cap \mathcal{N}_h. \end{cases}$$

When an edge $\mathrm{E} \in \mathcal{E}_{\Gamma_r^{(k)}}$ contains some nodes, we define the edge basis $\varphi_{\mathrm{E}} \in V_h(\Gamma)$ by

$$\varphi_{\mathrm{E}}(p) = \begin{cases} 1, & \text{if } \ p \in \mathrm{E} \cap \mathcal{N}_h, \\ 0, & \text{if } \ p \in (\Gamma \backslash \mathrm{E}) \cap \mathcal{N}_h. \end{cases}$$

Similarly, when a vertex $\mathrm{V} \in \mathcal{V}_{\Gamma_r^{(k)}}$ is just a node, we define the vertex basis $\varphi_{\mathrm{V}} \in V_h(\Gamma)$ by

$$\varphi_{\mathrm{V}}(p) = \begin{cases} 1, & \text{if node } \ p = \mathrm{V}, \\ 0, & \text{if node } \ p \neq \mathrm{V}. \end{cases}$$

Now, we define the coarse subspace

$$V_h^0(\Gamma_r^{(k)}) = span\{\varphi_{\mathrm{F}},\ \varphi_{\mathrm{E}},\ \varphi_{\mathrm{V}}:\ \mathrm{F} \in \mathcal{F}_{\Gamma_r^{(k)}},\ \mathrm{E} \in \mathcal{E}_{\Gamma_r^{(k)}},\ \mathrm{V} \in \mathcal{V}_{\Gamma_r^{(k)}}\}$$

$$(k = 0, \cdots, J-1;\ r = 1, \cdots, \textsc{m}_k).$$

Remark 2. In most situations, there is no node on an edge E, and a vertex V is not a node. Then, the coarse subspace reduces to

$$V_h^0(\Gamma_r^{(k)}) = span\{\varphi_{\mathrm{F}}:\ \mathrm{F} \in \mathcal{F}_{\Gamma_r^{(k)}}\}.$$

In such case, we have that $dim(V_h^0(\Gamma_r^{(k)})) = m_{k+1}$.

With the local subspaces and the coarse subspaces defined above, we get the multilevel space decomposition of $V_h(\Gamma)$

$$V_h(\Gamma) = \sum_{k=0}^{J-1} \sum_{r=1}^{M_k} V_h^0(\Gamma_r^{(k)}) + \sum_{r=1}^{M_J} V_h(\Gamma_r^{(J)}).$$

Remark 3. In applications, the above multilevel decomposition would be generated in a suitable manner such that both each local subspace $V_h(\Gamma_r^{(J)})$ and each coarse subspace $V_h^0(\Gamma_r^{(k)})$ have a low dimension.

[1] Thanks to Prof. R. Hiptmair, who told the author that the basis φ_{F} can be also defined using an aggregation framework. Our method seems to be cheaper than the aggregation method (refer to Remark 1).

3.3 Main Result

Let $\langle\cdot,\ \cdot\rangle$ denote the inner product on Γ. For ease of notation, we define

$$\|\varphi_h\|^2_{*,\Gamma} = \langle S_h\varphi_h, \varphi_h\rangle \cong (a_1 + a_2)|\varphi_h|^2_{H^{\frac{1}{2}}_{00}(\Gamma)} \qquad \varphi_h \in V_h(\Gamma).$$

The following result follows from [4].

Theorem 1. *For any $\phi_h \in V_h(\Gamma)$, there exist functions*

$$\phi^{(k)}_{r,\ 0} \in W^0_h(\Gamma^{(k)}_r) \quad (0 \le k \le J-1) \quad \textit{and} \quad \phi^{(J)}_r \in V_h(\Gamma^{(J)}_r)$$

such that

$$\phi_h = \sum_{k=0}^{J-1}\sum_{r=1}^{M_k} \phi^{(k)}_{r\ ,0} + \sum_{r=1}^{M_J} \phi^{(J)}_r \tag{4}$$

and

$$\sum_{k=0}^{J-1}\sum_{r=1}^{M_k} \|\phi^{(k)}_{r\ ,0}\|^2_{*,\Gamma} + \sum_{r=1}^{M_J} \|\phi^{(J)}_r\|^2_{*,\Gamma} \lesssim J[1 + \log(1/h)]^2 \|\phi_h\|^2_{*,\Gamma} \quad (J \ge 1). \tag{5}$$

4 Multilevel Preconditioner for S_h

In this section, we construct a multilevel preconditioner for S_h based on the multilevel decomposition introduced in the previous section.

4.1 Coarse Solvers

We want to consider a coarse solver $M^{(k)}_{r,\ 0}:\ V^0_h(\Gamma^{(k)}_r) \rightarrow V^0_h(\Gamma^{(k)}_r)$ satisfying

$$\langle (M^{(k)}_{r,\ 0})^{-1} S_h\phi_h, S_h\phi_h\rangle \cong \langle \phi_h, S_h\phi_h\rangle, \quad \forall \phi_h \in V^0_h(\Gamma^{(k)}_r).$$

The desired coarse solver can be defined by

$$\begin{aligned}(M^{(k)}_{r,\ 0})^{-1}\phi_h = {} & \frac{1}{\lambda'_k} \sum_{\mathrm{F}\in\mathcal{F}_{\Gamma^{(k)}_r}} \langle \phi_h, \varphi_{\mathrm{F}}\rangle \varphi_{\mathrm{F}} + \sum_{\mathrm{E}\in\mathcal{E}_{\Gamma^{(k)}_r}} \frac{1}{\lambda^k_{\mathrm{E}}} \langle \phi_h, \varphi_{\mathrm{E}}\rangle \varphi_{\mathrm{E}} \\ & + \frac{1}{\lambda''_k} \sum_{\mathrm{V}\in\mathcal{V}_{\Gamma^{(k)}_r}} \langle \phi_h, \varphi_{\mathrm{V}}\rangle \varphi_{\mathrm{V}}, \quad \phi_h \in V^0_h(\Gamma^{(k)}_r).\end{aligned}$$

Here,

$$\lambda'_k = (a_1 + a_2) d_k \log(d_k/h) \cong \langle S_h\varphi_{\mathrm{F}}, \varphi_{\mathrm{F}}\rangle,$$
$$\lambda^k_{\mathrm{E}} = (a_1 + a_2)\|\varphi_{\mathrm{E}}\|^2_{0,\ \mathrm{E}} \cong \langle S_h\varphi_{\mathrm{E}}, \varphi_{\mathrm{E}}\rangle$$

and

$$\lambda''_k = h(a_1 + a_2) \cong \langle S_h\varphi_{\mathrm{V}}, \varphi_{\mathrm{V}}\rangle.$$

4.2 Local Solvers

Inspired by the ideas in [3], we define the inverse of a local solver instead of the local solver itself.

Precisely, let us define the operator

$$K\varphi(q) = \frac{1}{4\pi}\int_{\Gamma} \frac{1}{|p-q|}\varphi(p)dp, \quad q \in \Gamma.$$

Since

$$\langle K\varphi, \varphi\rangle \cong \|\varphi\|^2_{-\frac{1}{2},\Gamma} \quad \forall \varphi \in H^{-\frac{1}{2}}(\Gamma),$$

we choose a local solver $M_r^{(J)} : V_h(\Gamma_r^{(J)}) \rightarrow V_h(\Gamma_r^{(J)})$ such that

$$(M_r^{(J)})^{-1} \cong (a_1 + a_2)^{-1} K|_{V_h(\Gamma_r^{(J)})}.$$

Thus, we can define $(M_r^{(J)})^{-1}$ by

$$\langle (M_r^{(J)})^{-1}\varphi_h, \psi_h\rangle = \frac{1}{4\pi(a_1+a_2)}\int_{\Gamma_r^{(J)}}\int_{\Gamma_r^{(J)}} \frac{\varphi_h(p)\psi_h(q)}{|p-q|}ds(p)ds(q),$$
$$\varphi_h \in V_h(\Gamma_r^{(J)}),\ \forall \psi_h \in V_h(\Gamma_r^{(J)}).$$

The above integrals can be calculated by the formulas introduced in [2]. Since each $\Gamma_r^{(J)}$ contains only a few nodes, it is cheap to calculate the stiffness matrix of $(M_r^{(J)})^{-1}$.

4.3 The Final Preconditioner

As usual, we define the L^2-projectors

$$Q_{r,\ 0}^{(k)} : V_h(\Gamma) \rightarrow V_h^0(\Gamma_r^{(k)}), \quad Q_r^{(J)} : V_h(\Gamma) \rightarrow V_h(\Gamma_r^{(J)}).$$

Then, the desired preconditioner can be defined as follows

$$M_J^{-1} = \sum_{k=0}^{J-1}\sum_{r=1}^{M_k} (M_{r,\ 0}^{(k)})^{-1} Q_{r,\ 0}^{(k)} + \sum_{r=1}^{M_J} (M_r^{(J)})^{-1} Q_r^{(J)}. \tag{6}$$

The following result can be proved as in [1] (by using Theorem 1).

Theorem 2. *Assume that the sequence $\{m_k\}$ is uniformly bounded. Then, we have*

$$cond(M_J^{-1}S_h) \leq CJ^2[1 + \log(1/h)]^2. \tag{7}$$

Hereafter, C is a constant independent of h, of d_k and of the jumps of the coefficient $a(p)$ across the interface.

Remark 4. Our method can be extended to the case of multiple subdomains and interfaces with "crossedges". The two main changes in this extension are that we need to construct a suitable coarse subspace involving the "crossedges", and a multilevel decomposition for each interface (see [4] for the details). For this general case, the term $\log(1/h)$ in (7) would be replaced by $\log(H/h)$, H being the "size" of the subdomains.

Remark 5. We conjecture that the factor J in (7) (and (5)) can be dropped (see the numerical results in Section 6). Unfortunately, we fail to prove this conjecture.

5 Computational Complexity

Let $n_\Gamma = O((1/h)^2)$ be the number of the nodes on Γ, and let $N_\Gamma(J)$ denote the computational complexity for implementing the action of $M^{-1}(J)$.

Proposition 1. *Let $m \geq 2$ be a given positive integer. Set $J = [\log_m n_\Gamma]$, and choose m_k by*

$$m_1 = m_2 = \cdots = m_J = m. \tag{8}$$

Then,

$$N_\Gamma(J) = O(n_\Gamma), \tag{9}$$

which is optimal.

6 Numerical Experiments

Consider the elliptic problem (1) with $\Omega = [0,\ 2] \times [0,\ 1]^2$, and

$$a(x,y,z) = \begin{cases} 10^{-5}, & \text{if } (x,y,z) \in [0,\ 1]^3, \\ 1, & \text{otherwise.} \end{cases}$$

The source function f is chosen in a suitable manner.

Decompose Ω into two cubes with edge length equal to 1, and use the standard $\mathbb{P}_1$ elements on each cube. Finally, decompose each $\Gamma_r^{(k)}$ $(k \leq J-1)$ into four squares with the same size (i.e., $m_k = 4$). We solve the interface equation (3) by PCG iteration with preconditioner M_J^{-1}, considering a tolerance $tol = 10^{-5}$. Some numerical results are reported in table 1.

Table 1. Number of iterations

$1/h$	$J=1$	$J=2$	$J=3$	$J=4$
8	11	11	/	/
16	15	16	15	/
32	19	20	21	20

Table 1 shows that the number of iterations for the new methods depend slightly on the ratio $1/h$ and is independent of the level J.

7 Conclusions

We have introduced a new multilevel preconditioner for Steklov-Poincaré operators. Here, the traditional nested grids are unnecessary. The preconditioner not only features almost optimal convergence, but also optimal computational complexity.

The future works will focus on developing a substructuring method with inexact solvers (almost finished, see [4] for an initial version), and on studying the preconditioning similar operators.

Acknowledgement. Thanks to Dr. Dudu Sun and Dr. Yang Liu for performing the numerical experiments. The work is supported by The Key Project of Natural Science Foundation of China G10531080 and National Basic Research Program of China G2005CB321702.

References

[1] J. Bramble, J. Pasciak, and J. Xu. Parallel multilevel preconditioners. *Math. Comput.*, 55:1–22, 1990.

[2] S. Erichsen and S. Sauter. Efficient automatic quadrature in 3-d Galerkin BEM. *Comput. Methods Appl. Mech. Engrg.*, 157:215–224, 1998.

[3] Q. Hu. Preconditioning Poincaré-Steklov operators arising from domain decompositions with mortar multipliers. *IMA J. Numer. Anal.*, 24:643–669, 2004.

[4] Q. Hu. Non-overlapping domain decomposition methods with a new class of multilevel solvers. Research report, ICMSEC (China), No. ICM-05-22, http://icmsec.cc.ac.cn, 2005.

[5] B. Smith and O. Widlund. A domain decomposition algorithm using hierarchical basis. *SIAM J. Sci. Stat. Comput.*, 11:1212–1220, 1990.

[6] C. Tong, T. Chan, and C. Kuo. A domain decomposition preconditioner based on a change to a multilevel nodal basis. *SIAM J. Sci. Stat. Comput.*, 12:1486–1495, 1991.

[7] A. Toselli and O. Widlund. *Domain Decomposition Methods – Algorithms and Theory.* Springer, Berlin, 2005.

[8] J. Xu and S. Zhang. Preconditioning the Steklov-Poincaré operator by using Green's function. *Math. Comp.*, 66:125–138, 1997.

[9] H. Yserentant. On the multi-level splitting of finite element spaces. *Numer. Math.*, 49:379–412, 1986.

MINISYMPOSIUM 10: Time Domain Decomposition Methods for Evolution Problems

Organizer: Martin J. Gander

Section de Mathématiques, Université de Genève, Switzerland.
martin.gander@math.unige.ch

Time Domain Decomposition methods are methods which decompose the time dimension of an evolution problem into time-subdomains, and then compute the solution trajectory in time simultaneously in all the time subdomains using an iteration. The advent of the parareal algorithm by Lions, Maday and Turinici in 2001 sparked renewed interest in these methods, and there are now several convergence results available for them. In particular, these methods exhibit superlinear convergence on bounded time intervals, a proof of which can be found in the paper of the plenary lecture given by Gander in this volume. While the speedup with parallelization in time is often less impressive than with parallelization in space, parallelization in time is for problems with few spatial components, or when using very many processors, often the only option, if results in real time need to be obtained. This reasoning also led to the name parareal (parallel in real time) of the new algorithm from 2001.

In the first paper, Bal and Wu show that the parareal algorithm applied to a Hamiltonian system is not symplectic, even if the fine and coarse solver in the parareal algorithm are symplectic. They then present a new type of time splitting, replacing the sum of coarse and fine approximate solutions in the parareal scheme by the composition of symplectic coarse and fine approximations. This leads to a symplectic time parallel method, and numerical experiments illustrate the effectiveness of this new approach.

In the second paper, Sarkis, Schaerer and Mathew present a parareal preconditioner for the solution of parabolic problems arising within an optimal control problem. Using results by Gander and Vandewalle, they prove that the parareal preconditioner is spectrally equivalent to the preconditioned problem, and numerical results confirm their analysis.

Symplectic Parareal

Guillaume Bal and Qi Wu

Department of Applied Physics and Applied Mathematics, Columbia University,
New York NY, 10027. {qw2107,gb2030}@columbia.edu

1 Introduction

The parareal algorithm, recalled in section 2 below, allows one to solve evolution equations on (possibly massively) parallel architectures. The two building blocks of the algorithms are a coarse-discretization predictor (solved sequentially) and a fine-discretization corrector (solved in parallel). First developed in [9] and slightly modified in [3], which is the algorithm presented below, the parareal algorithm has received quite a bit of attention lately; see e.g. [1, 2, 4, 5, 6, 7, 10] and their references. Section 3 recalls some results on the parareal algorithm when it is used to solve ordinary and partial differential equations. One of the main shortcomings of the parareal algorithm is that, as a predictor corrector scheme, it may generate high-frequency instabilities.

An area of great potential for the parareal algorithm may thus be the long time evolution of not-too-large systems of ordinary differential equations as they may arise e.g. in molecular dynamics and in the Keplerian problem. The parareal algorithm, however, does not preserve geometric properties such as the symplecticity of the continuous flow of a Hamiltonian system.

We propose in this note a framework to construct a symplectic parareal-type algorithm. The framework is based on the introduction of an interpolating step between the predicting step and the correcting step. The resulting Interpolated Predictor Corrector (IPC) scheme is presented in section 4. We first derive an IPC scheme for arbitrary systems of ordinary differential equations. We then show how the IPC can be rendered symplectic by using the interpolation of appropriate generating functions. Section 5 provides proof of concept by showing numerical simulations for a simple one-dimensional Hamiltonian system.

2 Parareal Algorithm

Let us consider a system of ordinary differential equations of the form

$$\frac{dX}{dt}(t) = b(t, X(t)), \qquad t \in [0, T], \qquad X(0) = X_0. \tag{1}$$

Here $X(t) \in \mathbb{R}^d$ for some finite d. We assume that the above system admits a unique solution. We set a time step $\Delta T > 0$ and a discretization $T^n = n\Delta T$ and introduce the solution operator $g(t,x)$ over the small interval ΔT given by $g(t,X) = X(t+\Delta T)$ when $X(t) = X$. Let now $g_\Delta(t,X)$ be a discretization of g and define the coarse solution

$$X_1^{n+1} = g_\Delta(T^n, X_1^n) \quad \text{for } 0 \le n \le N-1; \qquad X_1^0 = X_0. \tag{2}$$

We introduce the correction operator $\delta g(T^n, X) = g(T^n, X) - g_\Delta(T^n, X)$.

Then we define iteratively the **parareal** approximations

$$X_{k+1}^{n+1} = g_\Delta(T^n, X_{k+1}^n) + \delta g(T^n, X_k^n), \qquad k \ge 1. \tag{3}$$

Note that all the terms $\delta g(T^n, X_k^n)$ for $0 \le n \le N-1$ may be performed in parallel. Let us define the error $\varepsilon_k^n = X_k^n - X(T^n)$. Provided that g_Δ provides a scheme of order m and is such that g_Δ and δg are Lipschitz continuous (see e.g. [2] for the details), we obtain the following estimate:

$$|\varepsilon_k^n| = |X_k^n - X(T^n)| \le C(\Delta T)^{k(m+1)} \binom{n}{k} (1 + |X_0|). \tag{4}$$

For $n = N$ and $k = O(1)$ (the case of interest in practice), we thus obtain:

$$|X_k^N - X(T)| \le CT(\Delta T)^{km}(1 + |X_0|). \tag{5}$$

The iterative scheme (3) replaces a discretization of order m by a discretization of order km after $k-1$ iterations, involving k coarse solutions and $k-1$ fine solutions that can be calculated in parallel.

3 Two Remarks on the Parareal Algorithm

Provided that we seek a final solution at time T with an accuracy of order δt, we have four parameters at our disposal: (i) the coarse time step ΔT; (ii) the number of parareal iterations k; (iii) the number N of successive uses of the parareal scheme over intervals of size $\tau = \frac{T}{N}$ and (iv) the number of available processors P. An analysis of the choices for these parameters that maximize speedup and system efficiency is presented in [2]. The main conclusions are as follows. When the number of available processors is unlimited, i.e., at least of order $(\delta t)^{-1/2}$, then an optimal speedup is attained when ΔT, k, and N are chosen as $\Delta T \approx (\delta T)^{1/2}$, $k = 2$, and $N = 1$. Assuming that the number of processors is smaller and that it takes the form $P = (\delta t)^{-\alpha}$ for some $0 < \alpha < 1/2$, then optimality in the system efficiency (i.e., in the use of all available processors) is achieved provided that the parameters are chosen as $\Delta T \approx (\delta T)^{(1+\alpha)/3}$, $k = 2$, and $N \approx (\delta t)^{-2(1-2\alpha)/2}$.

The parareal algorithm is therefore quite efficient when the number of parareal iterations is $k = 2$, which means that the coarse solver is used twice in a sequential fashion and that the fine solver is used once in parallel. Larger values of k may be beneficial to obtain a better accuracy or to allow for more conservative choices of the (a priori unknown) parameters ΔT and N. Subsequent modifications of the parareal algorithm in this paper implicitly recognize that $k = 2$ is a reasonable choice.

The second remark pertains to the use of the parareal algorithm to solve partial differential equations. Several studies have shown that the parareal algorithm performed well for parabolic equations but showed some instabilities for hyperbolic equations; see e.g. [4, 6]. Analytical calculations performed for simple examples of partial differential equations in [3, 1] provide some explanations for this behavior. In the framework of equations with constant coefficients, we obtain in the Fourier domain the following evolution equation

$$\frac{\partial \hat{u}}{\partial t}(t,\xi) + P(\xi)\hat{u}(t,\xi) = 0,\ \xi \in \mathbb{R},\ t > 0, \qquad \hat{u}(0,\xi) = \hat{u}_0(\xi),\ \xi \in \mathbb{R}. \tag{6}$$

We define $\delta(\xi) = P(\xi)\varDelta T$ and the propagator $g(\delta(\xi)) = e^{-\delta(\xi)}$.

Assume that the symbol $P(\xi)$ is approximated by $P_H(\xi)$ to model spatial discretization and that the time propagator $g(\delta)$ is approximated by $g_\varDelta(\delta_H)$, where $\delta_H(\xi) = P_H(\xi)\varDelta T$. We then define the parareal scheme as:

$$\hat{u}^{n+1}_{k+1}(\xi) = g_\varDelta(\delta_H(\xi))\hat{u}^n_{k+1}(\xi) + \delta g(\xi)\hat{u}^n_k(\xi),\ \ \delta g(\xi) = g(\delta(\xi)) - g_\varDelta(\delta_H(\xi)), \tag{7}$$

with $\hat{u}^0_{k+1}(\xi) = \hat{u}_0(\xi)$ and $\hat{u}^n_0(\xi) \equiv 0$. We verify that we have:

$$\hat{u}^n_{k+1}(\xi) = \sum_{m=0}^{k} \binom{n}{m} (\delta g(\xi))^m g_\varDelta^{n-m}(\delta_H(\xi))\hat{u}_0(\xi). \tag{8}$$

The error term $\varepsilon^n_k(\xi) = \hat{u}^n(\xi) - \hat{u}^n_k(\xi)$ satisfies the following equation: $\varepsilon^{n+1}_{k+1}(\xi) = g_\varDelta(\delta_H(\xi))\varepsilon^n_{k+1}(\xi) + (g(\delta(\xi)) - g_\varDelta(\delta_H(\xi)))\varepsilon^n_k(\xi)$, with boundary conditions $\varepsilon^0_{k+1}(\xi) = 0$ and $\varepsilon^n_0(\xi) = \hat{u}^n(\xi)$. We may prove by induction that:

$$\varepsilon^n_k(\xi) = (\delta g(\xi))^k \sum_{p_1=1}^{n-1} \cdots \sum_{p_{k-1}=1}^{p_{k-2}-1} \sum_{p_k=0}^{p_{k-1}-1} g^{p_k}(\delta) g_\varDelta^{n-p_k-k}(\delta_H)\hat{u}_0(\xi). \tag{9}$$

This provides the following bound for the error estimate

$$|\varepsilon^n_k(\xi)| \lesssim |\delta g(\xi)|^k \binom{n}{k} \sup_p |g_\varDelta|^{n-p-k}(\delta_H)|g|^p(\delta)|\hat{u}_0(\xi)|. \tag{10}$$

The above equation shows a different behavior of the error estimate for low and for high frequencies. For low frequencies, $|\delta g(\xi)|^k$ is small by consistency and the error term $|\varepsilon^n_k(\xi)|$ is of the same order as in (4)-(5). For high frequencies however, all we can expect from $|\delta g(\xi)|^k$ is that it is bounded. The term $\binom{n}{k} \approx n^k$ for $k \ll n$ thus creates instabilities.

The lack of stability of the parareal scheme may be seen in (8). We observe that for $k+1 \geq 2$, the large term $\binom{n}{k} \approx n^k$ can be compensated in three ways: when $|\delta g(\xi)|^k$ is small, which happens for sufficiently small frequencies; when $|g_\varDelta|(\delta_H(\xi))$ is small because the scheme is sufficiently damping at high frequencies; or when $\hat{u}_0(\xi)$ is small because $u_0(x)$ is sufficiently smooth. There are however many schemes $g_\varDelta(\delta_H)$, which are stable, in the sense that u^n_1 remains bounded uniformly in n, and yet which generate unstable parareal schemes; we refer to e.g. [1] for additional details.

4 Interpolated Predictor Corrector Scheme

A reasonable conclusion that can be drawn from what we have seen so far is that the parareal algorithm is adapted to solving small systems of equations over long times. Such systems do not possess instabilities caused by high frequencies and could greatly benefit from the high accuracy obtained by the parareal algorithm. In several practical applications of long term evolutions however, accuracy is not the only constraint. Users may also want their numerical solutions to satisfy some of the geometric constraints that the exact solutions verify. One such geometric constraint is symplecticity in the solution of Hamiltonian evolution equations:

$$\dot{\mathbf{q}} = \nabla_{\mathbf{p}} H(\mathbf{p}, \mathbf{q}), \qquad \dot{\mathbf{p}} = -\nabla_{\mathbf{q}} H(\mathbf{p}, \mathbf{q}), \tag{11}$$

where the symplectic two-form $d\mathbf{p} \wedge d\mathbf{q}$ is preserved by the flow.

It turns out that the parareal algorithm is not symplectic, even when g and g_Δ are symplectic. The reason is that the sum of symplectic operators appearing in (3) is in general not symplectic. In order to make a parallel algorithm such as parareal symplectic, we need to replace the addition of jumps in (3) by compositions of symplectic maps (since composition of symplectic maps is indeed clearly symplectic).

One way to do this gives rise to the following **Interpolated Predictor Corrector** (IPC) scheme. Let us forget about symplectic structures for the moment and consider an arbitrary system of ordinary differential equations such as (1). We still define the coarse predictor X_1^n as the solution of (2). Now instead of viewing the exact propagator as $g = g_\Delta + (g - g_\Delta)$, which is the main ingredient used in the parareal algorithm (3), we consider the following decomposition;

$$g = \psi_\Delta \circ g_\Delta, \qquad \psi_\Delta \equiv g \circ g_\Delta^{-1}. \tag{12}$$

This definition assumes that the approximation of identity g_Δ is indeed invertible on $\mathbb{R}^d$. We suppress explicit time dependency to simplify.

Once X_1^n is calculated sequentially for $n \geq 0$, we can calculate $\psi_\Delta(X_1^{n+1})$ for all $n \geq 0$ with the requested accuracy and in parallel for the sequence $0 \leq n \leq N-1$ provided that N processors are available. In the second step of the predictor-corrector algorithm, we need to be able to evaluate $\psi_\Delta \circ g_\Delta$ at the points X_2^n sequentially. Since $\psi_\Delta \circ g_\Delta$ has only been evaluated at the points X_1^n, an interpolation step is necessary.

Let us assume that the dynamical system has sufficiently smooth trajectories. Then ψ_Δ is a smooth function on $\mathbb{R}^d$. In fact, it is an approximation of Identity of order $(\Delta T)^{m+1}$ if the coarse scheme g_Δ is of order m. The function $\psi_\Delta : \mathbb{R}^d \to \mathbb{R}^d$ can then be approximated by an interpolated function, which we will denote by $\mathcal{I}(\psi_\Delta)$. Such an interpolation is chosen so that $\mathcal{I}(\psi_\Delta)(X_1^n) = \psi_\Delta(X_1^n)$ for all $0 \leq n \leq N-1$.

Once an interpolation $\mathcal{I}(\psi_\Delta)$ is chosen, we *define* the IPC scheme as:

$$X_2^{n+1} = \mathcal{I}(\psi_\Delta) \circ g_\Delta(X_2^n), \quad n \geq 0, \qquad X_2^0 = X_0. \tag{13}$$

See Fig. 1. We obtain the following result.

Theorem 1. *Let us assume that $\mathcal{I}(\psi_\Delta) - \psi_\Delta$ is a Lipschitz function on $\mathbb{R}^d$ with Lipschitz constant of order $(\Delta T)^{M+1}$. Then the IPC scheme is an accurate scheme of order M, so that e.g. $|X(N\Delta T) - X_2^N| \leq CT(\Delta T)^M(1 + |X_0|)$.*

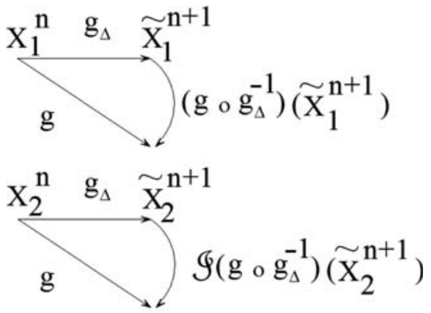

X_1^n
g_Δ
$\tilde{X}_1^{n+1}$
g
$(g \circ g_\Delta^{-1})(\tilde{X}_1^{n+1})$
X_2^n
g_Δ
$\tilde{X}_2^{n+1}$
g
$\mathfrak{I}(g \circ g_\Delta^{-1})(\tilde{X}_2^{n+1})$

Fig. 1. Construction of the IPC scheme

The proof is classical: $\mathcal{I}(\psi_\Delta) \circ g_\Delta$ is consistent with an accuracy of order $(\Delta T)^{M+1}$ while $\mathcal{I}(\psi_\Delta) \circ g_\Delta$ generates a stable, thus convergent, scheme.

The main ingredient in the construction remains to find an appropriate choice for the interpolating operator $\mathcal{I}$. Note however that ψ_Δ is a smooth map of size $O(\Delta T^{m+1})$, which is known at N nearby points along a trajectory. Under sufficient geometric constraints, we may thus hope that polynomial interpolations may converge to the true map ψ_Δ with spectral accuracy in the vicinity of the discrete trajectory X_1^n. What would be the most accurate and least expensive way to obtain this interpolation remains to be investigated. Note that M above is arbitrary and not necessarily of the form $2m$ as for the parareal algorithm with $k = 2$. The two-step IPC scheme can be arbitrarily accurate provided that the flow is sufficiently smooth and the interpolation sufficiently accurate.

Symplectic scheme. We now come back to the original problem of devising a parallel scheme that would preserve the symplecticity of the continuous equations. The operator ψ_Δ constructed above is clearly symplectic as a composition of symplectic maps. The interpolation $\mathcal{I}$ however may not preserve symplecticity if e.g. polynomial approximation is used. In order to construct a symplectic interpolation, we use the concept of *generating function*; see [8].

We now assume that $X = (\mathbf{q}, \mathbf{p}) \in \mathbb{R}^{2n}$ solves an equation of the form (11). Because ψ_Δ is an approximation of identity on $\mathbb{R}^{2d}$, there exists, at least locally [8], a generating function $S(\mathbf{q}^*, \mathbf{p}) = \mathbf{q}^* \cdot \mathbf{p} + \delta(\mathbf{q}^*, \mathbf{p})$, where δ maps a subset in $\mathbb{R}^{2d}$ to $\mathbb{R}$ and where $(\mathbf{q}^*, \mathbf{p}^*) = \psi_\Delta(\mathbf{q}, \mathbf{p})$. We assume here that S and δ are defined globally; an assumption that can be alleviated by appropriate partition of unity of $\mathbb{R}^{2d}$. The maps ψ_Δ and S are then related by the following equations

$$\mathbf{q}^* = \mathbf{q} - \frac{\partial \delta}{\partial \mathbf{p}}(\mathbf{q}^*, \mathbf{p}), \qquad \mathbf{p}^* = \mathbf{p} + \frac{\partial \delta}{\partial \mathbf{q}^*}(\mathbf{q}^*, \mathbf{p}). \tag{14}$$

The coarse scheme provides the set of N points $(g_\Delta(X_1^n), \psi_\Delta(g_\Delta(X_1^n)))$ of the form $((\mathbf{q}, \mathbf{p}), (\mathbf{q}^*, \mathbf{p}^*))$. We find an interpolation $\mathcal{I}(\delta)(\mathbf{q}^*, \mathbf{p})$ of $\delta(\mathbf{q}^*, \mathbf{p})$ so that (14) is exactly satisfied at such a set of points. Owing to (14), the interpolated generating function $\mathcal{I}(\delta)$ now implicitly generates a map on $\mathbb{R}^{2d}$, which we will call $\mathcal{I}(\psi_\Delta)$. This map is by construction symplectic, and provided that the interpolation $\mathcal{I}(\delta)$ of δ is accurate (say of order ΔT^{M+1}), then so is the interpolation $\mathcal{I}(\psi_\Delta)$. We may then apply Theorem 1 and obtain a **symplectic IPC** scheme of order M.

Note that the interpolation of the generating function may be performed locally by appropriate choice of a partition of unity. The interpolated map $\mathcal{I}(\delta)$ would then take the form $\sum_{i\in I}\mathcal{I}_i(\delta_i)\phi_i$ with obvious notation. The astute reader may also have noticed that the symplectic map $\mathcal{I}(\psi_\Delta)$ so constructed depends on the coarse trajectory X_1^n and thus on its seed X_0. When several trajectories are considered, then the interpolations cannot be performed independently if one wants a truly symplectic scheme. One may either perform one interpolation based on all coarse trajectories, or make sure that the interpolation performed on a new trajectory is compatible with the interpolations obtained from previous trajectories. Such complications also arise when the symplectic IPC is restarted in the sense considered in section 3. When the number N of successive uses of the symplectic IPC is greater than 2, then we need to ensure that the interpolations generated at each restart of the algorithm are compatible with each-other.

As we have noted earlier, the optimal way to perform the interpolation step is still research to be done, whether in the framework of symplectic maps or that of more general maps. In the next section, we show proof of concept by considering a one-dimensional Hamiltonian system and a symplectic IPC schemes based on a global interpolation. Such an interpolation is not optimal and may be computationally prohibitively expensive in higher dimensions.

5 Numerical Simulations

We consider the one-dimensional Hamiltonian (pendulum) system (11) with

$$H(q,p) = \frac{1}{2}p^2 + \sin q. \tag{15}$$

We choose a discretization g_Δ which is second-order and symplectic. The $N = 50$ locations of the parareal solution X_2^n presented in section 2 for $1 \le n \le N$ are shown for several choices of the coarse time step $\Delta T = 0.5$, $\Delta T = 0.65$ and $\Delta T = 0.7$, respectively, in Fig. 2 (they correspond to different final times). The fine time step is chosen sufficiently small so that the operator g is estimated almost exactly, also by the second-order symplectic scheme. The parareal solution significantly departs from the surface of constant Hamiltonian for large values of ΔT (as it would for larger times and smaller values of ΔT). This is an indication that the parareal scheme

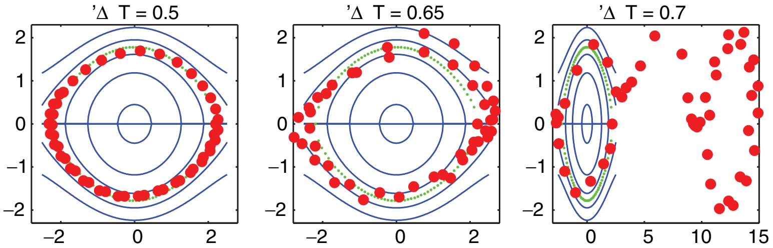


Fig. 2. Parareal solution X_2^n for $1 \le n \le 50$ and $\Delta T = 0.5$, 0.65, and 0.7.

looses the symplectic structure of the flow, and this even though both g and g_Δ are symplectic.

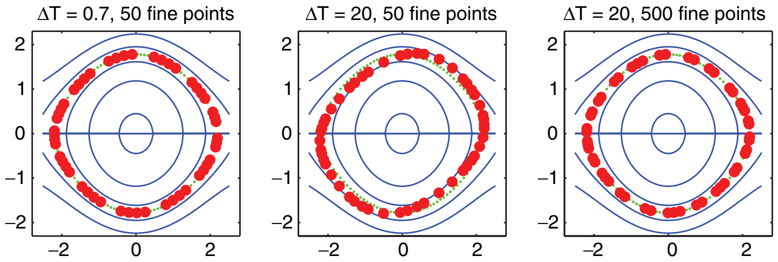


Fig. 3. Symplectic IPC parareal X_2^n for $1 \leq n \leq 50$ and $\Delta T = 0.7$, 20, and 40.

Let now M be the number of discretization points per ΔT for the fine solution operator g. The solution of the IPC scheme X_2^n presented in section 4 is shown in Fig. 3 for values of $(\Delta T, M)$ equal to $(0.7, 50)$, $(20, 50)$, and $(20, 500)$, respectively. The generating function $S(q^*, p)$ is constructed globally on the square $(-2.8, 2.8) \times (-2.3, 2.3)$. Its interpolation is a polynomial of sufficiently high degree so that the $2N$ constraints in (14) generate an under-determined system of linear equations, which is solved by standard least squares. The pseudo-inversion ensures that the resulting interpolation satisfies the constraints exactly and is smooth. The IPC scheme preserves symplecticity independent of ΔT and M. When the fine calculation is not sufficiently accurate (M is too small), ψ_Δ is not estimated accurately and the resulting trajectory may deviate from the true trajectory. With $M = 500$, the estimate of ψ_Δ becomes more accurate and so is its (global) interpolation.

Acknowledgement. This work was supported in part by NSF grant DMS-0239097. GB acknowledges support from an Alfred P. Sloan fellowship.

References

[1] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lect. Notes Comput. Sci. Eng.*, pages 425–432. Springer, Berlin, 2005.

[2] G. Bal. Parallelization in time of (stochastic) ordinary differential equations, 2006. Submitted; `www.columbia.edu/~gb2030/PAPERS/Paralleltime.pdf`.

[3] G. Bal and Y. Maday. A "parareal" time discretization for non-linear PDE's with application to the pricing of an american put. In *Recent Developments in Domain Decomposition Methods (Zürich, 2001)*, volume 23 of *Lect. Notes in Comput. Sci. Eng.*, pages 189–202. Springer, Berlin, 2002.

[4] C. Farhat and M. Chandesris. Time-decomposed parallel time-integrators: theory and feasibility studies for fluid, structure, and fluid-structure applications. *Internat. J. Numer. Methods Engrg.*, 58(9):1397–1434, 2003.

[5] C. Farhat, J. Cortial, C. Dastillung, and H. Bavestrello. Time-parallel implicit integrators for the near-real-time prediction of linear structural dynamic responses. *Internat. J. Numer. Methods Engrg.*, 67(5):697–724, 2006.

[6] P. F. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lect. Notes Comput. Sci. Engrg.*, pages 433–440. Springer, Berlin, 2005.

[7] M. J. Gander and S. Vandewalle. On the superlinear and linear convergence of the parareal algorithm. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lect Notes Comput. Sci. Engrg.*, pages 291–298. Springer, Berlin, 2007.

[8] E. Hairer, L. Christian, and G. Wanner. *Geometric Numerical Integration.* Springer-Verlag, Berlin, 2002.

[9] J.-L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps "pararéel". *C.R. Acad. Sci. Paris Sér. I Math.*, 332(7):661–668, 2000.

[10] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *C.R. Acad. Sci. Paris Sér. I Math.*, 335:387–391, 2002.

Block Diagonal Parareal Preconditioner for Parabolic Optimal Control Problems

Marcus Sarkis[1,2], Christian E. Schaerer[1], and Tarek Mathew[1]

[1] IMPA, Dona Castorina 110, Rio de Janeiro, RJ 22460-320, Brazil.
{msarkis,cschaer}@impa.br
[2] WPI, 100 Institute Road, Worcester, MA 01609, USA.
{tmathew}@poonithara.org

Summary. We describe a block matrix iterative algorithm for solving a linear-quadratic parabolic optimal control problem (OCP) on a finite time interval. We derive a reduced symmetric indefinite linear system involving the control variables and auxiliary variables, and solve it using a preconditioned MINRES iteration, with a symmetric positive definite block diagonal preconditioner based on the parareal algorithm. Theoretical and numerical results show that the preconditioned algorithm converges at a rate independent of the mesh size h, and has parallel scalability.

1 Introduction

Let (t_0, t_f) denote a time interval, let $\Omega \subset \mathbb{R}^2$ be a polygonal domain of size of order $O(1)$ and let $\mathcal{A}$ be a coercive map from a Hilbert space $L^2(t_o, t_f; Y)$ to $L^2(t_o, t_f; Y')$, where $Y = H_0^1(\Omega)$ and $Y' = H^{-1}(\Omega)$, i.e., the dual of Y with respect to the pivot space $H = L^2(\Omega)$; see [2]. Denote the state variable space as $\mathcal{Y} = \{z \in L^2(t_o, t_f; Y) : z_t \in L^2(t_o, t_f; Y')\}$, where it can be shown that $\mathcal{Y} \subset \mathcal{C}^0([t_o, t_f]; H)$; see [2]. Given $y_o \in H$, we consider the following state equation on (t_0, t_f) with $z \in \mathcal{Y}$:

$$\begin{cases} z_t + \mathcal{A}z = \mathcal{B}v \quad \text{for} \quad t_o < t < t_f, \\ \qquad z(0) = y_o. \end{cases} \tag{1}$$

The distributed control v belongs to an admissible space $\mathcal{U} = L^2(t_o, t_f; U)$, where in our application $U = L^2(\Omega)$, and $\mathcal{B}$ is an operator in $\mathcal{L}(\mathcal{U}, L^2(t_o, t_f; H))$. It can be shown that the problem (1) is well posed, see [2], and we indicate the dependence of z on $v \in \mathcal{U}$ using the notation $z(v)$. Given a target function $\hat{y}$ in $L^2(t_o, t_f; H)$ and parameters $q > 0$, $r > 0$, we shall employ the following cost function which we associate with the state equation (1):

$$J(z(v), v) := \frac{q}{2} \int_{t_0}^{t_f} \|z(v)(t, .) - \hat{y}(t, \cdot)\|_{L^2(\Omega)}^2 \, dt + \frac{r}{2} \int_{t_0}^{t_f} \|v(t, \cdot)\|_{L^2(\Omega)}^2 \, dt. \tag{2}$$

For simplicity of presentation, we assume that $y_o \in Y$ and $\hat{y} \in L^2(t_o, t_f; Y)$, and normalize $q = 1$. The optimal control problem for equation (1) consists of finding a controller $u \in \mathcal{U}$ which *minimizes* the cost function (2):

$$J(y,u) \;:=\; \min_{v \in \mathcal{U}} J(z(v),v). \tag{3}$$

Since $q, r > 0$, the optimal control problem (3) is well posed, see [2].

Our presentation is organized as follows: In § 2 we discretize (3) using a finite element method and backward Euler discretization, yielding a large scale saddle point system. In § 3, we introduce and analyze a symmetric positive definite block diagonal preconditioner for the saddle point system, based on the *parareal* algorithm [3]. In § 4, we present numerical results which illustrate the scalability of the algorithm.

2 The Discretization and the Saddle Point System

To discretize the state equation (1) in space, we apply the finite element method to its weak formulation for each fixed $t \in (t_o, t_f)$. We choose a quasi-uniform triangulation $\mathcal{T}_h(\Omega)$ of Ω, and employ the $\mathbb{P}_1$ conforming finite element space $Y_h \subset Y$ for $z(t,\cdot)$, and the $\mathbb{P}_0$ finite element space $U_h \subset U$ for approximating $v(t,\cdot)$. Let $\{\phi_j\}_{j=1}^{\hat{q}}$ and $\{\psi_j\}_{j=1}^{\hat{p}}$ denote the standard basis functions for Y_h and U_h, respectively. Throughout the paper we use the same notation $z \in Y_h$ and $z \in \mathbb{R}^{\hat{q}}$, or $v \in U_h$ and $v \in \mathbb{R}^{\hat{p}}$, to denote both a finite element function in space and its corresponding vector representation. To indicate their time dependence we denote $\underline{z}$ and $\underline{v}$.

A discretization in space of the continuous time linear-quadratic optimal control problem will seek to minimize the following quadratic functional:

$$J_h(\underline{z},\underline{v}) := \frac{1}{2}\int_{t_o}^{t_f} (\underline{z}-\underline{\hat{y}})^T(t) M_h (\underline{z}-\underline{\hat{y}})(t)\,dt + \frac{r}{2}\int_{t_o}^{t_f} \underline{v}^T(t) R_h \underline{v}(t)\,dt \tag{4}$$

subject to the *constraint* that $\underline{z}$ satisfies the discrete equation of state:

$$M_h \dot{\underline{z}} + A_h\, \underline{z} = B_h \underline{v}, \quad \text{for} \;\; t_o < t < t_f; \quad \text{and} \;\; \underline{z}(t_o) = y_o^h. \tag{5}$$

Here $(\underline{z} - \underline{\hat{y}}^h)(t)$ denotes the tracking error, where $\underline{\hat{y}}^h(t)$ and y_0^h belong to Y_h and are approximations of $\underline{\hat{y}}(t)$ and y_o (for instance, use $L^2(\Omega)$-projections into Y_h). The matrices $M_h, A_h \in \mathbb{R}_h^{\hat{q}\times\hat{q}}$, $B_h \in \mathbb{R}^{\hat{q}\times\hat{p}}$ and $R_h \in \mathbb{R}^{\hat{p}\times\hat{p}}$ have entries $(M_h)_{ij} := (\phi_i, \phi_j)$, $(A_h)_{ij} := (\phi_i, \mathcal{A}\phi_j)$, and $(B_h)_{ij} := (\phi_i, \mathcal{B}\psi_j)$ and $(R_h)_{ij} := (\psi_i, \psi_j)$, where $(\cdot,\cdot)$ denotes the $L^2(\Omega)$ inner product.

To obtain a temporal discretization of (4) and (5), we partition $[t_o, t_f]$ into $\hat{l}$ equal sub-intervals with time step size $\tau = (t_f - t_o)/\hat{l}$. We denote $t_l = t_o + l\,\tau$ for $0 \le l \le \hat{l}$. Associated with this partition, we assume that the state variable $\underline{z}$ is continuous in $[t_o, t_f]$ and linear in each sub-interval $[t_{l-1}, t_l]$, $1 \le l \le \hat{l}$ with associated basis functions $\{\vartheta_l\}_{l=0}^{\hat{l}}$. Denoting $z_l \in \mathbb{R}^{\hat{q}}$ as the nodal representation of $\underline{z}(t_l)$ we have $\underline{z}(t) = \sum_{l=0}^{\hat{l}} z_l \vartheta_l(t)$. The control variable $\underline{v}$ is assumed to be a discontinuous function and constant in each sub-interval (t_{l-1}, t_l) with associated basis functions $\{\chi_l\}_{l=1}^{\hat{l}}$. Denoting $v_l \in \mathbb{R}^{\hat{p}}$ as the nodal representation of $\underline{v}(t_l - (\tau/2))$, we have $\underline{v}(t) = \sum_{l=1}^{\hat{l}} v_l \chi_l(t)$.

The corresponding discretization of the expression (4) results in:

$$J_h^{\tau}(\mathbf{z},\mathbf{v}) = \frac{1}{2}\,(\mathbf{z}-\hat{\mathbf{y}})^T \mathbf{K}(\mathbf{z}-\hat{\mathbf{y}}) + \frac{1}{2}\,\mathbf{v}^T \mathbf{G}\mathbf{v} + (\mathbf{z}-\hat{\mathbf{y}})^T \mathbf{g}. \tag{6}$$

The block vectors $\mathbf{z} := [z_1^T, \ldots, z_{\hat{l}}^T]^T \in \mathbb{R}^{\hat{l}\hat{q}}$ and $\mathbf{v} := [v_1^T, \ldots, v_{\hat{l}}^T]^T \in \mathbb{R}^{\hat{l}\hat{p}}$ denote the state and control variables, respectively, at all the discrete times. The discrete target is $\hat{\mathbf{y}} := [\hat{y}_1^T, \ldots, \hat{y}_{\hat{l}}^T]^T \in \mathbb{R}^{\hat{l}\hat{q}}$ with target error $e_l = (z_l - \hat{y}_l^h)$ for $0 \leq l \leq \hat{l}$. Matrix $\mathbf{K} = D_\tau \otimes M_h \in \mathbb{R}^{(\hat{l}\hat{q})\times(\hat{l}\hat{q})}$, where $D_\tau \in \mathbb{R}^{\hat{l}\times\hat{l}}$ has entries $(D_\tau)_{ij} := \int_{t_o}^{t_f} \vartheta_i(t)\vartheta_j(t)dt$, for $1 \leq i, j \leq \hat{l}$, while $\mathbf{G} = r\tau I_{\hat{l}} \otimes R_h \in \mathbb{R}^{(\hat{l}\hat{p})\times(\hat{l}\hat{p})}$, where $\otimes$ stands for the Kronecker product and $I_{\hat{l}} \in \mathbb{R}^{\hat{l}\times\hat{l}}$ is an identity matrix. The vector $\mathbf{g} = (g_1^T, 0^T, \ldots, 0^T)^T$ where $g_1 = \frac{\tau}{6} M_h e_0$. Note that g_1 does not necessarily vanish because it is not assumed that $y_0^h = \hat{y}_0^h$.

Employing the backward Euler discretization of (5) in time, yields:

$$\mathbf{E}\,\mathbf{z} + \mathbf{N}\,\mathbf{v} = \mathbf{f}, \tag{7}$$

where the input vector is $\mathbf{f} := [(M_h y_0^h)^T, 0^T, ..., 0^T]^T \in \mathbb{R}^{\hat{l}\hat{q}}$. The block lower bidiagonal matrix $\mathbf{E} \in \mathbb{R}^{(\hat{l}\hat{q})\times(\hat{l}\hat{q})}$ is given by:

$$\mathbf{E} = \begin{bmatrix} F_h & & & \\ -M_h & F_h & & \\ & \ddots & \ddots & \\ & & -M_h & F_h \end{bmatrix}, \tag{8}$$

where $F_h = (M_h + \tau A_h) \in \mathbb{R}^{\hat{q}\times\hat{q}}$. The block diagonal matrix $\mathbf{N} \in \mathbb{R}^{(\hat{l}\hat{q})\times(\hat{l}\hat{p})}$ is given by $\mathbf{N} = -\tau I_{\hat{l}} \otimes B_h$. The Lagrangian $\mathcal{L}_h(\mathbf{z}, \mathbf{v}, \mathbf{q})$ for minimizing (6) subject to constraint (7) is:

$$\mathcal{L}_h^\tau(\mathbf{z}, \mathbf{v}, \mathbf{q}) = J_h^\tau(\mathbf{z}, \mathbf{v}) + \mathbf{q}^T(\mathbf{E}\mathbf{z} + \mathbf{N}\mathbf{v} - \mathbf{f}). \tag{9}$$

To obtain a discrete saddle point formulation of (9), we apply optimality conditions for $\mathcal{L}_h(\cdot, \cdot, \cdot)$. This yields the symmetric indefinite linear system:

$$\begin{bmatrix} \mathbf{K} & \mathbf{0} & \mathbf{E}^T \\ \mathbf{0} & \mathbf{G} & \mathbf{N}^T \\ \mathbf{E} & \mathbf{N} & \mathbf{0} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{K}\hat{\mathbf{y}} - \mathbf{g} \\ \mathbf{0} \\ \mathbf{f} \end{bmatrix}, \tag{10}$$

where $\hat{\mathbf{y}} := [(\hat{y}_1^h)^T, \ldots, (\hat{y}_{\hat{l}}^h)^T]^T \in \mathbb{R}^{\hat{l}\hat{q}}$. Eliminating $\mathbf{y}$ and $\mathbf{p}$ in (10), and defining $\mathbf{b} := \mathbf{N}^T\mathbf{E}^{-T}\left(\mathbf{K}\mathbf{E}^{-1}\mathbf{f} - \mathbf{K}\hat{\mathbf{y}} + \mathbf{g}\right)$ yields the *reduced* Hessian system:

$$(\mathbf{G} + \mathbf{N}^T\mathbf{E}^{-T}\mathbf{K}\mathbf{E}^{-1}\mathbf{N})\mathbf{u} = \mathbf{b}. \tag{11}$$

The matrix $\mathbf{H} := \mathbf{G} + \mathbf{N}^T\mathbf{E}^{-T}\mathbf{K}\mathbf{E}^{-1}\mathbf{N}$ is symmetric positive definite and $(\mathbf{u}, \mathbf{G}\mathbf{u}) \leq (\mathbf{u}, \mathbf{H}\mathbf{u}) \leq \mu(\mathbf{u}, \mathbf{G}\mathbf{u})$, where $\mu = O(1 + \frac{1}{r})$; for details see [4]. As a result, the Preconditioned Conjugate Gradient method (PCG) can be used to solve (11), but each matrix-vector product with $\mathbf{H}$ requires the solution of two linear systems, one with $\mathbf{E}$ and one with $\mathbf{E}^T$. To avoid double iterations, we define the auxiliary variable $\mathbf{w} := -\mathbf{E}^{-T}\mathbf{K}\mathbf{E}^{-1}\mathbf{N}\mathbf{u}$. Then (11) will be equivalent to the symmetric indefinite system:

$$\begin{bmatrix} \mathbf{E}\mathbf{K}^{-1}\mathbf{E}^T & \mathbf{N} \\ \mathbf{N}^T & -\mathbf{G} \end{bmatrix} \begin{bmatrix} \mathbf{w} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{0} \\ -\mathbf{b} \end{bmatrix}. \tag{12}$$

The system (12) is ill-conditioned and will be solved using the MINRES algorithm with a preconditioner of the form $\mathbf{P} := \text{diag}(\mathbf{E}_n^{-T}\hat{\mathbf{K}}\mathbf{E}_n^{-1}, \mathbf{G}^{-1})$; see [5]. For a fixed number of parareal sweeps n, $\mathbf{E}_n^{-1}$ and $\mathbf{E}_n^{-T}$ are linear operators. We next define the operator $\mathbf{E}_n^{-1}$ and then analyze the spectral equivalence between $\mathbf{E}^{-T}\mathbf{K}\mathbf{E}^{-1}$ and $\mathbf{E}_n^{-T}\hat{\mathbf{K}}\mathbf{E}_n^{-1}$.

3 Parareal Approximation $\mathbf{E}_n^{-T}\hat{\mathbf{K}}\mathbf{E}_n^{-1}$

An application of $\mathbf{E}_n^{-T}\hat{\mathbf{K}}\mathbf{E}_n^{-1}$ to a vector $\mathbf{s} \in \mathbb{R}^{(\hat{l}\hat{q})\times(\hat{l}\hat{q})}$ is performed as follows: Step 1, apply $\mathbf{E}_n^{-1}\mathbf{s} :\rightarrow \hat{\mathbf{z}}^n$ using n applications of the parareal method described below. Step 2, multiply $\hat{\mathbf{K}}\mathbf{z}^n :\rightarrow \hat{\mathbf{t}}$ where $\hat{\mathbf{K}} := \hat{D}_\tau \otimes M_h$, $\hat{D}_\tau :=$ blockdiag$(\hat{D}_\tau^1, \ldots, \hat{D}_\tau^{\hat{k}})$, and the $\hat{D}_\tau^k$ are the time mass matrices associated to the sub-intervals $[T_{k-1}, T_k]$. And Step 3, apply $\mathbf{E}_n^{-T}\hat{\mathbf{t}}^n :\rightarrow \mathbf{x}$, i.e., the transpose of Step 1.

To describe $\mathbf{E}_n$, we partition the time interval $[t_o, t_f]$ into $\hat{k}$ *coarse* sub-intervals of length $\Delta T = (t_f - t_o)/\hat{k}$, setting $T_0 = t_o$ and $T_k = t_o + k\Delta T$ for $1 \le k \le \hat{k}$. We define fine and coarse propagators F and G as follows. The local solution at T_k is defined marching the backward Euler method from T_{k-1} to T_k on the fine triangulation τ with an initial data Z_{k-1} at T_{k-1}. Let $\hat{m} = (T_k - T_{k-1})/\tau$ and $j_{k-1} = \frac{T_{k-1}-T_0}{\tau}$. It it is easy to see that:

$$M_h Z_k = F Z_{k-1} + S_k, \tag{13}$$

where $F := (M_h F_h^{-1})^{\hat{m}} M_h \in \mathbb{R}^{\hat{q}\times\hat{q}}$, $S_k := \sum_{m=1}^{\hat{m}} \left(M_h F_h^{-1}\right)^{\hat{m}-m+1} s_{j_{k-1}+m}$ with $Z_0 = 0$. Imposing the continuity condition at time T_k, for $1 \le k \le \hat{k}$, i.e., $M_h Z_k - F Z_{k-1} - S_k = 0$, we obtain the system:

$$\begin{bmatrix} M_h & & & \\ -F & M_h & & \\ & \ddots & \ddots & \\ & & -F & M_h \end{bmatrix} \begin{bmatrix} Z_1 \\ Z_2 \\ \vdots \\ Z_{\hat{k}} \end{bmatrix} = \begin{bmatrix} S_1 \\ S_2 \\ \vdots \\ S_{\hat{k}} \end{bmatrix}. \tag{14}$$

The coarse solution at T_k with initial data $Z_{k-1} \in \mathbb{R}^{\hat{q}}$ at T_{k-1} is given by one coarse time step of the backward Euler method $M_h Z_k = G Z_{k-1}$ where $G := M_h(M_h + A_h\Delta T)^{-1} M_h \in \mathbb{R}^{\hat{q}\times\hat{q}}$. In the parareal algorithm, the coarse propagator G is used for preconditioning the system (14) via:

$$\begin{bmatrix} Z_1^{n+1} \\ Z_2^{n+1} \\ \vdots \\ Z_{\hat{k}}^{n+1} \end{bmatrix} = \begin{bmatrix} Z_1^{n} \\ Z_2^{n} \\ \vdots \\ Z_{\hat{k}}^{n} \end{bmatrix} + \left(\begin{bmatrix} M_h & & & \\ -G & M_h & & \\ & \ddots & \ddots & \\ & & -G & M_h \end{bmatrix} \right)^{-1} \begin{bmatrix} R_1^{n} \\ R_2^{n} \\ \vdots \\ R_{\hat{k}}^{n} \end{bmatrix}, \tag{15}$$

where the residual vector $\mathbf{R}^n := [R_1^{nT}, ..., R_{\hat{k}}^{nT}]^T \in \mathbb{R}^{\hat{k}\hat{q}}$ is defined in the usual way from the equation (14).

We are now in position to define $\hat{\mathbf{z}}^n := \mathbf{E}_n^{-1}\mathbf{s}$. Let $\hat{\mathbf{z}}^n$ be the nodal representation of a piecewise linear function $\underline{\hat{z}}^n$ in time with respect to the fine triangulation τ on $[t_o, t_f]$, however continuous only inside each coarse sub-interval $[T_{k-1}, T_k]$, i.e., the function $\underline{\hat{z}}^n$ can be discontinuous across the points T_k, $1 \le k \le \hat{k}-1$, therefore, $\hat{\mathbf{z}}^n \in \mathbb{R}^{(\hat{l}+\hat{k}-1)\hat{q}}$. On each sub-interval $[T_{k-1}, T_k]$, $\underline{\hat{z}}^n$ is defined marching the backward Euler method from T_{k-1} to T_k on the fine triangulation τ with initial condition Z_{k-1}^n at T_{k-1}.

Theorem 1. *For any* $\mathbf{s} \in \mathbb{R}^{(\hat{l}\hat{q})\times(\hat{l}\hat{q})}$ *and* $\epsilon \in (0, 1/2)$*, we have:*

$$\gamma_{\min}\left(\mathbf{E}^{-1}\mathbf{s}, \mathbf{K}\mathbf{E}^{-1}\mathbf{s}\right) \le \left(\mathbf{E}_n^{-1}\mathbf{s}, \hat{\mathbf{K}}\mathbf{E}_n^{-1}\mathbf{s}\right) \le \gamma_{\max}\left(\mathbf{E}^{-1}\mathbf{s}, \mathbf{K}\mathbf{E}^{-1}\mathbf{s}\right),$$

$$\textit{where}\begin{cases}\gamma_{\max} := (1+\frac{\rho_n^2(t_f-t_o)}{\tau\epsilon}+2\epsilon)/(1-2\epsilon),\\ \gamma_{\min} := (1-\frac{\rho_n^2(t_f-t_o)}{\tau\epsilon}-2\epsilon)/(1+2\epsilon).\end{cases}$$

Proof. Let $V_h := [v_1, ..., v_{\hat{q}}]$ and $\Lambda_h := \text{diag}\{\lambda_1, ..., \lambda_{\hat{q}}\}$ be the generalized eigenvectors and eigenvalues of A_h with respect to M_h, i.e., $A_h = M_h V_h \Lambda_h V_h^{-1}$. Let $\mathbf{z} := \mathbf{E}^{-1}\mathbf{s}$ with $\underline{z}(t) = \sum_{q=1}^{\hat{q}} \alpha_q(t) v_q$, and $\hat{\mathbf{z}}^n := \mathbf{E}_n^{-1}\mathbf{s}$ with $\underline{\hat{z}}^n(t) = \sum_{q=1}^{\hat{q}} \alpha_q^n(t) v_q$. We note that α_q^n might be discontinuous across the T_k. Then:

$$(\mathbf{E}^{-1}\mathbf{s}, \mathbf{K}\mathbf{E}^{-1}\mathbf{s}) = \|\underline{z}\|^2_{L^2(t_o,t_f;L^2(\Omega))} = \textstyle\sum_{q=1}^{\hat{q}} \|\alpha_q\|^2_{L^2(t_o,t_f)},$$

$$(\mathbf{E}_n^{-1}\mathbf{s}, \hat{\mathbf{K}}\mathbf{E}_n^{-1}\mathbf{s}) = \|\underline{\hat{z}}^n\|^2_{L^2(t_o,t_f;L^2(\Omega))} = \textstyle\sum_{q=1}^{\hat{q}} \|\alpha_q^n\|^2_{L^2(t_o,t_f)},$$

and therefore:

$$\begin{aligned}\|\alpha_q^n\|^2_{L^2(t_o,t_f)} &= \left(\alpha_q^n - \alpha_q, \alpha_q^n + \alpha_q\right)_{L^2(t_o,t_f)} + \|\alpha_q\|^2_{L^2(t_o,t_f)}\\ &\leq \frac{1}{4\epsilon}\|\alpha_q^n - \alpha_q\|^2_{L^2(t_o,t_f)} + \epsilon\|\alpha_q^n + \alpha_q\|^2_{L^2(t_o,t_f)} + \|\alpha_q\|^2_{L^2(t_o,t_f)}\\ &\leq \frac{1}{4\epsilon}\|\alpha_q^n - \alpha_q\|^2_{L^2(t_o,t_f)} + 2\epsilon\|\alpha_q^n\|^2_{L^2(t_o,t_f)} + (1+2\epsilon)\|\alpha_q\|^2_{L^2(t_o,t_f)},\end{aligned}$$

which reduces to:

$$(1-2\epsilon)\|\alpha_q^n\|^2_{L^2(t_o,t_f)} \leq (1+2\epsilon)\|\alpha_q\|^2_{L^2(t_o,t_f)} + \tfrac{1}{4\epsilon}\|\alpha_q^n - \alpha_q\|^2_{L^2(t_o,t_f)}.$$

For each $t_l \in [T_{k-1}, T_k]$ we have:

$$|\alpha_q^n(t_l) - \alpha_q(t_l)| = (1+\tau\lambda_q)^{-(t_l - T_{k-1})/\tau}|\alpha_q^n(T_{k-1}) - \alpha_q(T_{k-1})|,$$

and since $\lambda_q > 0$ implies $(1+\tau\lambda_q)^{-(t_l-T_{k-1})/\tau} \leq 1$, we obtain:

$$\|\alpha_q^n - \alpha_q\|^2_{L^2(T_{k-1},T_k)} \leq \Delta T|\alpha_q^n(T_{k-1}) - \alpha_q(T_{k-1})|^2.$$

Hence:

$$(1-2\epsilon)\|\alpha_q^n\|^2_{L^2(t_o,t_f)} \leq (1+2\epsilon)\|\alpha_q\|^2_{L^2(t_o,t_f)} + \frac{t_f - t_o}{4\epsilon}\max_{0\leq k\leq \hat{k}}|\alpha_q^n(T_k) - \alpha_q(T_k)|^2.$$

Using the Lemma 1 (see below) with $\alpha_q(T_0) = 0$ and initial guess $\alpha_q^0(T_k) = 0$, and using

$$\max_{0\leq k\leq \hat{k}}|\alpha_q(T_k)|^2 = |\alpha_q(T_{k'})|^2 \leq \frac{4}{\tau}\min_{\beta}\|\alpha_q(T_{k'}) + \beta t\|^2_{L^2(T_{k'},T_{k'}+\tau)}$$

we obtain:

$$\max_{0\leq k\leq \hat{k}}|\alpha_q^n(T_k) - \alpha_q(T_k)|^2 \leq \rho_n^2\max_{0\leq k\leq \hat{k}}|\alpha_q(T_k)|^2 \leq \frac{4\rho_n^2}{\tau}\|\alpha_q\|^2_{L^2(t_o,t_f)},$$

and the upper bound (16) follows. The lower bound follows similarly.

Remark 1. Performing straightforward computations we obtain:

$$\min_{\epsilon} \gamma_{\max}(\epsilon) = 1 + \frac{4}{\sqrt{1+\frac{\tau}{\rho_n^2(t_f-t_o)}}-1}.$$

Hence, for small values of ρ_n, we have $\gamma_{\max} - 1 \approx 4\sqrt{\frac{\rho_n^2(t_f-t_o)}{\tau}}$. The dependence of $\gamma_{max} - 1$ with respect to τ is sharp as evidenced in Table 1 (see below) since it increases by a $\sqrt{2}$ factor when τ is refined by half.

Decompose $Z_k = \sum_{q=1}^{\hat{q}} \alpha_q(T_k)v_q$ and $Z_k^n = \sum_{q=1}^{\hat{q}} \alpha_q^n(T_k)v_q$, and denote $\zeta_q^n(T_k) := \alpha_q(T_k) - \alpha_q^n(T_k)$. The convergence of the parareal algorithm for systems follows from the next lemma which it is an extension of the results presented in [1].

Lemma 1. *Let $\Delta T = (t_f - t_o)/\hat{k}$ and $T_k = t_o + k\Delta T$ for $0 \le k \le \hat{k}$. Then,*

$$\max_{1\le k\le \hat{k}} |\alpha_q(T_k) - \alpha_q^n(T_k)| \le \rho_n \max_{1\le k\le \hat{k}} |\alpha_q(T_k) - \alpha_q^0(T_k)|,$$

where $\rho_n := \sup_{0<\beta<1}\left(e^{1-1/\beta} - \beta\right)^n \frac{1}{n!}\left|\frac{d^{n-1}}{d\beta^{n-1}}\left(\frac{1-\beta^{\hat{k}-1}}{1-\beta}\right)\right| \le 0.2984^n.$

Proof. Using Theorem 2 from [1] we obtain:

$$\zeta_q^n = \left((1+\lambda_q\tau)^{-\Delta T/\tau} - \beta_q\right)\mathcal{T}(\beta_q)\zeta_q^{n-1}, \tag{16}$$

where $\beta_q := (1+\lambda_q\Delta T)^{-1}$ and $\mathcal{T}(\beta) := \left\{\beta^{j-i-1} \text{ if } j > i,\ 0 \text{ otherwise}\right\}$ is a Toeplitz matrix of size $\hat{k}$. Applying (16) recursively we obtain:

$$\max_{1\le k\le \hat{k}} |\zeta_q^n| \le \rho_n^q \max_{1\le k\le \hat{k}} |\zeta_q^0|,$$

where:

$$\rho_n^q := \left\|\left((1+\lambda_q\tau)^{-\Delta T/\tau} - \beta_q\right)^n \mathcal{T}^n(\beta_q)\right\|_{L^\infty}. \tag{17}$$

Since $\lambda_q > 0$ and $\beta_q \le (1+\lambda_q\Delta T)^{-\Delta T/\tau} \le e^{-\lambda_q\Delta T}$, we obtain

$$|\,(1+\lambda_q\tau)^{-\Delta T/\tau} - \beta_q| \le |e^{-\lambda_q\Delta T} - \beta_q| = |e^{1-1/\beta_q} - \beta_q|, \tag{18}$$

which yields:

$$\rho_n^q \le |e^{1-1/\beta_q} - \beta_q|^n \|\mathcal{T}^n(\beta_q)\|_{L^\infty} \le \sup_{0<\beta<1} |e^{1-1/\beta} - \beta|^n \|\mathcal{T}^n(\beta)\|_{L^\infty}.$$

By considering $\|\mathcal{T}^n(\beta)\|_\infty \le \|\mathcal{T}(\beta)\|_\infty^n = \left|\frac{1-\beta^{\hat{k}-1}}{1-\beta}\right|^n$, a simpler upper bound for ρ_n can be obtained:

$$\sup_{0<\beta<1}\left|e^{1-1/\beta} - \beta\right|^n \left|\frac{1-\beta^{\hat{k}-1}}{1-\beta}\right|^n \le \left(\sup_{0<\beta<1}\frac{e^{1-1/\beta}-\beta}{1-\beta}\right)^n \approx 0.2984^n,$$

and the maximum is attained around $\beta_* = 0.358$, independently of n and $\hat{k}$ (β_* presents slight variation for $1 \le n$ and $6 \le \hat{k}$, cases of practical interest).

4 Numerical Experiments

The optimal control problem we consider involves the 1D-heat equation:

$$z_t - z_{xx} = v, \quad 0 < x < 1, \quad 0 < t < 1,$$

with boundary conditions $z(t,0) = z(t,1) = 0$ for $t \in [0,1]$, and initial data $z(0,x) = 0$ for $x \in [0,1]$. The control variable $v(\cdot)$ corresponds to the forcing term, and the target function is the nodewise interpolation of the function $\hat{y}(t,x) = x(1-x)e^{-x}$. We choose a tolerance $tol \leq 10^{-6}$ for the left preconditioned MINRES.

Table 1 lists the value of $(\gamma_{\max} - 1)$ for different values of τ and n. The results confirm Remark 1. Table 2 lists the number of MINRES iterations as ΔT and τ vary while $(\Delta T/\tau)$ remains constant. Choosing $n = 2, 4, 7$ iterations for the Parareal, the number of iterations for the MINRES basically remains constant when h or τ are refined, and so the results indicate scalability. Table 3 lists the number of MINRES iterations for $n = 2$ and $\tau = (1/512)$ for different values of $(\Delta T/\tau)$. It indicates also scalability with respect to ΔT. Like in [4], we observe numerically that the number of MINRES iterations grows logarithmically with respect to $1/r$.

Table 1. Values of $\gamma_{max} - 1$ when τ is refined. Parameters $h = 1/10$ and $\Delta T = 1/20$.

$n \setminus \hat{l}$	200	400	800	1600
$n = 1$	0.864415	1.449299	2.473734	4.371709
$n = 2$	0.070835	0.097852	0.136802	0.193845
$n = 3$	0.007760	0.010765	0.015141	0.021165
$n = 4$	0.000865	0.001224	0.001715	0.002397

Table 2. MINRES iterations using a parareal with $n = 2/4/7$ as preconditioners. Parameters $r = 0.0001$ and $\Delta T/\tau = 16$.

$\hat{k}$	4	8	16	32
$\hat{l}$	64	128	256	512
$h = 1/16$	62 / 40 / 42	58 / 44 / 44	60 / 50 / 44	60 / 50 / 44
$h = 1/32$	60 / 42 / 42	58 / 44 / 44	60 / 50 / 44	62 / 50 / 44
$h = 1/64$	60 / 42 / 42	58 / 44 / 44	60 / 50 / 44	62 / 50 / 44

References

[1] M. J. Gander and S. Vandewalle. On the super linear and linear convergence of the parareal algorithm. In *Domain Decomposition Methods in Science and*

Table 3. MINRES iterations using the Parareal algorithm with $n = 2$ as preconditioner. Parameters $r = 0.001/0.0001/0.00001$ and $\tau = 1/512$.

$\hat{k}$	8	16	32	64
$\Delta T/\tau$	64	32	16	8
$h = 1/16$	32 / 62 / 136	32 / 62 / 136	32 / 60 / 132	32 / 60 / 132
$h = 1/32$	32 / 62 / 136	32 / 62 / 136	32 / 62 / 132	32 / 60 / 132
$h = 1/64$	32 / 62 / 136	32 / 62 / 136	32 / 62 / 132	32 / 60 / 132

Engineering XVI, volume 55 of *Lect. Notes Comput. Sci. Engrg.*, pages 291–298. Springer, Berlin, 2007.

[2] J. L. Lions. *Optimal Control of Systems Governed by Partial Differential Equations.* Springer-Verlag, Berlin-Heidelberg-New York, 1971.

[3] J. L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps pararéel. *C. R. Acad. Sci. Paris Sér. I Math.*, 332(7):661–668, 2001.

[4] T. P. Mathew, M. Sarkis, and C. E. Schaerer. Block iterative algorithms for the solution of parabolic optimal control problems. In M. Daydé et al., editor, *High Performance Computing for Computational Science - VECPAR 2006*, pages 452–465. Springer, Lect. Notes Comput. Sci., 4395, 2007.

[5] C. C. Paige and M. A. Saunders. Solutions of sparse indefinite systems of linear equations. *SIAM J. Numer. Anal.*, 12(4):617–629, 1975.

Part III

Contributed Presentations

Boundary Value Problems in Ramified Domains with Fractal Boundaries

Yves Achdou[1] and Nicoletta Tchou[2]

[1] UFR Mathématiques, Université Paris 7, Case 7012, 75251 Paris Cedex 05, France and Laboratoire Jacques-Louis Lions, Université Paris 6, 75252 Paris Cedex 05. achdou@math.jussieu.fr

[2] IRMAR, Université de Rennes 1, Rennes, France. nicoletta.tchou@univ-rennes1.fr

Introduction

This work deals with some Poisson problems in a self-similar ramified domain of $\mathbb{R}^2$ with a fractal boundary (see Figure 1). We consider generalized Neumann condition on the fractal boundary. The first goal is to give a rigorous functional setting. The second goal is to propose a strategy for computing the solutions in simple subdomains obtained by stopping the construction after a finite number of steps. When the Neumann data belongs to the Haar basis associated to a dyadic decomposition of the fractal boundary, we show that the solution can be found by solving a sequence of boundary value problems in an elementary cell, with nonhomogeneous and nonlocal boundary conditions. For a general Neumann data g, the idea is to expand g on the Haar basis and use the linearity of the problem for deriving an expansion of the solution.
This work is an extension of [1], where the Hausdorff dimension of the fractal boundary was 1. Related results for the Helmholtz equation are contained in [2]. The proofs of the theoretical results below are given in [3].

1 The Geometry

Let a be a positive parameter. Consider the points of $\mathbb{R}^2$: $P_1 = (-1, 0)$, $P_2 = (1, 0)$, $P_3 = (-1, 1)$, $P_4 = (1, 1)$, $P_5 = (-1+a\sqrt{2}, 1+a\sqrt{2})$ and $P_6 = (1-a\sqrt{2}, 1+a\sqrt{2})$. Let Y^0 and F_i, $i = 1, 2$ be respectively the hexagonal subset of $\mathbb{R}^2$ and the similitudes defined by the following:

$$
\begin{aligned}
&Y^0 = \text{Interior}\Big(\ \text{Conv}(P_1, P_2, P_3, P_4, P_5, P_6)\Big),\\
&F_i(x) = \Big((-1)^i\Big(1-\tfrac{a}{\sqrt{2}}\Big) + \tfrac{a}{\sqrt{2}}\Big(x_1 + (-1)^i x_2\Big), 1 + \tfrac{a}{\sqrt{2}} + \tfrac{a}{\sqrt{2}}\Big(x_2 + (-1)^{i+1} x_1\Big)\Big)\,.
\end{aligned}
$$

The similitude F_i has the dilation ratio a and the rotation angle $(-1)^{i+1}\pi/4$. To prevent $F_1(Y^0)$ and $F_2(Y^0)$ from overlapping, one must choose $a \leq \sqrt{2}/2$.

For $n \geq 1$, we call $\mathcal{A}_n$ the set containing all the 2^n mappings from $\{1,\dots,n\}$ to $\{1,2\}$. We define

$$\mathcal{M}_\sigma = F_{\sigma(1)} \circ \cdots \circ F_{\sigma(n)} \quad \text{for } \sigma \in \mathcal{A}_n, \tag{1}$$

and the ramified open domain, see Figure 1,

$$\Omega = \text{Interior}\left(\overline{Y^0} \cup \left(\bigcup_{n=1}^{\infty} \bigcup_{\sigma\in\mathcal{A}_n} \mathcal{M}_\sigma(\overline{Y^0})\right)\right). \tag{2}$$

Stronger constraints must be imposed on a to prevent the sets $\mathcal{M}_\sigma(\overline{Y^0})$, $\sigma \in \mathcal{A}_n$, $n > 0$, from overlapping. It can be shown that the condition is $2\sqrt{2}a^5 + 2a^4 + 2a^2 + \sqrt{2}a - 2 \leq 0$, *i.e.* , $a \leq a^* \sim 0.593465\dots$

We call Γ^∞ the self similar set associated to the similitudes F_1 and F_2, *i.e.* the unique compact subset of $\mathbb{R}^2$ such that $\Gamma^\infty = F_1(\Gamma^\infty) \cup F_2(\Gamma^\infty)$. The Hausdorff dimension of Γ^∞ can be computed since Γ^∞ satisfies the Moran condition (open set condition) (see [6, 5]): $\dim_H(\Gamma^\infty) = -\log 2/\log a$. For instance, if $a = a^*$, then $\dim_H(\Gamma^\infty) \sim 1.3284371$.

We split the boundary of Ω into Γ^∞, $\Gamma^0 = [-1,1]\times\{0\}$ and $\Sigma = \partial\Omega\backslash(\Gamma^0 \cup \Gamma^\infty)$. We define the polygonal open domain Y^N obtained by stopping the above construction at step $N+1$,

$$Y^N = \text{Interior}\left(\overline{Y^0} \cup \left(\bigcup_{n=1}^{N} \bigcup_{\sigma\in\mathcal{A}_n} \mathcal{M}_\sigma(\overline{Y^0})\right)\right). \tag{3}$$

We also define the sets $\Gamma^\sigma = \mathcal{M}_\sigma(\Gamma^0)$ and $\Gamma^N = \cup_{\sigma\in\mathcal{A}_N}\Gamma^\sigma$.

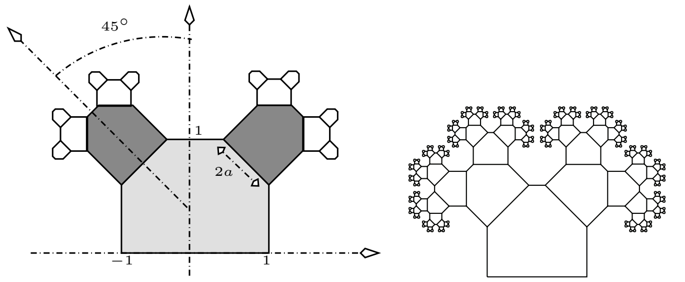


Fig. 1. The ramified domain Ω (only a few generations are displayed).

2 Functional Setting

Let $H^1(\Omega)$ be the space of functions in $L^2(\Omega)$ with first order partial derivatives in $L^2(\Omega)$. We also define

$$\mathcal{V}(\Omega) = \left\{v \in H^1(\Omega); v|_{\Gamma^0} = 0\right\} \quad \text{and} \quad \mathcal{V}(^n) = \left\{v \in H^1(Y^n); v|_{\Gamma^0} = 0\right\}.$$

Theorem 1. *There exists a constant $C > 0$, such that*

$$\forall u \in H^1(\Omega), \qquad \|u\|^2_{L^2(\Omega)} \leq C \left(\|\nabla u\|^2_{L^2(\Omega)} + \|u|_{\Gamma^0}\|^2_{L^2(\Gamma^0)} \right). \tag{4}$$

The embedding of $H^1(\Omega)$ in $L^2(\Omega)$ is compact.

For defining traces on Γ^∞, we need the classical result, see [4]:

Theorem 2. *There exists a unique Borel regular probability measure μ on Γ^∞ such that for any Borel set $A \subset \Gamma^\infty$,*

$$\mu(A) = 1/2\mu \left(F_1^{-1}(A)\right) + 1/2\mu \left(F_2^{-1}(A)\right). \tag{5}$$

The measure μ is called the *self-similar measure defined in the self similar triplet* $(\Gamma^\infty, F_1, F_2)$. Let L^2_μ be the Hilbert space of the functions on Γ^∞ that are μ-measurable and square integrable w.r.t. μ, with the norm $\|u\|_{L^2_\mu} = \sqrt{\int_{\Gamma^\infty} u^2 d\mu}$. A Hilbertian basis of L^2_μ can be constructed with e.g. Haar wavelets.
Consider the sequence of linear operators $\ell^n : H^1(\Omega) \to L^2_\mu$,

$$\ell^n(u) = \sum_{\sigma \in \mathcal{A}_n} \left(1/|\Gamma^\sigma| \int_{\Gamma^\sigma} u \, dx \right) \mathbf{1}_{\mathcal{M}_\sigma(\Gamma^\infty)}, \tag{6}$$

where $|\Gamma^\sigma|$ is the Lebesgue measure of Γ^σ.

Lemma 1. *The sequence $(\ell^n)_n$ converges in $\mathcal{L}(H^1(\Omega), L^2_\mu)$, to an operator that we call ℓ^∞. The operator ℓ^∞ can be seen as a renormalized trace operator.*

3 A Class of Poisson Problems

Take $g \in L^2_\mu$ and $u \in H^{\frac{1}{2}}(\Gamma^0)$. We look for $U(u, g) \in H^1(\Omega)$ s.t.

$$(U(u,g))|_{\Gamma^0} = u, \text{ and } \int_\Omega \nabla(U(u,g)) \cdot \nabla v = \int_{\Gamma^\infty} g\ell^\infty(v) \, d\mu, \ \forall v \in \mathcal{V}(\Omega). \tag{7}$$

If it exists, then $(U(u,g))$ satisfies $\Delta(U(u,g)) = 0$ in Ω, and $\partial_n(U(u,g)) = 0$ on Σ. We shall discuss the boundary condition on Γ^∞ after the following:

Theorem 3. *For $g \in L^2_\mu$ and $u \in H^{\frac{1}{2}}(\Gamma^0)$, (7) has a unique solution.*
Furthermore, if $g = \ell^\infty(\tilde{g})$, $\tilde{g} \in \mathcal{C}^1(\overline{\Omega})$, if $w_q \in H^1(Y^q)$ is the solution of:

$$\begin{aligned} &\Delta w_q = 0 \quad \textit{in } Y^q, \qquad w_q|_{\Gamma^0} = u, \qquad \partial_n w_q = 0 \quad \textit{on } \partial Y_q \backslash (\Gamma^0 \cup \Gamma^{q+1}), \\ &\partial_n w_q = (1/|\Gamma^{q+1}|)\tilde{g}|_{\Gamma^{q+1}} \quad \textit{on } \Gamma^{q+1}, \end{aligned}$$

then $\lim_{q \to \infty} \|(U(u,g))|_{Y^q} - w_q\|_{H^1(Y^q)} = 0$.

Theorem 3 says in particular that (7) has an intrinsic meaning for a large class of data g. From the definition of w_q, we may say that $U(u,g)$ satisfies a Neumann condition on Γ^∞ with datum g.

4 A Strategy for Computing $U(u,g)|_{Y^n}$

4.1 The Case when $g = 0$.

We use the notation $\mathcal{H}(u) = U(u, 0)$. Call T the Dirichlet-Neumann operator from $H^{\frac{1}{2}}(\Gamma^0)$ to $(H^{\frac{1}{2}}(\Gamma^0))'$, $Tu = \partial_n \mathcal{H}(u)|_{\Gamma^0}$. We remark that $T \in \mathbb{O}$, the cone containing the self-adjoint, positive semi-definite, bounded linear operators from $H^{\frac{1}{2}}(\Gamma^0)$ to $(H^{\frac{1}{2}}(\Gamma^0))'$ which vanish on the constants.
If T is available, the self-similarity implies that $\mathcal{H}(u)|_{Y^0} = w$, where w is s.t.

$$\Delta w = 0 \quad \text{in } Y^0, \qquad \frac{\partial w}{\partial n}|_{\partial Y^0 \setminus (\Gamma^0 \cup \Gamma^1)} = 0, \tag{8}$$

$$w|_{\Gamma^0} = u, \tag{9}$$

$$\frac{\partial w}{\partial n} + \frac{1}{a}\left(T(w|_{F_i(\Gamma^0)} \circ F_i)\right) \circ F_i^{-1} = 0 \quad \text{on } F_i(\Gamma^0),\ i = 1, 2. \tag{10}$$

We stress the fact that (8)-(10) is well posed, from the observation on T above. Since (10) allows computing $\mathcal{H}(u)|_{Y^0}$, it is called a transparent boundary condition. The construction may be generalized to $\mathcal{H}(u)|_{Y^{n-1}}$, $n \geq 1$:

Proposition 1. *For $u \in H^{\frac{1}{2}}(\Gamma^0)$, $\mathcal{H}(u)|_{Y^{n-1}}$ can be found by successively solving $1 + 2 + \cdots + 2^{n-1}$ boundary value problems in Y^0:*

```
• Loop: for $p = 0$ to $n - 1$,
  •• Loop : for $\sigma \in \mathcal{A}_p$, (at this point, if $p \geq 1$, $(\mathcal{H}(u))|_{\Gamma^\sigma}$ is known)
    ••• Find $w \in H^1(Y^0)$ satisfying the boundary value problem (8), (10), and either (9) if $p = 0$, or $w|_{\Gamma^0} = \mathcal{H}(u)|_{\Gamma^\sigma} \circ \mathcal{M}_\sigma$ if $p > 0$.
    ••• Set $\mathcal{H}(u)|_{Y^0} = w$ if $p = 0$. If $p > 0$, set $\mathcal{H}(u)|_{\mathcal{M}_\sigma(Y^0)} = w \circ (\mathcal{M}_\sigma)^{-1}$.
```

We are left with computing T: in Theorem 4 below, we show that T can be obtained as the limit of a sequence of operators constructed by a simple induction. This is the consequence of the following result:

Proposition 2. *There exists a constant $\rho < 1$ such that for any $u \in H^{\frac{1}{2}}(\Gamma^0)$,*

$$\sum_{\sigma \in \mathcal{A}_p} \int_{\Omega^\sigma} |\nabla \mathcal{H}(u)|^2 \leq \rho^p \int_{\Omega} |\nabla \mathcal{H}(u)|^2, \quad \forall p > 0. \tag{11}$$

In order to compute T, we introduce the mapping $\mathbb{M} : \mathbb{O} \mapsto \mathbb{O}$: for any $Z \in \mathbb{O}$,

$$\forall u \in H^{\frac{1}{2}}(\Gamma^0), \qquad \mathbb{M}(Z)u = \partial_n w|_{\Gamma^0}, \tag{12}$$

where w satisfies (8), (9) and $\frac{\partial w}{\partial n} + \frac{1}{a}\left(Z(w|_{F_i(\Gamma^0)} \circ F_i)\right) \circ F_i^{-1} = 0$ on $F_i(\Gamma^0)$.

Theorem 4. *The operator T is the unique fixed point of $\mathbb{M}$. Moreover, if ρ, $0 < \rho < 1$, is the constant appearing in (11), then, for all $Z \in \mathbb{O}$, $\exists C > 0$ s.t.*

$$\|\mathbb{M}^p(Z) - T\| \leq C\rho^{\frac{p}{4}}, \quad \forall p \geq 0. \tag{13}$$

In what follows, we propose a method for computing $(U(0,g))|_{Y^{n-1}}$, (n is some fixed positive integer). We first distinguish the case when g belongs to the Haar basis associated to the dyadic decomposition of Γ^∞.

4.2 The Case when g Belongs to the Haar Basis

The case when g is a Haar wavelet is particularly favorable because transparent boundary conditions may be used, thanks to self-similarity.
Let us call $e_F = U(0, 1_{\Gamma^\infty})$.
We introduce the linear operator B, bounded from $(H^{\frac{1}{2}}(\Gamma^0))'$ to $L^2(\Gamma^0)$, by: $Bz = -\frac{\partial w}{\partial x_2}|_{\Gamma^0}$, where $w \in \mathcal{V}(Y^0)$ is the unique weak solution to

$$\Delta w = 0 \quad \text{in } Y^0, \qquad \frac{\partial w}{\partial n} = 0 \quad \text{on } \partial Y_0 \backslash (\Gamma^0 \cup \Gamma^1), \tag{14}$$

$$\frac{\partial w}{\partial x_2}|_{F_i(\Gamma^0)} + \frac{1}{a}\left(T(w|_{F_i(\Gamma^0)} \circ F_i)\right) \circ F_i^{-1} = -z \circ F_i^{-1}, \qquad i = 1, 2. \tag{15}$$

The self-similarity in the geometry and the scale-invariance of the equations are the fundamental ingredients for proving the following theorem:

Theorem 5. *The normal derivative y_F of e_F on Γ^0 belongs to $L^2(\Gamma^0)$ and is the unique solution to: $y_F = By_F$ and $\int_{\Gamma^0} y_F = -1$.*
For all $n \geq 1$, the restriction of e_F to Y^{n-1} can be found by successively solving $1 + 2 + \cdots + 2^{n-1}$ boundary value problems in Y^0, as follows:
•Loop: for $p = 0$ to $n - 1$,
• • Loop : for $\sigma \in \mathcal{A}_p$, (at this point, if $p > 0$, $e_F|_{\Gamma^\sigma}$ is known)
• •• Solve the boundary value problem in Y^0: find $w \in H^1(\Omega)$ satisfying (14), with $w|_{\Gamma^0} = 0$ if $p = 0$, $w|_{\Gamma^0} = e_F|_{\Gamma^\sigma} \circ \mathcal{M}_\sigma$ if $p > 0$, and

$$\frac{\partial w}{\partial n} + \frac{1}{a}\left(T(w|_{F_i(\Gamma^0)} \circ F_i)\right) \circ F_i^{-1} = -\frac{1}{2^{p+1}a} y_F \circ F_i^{-1}, \quad \text{on } F_i(\Gamma^0),\ i = 1, 2.$$

• •• Set $e_F|_{Y^0} = w$ if $p = 0$, else set $e_F|_{\mathcal{M}_\sigma(Y^0)} = w \circ (\mathcal{M}_\sigma)^{-1}$.

When g is a Haar wavelet on Γ^∞, the knowledge of T, e_F and y_F permits $U(0, g)$ to be computed: call $g^0 = 1_{F_1(\Gamma^\infty)} - 1_{F_2(\Gamma^\infty)}$ the Haar mother wavelet, and define $e^0 = U(0, g^0)$. One may compute $e^0|_{Y^n}$ by using the following:

Proposition 3. *We have $e^0|_{Y^0} = w$, where $w \in \mathcal{V}(Y^0)$ satisfies (14) and*

$$\frac{\partial w}{\partial n} + \frac{1}{a}\left(T(w|_{F_i(\Gamma^0)} \circ F_i)\right) \circ F_i^{-1} = \frac{(-1)^i}{2a} y_F \circ F_i^{-1} \text{ on } F_i(\Gamma^0),\ i = 1, 2, \tag{16}$$

Furthermore, for $i = 1, 2$,

$$e^0|_{F_i(\Omega_0)} = (-1)^{i+1}/2\, e_F \circ F_i^{-1} + \left(\mathcal{H}(e^0|_{F_i(\Gamma_0)} \circ F_i)\right) \circ F_i^{-1}. \tag{17}$$

For a positive integer p , take $\sigma \in \mathcal{A}_p$. Call g^σ the Haar wavelet on Γ^∞, defined by $g^\sigma|_{\mathcal{M}_\sigma(\Gamma^\infty)} = g^0 \circ \mathcal{M}_\sigma^{-1}$, and $g^\sigma|_{\Gamma^\infty \backslash \mathcal{M}_\sigma(\Gamma^\infty)} = 0$, and call $e^\sigma = U(0, g^\sigma)$, and y^σ (resp. y^0) the normal derivative of e^σ (resp. e^0) on Γ^0. The following result shows that (e^σ, y^σ) can be computed by induction:

Proposition 4. *The family (e^σ, y^σ) is defined by induction: assume that $\mathcal{M}_\sigma = F_i \circ \mathcal{M}_\eta$ for some $i \in \{1, 2\}$, $\eta \in \mathcal{A}_{p-1}$, $p > 1$. Then $e^\sigma|_{Y^0} = w$, where $w \in \mathcal{V}(Y^0)$ satisfies (14) and*

$$\frac{\partial w}{\partial n} + \frac{1}{a}\left(T(w|_{F_i(\Gamma^0)} \circ F_i)\right) \circ F_i^{-1} = -\frac{1}{2a} y^\eta \circ F_i^{-1} \quad \text{on } F_i(\Gamma^0),\ i = 1, 2. \tag{18}$$

Then, with $j = 1 - i$, $e^\sigma|_{\Omega\setminus Y^0}$ is given by

$$\begin{aligned} e^\sigma|_{F_i(\Omega)} &= \frac{1}{2}e^\eta \circ F_i^{-1} + \big(\mathcal{H}(e^\sigma|_{F_i(\Gamma_0)} \circ F_i)\big) \circ F_i^{-1}, \\ e^\sigma|_{F_j(\Omega)} &= \big(\mathcal{H}(e^\sigma|_{F_j(\Gamma_0)} \circ F_j)\big) \circ F_j^{-1}. \end{aligned} \tag{19}$$

If $\mathcal{M}_\sigma = F_i$, $i = 1, 2$, then y^η (resp. e^η) must be replaced by y^0 (resp. e^0) in (18), (resp.(19)).

What follows indicates that for $n \geq 0$ fixed, $\|\nabla e^\sigma\|_{L^2(Y^n)}$, $\sigma \in \mathcal{A}_p$, decays exponentially as $p \to \infty$:

Theorem 6. *$\exists C > 0$ and ρ, $0 < \rho < 1$ s.t.*

$$\|\nabla e^\sigma\|_{L^2(Y^n)} \leq C 2^{-n} \rho^{p-n}, \quad \forall \sigma \in \mathcal{A}_p, 0 \leq n < p - 1. \tag{20}$$

4.3 The General Case

Consider now the case when g is a general function in L^2_μ. It is no longer possible to use the self-similarity in the geometry for deriving transparent boundary conditions for $U(0, g)$. The idea is different: one expands g on the Haar basis, and use the linearity of (7) with respect to g for obtaining an expansion of $U(0, g)$ in terms of e_F, e^0, and e^σ, $\sigma \in \mathcal{A}_p$, $p > 1$. Indeed, one can expand $g \in L^2_\mu$ as follows:

$$g = \alpha_F 1_{\Gamma^\infty} + \alpha_0 g^0 + \sum_{p=1}^{\infty} \sum_{\sigma \in \mathcal{A}_p} \alpha_\sigma g^\sigma. \tag{21}$$

The following result, which is a consequence of Theorem 6, says that $(U(0, g))|_{Y^n}$ can be expanded in terms of $e_F|_{Y^n}$, $e^0|_{Y^n}$, and $e^\sigma|_{Y^n}$, $\sigma \in \mathcal{A}_p$, $p \geq 1$. Moreover, a few terms in the expansion are enough to approximate $(U(0, g))|_{Y^n}$ with a good accuracy:

Proposition 5. *Assume (21) and call r^P the error $r^P = U(0, g) - \alpha_F e_F - \alpha_0 e^0 - \sum_{p=1}^{P} \sum_{\sigma \in \mathcal{A}_p} \alpha_\sigma e^\sigma$. $\exists C$ (independent of g) s.t.*

$$\|r^P\|_{H^1(Y^n)} \leq C\sqrt{2^{-P}} \rho^{P-n} \|g\|_{L^2_\mu}, \quad \forall n, P,\ 0 \leq n < P - 1. \tag{22}$$

Generalizations. Here we discuss possible generalizations of the example above. The geometrical construction only depends on three basic elements: the elementary cell Y^0 and the similitudes F_1 and F_2 (dilation ratii a_1 and a_2, $0 < a_i < 1$, rotation angles α_1 and α_2). The following conditions must be satisfied: 1) the elementary cell Y^0 is a Lipschitz domain. 2) The domain Ω defined by (2) is a connected open set. 3) For $\sigma_1, \sigma_2 \in \cup_{n\in\mathbb{N}} \mathcal{A}_n$, $\sigma_1 \neq \sigma_2$, $\mathcal{M}_{\sigma_1}(Y^0) \cap \mathcal{M}_{\sigma_2}(Y^0) = \emptyset$. If these conditions are fulfilled, all the above results apply. The important point is to use the measure μ defined in Theorem 2. Of course, one can consider constructions with to more than two similitudes, i.e. F_i, $i = 1, \ldots, p$, with respective dilation ratio $a_i > 0$ and angles α_i.

5 Numerical Results

To transpose the strategies described above to finite element methods, one needs to use self-similar triangulations of Ω: we first consider a regular family of meshes $\mathcal{T}_h^0$ of Y^0, with the special property that for $i = 1, 2$, the set of nodes of $\mathcal{T}_h^0$ lying on $F_i(\Gamma^0)$ is the image by F_i of the set of nodes lying on Γ^0. Then one can construct self-similar meshes of Ω by $\mathcal{T}_h = \cup_{p=0}^{\infty} \cup_{\sigma \in \mathcal{A}_p} \mathcal{M}_\sigma(\mathcal{T}_h^0)$, with self-explanatory notations. With such meshes and conforming finite elements, one can transpose everything to the discrete level.

An Example. The aim is to compute $U(0, g)|_{Y^5}$, with $g(s) = (1_{s<0} - 1_{s>0}) \cos(3\pi s/2)$, where $s \in [-1, 1]$ is a parametrization of Γ^∞. We first compute the operator T by the method in § 4.1 and $e^\sigma|_{Y^5}$, for $\sigma \in \mathcal{A}_p$, $p \leq 5$ by the method in § 4.2. Then we expand g on the Haar basis and use the expansion in Proposition 5.

In the top of Figure 2, we plot two approximations of $U(0, g)|_{Y^5}$; we have used the expansion in Proposition 5., with $P = 5$ on the left, and $P = 2$ on the right. We see that taking $P = 2$ is enough for approximating $U(0, g)|_{Y^0}$, but not for $U(0, g)|_{Y^j}$, $j \geq 1$. In the bottom of Figure 2, we plot (in log scales) the errors $\| \sum_{p=i}^{5} \sum_{\sigma \in \mathcal{A}_p} \alpha^\sigma e_h^\sigma \|_{L^2(Y^j)}$, for $i = 2, 3, 4$ and $j = 0, 1, 2, 3, 4$, where α^σ are the coefficients of the wavelet expansion of g. The behavior is the one predicted by Proposition 5.

Again, we stress that there is no error from the domain truncation, and that we did not solve any boundary value problem in Y^5, but a sequence of boundary problems in Y^0. Nevertheless, the function smoothly matches at the interfaces between the subdomains.

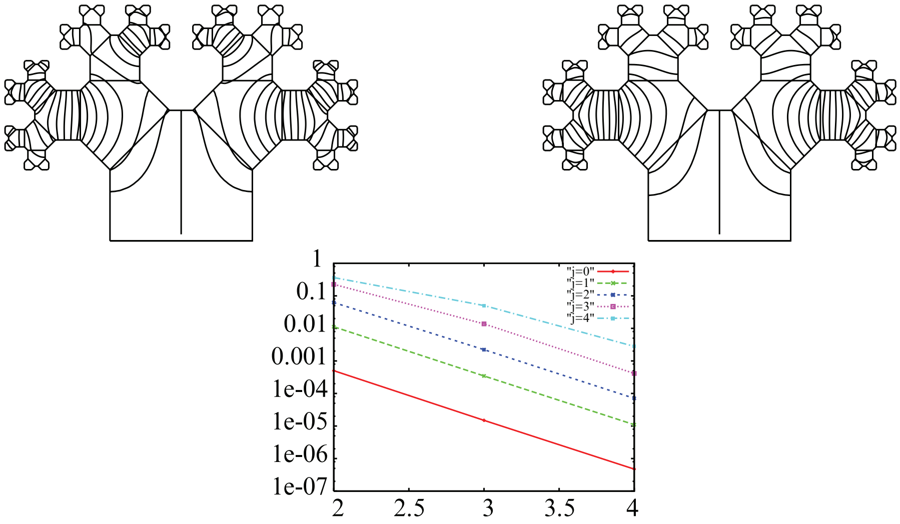


Fig. 2. Top: Contours of the approximations of $U(0, g)|_{Y^5}$ by taking $P = 5$(left) and $P = 2$(right). Bottom: $\| \sum_{p=i}^{5} \sum_{\sigma \in \mathcal{A}_p} \alpha^\sigma e_h^\sigma \|_{L^2(Y^j)}$ for $i = 2, 3, 4$ and $j = 0, 1, 2, 3, 4$.

References

[1] Y. Achdou, C. Sabot, and N. Tchou. A multiscale numerical method for Poisson problems in some ramified domains with a fractal boundary. *Multiscale Model. Simul.*, 5(3):828–860, 2006.

[2] Y. Achdou, C. Sabot, and N. Tchou. Transparent boundary conditions for the Helmholtz equation in some ramified domains with a fractal boundary. *J. Comput. Phys.*, 220(2):712–739, 2007.

[3] Y. Achdou and N. Tchou. Neumann conditions on fractal boundaries. *Asymptot. Anal.*, 53(1-2):61–82, 2007.

[4] Kenneth Falconer. *Techniques in Fractal Geometry.* John Wiley & Sons Ltd., Chichester, 1997.

[5] J. Kigami. *Analysis on Fractals*, volume 143 of *Cambridge Tracts in Mathematics.* Cambridge University Press, Cambridge, 2001.

[6] P.A.P. Moran. Additive functions of intervals and Hausdorff measure. *Proc. Cambridge Philos. Soc.*, 42:15–23, 1946.

An Additive Schwarz Method for the Constrained Minimization of Functionals in Reflexive Banach Spaces

Lori Badea

Institute of Mathematics of the Romanian Academy, P.O. Box 1-764, 014700
Bucharest, Romania. lori.badea@imar.ro

Summary. In this paper, we show that the additive Schwarz method proposed in [3] to solve one-obstacle problems converges in a much more general framework. We prove that this method can be applied to the minimization of functionals over a general enough convex set in a reflexive Banach space. In the Sobolev spaces, the proposed method is an additive Schwarz method for the solution of the variational inequalities coming from the minimization of non-quadratic functionals. Also, we show that the one-, two-level variants of the method in the finite element space converge, and we explicitly write the constants in the error estimations depending on the overlapping and mesh parameters.

1 Introduction

The literature on the domain decomposition methods is very large. We can see, for instance, the papers in the proceedings of the annual conferences on domain decomposition methods starting with [5], or those cited in the books [10, 11] and [13]. The multilevel or multigrid methods can be viewed as domain decomposition methods and we can cite, for instance, the results obtained in [7, 9, 11].

In [3], an additive Schwarz method has been proposed for symmetric variational inequalities. Although this method does not assume a decomposition of the convex set according to the domain decomposition, the convergence proof is given only for the one-obstacle problems. In Section 2 of this paper, we prove that the method converges in a much more general framework, i.e. we can apply it to the minimization of functionals over a general enough convex set in a reflexive Banach space. In Section 3, we show that, in the Sobolev spaces, the proposed method is an additive Schwarz method and it converges for variational inequalities coming from the minimization of non-quadratic functionals. Also, in Section 4, we show that the one-, two-level variants of the method in the finite element space converge, and we explicitly write the constants in the error estimations depending on the overlapping and mesh parameters. The convergence rates we find are similar with those obtained in the literature for symmetric inequalities or equations, i.e. they are almost independent on the overlapping and mesh parameters in the case of the two-level method.

2 General Convergence Result

Let us consider a reflexive Banach space V, some closed subspaces of V, $V_1, \cdots, V_m$, and $K \subset V$ a non empty closed convex subset. We make the following

Assumption 1 *There exists a constant $C_0 > 0$ such that for any $w, v \in K$ there exist $v_i \in V_i$, $i = 1, \ldots, m$, which satisfy*

$$v - w = \sum_{i=1}^{m} v_i, \quad w + v_i \in K \textit{ and } \sum_{i=1}^{m} ||v_i|| \leq C_0 ||v - w||.$$

We consider a Gâteaux differentiable functional $F : V \to \mathbf{R}$, which is assumed to be coercive on K, in the sense that $\frac{F(v)}{||v||} \to \infty$, as $||v|| \to \infty$, $v \in K$, if K is not bounded. Also, we assume that there exist two real numbers p, $q > 1$ such that for any real number $M > 0$ there exist α_M, $\beta_M > 0$ for which

$$\begin{aligned} &\alpha_M ||v - u||^p \leq \langle F'(v) - F'(u), v - u \rangle \text{ and} \\ &||F'(v) - F'(u)||_{V'} \leq \beta_M ||v - u||^{q-1} \end{aligned} \tag{1}$$

for any $u, v \in V$ with $||u||, ||v|| \leq M$. Above, we have denoted by F' the Gâteaux derivative of F, and we have marked that the constants α_M and β_M may depend on M. It is evident that if (1) holds, then for any $u, v \in V$, $||u||, ||v|| \leq M$, we have $\alpha_M ||v - u||^p \leq \langle F'(v) - F'(u), v - u \rangle \leq \beta_M ||v - u||^q$. Following the way in [6], we can prove that for any $u, v \in V$, $||u||, ||v|| \leq M$, we have

$$\langle F'(u), v - u \rangle + \frac{\alpha_M}{p} ||v - u||^p \leq F(v) - F(u) \leq \langle F'(u), v - u \rangle + \frac{\beta_M}{q} ||v - u||^q. \tag{2}$$

We point out that since F is Gâteaux differentiable and satisfies (1), then F is a convex functional (see Proposition 5.5 in [4], p. 25).

We consider the minimization problem

$$u \in K : F(u) \leq F(v), \text{ for any } v \in K, \tag{3}$$

and since the functional F is convex and differentiable, it is equivalent with the variational inequality

$$u \in K : \langle F'(u), v - u \rangle \geq 0, \text{ for any } v \in K. \tag{4}$$

We can use, for instance, Theorem 8.5 in [8], p. 251, to prove that problem (3) has a unique solution if F has the above properties. In view of (2), for a given $M > 0$ such that the solution $u \in K$ of (3) satisfies $||u|| \leq M$, we have

$$\frac{\alpha_M}{p} ||v - u||^p \leq F(v) - F(u) \text{ for any } v \in K, \ ||v|| \leq M. \tag{5}$$

To solve the minimization problem (3), we propose the following additive subspace correction algorithm corresponding to the subspaces $V_1, \ldots, V_m$ and the convex set K.

Algorithm 1 *We start the algorithm with an arbitrary $u^0 \in K$. At iteration $n+1$, having $u^n \in K$, $n \geq 0$, we solve the inequalities*

$$\begin{aligned}&w_i^{n+1} \in V_i,\ u^n + w_i^{n+1} \in K : \langle F'(u^n + w_i^{n+1}), v_i - w_i^{n+1}\rangle \geq 0,\\ &\textit{for any } v_i \in V_i,\ u^n + v_i \in K,\end{aligned} \tag{6}$$

for $i = 1, \cdots, m$, and then we update $u^{n+1} = u^n + \rho \sum_{i=1}^{m} w_i^{n+1}$, where ρ is chosen such that $u^{n+1} \in K$ for any $n \geq 0$.

A possible choice of ρ to get $u^{n+1} \in K$, is $\rho \leq \frac{1}{m}$. Indeed, if we write $0 < r = \rho m \leq 1$, then $u^{n+1} = (1-r)u^n + r\sum_{i=1}^{m} \frac{1}{m}(u^n + w_i^{n+1}) \in K$. Evidently, problem (6) has an unique solution and it is equivalent with

$$\begin{aligned}&w_i^{n+1} \in V_i,\ u^n + w_i^{n+1} \in K : F(u^n + w_i^{n+1}) \leq F(u^n + v_i),\\ &\text{for any } v_i \in V_i,\ u^n + v_i \in K.\end{aligned} \tag{7}$$

Let us now give the convergence result of Algorithm 1.

Theorem 1. *We consider that V is a reflexive Banach, $V_1, \cdots, V_m$ are some closed subspaces of V, K is a non empty closed convex subset of V satisfying Assumption 1, and F is a Gâteaux differentiable functional on K which is supposed to be coercive if K is not bounded, and satisfies (1). On these conditions, if u is the solution of problem (3) and u^n, $n \geq 0$, are its approximations obtained from Algorithm 1, then there exists $M > 0$ such that the following error estimations hold:*

(i) if $p = q$ we have

$$F(u^n) - F(u) \leq \left(\frac{C_1}{C_1+1}\right)^n \left[F(u^0) - F(u)\right], \tag{8}$$

$$||u^n - u||^p \leq \frac{p}{\alpha_M}\left(\frac{C_1}{C_1+1}\right)^n \left[F(u^0) - F(u)\right], \tag{9}$$

where C_1 is given in (14), and

(ii) if $p > q$ we have

$$F(u^n) - F(u) \leq \frac{F(u^0) - F(u)}{\left[1 + nC_2\left(F(u^0) - F(u)\right)^{\frac{p-q}{q-1}}\right]^{\frac{q-1}{p-q}}}, \tag{10}$$

$$||u - u^n||^p \leq \frac{p}{\alpha_M}\frac{F(u^0) - F(u)}{\left[1 + nC_2\left(F(u^0) - F(u)\right)^{\frac{p-q}{q-1}}\right]^{\frac{q-1}{p-q}}}, \tag{11}$$

where C_2 is given in (18).

Proof. We first prove that the approximation sequence $(u^n)_{n\geq 0}$ of u obtained from Algorithm 1 is bounded for $\rho = \frac{r}{m}$, $0 \leq r \leq 1$. In view of the convexity of F and equation (7), we get

$$F(u^{n+1}) = F(u^n + \frac{r}{m}\sum_{i=1}^m w_i^{n+1}) = F((1-r)u^n + \sum_{i=1}^m \frac{r}{m}(u^n + w_i^{n+1}))$$
$$\leq (1-r)F(u^n) + \frac{r}{m}\sum_{i=1}^m F(u^n + w_i^{n+1}) \leq F(u^n).$$

Consequently, using (3), we have $F(u) \leq F(u^{n+1}) \leq F(u^n) \leq \cdots \leq F(u^0)$, and, from the coercivity of F if K is not bounded, we get that there exists $M > 0$, such that $||u|| \leq M$ and $||u^n|| \leq M$, $n \geq 0$.

In view of (2) and (6), we get $\frac{\alpha_M}{p}||w_i^{n+1}||^p \leq F(u^n) - F(u^n + w_i^{n+1})$. Using this equation in the place of (7), with a proof similar with the above one, we get

$$\rho\frac{\alpha_M}{p}\sum_{i=1}^m ||w_i^{n+1}||^p \leq F(u^n) - F(u^{n+1}) \tag{12}$$

Now, writing $\bar{u}^{n+1} = u^n + \sum_{i=1}^m w_i^{n+1}$ in view of the convexity of F, we have

$$F(u^{n+1}) = F(u^n + \frac{r}{m}\sum_{i=1}^m w_i^{n+1}) = F((1-\frac{r}{m})u^n + \frac{r}{m}(u^n + \sum_{i=1}^m w_i^{n+1})$$
$$\leq (1-\frac{r}{m})F(u^n) + \frac{r}{m}F(u^n + \sum_{i=1}^m w_i^{n+1}) \leq (1-\frac{r}{m})F(u^n) + \frac{r}{m}F(\bar{u}^{n+1}).$$

With $v := u$ and $w := u^n$, we get a decomposition $v_i^n \in V_i$ of $u - u^n$ satisfying the conditions of Assumption 1. Using this decomposition, the above equation, (2) and inequalities (6),

$$\begin{aligned}
&F(u^{n+1}) - F(u) + \rho\tfrac{\alpha_M}{p}||\bar{u}^{n+1} - u||^p \\
&\leq (1-\rho)(F(u^n) - F(u)) + \rho\left(F(\bar{u}^{n+1}) - F(u) + \tfrac{\alpha_M}{p}||\bar{u}^{n+1} - u||^p\right) \\
&\leq (1-\rho)(F(u^n) - F(u)) + \rho\langle F'(\bar{u}^{n+1}), \bar{u}^{n+1} - u\rangle \\
&= (1-\rho)(F(u^n) - F(u)) + \rho\sum_{i=1}^m \langle F'(\bar{u}^{n+1}), w_i^{n+1} - v_i^n\rangle \\
&\leq (1-\rho)(F(u^n) - F(u)) + \rho\sum_{i=1}^m \langle F'(u^n + w_i^{n+1}) - F'(\bar{u}^{n+1}), v_i^n - w_i^{n+1}\rangle \\
&\leq (1-\rho)(F(u^n) - F(u)) + \rho\beta_M\left(\sum_{i=1}^m ||w_i^{n+1}||\right)^{q-1}\sum_{i=1}^m ||v_i^n - w_i^{n+1}|| \\
&\leq (1-\rho)(F(u^n) - F(u)) + \rho\beta_M m^{\frac{(p-1)(q-1)}{p}}\left(\sum_{i=1}^m ||w_i^{n+1}||^p\right)^{\frac{q-1}{p}}\sum_{i=1}^m (||v_i^n|| + ||w_i^{n+1}||) \\
&\leq (1-\rho)(F(u^n) - F(u)) \\
&\quad + \rho\beta_M m^{\frac{(p-1)(q-1)}{p}}\left(\sum_{i=1}^m ||w_i^{n+1}||^p\right)^{\frac{q-1}{p}}\left(C_0||u - u^n|| + \sum_{i=1}^m ||w_i^{n+1}||\right) \\
&\leq (1-\rho)(F(u^n) - F(u)) \\
&\quad + \rho\beta_M m^{\frac{(p-1)(q-1)}{p}}\left(\sum_{i=1}^m ||w_i^{n+1}||^p\right)^{\frac{q-1}{p}}\left(C_0||u - \bar{u}^{n+1}|| + (1+C_0)\sum_{i=1}^m ||w_i^{n+1}||\right)
\end{aligned}$$

$$\leq (1-\rho)(F(u^n)-F(u)) + \rho\beta_M C_0 m^{\frac{(p-1)(q-1)}{p}} \left(\sum_{i=1}^{m} ||w_i^{n+1}||^p\right)^{\frac{q-1}{p}} ||u-\bar{u}^{n+1}||$$
$$+ \rho\beta_M(1+C_0) m^{\frac{(p-1)q}{p}} \left(\sum_{i=1}^{m} ||w_i^{n+1}||^p\right)^{\frac{q}{p}} .$$

But, for any $\varepsilon > 0$ $r > 1$ and $x, y \geq 0$, we have $x^{\frac{1}{r}} y \leq \varepsilon x + \frac{1}{\varepsilon^{\frac{1}{r-1}}} y^{\frac{r}{r-1}}$. Consequently, we get

$$F(u^{n+1}) - F(u) + \rho\tfrac{\alpha_M}{p} ||\bar{u}^{n+1} - u||^p$$
$$\leq (1-\rho)(F(u^n)-F(u)) + \rho\beta_M(1+C_0) m^{\frac{(p-1)q}{p}} \left(\sum_{i=1}^{m} ||w_i^{n+1}||^p\right)^{\frac{q}{p}}$$
$$+ \rho\beta_M C_0 \frac{m^{\frac{(p-1)(q-1)}{p}}}{\varepsilon^{\frac{1}{p-1}}} \left(\sum_{i=1}^{m} ||w_i^{n+1}||^p\right)^{\frac{q-1}{p-1}} + \rho\beta_M C_0 \varepsilon m^{\frac{(p-1)(q-1)}{p}} ||u-\bar{u}^{n+1}||^p .$$

With $\varepsilon = \frac{\alpha_M}{p} \frac{1}{\beta_M C_0 m^{\frac{(p-1)(q-1)}{p}}}$, the above equations become

$$F(u^{n+1}) - F(u) \leq \tfrac{1-\rho}{\rho}(F(u^n) - F(u^{n+1})) + \beta_M(1+C_0) m^{\frac{(p-1)q}{p}} \left(\sum_{i=1}^{m} ||w_i^{n+1}||^p\right)^{\frac{q}{p}}$$
$$+ \left(\beta_M C_0 m^{\frac{(p-1)(q-1)}{p}}\right)^{\frac{p}{p-1}} \left(\frac{p}{\alpha_M}\right)^{\frac{1}{p-1}} \left(\sum_{i=1}^{m} ||w_i^{n+1}||^p\right)^{\frac{q-1}{p-1}} .$$

In view of this equation and (12), we have

$$\begin{aligned} F(u^{n+1}) - F(u) \leq{} & \frac{1-\rho}{\rho}\left(F(u^n) - F(u^{n+1})\right) \\ & + \frac{1}{\rho^{\frac{q}{p}}} \frac{\beta_M(1+C_0) m^{\frac{(p-1)q}{p}}}{(\frac{\alpha_M}{p})^{\frac{q}{p}}} \left(F(u^n) - F(u^{n+1})\right)^{\frac{q}{p}} \\ & + \frac{1}{\rho^{\frac{q-1}{p-1}}} \frac{\left(\beta_M C_0 m^{\frac{(p-1)(q-1)}{p}}\right)^{\frac{p}{p-1}}}{\left(\frac{\alpha_M}{p}\right)^{\frac{q}{p-1}}} \left(F(u^n) - F(u^{n+1})\right)^{\frac{q-1}{p-1}} . \end{aligned} \tag{13}$$

We notice that because of (2) we must have $p \geq q$. Also, using (5), we see that error estimations in (9) and (11) can be obtained from (8) and (10), respectively. Now, if $p = q$, from the above equation, we easily get equation (8), where

$$C_1 = \frac{1}{\rho}\left(1 - \rho + m^{p-1} \frac{\beta_M(1+C_0)}{\frac{\alpha_M}{p}} + m^{p-1} \left(\frac{\beta_M C_0}{\frac{\alpha_M}{p}}\right)^{\frac{p}{p-1}}\right) . \tag{14}$$

Finally, if $p > q$, from (13), we have

$$F(u^{n+1}) - F(u) \leq C_3 \left(F(u^n) - F(u^{n+1})\right)^{\frac{q-1}{p-1}} \tag{15}$$

where

$$C_3 = \frac{1-\rho}{\rho}\left(F(u^0)-F(u)\right)^{\frac{p-q}{p-1}} + \frac{m^{\frac{(p-1)q}{p}}}{\rho^{\frac{q}{p}}}\frac{\beta_M(1+C_0)}{(\frac{\alpha_M}{p})^{\frac{q}{p}}}\left(F(u^0)-F(u)\right)^{\frac{p-q}{p(p-1)}}$$
$$+\frac{m^{q-1}}{\rho^{\frac{q-1}{p-1}}}\frac{(\beta_M C_0)^{\frac{p}{p-1}}}{\left(\frac{\alpha_M}{p}\right)^{\frac{q}{p-1}}}\,. \tag{16}$$

From (15), we get $F(u^{n+1}) - F(u) + \frac{1}{C_3^{\frac{p-1}{q-1}}}\left(F(u^{n+1})-F(u)\right)^{\frac{p-1}{q-1}} \le F(u^n) - F(u)$, and we know (see Lemma 3.2 in [12]) that for any $r > 1$ and $c > 0$, if $x \in (0, x_0]$ and $y > 0$ satisfy $y + cy^r \le x$, then $y \le \left(\frac{c(r-1)}{crx_0^{r-1}+1} + x^{1-r}\right)^{\frac{1}{1-r}}$. Consequently, we have $F(u^{n+1}) - F(u) \le \left[C_2 + (F(u^n)-F(u))^{\frac{q-p}{q-1}}\right]^{\frac{q-1}{q-p}}$, from which,

$$F(u^{n+1}) - F(u) \le \left[(n+1)C_2 + \left(F(u^0)-F(u)\right)^{\frac{q-p}{q-1}}\right]^{\frac{q-1}{q-p}}, \tag{17}$$

where

$$C_2 = \frac{p-q}{(p-1)\left(F(u^0)-F(u)\right)^{\frac{p-q}{q-1}} + (q-1)C_3^{\frac{p-1}{q-1}}}. \tag{18}$$

Equation (17) is another form of equation (10).

3 Additive Schwarz Method as a Subspace Correction Method

The proofs of the results in this section are similar with those in the case of the multiplicative Schwarz method which are given in [1] for the infinite dimensional case, and in [2] for the one- and two-level methods. Detailed proofs for the additive method will be given in a forthcoming paper.

Let Ω be an open bounded domain in $\mathbb{R}^d$ with Lipschitz continuous boundary $\partial\Omega$. We take $V = W_0^{1,s}(\Omega)$, $1 < s < \infty$, and a convex closed set $K \subset V$ satisfying

Property 1. If $v, w \in K$ and $\theta \in C^1(\bar{\Omega})$, with $0 \le \theta \le 1$, then $\theta v + (1-\theta)w \in K$.

We consider an overlapping decomposition of the domain Ω, $\Omega = \cup_{i=1}^m \Omega_i$, in which Ω_i are open subdomains with Lipschitz continuous boundary. We associate to this domain decomposition the subspaces $V_i = W_0^{1,s}(\Omega_i)$, $i = 1, \dots, m$. In this case, Algorithm 1 represents an additive Schwarz method. We can show that Assumption 1 holds for any convex set K having Property 1. Consequently, the additive Schwarz method geometrically converges if the convex set has this property, but the constant C_0 in Assumption 1 depends on the domain decomposition parameters. Therefore, since the constants C_1 and C_2 in the error estimations in Theorem 1 depend on C_0, then these estimations will depend on the domain decomposition parameters, too.

When we use the linear finite element spaces we introduce similar spaces to the above ones, V_h and V_h^i, $i = 1, \dots, m$, which are considered as subspaces of $W_0^{1,s}$. For the one- and two-level additive Schwarz methods, we can show that Assumption 1 also holds for any closed convex set K_h satisfying

Property 2. If $v, w \in K_h$, and if $\theta \in C^0(\bar{\Omega})$, $\theta|_\tau \in C^1(\tau)$ for any $\tau \in \mathcal{T}_h$, and $0 \leq \theta \leq 1$, then $L_h(\theta v + (1-\theta)w) \in K_h$.

We have denoted by $\mathcal{T}_h$ the mesh partition of the domain, and by L_h the $\mathbb{P}_1$-Lagrangian interpolation operator which uses the function values at the mesh nodes. We can prove that Assumption 1 holds for any convex set K_h having Property 2. Moreover, in this case, we are able to explicitly write the dependence of C_0 on the domain decomposition and mesh parameters.

In the case of the one-level method, this constant can be written as

$$C_0 = Cm\,(1 + 1/\delta)\,, \tag{19}$$

where δ is the overlapping parameter and C is independent of the mesh parameter and the domain decomposition. In the case of the two-level method, we introduce a new subspace V_H^0 associated with the coarse mesh $\mathcal{T}_H$. The constant C_0 can be written as

$$C_0 = C(m+1)\,(1 + H/\delta)\,C_{d,s}(H,h), \tag{20}$$

where

$$C_{d,s}(H,h) = \begin{cases} 1 & \text{if } d = s = 1 \text{ or } 1 \leq d < s \leq \infty \\ \left(\ln \frac{H}{h} + 1\right)^{\frac{d-1}{d}} & \text{if } 1 < d = s < \infty \\ \left(\frac{H}{h}\right)^{\frac{d-s}{s}} & \text{if } 1 \leq s < d < \infty. \end{cases} \tag{21}$$

We notice that, if the overlapping size δ and the mesh sizes H and h are chosen such that H/h and H/δ are constant, then the convergence rate of the two-level additive Schwarz method is independent of the mesh and domain decomposition parameters.

Acknowledgement. The author acknowledges the partial financial support of IMAR under the contracts CEEX05-D11-23/2005 and CEEX06-11-12/2006.

References

[1] L. Badea. Convergence rate of a multiplicative Schwarz method for strongly nonlinear inequalities. In *Analysis and Optimization of Differential Systems (also available from http://www.imar.ro/lbadea)*, pages 31–42. Kluwer Academic Publishers, Boston, 2003.

[2] L. Badea. Convergence rate of a Schwarz multilevel method for the constrained minimization of nonquadratic functionals. *SIAM J. Numer. Anal.*, 44(2):449–477, 2006.

[3] L. Badea and J. Wang. An additive Schwarz method for variational inequalities. *Math. Comp.*, 69(232):1341–1354, 1999.

[4] I. Ekeland and R. Temam. *Analyse Convexe et Problèmes Variationnels.* Dunod, Paris, 1974.

[5] R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périeux (Eds.). *First Int. Symp. on Domain Decomposition Methods.* SIAM, Philadelphia, 1988.

[6] R. Glowinski, J. L. Lions, and R. Trémolières. *Analyse Numérique des Inéquations Variationnelles.* Dunod, Paris, 1976.

[7] R. Kornhuber. *Adaptive Monotone Multigrid Methods for Nonlinear Variational Problems.* Teubner-Verlag, Stuttgart, 1997.

[8] J. L. Lions. *Quelques Méthodes de Résolution des Problèmes aux Limites non Linéaires.* Dunod, Paris, 1969.

[9] J. Mandel. A multilevel iterative method for symmetric, positive definite linear complementary problems. *Appl. Math. Optimization*, 11:77–95, 1984.

[10] A. Quarteroni and A. Valli. *Domain Decomposition Methods for Partial Differential Equations.* Oxford Science Publications, 1999.

[11] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition.* Cambridge University Press, 1996.

[12] X.-C. Tai and J. Xu. Global and uniform convergence of subspace correction methods for some convex optimization problems. *Math. Comp.*, 71:105–124, 2002.

[13] A. Toselli and O. Widlund. *Domain Decomposition Methods – Algorithms and Theory.* Springer-Verlag, Berlin, 2005.

Completions to Sparse Shape Functions for Triangular and Tetrahedral p-FEM

Sven Beuchler[1] and Veronika Pillwein[2]

[1] Institute for Comp. Mathematics, JKU Linz, Altenberger Straße 69, A 4040 Linz, Austria. sven.beuchler@jku.at

[2] SFB F013, Altenberger Straße 69, A 4040 Linz, Austria. veronika.pillwein@sfb013.uni-linz.ac.at

1 Introduction

In this paper, we investigate the following boundary value problem: Let $\Omega \subset \mathbb{R}^d$, $d = 2, 3$ be a bounded domain and let $\mathcal{A}$ be a matrix which is symmetric and uniformly positive definite in Ω. Find $u \in H^1_{\Gamma_1}(\Omega) = \{u \in H^1(\Omega), u = 0 \text{ on } \Gamma_1\}$, $\Gamma_1 \cap \Gamma_2 = \emptyset$, $\Gamma_1 \cup \Gamma_2 = \partial\Omega$ such that

$$a_{\triangle}(u,v) := \int_\Omega (\nabla u)^T \mathcal{A} \nabla v = \int_\Omega fv + \int_{\Gamma_2} f_1 v := \langle f, v\rangle_\Omega + \langle f_1, v\rangle_{\Gamma_2} \qquad (1)$$

holds for all $v \in H^1_{\Gamma_1}(\Omega)$. Problem (1) will be discretized by means of the hp-version of the finite element method using triangular/tetrahedral elements $\triangle_s$, $s = 1, \ldots, nel$, see e.g. [5, 8]. Let $\hat{\triangle}_d$, $d = 2, 3$ be the reference triangle (tetrahedron) and $F_s : \hat{\triangle} \to \triangle_s$ be the (possibly nonlinear) isoparametric mapping to the element $\triangle_s$. We define the finite element space $\mathbb{M} := \{u \in H^1_{\Gamma_1}(\Omega), u \mid_{\triangle_s} = \tilde{u}(F_s^{-1}(x,y,z)), \tilde{u} \in \mathbb{P}_p\}$, where $\mathbb{P}_p$ is the space of all polynomials of maximal total degree p. By $\Psi = (\psi_1, \ldots, \psi_N)$, we denote a basis for $\mathbb{M}$. The Galerkin projection of (1) onto $\mathbb{M}$ leads to the linear system of algebraic finite element equations

$$\mathcal{K}_\Psi \underline{u} = \underline{f}, \quad \text{where} \quad \mathcal{K}_\Psi = [a_\Delta(\psi_j, \psi_i)]_{i,j=1}^N, \quad \underline{f}_p = [\langle f, \psi_i\rangle + \langle f_1, \psi_i\rangle_{\Gamma_2}]_{i=1}^N. \qquad (2)$$

The global stiffness matrix $\mathcal{K}_\Psi$ can be expressed by the local stiffness matrices on the elements, i.e.

$$\mathcal{K}_\Psi = \sum_{s=1}^{nel} R_s^T K_s R_s, \qquad (3)$$

where K_s is the stiffness matrix on the element $\triangle_s$ and R_s denotes the connectivity matrix for the numbering of the shape functions on $\triangle_s$ and Ω. In the 2D and 3D case, the choice of a basis which is optimal due to condition number and sparsity of $\mathcal{K}_\Psi$ is not so clear. In [7], a new basis for triangular and tetrahedral elements has been proposed. This basis is optimal w.r.t. the number of nonzero entries of the element stiffness matrix, see [6]. A proof for the sparsity of the element stiffness matrix with $\mathcal{O}(p^d)$ nonzero entries is not known in the literature. In [4], another basis for the triangular case is proposed. Moreover, it is proved that the element stiffness matrix has $\mathcal{O}(p^2)$ nonzero entries. This paper is a completion to the papers [4] and [3]. We will prove the sparsity for the Karniadakis-Sherwin basis, [7]. This proof is similar to the proof for the basis in [4]. However, the proof requires some additional relations for Jacobi polynomials which makes the proof more technical.

The outline of the paper is the following. In Section 2, we summarize the most important properties for Jacobi polynomials. In Section 3, the 2D case is investigated. In Section 4, the 3D case is investigated.

2 Properties of Jacobi Polynomials

For the definition of our basis functions on the reference element, Jacobi polynomials are required, see [1, 2, 9] for more details.

Let

$$p_n^\alpha(x) = \frac{1}{2^n n!(1-x)^\alpha} \frac{\mathrm{d}^n}{\mathrm{d}x^n} \left((1-x)^\alpha (x^2-1)^n\right) \quad n \in \mathbb{N}_0, \alpha > -1 \tag{4}$$

be the nth Jacobi polynomial with respect to the weight function $(1-x)^\alpha$. $p_n^\alpha(x)$ is a polynomial of degree n, i.e. $p_n^\alpha \in \mathbb{P}_n((-1,1))$, where $\mathbb{P}_n$ is the space of all polynomials of degree n on the interval. Moreover, let

$$\hat{p}_n^\alpha(x) = \int_{-1}^{x} p_{n-1}^\alpha(y)\,\mathrm{d}y \quad n \geq 1, \quad \hat{p}_0^\alpha(x) = 1 \tag{5}$$

be the nth integrated Jacobi polynomial.

Lemma 1. *Let p_n^α be defined via (4). Moreover, let $j, l \in \mathbb{N}_0$ and $\alpha > -1$. Then, we have*

$$p_n^{\alpha-1}(x) = \frac{1}{\alpha + 2n}\left[(\alpha+n)p_n^\alpha(x) - n p_{n-1}^\alpha(x)\right]. \tag{6}$$

Moreover, the integral relations

$$\int_{-1}^{1} (1-x)^\alpha p_j^\alpha(x) p_l^\alpha(x)\,\mathrm{d}x = \rho_j^\alpha \delta_{jl}, \quad \textit{where } \rho_j^\alpha = \frac{2^{\alpha+1}}{2j+\alpha+1}, \tag{7}$$

$$\int_{-1}^{1} (1-x)^\alpha p_j^\beta(x) q_l(x)\,\mathrm{d}x = 0 \quad \forall q_l \in \mathbb{P}_l, \alpha-\beta \in \mathbb{N}_0, j > l+\alpha-\beta \tag{8}$$

are valid.

Proof. A proof can be found in [4].

The next lemma considers properties of the integrated Jacobi polynomials (5).

Lemma 2. *Let $l, j \in \mathbb{N}_0$. Let p_n^α and $\hat{p}_n^\alpha$ be defined via (4) and (5). Then, the identities*

$$\hat{p}_n^\alpha(x) = \frac{2n+2\alpha}{(2n+\alpha-1)(2n+\alpha)} p_n^\alpha(x) + \frac{2\alpha}{(2n+\alpha-2)(2n+\alpha)} p_{n-1}^\alpha(x)$$

$$-\frac{2n-2}{(2n+\alpha-1)(2n+\alpha-2)}p_{n-2}^{\alpha}(x), \quad n \geq 2, \tag{9}$$

$$\hat{p}_n^{\alpha}(x) = \frac{2}{2n+\alpha-1}\left(p_n^{\alpha-1}(x) + p_{n-1}^{\alpha-1}(x)\right), \quad n \geq 1 \tag{10}$$

are valid.

Proof. The proof can be found in [4].

Finally, we present two properties of the Jacobi polynomials which have not been presented in [4].

Lemma 3. *Let $l, j \in \mathbb{N}_0$. Let p_n^{α} and $\hat{p}_n^{\alpha}$ be defined via* (4) *and* (5)*. Then, the following assertions are valid for $\alpha > -1, j > 1$:*

$$(\alpha-1)\hat{p}_j^{\alpha}(y) = (1-y)p_{j-1}^{\alpha}(y) + 2p_j^{\alpha-2}(y), \tag{11}$$

$$\begin{aligned}(1-y)\left((2-2j)p_{j-2}^{\alpha}(y) + \alpha p_{j-1}^{\alpha}(y)\right) \\ +(\alpha+2j-2)(\alpha-1)\hat{p}_j^{\alpha}(y) = 4(\alpha+j-2)p_{j-1}^{\alpha-2}(y) + (2\alpha-4)p_j^{\alpha-2}(y).\end{aligned} \tag{12}$$

Proof. The proof can be found in [3].

3 The Triangular Case

In this section, we consider the case $d = 2$. Let $\hat{\triangle}_2$ be the reference triangle with the vertices $(-1,-1)$, $(1,-1)$ and $(0,1)$. We introduce

$$\phi_{ij}(x,y) = \hat{p}_i^0\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i \hat{p}_j^{2i}(y), \quad i+j \leq p, i \geq 2, j \geq 1, \tag{13}$$

as the interior bubble functions on $\hat{\triangle}_2$. This is the basis proposed in [7], whereas the basis with $\hat{p}_j^{2i-1}(y)$ instead of $\hat{p}_j^{2i}(y)$ in (13) has been investigated in [4]. The vertex functions and edge bubbles are taken from [4]. Let $A = \begin{bmatrix} a & b \\ b & c \end{bmatrix}$ and let

$$\hat{K}_2 = [a_{ij,kl}]_{(i,j);(k,l)} = \left[\int_{\hat{\triangle}_2} (\nabla\phi_{ij}(x,y))^T A \nabla\phi_{kl}(x,y)\, \mathrm{d}(x,y)\right]_{(i,j);(k,l)} \tag{14}$$

be the stiffness matrix on $\hat{\triangle}_2$ with respect to the basis (13).

Theorem 1. *Let $\hat{K}_2$ be defined via* (13)-(14)*. Then, the matrix $\hat{K}_2$ has $\mathcal{O}(p^2)$ nonzero matrix entries. More precisely, $a_{ij,kl} = 0$ if $|i-k| > 2$ or $|i-k+j-l| > 1$.*

Proof. First, we compute $\nabla\phi_{ij}$. A simple computation shows that

$$\nabla\phi_{ij} = \begin{bmatrix} p_{i-1}^0\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^{i-1}\hat{p}_j^{2i}(y) \\ \frac{1}{2}p_{i-2}^0\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^{i-1}\hat{p}_j^{2i}(y) + \hat{p}_i^0\left(\frac{2x}{1-y}\right)\left(\frac{1-y}{2}\right)^i p_{j-1}^{2i}(y)\end{bmatrix}, \tag{15}$$

see [4]. Let

$$a_{ij,kl}^{(y)} = \int_{\hat{\triangle}_2} \frac{\partial\phi_{ij}}{\partial y}(x,y)\frac{\partial\phi_{kl}}{\partial y}(x,y)\, \mathrm{d}(x,y). \tag{16}$$

Using (15) and the Duffy transform $z = \frac{2x}{1-y}$, we obtain

$$\begin{aligned} a^{(y)}_{ij,kl} &= \frac{1}{4} \int_{-1}^{1} p^0_{i-2}(z) p^0_{k-2}(z) \; \mathrm{d}z \int_{-1}^{1} \left(\frac{1-y}{2} \right)^{i+k-1} \hat{p}^{2i}_j(y) \hat{p}^{2k}_l(y) \; \mathrm{d}y \\ &+ \int_{-1}^{1} \hat{p}^0_i(z) \hat{p}^0_k(z) \; \mathrm{d}z \int_{-1}^{1} \left(\frac{1-y}{2} \right)^{i+k+1} p^{2i}_{j-1}(y) p^{2k}_{l-1}(y) \; \mathrm{d}y \\ &+ \frac{1}{2} \int_{-1}^{1} p^0_{i-2}(z) \hat{p}^0_k(z) \; \mathrm{d}z \int_{-1}^{1} \left(\frac{1-y}{2} \right)^{i+k} \hat{p}^{2i}_j(y) p^{2k}_{l-1}(y) \; \mathrm{d}y \\ &+ \frac{1}{2} \int_{-1}^{1} \hat{p}^0_i(z) p^0_{k-2}(z) \; \mathrm{d}z \int_{-1}^{1} \left(\frac{1-y}{2} \right)^{i+k} p^{2i}_{j-1}(y) \hat{p}^{2k}_l(y) \; \mathrm{d}y \\ &=: a^{(y,1)}_{ij,kl} + a^{(y,2)}_{ij,kl} + a^{(y,3)}_{ij,kl} + a^{(y,4)}_{ij,kl}. \end{aligned}$$

With (9) for $\alpha = 0$ we arrive at $a^{(y)}_{ij,kl} = 0$ if $|i-k| \notin \{0, 2\}$.

Let $k = i-2$. Then $a^{(y,1)}_{i,j,i-2,l}$ and $a^{(y,4)}_{i,j,i-2,l}$ vanish by repeated application of (7) and (9). The remaining integrals can be simplified using (9) and by (7) be evaluated to

$$\begin{aligned} \int_{-1}^{1} \hat{p}^0_i(z) \hat{p}^0_{i-2}(z) \; \mathrm{d}z &= -\frac{2}{(2i-1)(2i-3)(2i-5)}, \\ \int_{-1}^{1} p^0_{i-2}(z) \hat{p}^0_{i-2}(z) \; \mathrm{d}z &= \frac{2}{(2i-3)(2i-5)}. \end{aligned}$$

We insert now these relations into the expressions for $a^{(y,2)}_{ij,kl}$ and $a^{(y,3)}_{ij,kl}$ and use relation (11). Then, we obtain

$$\begin{aligned} a^{(y,2)}_{i,j,i-2,l} + a^{(y,3)}_{i,j,i-2,l} &= 2c_0 \int_{-1}^{1} \left(\frac{1-y}{2} \right)^{2i-1} p^{2i}_{j-1}(y) p^{2i-4}_{l-1}(y) \; \mathrm{d}y \\ & -(2i-1)c_0 \int_{-1}^{1} \left(\frac{1-y}{2} \right)^{2i-2} \hat{p}^{2i}_j(y) p^{2i-4}_{l-1}(y) \; \mathrm{d}y \end{aligned}$$

$$\begin{aligned} &= c_0 \int_{-1}^{1} \left(\frac{1-y}{2} \right)^{2i-2} \left[(1-y) p^{2i}_{j-1}(y) - (2i-1) \hat{p}^{2i}_j(y) \right] p^{2i-4}_{l-1}(y) \; \mathrm{d}y \\ &= 2c_0 \int_{-1}^{1} \left(\frac{1-y}{2} \right)^{2i-2} p^{2i-2}_j(y) p^{2i-4}_{l-1}(y) \; \mathrm{d}y \end{aligned}$$

with $c_0^{-1} = -(2i-1)(2i-3)(2i-5)$. Now, we apply (8) and obtain $a_{i,j,i-2,l} \neq 0$ if $-3 \leq j-l \leq -1$.

The case $k = i+2$ can be proved by the same arguments. For $i = k$, we investigate each term $a^{y,s}_{ij,kl}$, $s = 1, 2, 3, 4$ separately. Using (6)-(10), the assertion can be proved. This proof is similar to the proof given in [4].

Next, we consider

$$a^{(xy)}_{ij,kl} = \int_{\hat{\triangle}_2} \frac{\partial \phi_{ij}}{\partial x}(x,y) \frac{\partial \phi_{kl}}{\partial y}(x,y) \; \mathrm{d}(x,y). \tag{17}$$

Using (15) and the Duffy transform $z = \frac{2x}{1-y}$ again, we obtain

$$\begin{aligned} a_{ij,kl}^{(xy)} &= \frac{1}{2}\int_{-1}^{1} p_{i-1}^{0}(z)p_{k-2}^{0}(z)\,\mathrm{d}z \int_{-1}^{1}\left(\frac{1-y}{2}\right)^{i+k-1}\hat{p}_j^{2i}(y)\hat{p}_l^{2k}(y)\,\mathrm{d}y \\ &\quad + \int_{-1}^{1} p_{i-1}^{0}(z)\hat{p}_{k}^{0}(z)\,\mathrm{d}z \int_{-1}^{1}\left(\frac{1-y}{2}\right)^{i+k}\hat{p}_j^{2i}(y)p_{l-1}^{2k}(y)\,\mathrm{d}y. \end{aligned}$$

Using (7) and (9), $a_{ij,kl}^{(xy)} = 0$ if $|i-k| \neq 1$. Let $k = i+1$. Then, we have

$$\begin{aligned} a_{i,j,i+1,l}^{(xy)} &= \frac{1}{4i^2-1}\int_{-1}^{1}\left(\frac{1-y}{2}\right)^{2i}\hat{p}_j^{2i}(y) \\ &\qquad \times \left[(2i+1)\hat{p}_l^{2i+2}(y) - (1-y)p_{l-1}^{2i+2}(y)\right]\,\mathrm{d}y \\ &= -\frac{2}{4i^2-1}\int_{-1}^{1}\left(\frac{1-y}{2}\right)^{2i}\hat{p}_j^{2i}(y)p_l^{2i}(y)\,\mathrm{d}y. \end{aligned}$$

Finally, we apply (7) and (9) to obtain $a_{i,j,i+1,l}^{(xy)} \neq 0$ if $-2 \leq j-l \leq 0$. The case $k = i-1$ follows by the same arguments.

Remark 1. Since $a_{i,j,i-2,l}^{(y,2)} \neq 0$ for all $j > l$, the sparsity of $\hat{K}$ cannot be proved with a direct evaluation of $a_{i,j,i-2,l}^{(y,2)}$ and $a_{i,j,i-2,l}^{(y,3)}$. Only for $i = k$, the terms $a_{i,j,i,l}^{(y,s)}$, $s = 1,2,3,4$, can be considered separately.

4 The Tetrahedral Case

Let $\hat{\triangle}_3$ be the reference tetrahedron with the vertices $(-1,-1,-1)$, $(1,-1,-1)$, $(0,1,-1)$ and $(0,0,1)$. The interior bubbles are

$$\begin{aligned} \phi_{ijk}(x,y,z) &= \hat{p}_i^0\left(\frac{4x}{1-2y-z}\right)\left(\frac{1-2y-z}{4}\right)^i \hat{p}_j^{2i}\left(\frac{2y}{1-z}\right) \\ &\qquad \times \left(\frac{1-z}{2}\right)^j \hat{p}_k^{2i+2j}(z), \quad i+j+k \leq p, i \geq 2, j,k \geq 1. \end{aligned} \tag{18}$$

The vertex functions, edge bubbles and face bubbles are taken from [3]. Let $\hat{\mathcal{A}}_3 = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{12} & a_{22} & a_{23} \\ a_{13} & a_{23} & a_{33} \end{bmatrix} \in \mathbb{R}^{3\times 3}$ be a diffusion matrix with constant coefficients. We introduce

$$\hat{K}_3 = [a_{ijk,i'j'k'}]_{i,i'=2,j,j',k,k'=1}^{i+j+k=p,i'+j'+k'=p} = \left[\int_{\hat{\triangle}_3} (\nabla\phi_{ijk})^T \hat{\mathcal{A}}_3 \nabla\phi_{i'j'k'}\right]_{i,i',j,j',k,k'} \tag{19}$$

as the part of the stiffness matrix that corresponds to the interior bubbles (18).

Theorem 2. *Let $\hat{K}_3$ be defined via* (19)*. Then, the matrix has $\frac{(p-1)(p-2)(p-3)}{6}$ rows and columns. Moreover, the entry $a_{ijk,i'j'k'}$ of the matrix $\hat{K}_3$ is zero if $|i-i'| > 2$, or $|i+j-i'-j'| > 3$, or $|i+j+k-i'-j'-k'| > 2$.*

Proof. The proofs for $\int_{\hat{\triangle}_3} \frac{\partial\phi_{ijk}}{\partial x}\frac{\partial\phi_{i'j'k'}}{\partial x}$ and $\int_{\hat{\triangle}_3} \frac{\partial\phi_{ijk}}{\partial y}\frac{\partial\phi_{i'j'k'}}{\partial y}$ are similar to the triangular case. For the computation of $\int_{\hat{\triangle}_3} \frac{\partial\phi_{ijk}}{\partial z}\frac{\partial\phi_{i'j'k'}}{\partial z}$, 16 different integrals have to be considered, see [3]. For $i = i'$, all y-integrals can be considered separately, whereas for $|i - i'| = 2$ we collect several integrals and use relation (12). To illustrate this procedure we consider the following three integrands,

$$\begin{aligned}
\hat{\mathcal{I}}^{(14)} &= -(j-1)c_1\, \hat{p}_i^0(x)\hat{p}_{i'}^0(x)\; w_{\gamma_y}(y)\, p_{j-2}^{2i}(y)p_{j'-1}^{2i'}(y) \\
&\quad \times w_{\gamma_z}(z)\, \hat{p}_k^{2i+2j}(z)\hat{p}_{k'}^{2i'+2j'}(z), \\
\hat{\mathcal{I}}^{(15)} &= i\, i'\; c_1\hat{p}_i^0(x)\hat{p}_{i'}^0(x)\; w_{\gamma_y}(y)\, p_{j-1}^{2i}(y)p_{j'-1}^{2i'}(y)\; w_{\gamma_z}(z)\, \hat{p}_k^{2i+2j}(z)\hat{p}_{k'}^{2i'+2j'}(z), \\
\hat{\mathcal{I}}^{(17)} &= -i'c_1\; p_{i-2}^0(x)\hat{p}_{i'}^0(x)\; w_{\gamma_y-1}(y)\, \hat{p}_j^{2i}(y)p_{j'-1}^{2i'}(y) \\
&\quad \times w_{\gamma_z}(z)\, \hat{p}_k^{2i+2j}(z)\hat{p}_{k'}^{2i'+2j'}(z),
\end{aligned}$$

where $w_\gamma(\zeta) = \left(\frac{1-\zeta}{2}\right)^\gamma$ is the weight function for Jacobi polynomials $p_n^\gamma(\zeta)$ and $\gamma_y = i + i' + 1$, resp. $\gamma_z = i + j + i' + j'$. The constant c_1 is given by $c_1^{-1} = 4(i+j-1)(i'+j'-1)$. The numbering of the terms $\hat{\mathcal{I}}^{(14)}$, $\hat{\mathcal{I}}^{(15)}$, and $\hat{\mathcal{I}}^{(17)}$ corresponds to the numbering in [3]. These integrands are obtained after taking the partial derivative in z-direction and performing the corresponding Duffy transformations, compare [3]. The x-integrals can be evaluated as in the triangular case and for $i' = i - 2$ one obtains for $h(y,z) = \int_{-1}^{1} \hat{\mathcal{I}}^{(14)} + \hat{\mathcal{I}}^{(15)} + \hat{\mathcal{I}}^{(17)}\, \mathrm{d}x$, the equation

$$\begin{aligned}
h(y,z) =& c_2\; w_{2i-2}(y) \Big[(1-y)\Big((j-1)p_{j-2}^{2i}(y) - ip_{j-1}^{2i}(y)\Big) \\
& \quad -(2i-1)(i+j-1)\hat{p}_j^{2i}(y)\Big]\, p_{j'-1}^{2i-4}(y) w_{\gamma_z}(z)\hat{p}_k^{2i+2j}(z)\hat{p}_{k'}^{2i'+2j'}(z) \\
=:& c_2\, h_1(y)\, w_{\gamma_z}(z)\hat{p}_k^{2i+2j}(z)\hat{p}_{k'}^{2i'+2j'}(z),
\end{aligned}$$

where $c_2^{-1} = 4(2i-5)(2i-3)(2i-1)(i+j-3)(i+j-1)/(i-2)$. The straightforward approach is to evaluate these integrals by exploiting the orthogonality relation (7). To do so, one has to rewrite all polynomials in terms of Jacobi polynomials with parameter α corresponding to the appearing weight, i.e. $\alpha = 2i - 1$ for the first two summands and $\alpha = 2i - 2$ for the third one. This can easily be achieved for $p_{j'-1}^{2i-4}(y)$ using identity (6) recursively. In order to expand $p_{j-2}^{2i}(y), p_{j-1}^{2i}(y)$ and $\hat{p}_j^{2i}(y)$ in the basis of Jacobi polynomials $p_m^{2i-1}(y)$ we need the coefficients a_m, b_m of

$$p_n^{2i}(y) = \sum_{m=0}^{n} a_m\, p_m^{2i-1}(y), \qquad \text{resp.} \qquad \hat{p}_n^{2i}(y) = \sum_{m=0}^{n} b_m\, p_m^{2i-1}(y).$$

But for both transformations all $n + 1$ coefficients are nonzero. Hence we consider these three integrals together and rewrite the expression in angular brackets using identity (12) yielding,

$$\begin{aligned}
h_1(y) &= \Big[(1-y)\Big((j-1)\, p_{j-2}^{2i}(y) - i\, p_{j-1}^{2i}(y)\Big) \\
&\quad -(2i-1)(i+j-1)\, \hat{p}_j^{2i}(y)\Big]\, p_{j'-1}^{2i-4}(y) \\
&= \Big[2(2i+j-2)\, p_{j-1}^{2i-2}(y) + (2i-2)\, p_j^{2i-2}(y)\Big]\, p_{j'-1}^{2i-4}(y).
\end{aligned}$$

Using this substitution the y-integrand of $h(y,z)$, $h_1(y)$ has the following form,

$$h_1(y) = w_{2i-2}(y) \Big[(2i+j-2)\, p_{j-1}^{2i-2}(y) + (i-1)\, p_j^{2i-2}(y)\Big]\, p_{j'-1}^{2i-4}(y). \qquad (20)$$

Finally we use identity (6) twice on $p_{j'-1}^{2i-4}(y)$,

$$\begin{aligned} p_{j'-1}^{2i-4}(y) = & \frac{(j'-2)(j'-1)}{2(i+j'-3)(2i+2j'-5)}\, p_{j'-3}^{2i-2}(y) \\ & - \frac{(j'-1)(2i+j'-4)}{2(i+j'-3)(i+j'-2)}\, p_{j'-2}^{2i-2}(y) \\ & + \frac{(2i+j'-4)(2i+j'-3)}{2(i+j'-2)(2i+2j'-5)}\, p_{j'-1}^{2i-2}(y). \end{aligned}$$

Hence we have to evaluate integrals of the form

$$\int_{-1}^{1} w_{2i-2}(y)\, p_m^{2i-2}(y) p_{m'}^{2i-2}(y)\ \mathrm{d}y,$$

where the polynomial degrees range from $m \in \{j-1, j\}$ and $m' \in \{j'-3, j'-2, j'-1\}$. Now by orthogonality relation (7) it easily follows that the integral over (20) is nonzero only for $j' = j, j+1, j+2, j+3$.
The evaluation of the z-integrals can be done by the same procedure as in the triangular case. To finish the proof one has to consider also the off-diagonal terms, i.e. integrals of the form $\int_{\hat{\triangle}_3} \frac{\partial \phi_{ijk}}{\partial \eta} \frac{\partial \phi_{i'j'k'}}{\partial \zeta}$, $\eta, \zeta \in \{x, y, z\}$, $\eta \neq \zeta$. These integrals can be treated in complete analogy.

Our practical computations were performed using a program written in the environment of the computer algebra software Mathematica. A description of the applied algorithm can be found in [3].

Acknowledgement. The work of the second named author has been supported by the Austrian Science Fund FWF through SFB F013/01.

References

[1] M. Abramowitz and I. Stegun. *Handbook of Mathematical Functions.* Dover Publications, 1965.

[2] G.E. Andrews, R. Askey, and R. Roy. Special Functions. In *Encyclopedia of Mathematics and its Applications*, volume 71. Cambridge University Press, 1999.

[3] S. Beuchler and V. Pillwein. Shape functions for tetrahedral p-FEM using integrated Jacobi polynomials. Technical Report 2006-34, SFB F013, JKU Linz, 2006.

[4] S. Beuchler and J. Schöberl. New shape functions for triangular p-FEM using integrated Jacobi polynomials. *Numer. Math.*, 103:339–366, 2006.

[5] C. Schwab. *$p-$ and $hp-$Finite Element Methods. Theory and Applications in Solid and Fluid Mechanics.* Clarendon Press, Oxford, 1998.

[6] S.J. Sherwin. Hierarchical hp finite elements in hybrid domains. *Finite Elem. Anal. Des.*, 27:109–119, 1997.

[7] S.J. Sherwin and G.E. Karniadakis. A new triangular and tetrahedral basis for high-order finite element methods. *Internat. J. Numer. Methods Engrg.*, 38:3775–3802, 1995.

[8] P. Solin, K. Segeth, and I. Dolezel. *Higher-Order Finite Element Methods*. Chapman and Hall, 2004.

[9] F.G. Tricomi. *Vorlesungen über Orthogonalreihen.* Springer, Berlin, Göttingen and Heidelberg, 1955.

Finite Volume Method for Nonlinear Transmission Problems

Franck Boyer and Florence Hubert

Université de Provence, Centre de Mathématiques et Informatique.
39, rue F. Joliot Curie, 13453 Marseille Cedex 13, France.
{fboyer,fhubert}@cmi.univ-mrs.fr

1 Introduction

Discrete Duality Finite Volume (DDFV) schemes have recently been developed to approximate monotone nonlinear diffusion problems

$$-\mathrm{div}(\boldsymbol{\varphi}(z, \nabla u(z))) = f(z), \quad \text{in } \Omega,\ u = 0, \quad \text{on } \partial\Omega, \tag{1}$$

on general 2D grids. The principle of such schemes is to introduce discrete unknowns both at centers and vertices of any given primal mesh. A discrete gradient operator is then built over the diamond cells associated to the mesh and finally, the discrete flux balance equations are written on the primal and dual control volumes (see Section 2). The main advantages of this approach is that few geometric assumptions are needed for the grid (non conformal grids are allowed for instance), and that the discrete problem inherits the main properties (monotonicity, symmetry, ...) of the continuous one. In [1], it is proved that the scheme is well-posed and convergent. Under suitable regularity assumptions on $\boldsymbol{\varphi}$ and u, some error estimates are also obtained.

Application of these schemes to nonlinear transmission problems, that is when $\boldsymbol{\varphi}$ presents some discontinuities with respect to the space variable z, were first investigated in [2] in the case where uniform growth and coercivity conditions for $\xi \mapsto \boldsymbol{\varphi}(z, \xi)$ are assumed to hold over the domain.

We propose here to generalize this analysis to the case where these growth and coercivity conditions are no more uniform on the domain. We can imagine for instance that $\boldsymbol{\varphi}$ is linear with respect to ξ on a subdomain and fully nonlinear on its complementary. Such situations arise for instance in bimaterial problems in elastic-plastic mechanics (see [5, 8, 9]).

Let us precise the situation under study. Let Ω be a bounded polygonal open set in $\mathbb{R}^2$, split into N open polygonal subdomains Ω_i :

$$\overline{\Omega} = \cup_{i=1}^{N} \overline{\Omega_i},\ \Omega_i \cap \Omega_j = \emptyset \text{ if } i \neq j,$$

and that $\boldsymbol{\varphi} : \Omega \times \mathbb{R}^2 \to \mathbb{R}^2$ in equation (1) is a Caratheodory function, constant with respect to z on each Ω_i: $\boldsymbol{\varphi}(z, \xi) = \varphi_i(\xi)$, for all $z \in \Omega_i$ and $\xi \in \mathbb{R}^2$. There exists a family $\boldsymbol{p} = (p_i)_{\{i=1,\cdots,N\}}$, $p_i \in]1, \infty[$ and a constant $C_\varphi > 0$ such that

- Monotonicity on each subdomain Ω_i: for all $(\xi,\eta) \in \mathbb{R}^2 \times \mathbb{R}^2$

$$\begin{aligned}(\varphi_i(\xi)-\varphi_i(\eta),\xi-\eta) &\geq C_\varphi|\xi-\eta|^2(1+|\xi|^{p_i}+|\eta|^{p_i})^{\frac{p_i-2}{p_i}}, \text{ if } p_i \leq 2.\\ (\varphi_i(\xi)-\varphi_i(\eta),\xi-\eta) &\geq C_\varphi|\xi-\eta|^{p_i}, \text{ if } p_i > 2.\end{aligned} \tag{$\mathcal{H}_1$}$$

- Coercivity on each subdomain Ω_i: for all $\xi \in \mathbb{R}^2$

$$(\varphi_i(\xi),\xi) \geq C_\varphi(|\xi|^{p_i}-1). \tag{$\mathcal{H}_2$}$$

- Growth conditions : for all $(\xi,\eta) \in \mathbb{R}^2 \times \mathbb{R}^2$,

$$\begin{aligned}&|\varphi_i(\xi)-\varphi_i(\eta)| \leq C_\varphi|\xi-\eta|^{p_i-1}, \text{ if } p_i \leq 2,\\ &|\varphi_i(\xi)-\varphi_i(\eta)| \leq C_\varphi\left(1+|\xi|^{p_i-2}+|\eta|^{p_i-2}\right)|\xi-\eta|, \text{ if } p_i > 2.\end{aligned} \tag{$\mathcal{H}_3$}$$

Remark that assumption $(\mathcal{H}_3)$ implies that

$$|\varphi_i(\xi)| \leq C_\varphi(|\xi|^{p_i-1}+1), \quad \forall \xi \in \mathbb{R}^2. \tag{$\mathcal{H}_4$}$$

We introduce $L^{\boldsymbol{p}}(\Omega) = \{u/u_{|\Omega_i} \in L^{p_i}(\Omega_i)\}$, $W_0^{1,\boldsymbol{p}}(\Omega) = \{u \in W_0^{1,1}(\Omega)/\nabla u \in (L^{\boldsymbol{p}}(\Omega))^2\}$, and for $\boldsymbol{q} = (q_i)_{i=1,\cdots,N}$, we denote $\|u\|_{L^{\boldsymbol{p}}}^{\boldsymbol{q}} = \sum_{i=1}^{N} \|u_{|\Omega_i}\|_{L^{p_i}(\Omega_i)}^{q_i}$. We finally note $\boldsymbol{p}_{\min} = \min(p_i)$ and $\boldsymbol{p}_{\max} = \max(p_i)$.

Theorem 1. *Under assumptions* $(\mathcal{H}_1)$, $(\mathcal{H}_2)$, $(\mathcal{H}_4)$, *the problem* (1) *admits for all* $f \in L^{\boldsymbol{p}'_{\min}}(\Omega)$ *a unique solution* $u \in W_0^{1,\boldsymbol{p}}(\Omega)$. *(See [8].)*

These problems can be approximated either by finite element method, whose study is undertaken in particular in [9], or by the m-DDFV ("modified" Discrete Duality Finite Volume) method developed for non-linear elliptic equations with discontinuities in [2].

2 The m-DDFV Scheme

Let $\mathfrak{M}_i$ be a finite volume mesh on Ω_i for $i = 1,\cdots,N$ and $\mathfrak{M} = \cup_{i=1}^{N}\mathfrak{M}_i$. Note that the mesh $\mathfrak{M}$ can present non standard edges in particular on the boundaries $\partial\Omega_i \cap \partial\Omega_j$. We associate to each control volume $\mathcal{K} \in \mathfrak{M}$ a point $x_{\mathcal{K}} \in \mathcal{K}$, called the center. Let $\mathfrak{M}^*$ be the dual mesh of $\mathfrak{M}$, that is the mesh whose control volumes $\mathcal{K}^* \in \mathfrak{M}^*$ are obtained by joining the centers of control volumes around a vertex $x_{\mathcal{K}^*}$ (see Fig. 1). Note $\mathcal{T} = (\mathfrak{M},\mathfrak{M}^*)$. The DDFV methods involve both unknowns $(u_{\mathcal{K}}) \in \mathbb{R}^{\mathfrak{M}}$ on $\mathfrak{M}$ and $(u_{\mathcal{K}^*}) \in \mathbb{R}^{\mathfrak{M}^*}$ on $\mathfrak{M}^*$, we note $u^{\mathcal{T}} = (u_{\mathcal{K}},u_{\mathcal{K}^*}) \in \mathbb{R}^{\mathfrak{M}} \times \mathbb{R}^{\mathfrak{M}^*}$. Integrating equation (1) on both $\mathcal{K} \in \mathfrak{M}$ and $\mathcal{K}^* \in \mathfrak{M}^*$, the classical DDFV scheme consists in approaching the nonlinear fluxes $\int_{\partial\mathcal{K}} (\boldsymbol{\varphi}(z,\nabla u(z)),\boldsymbol{\nu}_{\mathcal{K}})\,dz$ and $\int_{\partial\mathcal{K}^*} (\boldsymbol{\varphi}(z,\nabla u(z)),\boldsymbol{\nu}_{\mathcal{K}^*})\,dz$ by using a discrete gradient $\nabla^{\mathcal{T}} u^{\mathcal{T}}$, piecewise constant on a partition $\mathfrak{D} = (\mathcal{D})_{\mathcal{D}\in\mathfrak{D}}$ called the diamond cells, and $\varphi_{\mathcal{D}}(\nabla_{\mathcal{D}}^{\mathcal{T}} u^{\mathcal{T}}) = \frac{1}{|\mathcal{D}|}\int_{\mathcal{D}} \boldsymbol{\varphi}(z,\nabla_{\mathcal{D}}^{\mathcal{T}} u^{\mathcal{T}})\,dz$. Each diamond cell is a quadrangle whose diagonals are some edge $\sigma = \mathcal{K}|\mathcal{L}$ and the edge $\sigma^* = (x_{\mathcal{K}},x_{\mathcal{L}})$. The set $\mathfrak{D}_{\Gamma_{ij}}$ specifies the

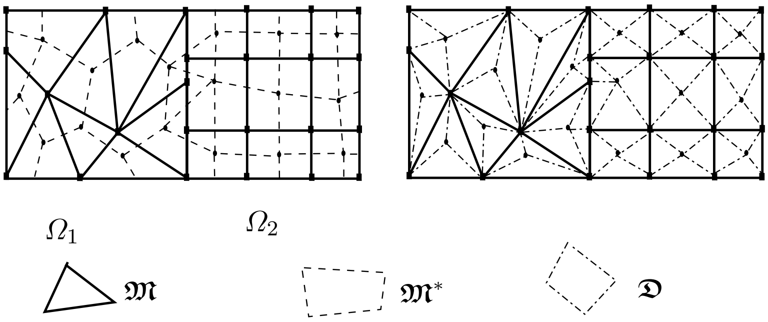


Fig. 1. The three meshes $\mathfrak{M}$, $\mathfrak{M}^*$, $\mathfrak{D}$

diamond cells lying across two distinct subdomains Ω_i and Ω_j and $\mathfrak{D}_\Gamma = \cup_{\substack{i,j \\ i\neq j}} \mathfrak{D}_{\Gamma_{ij}}$.

The discrete gradient introduced in [3, 7, 4] reads

$$\nabla^{\mathcal{T}} u^{\mathcal{T}} = \sum_{\mathcal{D}\in\mathfrak{D}} \nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} 1_{\mathcal{D}}, \ \nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} = \frac{1}{\sin\alpha_{\mathcal{D}}} \left(\frac{u_{\mathcal{L}} - u_{\mathcal{K}}}{|\sigma^*|} \boldsymbol{\nu} + \frac{u_{\mathcal{L}^*} - u_{\mathcal{K}^*}}{|\sigma|} \boldsymbol{\nu}^* \right) \tag{2}$$

with the notations of Fig. 2. The DDFV scheme is then defined by

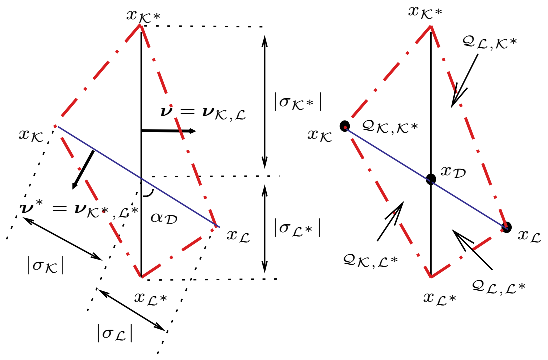

$x_{\mathcal{K}^*}$
$\boldsymbol{\nu} = \boldsymbol{\nu}_{\mathcal{K},\mathcal{L}}$
$|\sigma_{\mathcal{K}^*}|$
$x_{\mathcal{K}}$
$\boldsymbol{\nu}^* = \boldsymbol{\nu}_{\mathcal{K}^*,\mathcal{L}^*}$
$\alpha_{\mathcal{D}}$
$x_{\mathcal{L}}$
$|\sigma_{\mathcal{L}^*}|$
$|\sigma_{\mathcal{K}}|$
$x_{\mathcal{L}^*}$
$|\sigma_{\mathcal{L}}|$
$x_{\mathcal{K}^*}$
$\mathcal{Q}_{\mathcal{L},\mathcal{K}^*}$
$x_{\mathcal{K}}$
$\mathcal{Q}_{\mathcal{K},\mathcal{K}^*}$
$x_{\mathcal{D}}$
$x_{\mathcal{L}}$
$\mathcal{Q}_{\mathcal{K},\mathcal{L}^*}$
$\mathcal{Q}_{\mathcal{L},\mathcal{L}^*}$
$x_{\mathcal{L}^*}$

Fig. 2. Notations in a diamond cell $\mathcal{D} = \cup_{\mathcal{Q}\in\mathfrak{Q}_{\mathcal{D}}} \mathcal{Q}$

$$\begin{cases} -\sum_{\mathcal{D}_{\sigma,\sigma^*}\cap\mathcal{K}\neq\emptyset} |\sigma| \left(\varphi_{\mathcal{D}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}}), \boldsymbol{\nu}_{\mathcal{K}}\right) = \int_{\mathcal{K}} f(z)\,dz, \quad \forall\mathcal{K}\in\mathfrak{M}, \\ -\sum_{\mathcal{D}_{\sigma,\sigma^*}\cap\mathcal{K}^*\neq\emptyset} |\sigma^*| \left(\varphi_{\mathcal{D}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}}), \boldsymbol{\nu}_{\mathcal{K}^*}\right) = \int_{\mathcal{K}^*} f(z)\,dz, \forall\mathcal{K}^*\in\mathfrak{M}^*, \end{cases} \tag{3}$$

and admits a unique solution. Convergence and error estimates in that case are given in [1]. These error estimates are no more valid as soon as $\varphi_i \neq \varphi_j$, since we loose the consistency of the nonlinear fluxes across the edges on $\partial\Omega_i \cap \partial\Omega_j$. To tackle this problem, we proposed in [2] in the case $p_i = p_j, \forall i, \forall j$ to change the approximation of the nonlinearity on the diamond cells $\varphi_{\mathcal{D}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}})$ into $\boldsymbol{\varphi}^{\mathcal{N}}_{\mathcal{D}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}})$ in such a way that enforce the consistency of the fluxes across all the edges. The new scheme reads

$$\begin{cases} -\sum_{\mathcal{D}_{\sigma,\sigma^*}\cap\mathcal{K}\neq\emptyset} |\sigma| \left(\boldsymbol{\varphi}^{\mathcal{N}}_{\mathcal{D}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}}), \boldsymbol{\nu}_{\mathcal{K}}\right) = \int_{\mathcal{K}} f(z)\,dz, \quad \forall \mathcal{K}\in\mathfrak{M}, \\ -\sum_{\mathcal{D}_{\sigma,\sigma^*}\cap\mathcal{K}^*\neq\emptyset} |\sigma^*| \left(\boldsymbol{\varphi}^{\mathcal{N}}_{\mathcal{D}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}}), \boldsymbol{\nu}_{\mathcal{K}^*}\right) = \int_{\mathcal{K}^*} f(z)\,dz, \forall \mathcal{K}^*\in\mathfrak{M}^*. \end{cases} \tag{4}$$

To define $\boldsymbol{\varphi}^{\mathcal{N}}_{\mathcal{D}}(\nabla^{\mathcal{T}} u^{\mathcal{T}})$, we introduce a new discrete gradient constant on the quarters $(\mathcal{Q})_{\mathfrak{Q}}$ of the diamond cells (see Fig. 2)

$$\nabla^{\mathcal{N}} u^{\mathcal{T}} = \sum_{\mathcal{Q}\in\mathfrak{Q}} \nabla^{\mathcal{N}}_{\mathcal{Q}} u^{\mathcal{T}},\; \nabla^{\mathcal{N}}_{\mathcal{D}} u^{\mathcal{T}} = \sum_{\mathcal{Q}\in\mathfrak{Q}_{\mathcal{D}}} 1_{\mathcal{Q}} \nabla^{\mathcal{N}}_{\mathcal{Q}} u^{\mathcal{T}},$$

with $\nabla^{\mathcal{N}}_{\mathcal{Q}} u^{\mathcal{T}} = \nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}}\delta_{\mathcal{D}}$, where $\delta_{\mathcal{D}} \in \mathbb{R}^{n_{\mathcal{D}}}$ are artificial unknowns ($n_{\mathcal{D}} = 4$ for interior diamond cells and $n_{\mathcal{D}} = 1$ for boundary diamond cells) and $(B_{\mathcal{Q}})_{\mathcal{Q}\in\mathfrak{Q}}$ a set of $2\times n_{\mathcal{D}}$ matrices defined for interior diamond cells by

$$B_{\mathcal{Q}_{\mathcal{K},\mathcal{K}^*}} = \frac{1}{|\mathcal{Q}_{\mathcal{K},\mathcal{K}^*}|}\left(|\sigma_{\mathcal{K}}|\boldsymbol{\nu}^*, 0, |\sigma_{\mathcal{K}^*}|\boldsymbol{\nu}, 0\right),\; B_{\mathcal{Q}_{\mathcal{L},\mathcal{L}^*}} = \frac{1}{|\mathcal{Q}_{\mathcal{L},\mathcal{L}^*}|}\left(0, -|\sigma_{\mathcal{L}}|\boldsymbol{\nu}^*, 0, -|\sigma_{\mathcal{L}^*}|\boldsymbol{\nu}\right),$$

$$B_{\mathcal{Q}_{\mathcal{K},\mathcal{L}^*}} = \frac{1}{|\mathcal{Q}_{\mathcal{K},\mathcal{L}^*}|}\left(-|\sigma_{\mathcal{K}}|\boldsymbol{\nu}^*, 0, 0, |\sigma_{\mathcal{L}^*}|\boldsymbol{\nu}\right),\; B_{\mathcal{Q}_{\mathcal{L},\mathcal{K}^*}} = \frac{1}{|\mathcal{Q}_{\mathcal{L},\mathcal{K}^*}|}\left(0, |\sigma_{\mathcal{L}}|\boldsymbol{\nu}^*, -|\sigma_{\mathcal{K}^*}|\boldsymbol{\nu}, 0\right).$$

Note that $B_{\mathcal{Q}}$ depends only on the geometry of the diamond cell under study.

For $\mathcal{Q}\subset\Omega_i$, we note $\boldsymbol{\varphi}_{\mathcal{Q}}(\xi) = \varphi_i(\xi)$ and

$$\boldsymbol{\varphi}^{\mathcal{N}}_{\mathcal{D}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}}) = \frac{1}{|\mathcal{D}|}\sum_{\mathcal{Q}\in\mathfrak{Q}_{\mathcal{D}}} |\mathcal{Q}|\boldsymbol{\varphi}_{\mathcal{Q}}(\nabla^{\mathcal{N}}_{\mathcal{Q}} u^{\mathcal{T}}). \tag{5}$$

For each $\mathcal{D}\in\mathfrak{D}$, we choose $\delta_{\mathcal{D}}\in\mathbb{R}^{n_{\mathcal{D}}}$ such that, the conservativity of the fluxes is achieved, that is

$$\begin{aligned} \left(\boldsymbol{\varphi}_{\mathcal{Q}_{\mathcal{K},\mathcal{K}^*}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}_{\mathcal{K},\mathcal{K}^*}}\delta_{\mathcal{D}}), \boldsymbol{\nu}^*\right) &= \left(\boldsymbol{\varphi}_{\mathcal{Q}_{\mathcal{K},\mathcal{L}^*}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}_{\mathcal{K},\mathcal{L}^*}}\delta_{\mathcal{D}}), \boldsymbol{\nu}^*\right) \\ \left(\boldsymbol{\varphi}_{\mathcal{Q}_{\mathcal{L},\mathcal{K}^*}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}_{\mathcal{L},\mathcal{K}^*}}\delta_{\mathcal{D}}), \boldsymbol{\nu}^*\right) &= \left(\boldsymbol{\varphi}_{\mathcal{Q}_{\mathcal{L},\mathcal{L}^*}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}_{\mathcal{L},\mathcal{L}^*}}\delta_{\mathcal{D}}), \boldsymbol{\nu}^*\right) \\ \left(\boldsymbol{\varphi}_{\mathcal{Q}_{\mathcal{K},\mathcal{K}^*}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}_{\mathcal{K},\mathcal{K}^*}}\delta_{\mathcal{D}}), \boldsymbol{\nu}\right) &= \left(\boldsymbol{\varphi}_{\mathcal{Q}_{\mathcal{L},\mathcal{K}^*}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}_{\mathcal{L},\mathcal{K}^*}}\delta_{\mathcal{D}}), \boldsymbol{\nu}\right) \\ \left(\boldsymbol{\varphi}_{\mathcal{Q}_{\mathcal{K},\mathcal{L}^*}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}_{\mathcal{K},\mathcal{L}^*}}\delta_{\mathcal{D}}), \boldsymbol{\nu}\right) &= \left(\boldsymbol{\varphi}_{\mathcal{Q}_{\mathcal{L},\mathcal{L}^*}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}} + B_{\mathcal{Q}_{\mathcal{L},\mathcal{L}^*}}\delta_{\mathcal{D}}), \boldsymbol{\nu}\right). \end{aligned} \tag{6}$$

We then only have to solve for each diamond cell in $\mathfrak{D}_{\Gamma}$ a nonlinear problem and $\nabla^{\mathcal{N}}_{\mathcal{D}} u^{\mathcal{T}}$ can be seen as a nonlinear implicit function of $\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}}$. Note that $\delta_{\mathcal{D}} = 0$ as soon as $\mathcal{D}\subset\Omega_i$ for some $i = 1,\cdots,N$.

Theorem 2. *Under assumptions* $(\mathcal{H}_1)$*-*$(\mathcal{H}_3)$*, for all* $u^{\mathcal{T}}\in\mathbb{R}^{\mathcal{T}}$ *and all diamond cell* $\mathcal{D}$*, there exists a unique* $\delta_{\mathcal{D}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T}})\in\mathbb{R}^{n_{\mathcal{D}}}$ *satisfying* (6)*. The scheme* (4)*-*(6) *admits a unique solution.*

For simplicity we state here error estimates obtained when u belongs to the space $E = \{u\in\mathcal{C}(\bar{\Omega}),\, u\in\mathcal{C}^2(\Omega_i)\,\forall i\}$, even though the result can be extended to the case where $u_{|_{\Omega_i}}\in W^{2,p_i}(\Omega_i)$. We consider a family of meshes with convex diamond cells. The geometrical regularity of the meshes is controlled by a quantity denoted by $\mathrm{reg}(\mathcal{T})$, see [2] for more details.

Theorem 3. *Assume that the flux $\boldsymbol{\varphi}$ satisfies* $(\mathcal{H}_1)$-$(\mathcal{H}_3)$. *Let* $f \in L^{\boldsymbol{p}'_{\min}}(\Omega)$ *and assume that the solution* u *to* (1) *belongs to* E.

There exists $C > 0$ *depending on* u, *on* $\|f\|_{L^{\boldsymbol{p}'_{\min}}}$ *and on* $\mathrm{reg}(\mathcal{T})$ *such that*

$$\|u - u^{\mathcal{T}}\|^{\mathbf{2}}_{L^{\boldsymbol{p}}} + \|\nabla u - \nabla^{\mathcal{N}} u^{\mathcal{T}}\|^{\mathbf{2}}_{L^{\boldsymbol{p}}} \leq C\,\mathrm{size}(\mathcal{T})^{2(\boldsymbol{p}_{\min}-1)},\ \textit{if}\ \boldsymbol{p}_{\max} \leq 2$$

$$\|u - u^{\mathcal{T}}\|^{\boldsymbol{p}}_{L^{\boldsymbol{p}}} + \|\nabla u - \nabla^{\mathcal{N}} u^{\mathcal{T}}\|^{\boldsymbol{p}}_{L^{\boldsymbol{p}}} \leq C\,\mathrm{size}(\mathcal{T})^{\frac{\boldsymbol{p}_{\max}}{\boldsymbol{p}_{\max}-1}},\ \textit{if}\ \boldsymbol{p}_{\min} \geq 2.$$

As usual, these error estimates (which do not use any geometric assumptions on the solution) are pessimistic and numerical results given in Section 3 show that we can expect a much better behavior of these schemes.

Theorems 2 and 3 can be proved by following similar arguments than the ones presented in [2] for the case $p_i = p_j$, $\forall i, \forall j$.

3 Numerical Results

3.1 An Iterative Method to Solve the m-DDFV Scheme

We propose to solve the fully nonlinear discrete problem (4)-(6) by the following decomposition-coordination algorithm (see [6, 2]). Let $\mathcal{A} = (A_{\mathcal{Q}})_{\mathcal{Q}\in\mathfrak{Q}}$ be a family of definite positive 2×2 matrices, playing the role of heterogeneous and anisotropic augmented parameters and $\gamma \in \left]0, \frac{1+\sqrt{5}}{2}\right]$. The algorithm acts in three steps:

- Step 1: Find $(u^{\mathcal{T},n}, \delta^n_{\mathcal{D}})$ solution of

$$\sum_{\mathcal{Q}\in\mathfrak{Q}} |\mathcal{Q}| A_{\mathcal{Q}}(\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T},n} + B_{\mathcal{Q}}\delta^n_{\mathcal{D}} - g^{n-1}_{\mathcal{Q}}, \nabla^{\mathcal{T}}_{\mathcal{D}} v^{\mathcal{T}}) \tag{7}$$

$$= \frac{1}{2}\sum_{\mathcal{K}} |\mathcal{K}| f_{\mathcal{K}} v_{\mathcal{K}} + \frac{1}{2}\sum_{\mathcal{K}^*} |\mathcal{K}^*| f_{\mathcal{K}^*} v_{\mathcal{K}^*} + \sum_{\mathcal{Q}\in\mathfrak{Q}} |\mathcal{Q}| (\lambda^{n-1}_{\mathcal{Q}}, \nabla^{\mathcal{T}}_{\mathcal{D}} v),\ \forall v^{\mathcal{T}} \in \mathbb{R}^{\mathcal{T}}.$$

$$\sum_{\mathcal{Q}\in\mathfrak{Q}_{\mathcal{D}}} |\mathcal{Q}|^t B_{\mathcal{Q}} A_{\mathcal{Q}}(B_{\mathcal{Q}}\delta^n_{\mathcal{D}} + \nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T},n} - g^{n-1}_{\mathcal{Q}}) - \sum_{\mathcal{Q}\in\mathfrak{Q}_{\mathcal{D}}} |\mathcal{Q}|^t B_{\mathcal{Q}} \lambda^{n-1}_{\mathcal{Q}} = 0,\ \forall \mathcal{D} \in \mathfrak{D}. \tag{8}$$

  Equation (8) gives, on each $\mathcal{D}$, a formula for $\delta^n_{\mathcal{D}}$ as a function of $\nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T},n}$. It follows that (7) is nothing but a global DDFV linear system to solve.
- Step 2: On each $\mathcal{Q}$, find $g^n_{\mathcal{Q}} \in \mathbb{R}^2$ solution of

$$\boldsymbol{\varphi}_{\mathcal{Q}}(g^n_{\mathcal{Q}}) + \lambda^{n-1}_{\mathcal{Q}} + A_{\mathcal{Q}}(g^n_{\mathcal{Q}} - \nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T},n} - B_{\mathcal{Q}}\delta^n_{\mathcal{D}}) = 0. \tag{9}$$

  This is the unique nonlinear part of the algorithm and can be solved independently on each quarter diamond cell, by using the Newton method for instance.
- Step 3: On each $\mathcal{Q}$, compute $\lambda^n_{\mathcal{Q}}$ by

$$\lambda^n_{\mathcal{Q}} = \lambda^{n-1}_{\mathcal{Q}} + \gamma A_{\mathcal{Q}}(g^n_{\mathcal{Q}} - \nabla^{\mathcal{T}}_{\mathcal{D}} u^{\mathcal{T},n} - B_{\mathcal{Q}}\delta^n_{\mathcal{D}}). \tag{10}$$

In [2] the following result is proven.

Theorem 4. *Let* $\mathcal{T}$ *be a DDFV mesh on* Ω. *For any family* $(\boldsymbol{\varphi}_{\mathcal{Q}})_{\mathcal{Q}}$ *of strictly monotonic continuous maps from* $\mathbb{R}^2$ *onto itself, for any augmentation matrices family* $\mathcal{A}$ *and any* $\gamma \in \left]0, \frac{1+\sqrt{5}}{2}\right]$, *the algorithm* (7)-(10) *converges, when* n *goes to infinity, towards the unique solution to the m-DDFV scheme* (4)-(6).

3.2 Numerical Examples

We illustrate the behavior of the m-DDFV scheme compared to the DDFV one, on two academic examples for $\Omega = \Omega_1 \cup \Omega_2$ with $\Omega_1 =]0, 0.5[\times]0, 1[$ and $\Omega_2 =]0.5[\times]0, 1[$ and $\varphi_i(\xi) = |\xi|^{p_i - 2}\xi$:

$$\underline{\text{Test 1 :}}\;\; u(x,y) = \begin{cases} x\left(\left(\lambda^{\frac{p_2-1}{p_1-1}} - 1\right)(2x-1) + 1\right) & \text{for } x \leq 0.5, \\ (1-x)((1+\lambda)(2x-1)+1) & \text{for } x \geq 0.5, \end{cases}$$

$$\underline{\text{Test 2 :}}\;\; u(x,y) = \begin{cases} \sin(k\pi y)\left(\left(2 - \frac{4}{\pi}\right)x + \left(\frac{4}{k\pi} - 1\right)\right) & \text{for } x \leq 0.5, \\ \sin(k\pi y)(1-x)\left(\left(2 + \frac{4}{k\pi}\right)x - 1\right) & \text{for } x \geq 0.5. \end{cases}$$

In both cases the functions u, $\boldsymbol{\varphi}(z, \nabla u) \cdot \boldsymbol{n}$ are continuous across the interface $\partial\overline{\Omega_1} \cap \partial\overline{\Omega_2}$. The source terms is then computed by $f = -\mathrm{div}(\boldsymbol{\varphi}(z, \nabla u))$. For large values of λ, test 1 provides an example of large jump of the gradient. Tables 1 and 2 show that the DDFV method is first order in $L^{\boldsymbol{p}}$ norm whereas the m-DDFV is second order for both meshes (see Fig. 3). Note that the order of the m-DDFV in $W^{1,\boldsymbol{p}}$ norm is better on the mesh 2 (1.31) which is refined in the subdomain where $p = 4$ than for the regular triangular one (mesh 1 : 1.07).

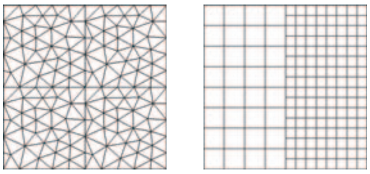


Fig. 3. Example of meshes : mesh 1 (left), mesh 2 (right)

Table 1. Norm of the error for test 1 on mesh 1 with $p_1 = 2$, $p_2 = 4$, $\lambda = 5.0$

mesh size	DDFV $L^{\boldsymbol{p}}(\Omega)$	m-DDFV $L^{\boldsymbol{p}}(\Omega)$	DDFV $W^{1,\boldsymbol{p}}(\Omega)$	m-DDFV $W^{1,\boldsymbol{p}}(\Omega)$
7.25E-02	4.70E-01	3.61E-02	2.5E+01	1.41
3.63E-02	2.36E-01	9.14E-02	2.03E+01	6.62E-01
1.81E-02	1.19E-01	2.24E-03	1.65E+01	3.11E-01
9.07E-03	6.01E-02	4.46E-04	1.34E+01	1.47E-01

Table 3 gives the convergence orders of the m-DDFV scheme for the test 2 for various values of (p_1, p_2) on the mesh 2. In this test the solution depends on both variables x and y, but ∇u is continuous at the interface which explains that DDFV and m-DDFV schemes have a similar behavior. Even though we get analogous

Table 2. Norm of the error for test 1 on mesh 2 for $p_1 = 2$, $p_2 = 4$, $\lambda = 5.0$

mesh size	DDFV $L^{\mathbf{p}}(\Omega)$	m-DDFV $L^{\mathbf{p}}(\Omega)$	DDFV $W^{1,\mathbf{p}}(\Omega)$	m-DDFV $W^{1,\mathbf{p}}(\Omega)$
8.83E-02	9.62E-01	9.93E-02	3.26E+01	2.52E+00
4.41E-02	4.82E-01	2.52E-02	2.62E+01	1.01E+00
2.21E-02	2.44E-01	6.31E-03	2.12E+01	4.09E-01
1.10E-02	1.23E-01	1.58E-03	1.71E+01	1.64E-01

convergence rate for $(p_1 = 2, p_2 = 4)$ and $(p_1 = 4, p_2 = 2)$, smaller global error is obtained in the case when the mesh is more refined in the domain where p_i is big.

4 Conclusions

We propose here a finite volume approach to approximate nonlinear transmission problems on general 2D grids. The m-DDFV scheme we study is solved by means of a decomposition-coordination method. Numerical results in the case of $p-$Laplacian like operators illustrate the good behavior of the scheme especially in case of big jumps of the gradient.

Table 3. Convergence rates in the two domains Ω_1 and Ω_2 for test 2 with $k = 5$

	$L^{p_1}(\Omega_1)$	$L^{p_2}(\Omega_2)$	$W^{1,p_1}(\Omega_1)$	$W^{1,p_2}(\Omega_2)$
$p_1 = 2$, $p_2 = 1.5$	2.00	1.99	1.49	1.69
$p_1 = 2$, $p_2 = 4$	2.00	2.00	1.56	1.20
$p_1 = 4$, $p_2 = 2$	2.04	1.98	1.20	1.60
$p_1 = 3$, $p_2 = 4$	2.11	2.02	1.30	1.19

References

[1] B. Andreianov, F. Boyer, and F. Hubert. Discrete duality finite volume schemes for Leray-Lions type elliptic problems on general 2D-meshes. *Numer. Methods Partial Differential Equations*, 23(1):145–195, 2007.

[2] F. Boyer and F. Hubert. Finite volume method for 2d linear and nonlinear elliptic problems with discontinuities. Technical report, Université de Provence, Marseille, France, 2006. `http://hal.archives-ouvertes.fr/hal-00110436`.

[3] Y. Coudière, J.-P. Vila, and P. Villedieu. Convergence rate of a finite volume scheme for a two-dimensional convection-diffusion problem. *M2AN Math. Model. Numer. Anal.*, 33(3):493–516, 1999.

[4] K. Domelevo and P. Omnes. A finite volume method for the Laplace equation on almost arbitrary two-dimensional grids. *M2AN Math. Model. Numer. Anal.*, 39(6):1203–1249, 2005.

[5] M. Feistauer and V. Sobotíková. Finite element approximation of nonlinear elliptic problems with discontinuous coefficients. *RAIRO Modél. Math. Anal. Numér.*, 24(4):457–500, 1990.

[6] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems.* Springer Series in Computational Physics. Springer-Verlag, New York, 1984.

[7] F. Hermeline. Approximation of diffusion operators with discontinuous tensor coefficients on distorted meshes. *Comput. Methods Appl. Mech. Engrg.*, 192(16-18):1939–1959, 2003.

[8] W. B. Liu. Degenerate quasilinear elliptic equations arising from bimaterial problems in elastic-plastic mechanics. *Nonlinear Anal.*, 35(4):517–529, 1999.

[9] W. B. Liu. Finite element approximation of a nonlinear elliptic equation arising from bimaterial problems in elastic-plastic mechanics. *Numer. Math.*, 86(3):491–506, 2000.

An Overlapping Domain Decomposition Method for Parameter Identification Problems

Xiao-Chuan Cai[1]*, Si Liu[1], and Jun Zou[2]

[1] Department of Computer Science, University of Colorado at Boulder, Boulder, CO 80309, USA, {cai,si.liu}@cs.colorado.edu

[2] Department of Mathematics, The Chinese University of Hong Kong, Shatin N. T., Hong Kong, zou@math.cuhk.edu.hk

Summary. A parallel fully coupled one-level Newton-Krylov-Schwarz method is investigated for solving the nonlinear system of algebraic equations arising from the finite difference discretization of inverse elliptic problems. Both L^2 and H^1 least squares formulations are considered with the H^1 regularization. We show numerically that the preconditioned iterative method is optimally scalable with respect to the problem size. The algorithm and our parallel software perform well on machines with modest number of processors, even when the level of noise is quite high.

1 Introduction

We consider an inverse elliptic problem [1, 6]: Find $\rho(x)$, such that

$$\begin{cases} -\nabla \cdot (\rho \nabla u) = f, \ x \in \Omega \\ \qquad u(x) = 0, \ x \in \partial\Omega. \end{cases} \tag{1}$$

When the measurement of $u(x)$ is given, denoted as $z(x)$, the inverse problem can be transformed into a minimization problem:

$$\text{minimize} \quad J(\rho, u) = \frac{1}{2}\int_\Omega (u-z)^2 dx + \frac{\beta}{2}\int_\Omega |\nabla \rho|^2 dx, \tag{2}$$

which is usually referred to as the "L^2 least squares formulation". When the measurement of $\nabla u(x)$ is given, denoted as $\nabla z(x)$, the inverse problem can be transformed into another minimization problem:

$$\text{minimize} \quad J(q, v) = \frac{1}{2}\int_\Omega \rho \, |\nabla u - \nabla z|^2 dx + \frac{\beta}{2}\int_\Omega |\nabla \rho|^2 dx, \tag{3}$$

which is usually referred to as the "H^1 least squares formulation". Both minimization problems (2) and (3) are subject to the constraint (1). We introduce the Lagrangian functional

* The work of XCC and SL was partially supported by DOE DE-FC02-01ER25479, NSF ACI-0305666 and ACI-0352334, the work of JZ was substantially supported by the Hong Kong RGC grant, Project 404105.

$$\mathcal{L}(\rho, u, \lambda) = \frac{1}{2}\int_{\Omega}(u-z)^2 dx + ((\nabla\lambda, \rho\nabla u) - (\lambda, f)) + \frac{\beta}{2}\int_{\Omega}|\nabla\rho|^2 dx \tag{4}$$

for the L^2 case, and

$$\mathcal{L}(\rho, u, \lambda) = \frac{1}{2}\int_{\Omega}\rho|\nabla u - \nabla z|^2 dx + ((\nabla\lambda, \rho\nabla u) - (\lambda, f)) + \frac{\beta}{2}\int_{\Omega}|\nabla\rho|^2 dx \tag{5}$$

for the H^1 case. The solution of both minimization problems can be obtained by solving the corresponding saddle-point problem: Find (ρ, u, λ) such that $(\nabla_\rho\mathcal{L})p = 0, (\nabla_u\mathcal{L})w = 0,$ and $(\nabla_\lambda\mathcal{L})\mu = 0$ for any (p, w, μ), which implies that

$$\begin{cases} -\beta\Delta\rho + \nabla u \cdot \nabla\lambda = 0 \\ -\nabla\cdot(\rho\nabla\lambda) + (u - z) = 0 \\ -\nabla\cdot(\rho\nabla u) - f = 0 \end{cases} \tag{6}$$

in the L^2 case. Similarly, in the H^1 case, we have

$$\begin{cases} -\beta\Delta\rho + \nabla u \cdot \nabla\lambda + \frac{1}{2}|\nabla u - \nabla z|^2 = 0 \\ -\nabla\cdot(\rho\nabla\lambda) + \nabla\cdot(\rho\nabla z) + f = 0 \\ -\nabla\cdot(\rho\nabla u) - f = 0. \end{cases} \tag{7}$$

Both systems share the same boundary conditions $\partial\rho/\partial n = 0, u = 0, \lambda = 0$ on $\partial\Omega$. A derivation of the boundary conditions is given in [3]. The rest of the paper is devoted to a Newton-Krylov-Schwarz method for solving the algebraic systems

$$F(U) = 0$$

arising from the finite difference discretization of (6) and (7) in a fully coupled fashion [3, 4].

2 Newton-Krylov-Schwarz Method

The family of Newton-Krylov-Schwarz (NKS) methods [2] is a general-purpose parallel algorithm for solving a system of nonlinear algebraic equations. NKS has three main components: an inexact Newton's method for the nonlinear system; a Krylov subspace linear solver for the Jacobian systems (restarted GMRES); and a Schwarz type preconditioner [7]. Other related techniques can be found in [5]. We carry out Newton iterations as following:

$$U_{k+1} = U_k - \lambda_k J(U_k)^{-1}F(U_k), \quad k = 0, 1, ... \tag{8}$$

where U_0 is an initial approximation to the solution and $J(U_k) = F'(U_k)$ is the Jacobian at U_k, and λ_k is the steplength determined by a linesearch procedure. The inexactness of Newton's method is reflected by the fact that we do not solve the Jacobian system exactly. The accuracy of the Jacobian solver is determined by some $\eta_k \in [0, 1)$ and the condition

$$\|F(U_k) + J(U_k)s_k\| \leq \eta_k \|F(U_k)\|. \tag{9}$$

The vector s_k is obtained by approximately solving the linear Jacobian system

$$J(U_k)M_k^{-1}(M_k s_k) = -F(U_k),$$

where M_k^{-1} is a one-level additive Schwarz right preconditioner. To formally define M_k^{-1}, we need to introduce a partition of Ω. We first partition the domain into non-overlapping substructures Ω_l, $l = 1, \cdots, N$. In order to obtain an overlapping decomposition of the domain, we extend each subregion Ω_l to a larger region Ω_l', i.e., $\Omega_l \subset \Omega_l'$. Only simple box decomposition is considered in this paper – all subdomains Ω_l and Ω_l' are rectangular and made up of integral numbers of fine mesh cells. The size of Ω_l is $H_x \times H_y$ and the size of Ω_l' is $H_x' \times H_y'$, where the H's are chosen so that the overlap, *ovlp*, is uniform in the number of fine mesh cells all around the perimeter, i.e., $ovlp = (H_x' - H_x)/2 = (H_y' - H_y)/2$ for interior subdomains. For boundary subdomains, we simply cut off the part that is outside Ω.

On each extended subdomain Ω_l', we construct a subdomain preconditioner B_l, whose elements are extracted from the matrix $J(U_k)$. Homogeneous Dirichlet boundary conditions are used on the internal subdomain boundary $\partial\Omega_l' \cap \Omega$, and the original boundary conditions are used on the physical boundary, if present. The additive Schwarz preconditioner can be written as

$$M_k^{-1} = I_1 B_1^{-1}(I_1)^T + \cdots + I_N B_N^{-1}(I_N)^T. \tag{10}$$

Let n be the total number of mesh points, and n_l' the total number of mesh points in Ω_l', then I_l is an $3n \times 3n_l'$ extension matrix that extends each vector defined on Ω_l' to a vector defined on the entire fine mesh by padding an $3n_l' \times 3n_l'$ identity matrix with zero rows. The factor of 3 is included because each mesh point has 3 unknowns.

3 Numerical Experiments

We study the performance of the proposed algorithm using the following test case with the observation function given as $z(x, y) = \sin(\pi x)\sin(\pi y)$, $\Omega = (0, 1) \times (0, 1)$, and the right-hand side f chosen so that the elliptic coefficient to be identified is $\rho = 1 + 100(xy(1-x)(1-y))^2$. To test the robustness of the algorithms, we add some noise to the observation data as

$$z^\delta = z + \delta\, rand(x, y) \tag{11}$$

or

$$\nabla z^\delta = \nabla z + \delta\,(rand(x, y), rand(x, y))^T, \tag{12}$$

depending on if the formulation is L^2 or H^1. Here $rand(x, y)$ defines a random scalar function. δ is responsible for the magnitude of the noise. Results with three different levels of noise ($\delta = 0\%, 1\%$ and 10%) will be presented. Since u needs to satisfy the elliptic equation, we assume that u and ∇u have some continuity and differentiability. Therefore, we smooth z in the L^2 formulation or ∇z in the H^1 formulation before we start the Newton iteration. This is necessary especially when the noise level is high. In particular, when the noise level is 10%, we replace the value of z or ∇z by the average value around it using the following weights

$$\begin{array}{ccccc}
\frac{1}{16} & & \frac{1}{8} & & \frac{1}{16}\\
 & \searrow & \downarrow & \swarrow & \\
\frac{1}{8} & \rightarrow & \frac{1}{4} & \leftarrow & \frac{1}{8}\\
 & \nearrow & \uparrow & \nwarrow & \\
\frac{1}{16} & & \frac{1}{8} & & \frac{1}{16}
\end{array}$$

We repeat this operation 3 times in all the experiments when $\delta = 10\%$. No smoothing is applied when δ is smaller than 10%.

To measure the accuracy of the algorithm, we assume the exact solution of the test problem is known, and $error_u$ and $error_\rho$ are the normalized discrete L^2 norms of the errors defined as

$$error_u = \sqrt{\sum (u_{ij} - u_{ij}^{exact})^2 h_x h_y} \quad \text{and} \quad error_\rho = \sqrt{\sum (\rho_{ij} - \rho_{ij}^{exact})^2 h_x h_y},$$

where h_x and h_y are mesh sizes along x and y directions, respectively.

In our experiments, we choose the stopping conditions as follows: The relative residual is less than 10^{-6} or the absolute residual is less than 10^{-10} for the nonlinear system. The relative residual is less than 10^{-6} or the absolute residual is less than 10^{-10} for each linear solve in the nonlinear iteration. We do not have a systematic way to pick β. Several values of β are tested in the range of 10^{-4} to 10^{-6}. In Newton's method, we use the initial guess $(\rho^{(0)}, u^{(0)}, \lambda^{(0)})^T = (1, z, 0)^T$ for the L^2 formulation. For the H^1 formulation, z is obtained as an integral of $\nabla_x z$ or $\nabla_y z$ along the x or y direction from one of the boundary points. In our experiments, at the mesh point (x_i, y_j),

$$z(x_i, y_j) = z(x_0, y_j) + \sum_{l=1}^{i} (\nabla_x z)|_{x_l} h_x$$

if we take the integral in the x direction, or a similar integral in the y direction.

We first test three meshes 40×40, 80×80, and 160×160. When the Jacobian systems are solved exactly with a Gaussian elimination, the total number of Newton iterations ranges from 3 to 6, and the iteration numbers are not sensitive to the level of noise, as shown in Table 1. The exact solution, and the numerical solutions for both L^2 and H^1 formulations with 3 levels of noise are shown in Fig. 1.

We next look at the performance of the algorithm, in particular, we would like to know how the convergence depends on the mesh size, the number of subdomains, and the overlapping size. We solve the problem on a 320×320 mesh using different number of processors (np), and the results, in terms of the iteration number and the total compute time, are in Table 2. The numbers of Newton iterations do not change when we change the number of processors or the overlapping size.

If we fix the number of subdomains, which is the same as the number of processors, and increase the overlapping size, the number of GMRES iterations decreases. The compute time decreases to a certain point and then begins to increase. This suggests that an optimal overlapping size exists if the objective is to minimize the total compute time when the number of processors is fixed. On a fixed mesh the number of GMRES iterations increases as we use more processors. This is expected since this is a single-level algorithm.

To check the $h-$scalability of the algorithm, we increase the mesh size and the number of processors at the same ratio in order for each processor to have the same number of mesh points. Table 3 shows the results with different mesh sizes for np=4, 16 and 64. Both the number of Newton iterations and the number of GMRES iterations are almost constants when the number of processors is fixed.

4 Final Remarks

We developed a fully parallel domain decomposition method for solving the system of nonlinear equations arising from the fully coupled finite difference discretization of some inverse elliptic problems. Traditionally this type of problems are solved by using Uzawa type of algorithms which split the system into two or three subsystems and each subsystem is solved individually. Subiterations are required between the subsystems. The subsystems are easier to solve than the global coupled system, but the iterations between subsystems are sequential in nature. The focus of this paper was to investigate a fully coupled approach without splitting the system into subsystems. Such an approach is more parallel than the splitting method. We showed numerically that with a powerful domain decomposition based preconditioner the convergence of the iterative methods can be obtained even for some difficult cases when the observation data has high level of noise. More details of the work will be included in a forthcoming paper [3].

Table 1. Errors and the number of Newton iterations for three different meshes and with different levels of noise.

		$error_u$	$error_\rho$	Newton
L^2 formulation 40×40	$\beta = 10^{-6}, \delta = 0$	0.000078	0.003163	3
	$\beta = 10^{-5}, \delta = 1\%$	0.000765	0.010723	3
	$\beta = 10^{-4}, \delta = 10\%$	0.008222	0.038667	3
L^2 formulation 80×80	$\beta = 10^{-6}, \delta = 0$	0.000073	0.003177	3
	$\beta = 10^{-5}, \delta = 1\%$	0.000532	0.010070	3
	$\beta = 10^{-4}, \delta = 10\%$	0.003849	0.029056	3
L^2 formulation 160×160	$\beta = 10^{-6}, \delta = 0$	0.000072	0.003203	3
	$\beta = 10^{-5}, \delta = 1\%$	0.000504	0.009908	3
	$\beta = 10^{-5}, \delta = 10\%$	0.002064	0.026190	4
H^1 formulation 40×40	$\beta = 10^{-5}, \delta = 0$	0.000362	0.001744	6
	$\beta = 10^{-5}, \delta = 1\%$	0.000355	0.006010	6
	$\beta = 10^{-4}, \delta = 10\%$	0.006980	0.022837	5
H^1 formulation 80×80	$\beta = 10^{-5}, \delta = 0$	0.000090	0.000406	4
	$\beta = 10^{-5}, \delta = 1\%$	0.000109	0.003842	4
	$\beta = 10^{-4}, \delta = 10\%$	0.001921	0.011741	4
H^1 formulation 160×160	$\beta = 10^{-5}, \delta = 0$	0.000023	0.000187	3
	$\beta = 10^{-5}, \delta = 1\%$	0.000030	0.002580	4
	$\beta = 10^{-4}, \delta = 10\%$	0.000473	0.007419	5

References

[1] H.T. Banks and K. Kunisch. *Estimation Techniques for Distributed Parameter Systems.* Birkhauser, Basel-Boston-Berlin, 1989.

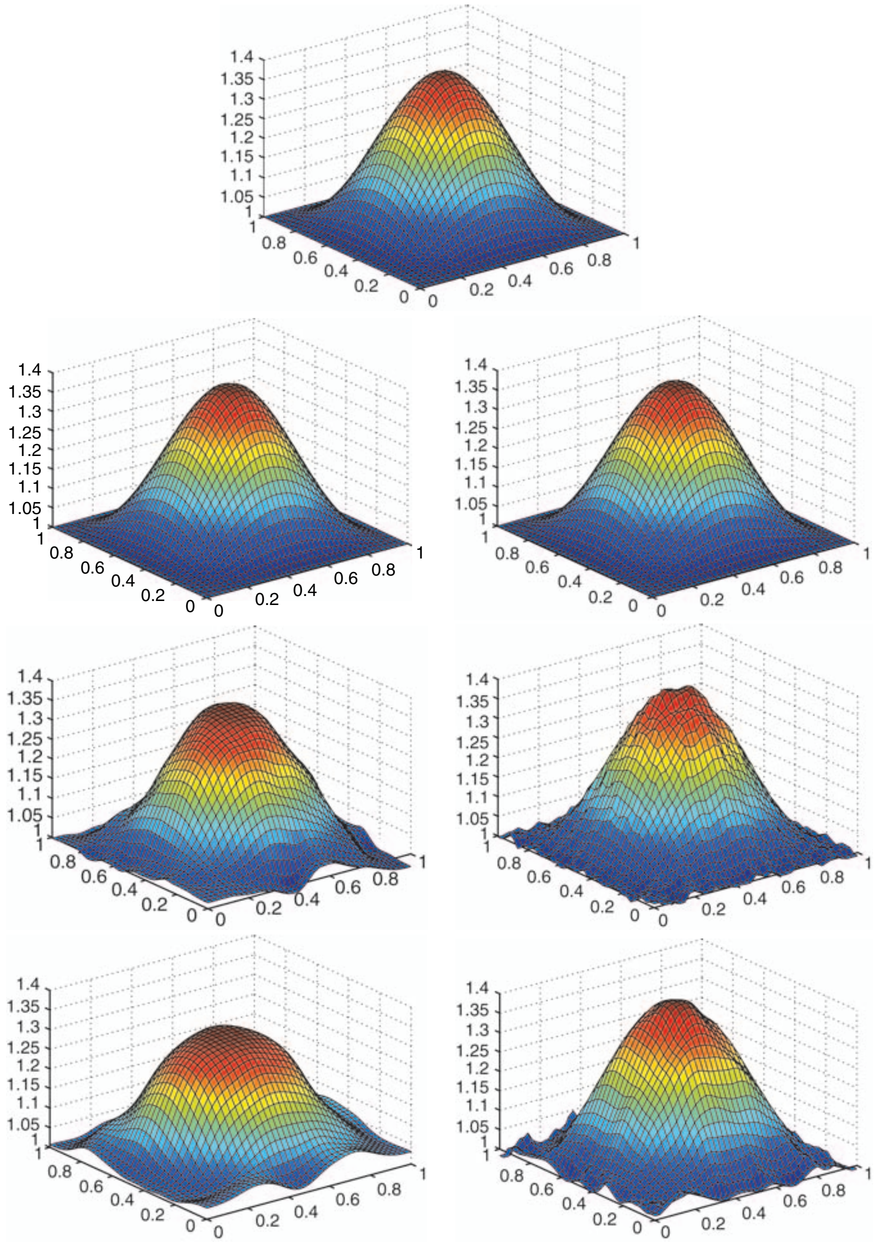


Fig. 1. The top picture is the exact solution ρ. The following six pictures are the numerical solution with $\delta = 0\%, 1\%, 10\%$ on a 40×40 mesh. The left three are for the L^2 formulation and the right three are for the H^1 formulation.

Table 2. The total number of Newton and the average number of GMRES iterations are shown below for a 320 × 320 mesh. The total compute time in seconds is in $(\cdot)$.

	np	$Newton$	$ovlp = 1$	$ovlp = 2$	$ovlp = 4$	$ovlp = 8$	$ovlp = 16$
L^2 formulation	1	3	1(374.33)	1(373.37)	1(375.98)	1(375.57)	1(374.62)
$\beta = 10^{-6}$	4	3	46(108.93)	33(97.62)	18(80.87)	13(79.21)	8(80.46)
$\delta = 0\%$	16	3	66(32.43)	46(26.39)	34(23.92)	22(22.66)	14(26.75)
	64	3	127(23.08)	92(19.22)	63(15.49)	42(14.83)	25(16.35)
L^2 formulation	1	3	1(374.98)	1(374.23)	1(372.92)	1(372.35)	1(374.21)
$\beta = 10^{-5}$	4	3	43(105.49)	26(86.60)	19(80.11)	14(79.02)	9(81.57)
$\delta = 1\%$	16	3	57(30.02)	45(25.89)	31(22.55)	22(23.44)	15(30.14)
	64	3	134(24.71)	94(19.50)	62(15.28)	45(15.09)	25(15.79)
L^2 formulation	1	5	1(623.39)	1(621.60)	1(627.58)	1(622.50)	1(629.40)
$\beta = 10^{-5}$	4	6	61(260.45)	47(225.89)	27(182.45)	18(168.59)	12(172.45)
$\delta = 10\%$	16	6	110(97.01)	81(77.46)	59(67.06)	39(59.56)	24(70.57)
	64	6	234(83.13)	162(62.44)	122(53.56)	78(45.28)	43(50.87)
H^1 formulation	1	3	1(382.09)	1(381.11)	1(384.03)	1(382.27)	1(380.59)
$\beta = 10^{-5}$	4	3	66(136.58)	41(106.42)	24(87.81)	17(84.60)	12(88.99)
$\delta = 0\%$	16	3	148(60.33)	96(43.64)	60(33.56)	37(30.11)	23(34.60)
	64	3	290(47.59)	212(38.34)	121(27.61)	92(25.11)	55(27.08)
H^1 formulation	1	4	1(505.06)	1(503.49)	1(501.99)	1(502.54)	1(504.08)
$\beta = 10^{-5}$	4	4	53(158.88)	34(129.94)	20(110.25)	15(107.46)	10(111.08)
$\delta = 1\%$	16	4	110(63.29)	72(47.44)	47(38.10)	29(34.19)	20(40.42)
	64	4	219(48.50)	142(35.01)	100(28.07)	58(22.82)	44(28.61)
H^1 formulation	1	5	1(624.17)	1(629.97)	1(627.58)	1(629.90)	1(628.54)
$\beta = 10^{-4}$	4	5	62(212.91)	47(178.81)	27(151.06)	18(139.06)	12(143.07)
$\delta = 10\%$	16	5	104(75.61)	82(65.45)	56(53.17)	36(47.70)	22(52.91)
	64	5	221(60.96)	161(49.38)	122(41.46)	71(33.36)	52(38.88)

Table 3. Newton and GMRES iteration numbers are shown below for three different meshes. The compute time in seconds is in $(\cdot)$. *ovlp* is 1/5 of the diameter of the subdomain.

	np	$Newton$	$GMRES$	$Newton$	$GMRES$	$Newton$	$GMRES$
		80 × 80 mesh		160 × 160 mesh		320 × 320 mesh	
L^2 formulation	4	3	6(2.62)	3	6(14.72)	3	6(100.44)
$\beta = 10^{-6}$	16	3	14(2.48)	3	14(6.33)	3	14(26.75)
$\delta = 0\%$	64	3	38(5.73)	3	40(7.28)	3	42(14.83)
L^2 formulation	4	3	7(2.41)	3	7(14.22)	3	6(100.23)
$\beta = 10^{-5}$	16	3	17(2.82)	3	16(6.60)	3	15(30.14)
$\delta = 1\%$	64	3	47(6.74)	3	45(7.68)	3	45(15.09)
L^2 formulation	4	3	9(3.03)	3	8(15.79)	3	8(100.47)
$\beta = 10^{-4}$	16	3	24(3.65)	3	23(8.02)	3	22(34.35)
$\delta = 10\%$	64	3	75(10.41)	3	72(11.47)	3	66(20.66)
H^1 formulation	4	4	8(3.43)	3	8(1.54)	3	8(106.40)
$\beta = 10^{-5}$	16	4	22(4.04)	3	24(7.47)	3	23(34.60)
$\delta = 0\%$	64	4	77(12.68)	3	81(12.14)	3	92(25.11)
H^1 formulation	4	4	8(3.43)	4	8(20.69)	4	7(131.44)
$\beta = 10^{-5}$	16	4	22(4.17)	4	19(9.25)	4	20(40.42)
$\delta = 1\%$	64	4	73(11.90)	4	75(11.89)	4	58(22.82)
H^1 formulation	4	4	8(3.85)	5	8(26.33)	5	8(163.20)
$\beta = 10^{-4}$	16	4	22(4.23)	5	21(12.14)	5	22(52.91)
$\delta = 10\%$	64	4	71(11.71)	5	69(16.88)	5	71(33.36)

[2] X.-C. Cai, W.D. Gropp, D.E. Keyes, R.G. Melvin, and D.P. Young. Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation. *SIAM J. Sci. Comput.*, 19(1):246–265, 1998.

[3] X.-C. Cai, S. Liu, and J. Zou. Parallel fully coupled algorithms for inverse elliptic problems, 2007. Submitted.

[4] E. Haber and U. Ascher. Preconditioned all-at-once methods for large, sparse parameter estimation problems. *Inverse Problems*, 17:1847–1864, 2001.

[5] M. Hanke. Regularizing properties of a truncated Newton-CG algorithm for nonlinear inverse problems. *Numer. Funct. Anal. Optim.*, 18:971–993, 1997.

[6] Yee Lo Keung and Jun Zou. An efficient linear solver for nonlinear parameter identification problems. *SIAM J. Sci. Comput.*, 22(5):1511–1526, 2000.

[7] A. Toselli and O.B. Widlund. *Domain Decomposition Methods—Algorithms and Theory.* Springer-Verlag, Berlin, 2005.

A Domain Decomposition Method for the Diffusion of an Age-structured Population in a Multilayer Environment

Caterina Cusulin[1] and Luca Gerardo-Giorda[2]

[1] Faculty of Mathematics, University of Vienna, Nordbergstraße 15, 1090 Vienna, Austria. caterina.cusulin@univie.ac.at

[2] Dipartimento di Matematica, Università di Trento - Italy. gerardo@science.unitn.it

1 Introduction

The spatial spread of an age-structured population in an isolated environment is commonly governed by a partial differential equation with zero-flux boundary condition for the spatial domain. The variables involved are time, age and space, which will be denoted in the following by t, a and x, respectively. We denote the spatial domain by $\Omega \subset \mathbb{R}^d$ $(d = 1, 2, 3)$, and we assume the age of the population to be bounded, *i.e.* there exists $a_\dagger > 0$ such that $a \in [0, a_\dagger]$. Denoting the population density at time t per unit volume and age by $p(t, a, x)$, the total population at time t is given by

$$P(t) = \int_\Omega \int_0^{a_\dagger} p(t, a, x)\, da\, dx.$$

Let then $T > 0$, the population density $p(t, a, x)$ satisfies the following model problem.

Find $p(t, a, x) \in C(0, T; L^2(0, a_\dagger; H^1(\Omega)))$ such that

$$\begin{array}{ll} p_t + p_a + \mu(a)\, p - \mathrm{div}\ (k(a, x)\nabla p) = g & \text{in } (0, T) \times (0, a_\dagger) \times \Omega \\ p(0, a, x) = p_0(a, x) & \text{in } (0, a_\dagger) \times \Omega \\ p(t, 0, x) = \displaystyle\int_0^{a_\dagger} \beta(a) p(t, a, x)\ da & \text{in } (0, T) \times \Omega \\ \mathbf{n} \cdot (k(a, x)\nabla p) = 0 & \text{on } (0, T) \times (0, a_\dagger) \times \partial\Omega, \end{array} \tag{1}$$

where $\mathbf{n}$ denotes the outward normal to $\partial\Omega$, $\beta(a)$ is the age-specific fertility, and $\mu(a)$ is the age-specific mortality, such that

$$\int_0^{a_\dagger} \mu(a)\, da = +\infty. \tag{2}$$

We refer to [7] and references therein for issues concerning existence and uniqueness for the solution of problem (1).

1.1 The Reduced Model

In order to avoid the difficulties entailed by the presence of an unbounded coefficient in (1), it is usual to introduce the *survival probability*

$$\varPi(a) = \exp\left(- \int_0^a \mu(s)\, ds \right),$$

and a new variable

$$u(t,a,x) = \frac{p(t,a,x)}{\varPi(a)}.$$

Owing to (2), the survival probability at age $a_\dagger$ vanishes, ensuring that no individual exceeds the maximal age.
With these positions, (1) is equivalent to the following reduced model problem.

Find $u(t,a,x) \in C(0,T; L^2(0,a_\dagger; H^1(\varOmega)))$ such that

$$\begin{array}{rll} u_t + u_a - \text{div}\,(k(a,x)\nabla u) = f & \text{in } (0,T) \times (0,a_\dagger) \times \varOmega \\ u(0,a,x) = u_0(a,x) & \text{in } (0,a_\dagger) \times \varOmega \\ u(t,0,x) = \displaystyle\int_0^{a_\dagger} m(a) u(t,a,x)\; da & \text{in } (0,T) \times \varOmega \\ \mathbf{n} \cdot (k(a,x)\, \nabla u) = 0 & \text{on } (0,T) \times (0,a_\dagger) \times \partial\varOmega, \end{array} \tag{3}$$

where $u_0(a,x) = p_0(a,x)/\varPi(a)$, $f = g/\varPi(a)$, and where $m(a) = \varPi(a)\,\beta(a)$ is called maternity function.

1.2 Space-time Discretization

Classical approaches to the numerical solution of (3) integrate along the characteristics in age and time (see for instance [4, 5, 6]). However, the presence of different time scales in the dynamics suggests the use of different steps in the discretization of time and age (see [1, 2]).
Let us consider a discretization of the interval $(0,T)$ into N subintervals of length $\varDelta t = T/N$ (for simplicity we consider a uniform discretization, adaptivity in time being beyond the scope of this paper). For equation (3) we advance in time by means of a backward Euler scheme, where the initial condition in age is computed at the previous time step. At each time step, we solve the parabolic problem in age and space:

Find $u^n \in L^2(0,a_\dagger; H^1(\varOmega))$ such that

$$\begin{cases} \dfrac{d}{da}\, \langle u^{n+1}, v \rangle + A(a; u^{n+1}, v) = (f,v) + \dfrac{1}{\varDelta t}(u^n, v) & \forall v \in H^1(\varOmega) \\ u^{n+1}(0,x) = \displaystyle\int_0^{a_\dagger} m(a) u^n(a,x)\; da \end{cases} \tag{4}$$

where $\langle \cdot, \cdot \rangle$ denotes the duality pairing between $H^1(\varOmega)$ and $H^{-1}(\varOmega)$, and where $A(a; \cdot, \cdot)$ is the bilinear form given by

$$A(a; w, v) = \int_\varOmega k(a,x) \nabla w \cdot \nabla v + \frac{1}{\varDelta t} \int_\varOmega wv.$$

We discretize equation (4) in space by means of finite elements (see e.g. [8] for an introduction to finite element methods). Let $\Omega = \bigcup_{j=1}^{N} K_j$, where each $K_j = T_{K_j}(E)$ is an element of the triangulation, E is the reference simplex, and T_{K_j} is an invertible affine map. The associated finite element space is then

$$V_h = \left\{ \varphi_h \in C^0(\Omega) \,|\, \varphi_{h|K_j} \circ T_{K_j} \in \mathbb{P}_1(E) \right\},$$

where $\mathbb{P}_1(E)$ is the space of polynomials of degree at most one in each variable on E. A semi-discrete problem in space is then obtained by applying a Galerkin procedure and choosing a finite element basis for V_h. Since the finite element basis functions depend only on space, we can rewrite problem (4) as

$$\begin{cases} M\dfrac{d\mathbf{u}_h^{n+1}}{da} + \mathcal{A}(a)\mathbf{u}_h^{n+1} = \mathbf{f} + \dfrac{1}{\Delta t} M\mathbf{u}^n \\ \mathbf{u}_h^{n+1}(0,x) = \displaystyle\int_0^{a_\dagger} m(a)\mathbf{u}_h^n(a,x)\,da \end{cases} \tag{5}$$

where M is the mass matrix ($M_{ij} = \int_\Omega \varphi_j \varphi_i \, dx$) and $\mathcal{A}(a)$ is the stiffness matrix associated to the bilinear form $A(a;\cdot,\cdot)$, ($(\mathcal{A}(a))_{ij} = A(a;\varphi_j,\varphi_i)$).

2 Diffusion in a Multilayer Environment and Domain Decomposition

We consider a population spreading in a stratified environment composed of m layers, with zero flux boundary conditions. We refer the interested reader to [10] for issues concerning the motivations of such model. We suppose that the age-specific fertility and the age-specific mortality depend only on the layer, while the diffusion coefficients depend both on the age and on the layer. On the interface between the j-th and the $(j+1)$-th layer we have to impose the continuity of the trace and the normal flux, thus the equation in the j-th layer reads

$$\begin{array}{rcl} \partial_t u_j + \partial_a u_j - \text{div}\ (k_j(a,x)\nabla u_j) = f_j & & \text{in } (0,T)\times(0,a_\dagger)\times\Omega_j \\ u_j(0,a,x) = u_{0,j}(a,x) & & \text{in } (0,a_\dagger)\times\Omega_j \\ u_j(t,0,x) = \displaystyle\int_0^{a_\dagger} m_j(a)u_j(t,a,x)\,da & & \text{in } (0,T)\times\Omega_j \\ \mathbf{n}_j\cdot(k_j(a,x)\nabla u_j) = 0 & & \text{on } (0,T)\times(0,a_\dagger)\times(\partial\Omega\cap\partial\Omega_j) \\ u_j(t,a,x) = u_{j-1}(t,a,x) & & \text{on } (0,T)\times(0,a_\dagger)\times(\overline{\Omega}_j\cap\overline{\Omega}_{j-1}) \\ u_j(t,a,x) = u_{j+1}(t,a,x) & & \text{on } (0,T)\times(0,a_\dagger)\times(\overline{\Omega}_j\cap\overline{\Omega}_{j+1}) \\ \mathbf{n}_j\cdot(k_j\nabla u_j) = \mathbf{n}_j\cdot(k_{j-1}\nabla u_{j-1}) & & \text{on } (0,T)\times(0,a_\dagger)\times(\overline{\Omega}_j\cap\overline{\Omega}_{j-1}) \\ \mathbf{n}_j\cdot(k_j\nabla u_j) = \mathbf{n}_j\cdot(k_{j+1}\nabla u_{j+1}) & & \text{on } (0,T)\times(0,a_\dagger)\times(\overline{\Omega}_j\cap\overline{\Omega}_{j+1}) \end{array} \tag{6}$$

A domain decomposition procedure to solve equation (6) is thus straightforward. After time discretization we have to solve a parabolic problem: the age-space domain is naturally decomposed in strips $(0,a_\dagger)\times\Omega_j$ and the global solution is obtained by a standard waveform relaxation procedure (see e.g. [9]) we outline in the following section.

2.1 A Waveform Relaxation Procedure

For sake of simplicity in presentation, we give here the two-domain formulation of the domain decomposition algorithm, and, in order to improve readability, we drop any index referring to time discretization. We set $\Omega = \Omega_1 \cup \Omega_2$, we denote the interface between the two subdomains by $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$, the space of traces on Γ of functions in $H^1(\Omega)$ by $\Lambda = H^{1/2}(\Gamma)$, and we set $V_i = H^1(\Omega_i)$ $(i = 1, 2)$. At each time step, the coupled problem reads as follows:

Find $u_1 \in L^2(0, a_\dagger; H^1(\Omega_1))$ and $u_2 \in L^2(0, a_\dagger; H^1(\Omega_2))$ such that

$$\begin{cases} \dfrac{d}{da}\langle u_1, v_1\rangle + A_1(a; u_1, v_1) = (f_1, v_1) & \forall v_1 \in V_1 \\ \dfrac{d}{da}\langle u_2, v_2\rangle + A_2(a; u_2, v_2) = (f_2, v_2) & \forall v_2 \in V_2 \\ u_1 = u_2 & \text{on } (0, a_\dagger) \times \Gamma \\ \dfrac{d}{da}\langle u_2, R_2\mu\rangle + A_2(a; u_2, R_2\mu) = & \forall \mu \in \Lambda \\ \qquad = (f, R_2\mu) + (f, R_1\mu) - \dfrac{d}{da}\langle u_1, R_1\mu\rangle - A_1(a; u_1, R_1\mu), & \end{cases} \tag{7}$$

where $A_i(a; \cdot, \cdot)$ denotes the restriction of the bilinear form $A(a; \cdot, \cdot)$ to Ω_i, whereas $R_i\mu$ denotes any possible extension of μ to Ω_i $(i = 1, 2)$.
We apply a balancing Neumann-Neumann waveform relaxation procedure to enforce the interface continuities of equation (7).

Step 1. At each time step, given an initial value $\lambda^0 \in L^2((0, a_\dagger) \times \Gamma)$, solve:

$$\begin{cases} \dfrac{d}{da}\langle u_1^{k+1}, v_1\rangle + A_1(a; u_1^{k+1}, v_1) = (f_1, v_1) & \forall v_1 \in V_1 \\ u_1^{k+1} = \lambda^k & \text{on } (0, a_\dagger) \times \Gamma \\ u_1^{k+1}(0, x) = u_1^0(x) & \end{cases}$$

and

$$\begin{cases} \dfrac{d}{da}\langle u_2^{k+1}, v_2\rangle + A_2(a; u_2^{k+1}, v_2) = (f_2, v_2) & \forall v_2 \in V_2 \\ u_2^{k+1} = \lambda^k & \text{on } (0, a_\dagger) \times \Gamma \\ u_2^{k+1}(0, x) = u_2^0(x). & \end{cases}$$

Step 2. Solve

$$\begin{cases} \dfrac{d}{da}\langle \psi_1^{k+1}, v_1\rangle + A_1(a; \psi_1^{k+1}, v_1) = (f_1, v_1) & \forall v_1 \in V_1 \\ \dfrac{d}{da}\langle \psi_1^{k+1}, R_1\mu\rangle + A_1(a; \psi_1^{k+1}, R_1\mu) = & \forall \mu \in \Lambda \\ \qquad = \dfrac{d}{da}\langle u_1^{k+1}, R_1\mu\rangle + \dfrac{d}{da}\langle u_2^{k+1}, R_2\mu\rangle & \\ \qquad\quad + A_1(a; u_1^{k+1}, R_1\mu) + A_2(a; u_2^{k+1}, R_2\mu) - (f, R_2\mu) - (f, R_1\mu) & \\ \psi_1^{k+1}(0, x) = 0 & \end{cases}$$

and

$$\begin{cases} \dfrac{d}{da}\,\langle \psi_2^{k+1}, v_2\rangle + A_2(a;\psi_2^{k+1}, v_2) = (f_2, v_2) & \forall v_2 \in V_2 \\ \dfrac{d}{da}\,\langle \psi_2^{k+1}, R_2\mu\rangle + A_2(a;\psi_2^{k+1}, R_2\mu) = & \forall \mu \in \Lambda \\ \qquad = \dfrac{d}{da}\,\langle u_1^{k+1}, R_1\mu\rangle + \dfrac{d}{da}\,\langle u_2^{k+1}, R_2\mu\rangle & \\ \qquad\quad + A_1(a; u_1^{k+1}, R_1\mu) + A_2(a; u_2^{k+1}, R_2\mu) - (f, R_2\mu) - (f, R_1\mu) & \\ \psi_2^{k+1}(0,x) = 0. & \end{cases}$$

Step 3. Set

$$\lambda^{k+1} = \lambda^k - \vartheta\left(\frac{k_1}{k_1+k_2}\psi_1^{k+1} - \frac{k_2}{k_1+k_2}\psi_2^{k+1}\right)_{|(0,a_\dagger)\times\Gamma}$$

and iterate until convergence.

For a more detailed description of the algorithm we refer to [3].

3 Numerical Results

In this section we consider a population spreading in a one dimensional environment constituted of two layers. We solve problem (7) on the domain $\Omega = [0,1]$, and we assume $a_\dagger = 100$ as maximal age. In the numerical tests we choose $\Delta a = 2$, as well as $\Delta t = 1$. We let $\Omega = \Omega_1 \cup \Omega_2$, with $\Omega_1 = (0,\alpha)$, $\Omega_2 = (\alpha, 1)$, and we discretize problem (7) in space via $\mathbb{P}_1$ finite elements. We solve each subproblem by computing the integral in (5) via a Simpson quadrature rule, and by advancing implicitly in age (5) via a backward Euler scheme. For a more detailed description of the numerical approximation of (3) in a single domain we refer to [2]. We consider diffusion coefficients that are uniform in age and heterogeneous in space, with the ratio $\delta k = k_1/k_2$ up to 10^4. The maternity function and initial profile are given in

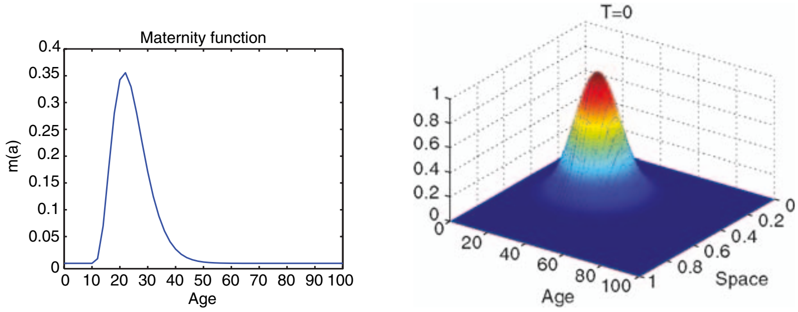


Fig. 1. Maternity function (left) and age-space initial profile (right) for the test cases

Fig. 1. In Table 1 and 2 we report the iteration counts at different time levels for two different positions of the interface, $\alpha = 0.5$ and $\alpha = 0.7$, with a mesh size of $h = 1/100$ in both subdomains. The stopping criterion is given by $\|\lambda^{k+1} - \lambda^k\|_0 / \|\lambda^k\|_0 < 10^{-6}$. The number of iterations increases with the amplitude of the jumps in the diffusive coefficients, but the algorithm appears to be robust with respect to the position of the interface and with respect to the evolution in time. In Table 3 we report the iteration counts at different time levels for different mesh sizes in Ω_1 and Ω_2. We choose $k_1 = .1$, $k_2 = .01$ and the interface is $\alpha = .6$. The algorithm appears insensitive to the difference of mesh sizes between the two subdomains. In Figure 2 we report the time evolution of the space profile of individuals of age 20 with $\delta k = 100$ and the evolution of the iteration counts (with $\alpha = 0.5$ and $\delta k = 1, 10, 100$) for a longer simulation, with stopping criterion set at 10^{-10}. The jump in the normal derivative due to the high heterogeneity of the spatial medium is clearly visible, and the robustness of the algorithm with respect to the evolution in time is evident. Finally, in Figure 3 we report the age-space profile of the solution at time $T = 5$, with $\delta k = 100$.

The numerical tests are performed with MATLAB® 6.1. A more detailed description of the test cases as well as further numerical results in two dimensions in space can be found in a forthcoming paper ([3]).

Table 1. Two subdomains, $\alpha = 0.5$, $h_1 = h_2 = 1/100$: iteration counts per time step: $\|\lambda^{k+1} - \lambda^k\|_0 / \|\lambda^k\|_0 < 10^{-6}$.

δk	$T = 1$	$T = 3$	$T = 6$	$T = 9$	$T = 12$	$T = 15$	$T = 20$
1	13	11	11	10	10	10	10
10	17	15	14	14	14	14	13
10^2	23	20	20	20	19	19	19
10^3	26	23	23	23	22	22	22
10^4	32	27	27	26	26	25	25

Table 2. Two subdomains, $\alpha = 0.7$, $h_1 = h_2 = 1/100$: iteration counts per time level: $\|\lambda^{k+1} - \lambda^k\|_0 / \|\lambda^k\|_0 < 10^{-6}$.

δk	$T = 1$	$T = 3$	$T = 6$	$T = 9$	$T = 12$	$T = 15$	$T = 20$
1	17	11	11	10	10	10	10
10	22	14	14	14	14	13	13
10^2	32	20	19	19	19	18	18
10^3	37	23	22	22	22	21	21
10^4	44	26	26	25	25	25	25

Table 3. Two subdomains, $\alpha = 0.6$, $\nu_1 = .1$, $\nu_2 = .01$: iteration counts per time level: $\|\lambda^{k+1} - \lambda^k\|_0 / \|\lambda^k\|_0 < 10^{-6}$.

h_1/h_2	$T = 1$	$T = 3$	$T = 6$	$T = 9$	$T = 12$	$T = 15$	$T = 20$
1	19	14	14	14	14	13	13
10	19	14	14	14	14	13	13
50	19	14	14	14	14	13	13

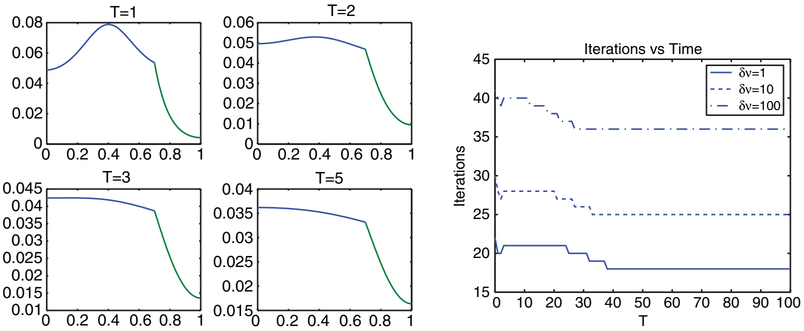


Fig. 2. Time evolution of the profile at age $a = 20$ (left, $\delta k = 100$) and Iterations count vs Time evolution (right, $\alpha = 0.5$, $\|\lambda^{k+1} - \lambda^k\|_0 / \|\lambda^k\|_0 < 10^{-10}$).

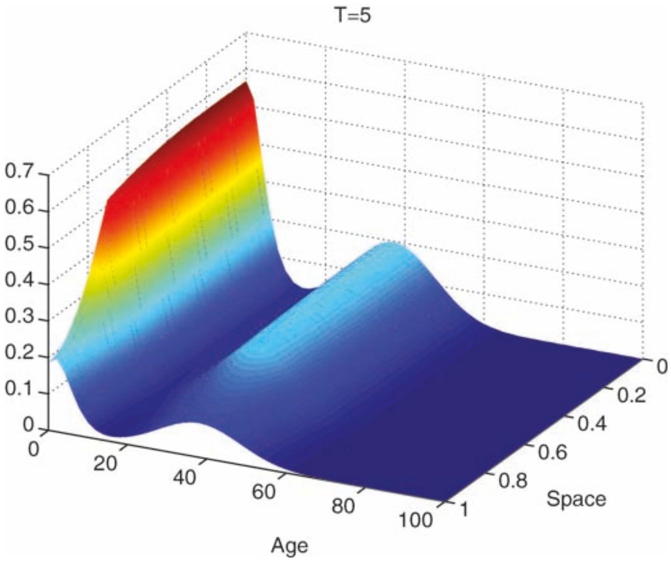


Fig. 3. Age-space profile at time $T = 5$, $\delta k = 100$.

4 Conclusions

We proposed here a balancing Neumann-Neumann procedure to approximate the solution of the diffusion of an age-structured population in a multilayer environment. The proposed algorithm appears to be very robust in terms of iteration counts with respect to the mesh size, the position of the interface, and the heterogeneities in the viscosity coefficients.

References

[1] B.P. Ayati and T. Dupont. Galerkin methods in age and space for a population model with nonlinear diffusion. *SIAM J. Numer. Anal.*, 40(3):1064–1076, 2002.

[2] C. Cusulin and L. Gerardo-Giorda. A FEM-Galerkin approximation for diffusion in age-structured population dynamics. Technical report, Dep. Math., University of Trento, 2006. In preparation.

[3] C. Cusulin and L. Gerardo-Giorda. Numerical approximation of the diffusion of an age-structured population in a multi-layer environment. Technical report, Dep. Math., University of Trento, 2006. In preparation.

[4] M.-Y. Kim. Galerkin methods for a model of population dynamics with nonlinear diffusion. *Numer. Methods Partial Differential Equations*, 12:59–73, 1996.

[5] M.-Y. Kim and E.-J. Park. Characteristic finite element methods for diffusion epidemic models with age-structured populations. *Comput. Math. Appl.*, 97:55–70, 1998.

[6] F.A. Milner. A numerical method for a model of population dynamics with spatial diffusion. *Comput. Math Appl.*, 19(31), 1990.

[7] A. Okubo and S.A. Levin. *Diffusion and Ecological Problems: Modern Perspectives.* Springer, New York, 2001.

[8] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations.* Springer-Verlag, Berlin, 1994.

[9] A. Quarteroni and A. Valli. *Domain Decompostion Methods for Partial Differential Equations.* Oxford University Press, 1999.

[10] N. Shigesada and K. Kawasaki. *Biological Invasions: Theory and Practice.* Oxford University Press, New York, 1997.

Why Classical Schwarz Methods Applied to Certain Hyperbolic Systems Converge Even Without Overlap

Victorita Dolean[1] and Martin J. Gander[2]

[1] Univ. de Nice Sophia-Antipolis, Laboratoire J.-A. Dieudonné, UMR CNRS No. 6621, Nice, France. dolean@math.unice.fr

[2] Section de Mathématiques, Université de Genève, CP 240, 1211 Genève. Martin.Gander@math.unige.ch

Summary. Overlap is essential for the classical Schwarz method to be convergent when solving elliptic problems. Over the last decade, it was however observed that when solving systems of hyperbolic partial differential equations, the classical Schwarz method can be convergent even without overlap. We show that the classical Schwarz method without overlap applied to the Cauchy-Riemann equations which represent the discretization in time of such a system, is equivalent to an optimized Schwarz method for a related elliptic problem, and thus must be convergent, since optimized Schwarz methods are well known to be convergent without overlap.

1 Introduction

The classical Schwarz method applied to scaler partial differential equations has been widely studied, as one can see from the many contributions in the proceedings of the international conferences on domain decomposition methods. Over the last decade, optimized variants of this method have been developed, which use absorbing conditions as transmission conditions at the interfaces between subdomains, and converge significantly faster than the classical Schwarz methods, see [7] and references therein. Less is known about the behavior of the classical Schwarz method applied to systems of partial differential equations; for the Euler equations, see [8, 9, 2] and [4, 5].

We show in this paper that the classical Schwarz method, which uses characteristic Dirichlet transmission conditions between subdomains, applied to the Cauchy Riemann equations is equivalent to an optimized Schwarz method applied to a well known equivalent elliptic problem. This explains why the classical Schwarz method in that case can be convergent even without overlap, and it allows us to develop more effective Schwarz methods for systems of partial differential equations. The extension of this idea to the more realistic case of Maxwell's equations, both in the time-harmonic and time-discretized case, can be found in [3].

2 Cauchy-Riemann Equations and Scalar Equivalent

To analyze the relationship between Schwarz methods for scalar partial differential equations (PDEs) and systems of PDEs, we use the Cauchy-Riemann equations

$$\mathcal{L}\mathbf{u} := \sqrt{\eta}\mathbf{u} + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \partial_x \mathbf{u} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \partial_y \mathbf{u} = \mathbf{f} := \begin{pmatrix} f \\ g \end{pmatrix}, \quad \mathbf{u} := \begin{pmatrix} u \\ v \end{pmatrix}, \tag{1}$$

on $\Omega = [0,1] \times \mathbb{R}$, with boundary conditions

$$v(0,y) = r(y), \qquad u(1,y) = s(y), \quad y \in \mathbb{R}. \tag{2}$$

The equations (1) can be interpreted as the time discretization of the hyperbolic system

$$\partial_t \mathbf{u} + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \partial_x \mathbf{u} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} \partial_y \mathbf{u} = 0, \text{on } \Omega = [0,1] \times \mathbb{R} \times \mathbb{R}_+.$$

At each time step, the resolution of equations of the type (1) is needed. Imposing the unknowns entering along the characteristics at the boundaries of the domain Ω like in (2) leads to a well-posed problem.

The scalar partial differential equation

$$\tilde{\mathcal{L}}\tilde{u} \equiv \eta\tilde{u} - \Delta\tilde{u} = \tilde{f}, \quad \text{in } \Omega, \tag{3}$$

with the boundary conditions

$$(\partial_x - \sqrt{\eta})\tilde{u}(0,y) = \tilde{r}(y), \quad \tilde{u}(1,y) = \tilde{s}(y), \quad y \in \mathbb{R} \tag{4}$$

is very much related to the Cauchy-Riemann equations:

Proposition 1. *If $\tilde{f} = (\sqrt{\eta} + \partial_x)f - \partial_y g$, $\tilde{r} = \partial_y r - f(0,\cdot)$ and $\tilde{s} = s$, then the velocity component u of the Cauchy-Riemann equations (1) with boundary conditions (2) coincides with the solution $\tilde{u}$ of the elliptic problem (3) with boundary conditions (4) for all $x, y \in \Omega$.*

A similar elliptic PDE can also be derived for v, but we will not need it for what follows.

3 Classical Schwarz Algorithm

We decompose the domain Ω into two overlapping or non-overlapping subdomains $\Omega_1 = (0,b) \times \mathbb{R}$ and $\Omega_2 = (a,1) \times \mathbb{R}$, and we denote the overlap by $L := b - a \geq 0$. A classical Schwarz algorithm for the Cauchy-Riemann equations (1) on these two subdomains is then defined by

$$\begin{array}{llll} \mathcal{L}\mathbf{u}_1^n = \mathbf{f}, & \text{in } \Omega_1, & \mathcal{L}\mathbf{u}_2^n = \mathbf{f}, & \text{in } \Omega_2, \\ v_1^n(0,y) = r(y), & y \in \mathbb{R}, & u_2^n(1,y) = s(y), & y \in \mathbb{R}, \\ u_1^n(b,y) = u_2^{n-1}(b,y), & y \in \mathbb{R}, & v_2^n(a,y) = v_1^{n-1}(a,y), & y \in \mathbb{R}, \end{array} \tag{5}$$

where $\mathbf{u}_j^n = (u_j^n, v_j^n)$ denotes the n-th iterate of $\mathbf{u}$ in domain Ω_j, $j = 1, 2$. Note that in this classical form of the Schwarz algorithm for the system of PDEs, we respected in the transmission conditions the information exchange along the characteristic directions, which is the most natural approach to follow when applying domain decomposition methods to hyperbolic problems, see for example [1, 9].

From the relation between the Cauchy-Riemann equations (1) and the associated elliptic problem (3) stated in Proposition 1, the related Schwarz algorithm for the elliptic problem is

$$\begin{aligned} \tilde{\mathcal{L}}\tilde{u}_1^n &= \tilde{f}, && \text{in } \Omega_1 & \tilde{\mathcal{L}}\tilde{u}_2^n &= \tilde{f}, && \text{in } \Omega_2 \\ \mathcal{B}\tilde{u}_1^n(0,y) &= \tilde{r}(y), && y\in\mathbb{R}, & \tilde{u}_2^n(1,y) &= \tilde{s}(y), && y\in\mathbb{R}, \\ \tilde{u}_1^n(b,y) &= \tilde{u}_2^{n-1}(b,y), && y\in\mathbb{R}, & \mathcal{B}\tilde{u}_2^n(a,y) &= \mathcal{B}\tilde{u}_1^{n-1}(a,y), && y\in\mathbb{R} \end{aligned} \tag{6}$$

where $\mathcal{B} = (\partial_x - \sqrt{\eta})$.

Theorem 1. *If algorithm (6) is started with the initial guess $\tilde{u}_1^0 = u_1^0$ and $\tilde{u}_2^0 = u_2^0$, then the iterates of algorithm (6) and algorithm (5) coincide, $u_l^n(x,y) = \tilde{u}_l^n(x,y)$ for all $(x,y)\in\Omega_l$, $l=1,2$ and $n\geq 1$.*

Proof. The proof is by induction. Proposition 1 shows the result for $n=1$. Assume then that the result is true at iteration $n-1$. Let $u^{1,n}$, $v^{1,n}$, $u^{2,n}$, and $v^{2,n}$ be the iterates of the Schwarz algorithm applied to the Cauchy-Riemann equations. We then have, on the one hand

$$u^{1,n}(b,y) = u^{2,n-1}(b,y) = \tilde{u}^{2,n-1}(b,y) = \tilde{u}^{1,n}(b,y).$$

On the other hand, differentiating the interface condition on v in (5) with respect to y and using the first Cauchy-Riemann equation, we get

$$(\partial_x - \sqrt{\eta})u^{2,n} - f = \partial_y v^{2,n} = \partial_y u^{1,n-1} = (\partial_x - \sqrt{\eta})u^{1,n-1} - f.$$

When evaluating the above expression at $x=a$, the f terms cancel, and we obtain

$$(\partial_x - \sqrt{\eta})u^{2,n} = (\partial_x - \sqrt{\eta})u^{1,n-1} = (\partial_x - \sqrt{\eta})\tilde{u}^{1,n-1} = (\partial_x - \sqrt{\eta})\tilde{u}^{2,n}.$$

Since the boundary conditions at $(0,y)$ and $(1,y)$ stay the same, the result follows from Proposition 1.

This theorem shows why the classical Schwarz algorithm (5) with characteristic Dirichlet transmission conditions for the Cauchy Riemann equations can converge even without overlap: it is equivalent to an optimized Schwarz method for a related elliptic PDE, and optimized Schwarz methods are also convergent without overlap, see [7].

We analyze now the convergence rate of Algorithm (5) when the domain is the entire plane, $\Omega = \mathbb{R}^2$, and the subdomains are $\Omega_1 = (-\infty, L)\times\mathbb{R}$ and $\Omega_2 = (0,\infty)\times\mathbb{R}$, $L\geq 0$. Let $\mathbf{e}_l^n(x,y) = (d_l^n(x,y), e_l^n(x,y))^t := \mathbf{u}(x,y) - \mathbf{u}_l^n(x,y)$, $l=1,2$ denote the error at iteration n. Then the $\mathbf{e}_l^n$ satisfy the homogeneous version of Algorithm (5), which after a Fourier transform $\mathcal{F}$ in y with parameter k, $\hat{\mathbf{e}}_l^n := \mathcal{F}(\mathbf{e}_l^n)$, gives

$$\begin{aligned} \hat{\mathcal{L}}\hat{\mathbf{e}}_1^n &= \mathbf{0}, && \text{in } \Omega_1, & \hat{\mathcal{L}}\hat{\mathbf{e}}_2^n &= \mathbf{0}, && \text{in } \Omega_2, \\ \hat{e}_1^n(-\infty,k) &= 0, && k\in\mathbb{R}, & \hat{d}_2^n(\infty,k) &= 0, && k\in\mathbb{R}, \\ \hat{d}_1^n(L,k) &= \hat{d}_2^{n-1}(L,k), && k\in\mathbb{R}, & \hat{e}_2^n(0,k) &= \hat{e}_1^{n-1}(0,k), && k\in\mathbb{R}, \end{aligned} \tag{7}$$

and $\hat{\mathcal{L}}$ denotes the action of the operator $\mathcal{L}$ after the Fourier transform in y, i.e.

$$\hat{\mathcal{L}}\hat{\mathbf{u}} := \mathcal{F}(\mathcal{L}\mathbf{u}) = \sqrt{\eta}\hat{\mathbf{u}} + \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix}\partial_x\hat{\mathbf{u}} + \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix} ik\hat{\mathbf{u}}.$$

Theorem 2. *If the initial error on the interfaces contains the Fourier components $\hat{\mathbf{e}}_1^0(L,k)$ and $\hat{\mathbf{e}}_2^0(0,k)$, $k \in \mathbb{R}$, then for any overlap $L \geq 0$, algorithm (5) converges for all k,*

$$|\hat{\mathbf{e}}_1^{2n}(L,k)| + |\hat{\mathbf{e}}_2^{2n}(0,k)| \leq (\rho(\eta,L,k))^2 \left(|\hat{\mathbf{e}}_1^0(L,k)| + |\hat{\mathbf{e}}_2^0(0,k)|\right), \tag{8}$$

and the convergence factor is given by

$$\rho(\eta,L,k) = \sqrt{\frac{\sqrt{\eta+k^2}-\sqrt{\eta}}{\sqrt{\eta+k^2}+\sqrt{\eta}} e^{-L\sqrt{\eta+k^2}}} < 1, \quad \forall k \in \mathbb{R}. \tag{9}$$

Proof. Solving (7) at iteration $n+1$, we obtain

$$\hat{\mathbf{e}}^{1,n+1} = \alpha^{n+1} e^{\lambda(x-L)} \begin{pmatrix} \sqrt{\eta+k^2}+\sqrt{\eta} \\ -ik \end{pmatrix}, \ \hat{\mathbf{e}}^{2,n+1} = \beta^{n+1} e^{-\lambda x} \begin{pmatrix} -ik \\ \sqrt{\eta+k^2}+\sqrt{\eta} \end{pmatrix}, \tag{10}$$

where $\lambda = \sqrt{\eta+k^2}$, and α^{n+1} and β^{n+1} are determined by the interface conditions to be

$$\alpha^{n+1} = \beta^n \frac{-ik}{\sqrt{\eta+k^2}+\sqrt{\eta}} e^{-\sqrt{\eta+k^2}L}, \quad \beta^{n+1} = \alpha^n \frac{-ik}{\sqrt{\eta+k^2}+\sqrt{\eta}} e^{-\sqrt{\eta+k^2}L}.$$

Performing a double step, this leads to the square of the convergence factor

$$\rho(\eta,L,k)^2 := \frac{\alpha^{n+1}}{\alpha^{n-1}} = \frac{\beta^{n+1}}{\beta^{n-1}} = -\frac{\sqrt{\eta+k^2}-\sqrt{\eta}}{\sqrt{\eta+k^2}+\sqrt{\eta}} e^{-2L\sqrt{\eta+k^2}},$$

which implies the result by induction on n.

4 Optimized Schwarz Algorithm

Algorithm (6) is a rather unusual optimized Schwarz algorithm for the elliptic problem (3), since it still uses Dirichlet transmission conditions at one of the interfaces. The guiding principle behind optimized Schwarz methods is to use absorbing transmission conditions, i.e. approximations of transparent boundary conditions at the interfaces between subdomains. The Robin transmission condition on one of the interfaces in (6) can be interpreted as a zeroth order low frequency approximation of a transparent condition, see [6]. In order to find better transmission conditions for the Cauchy-Riemann equations, we now derive their associated transparent boundary conditions.

To this end, we consider the Cauchy-Riemann equations (1) on the domain $\Omega = (0,1) \times \mathbb{R}$, with $\mathbf{f} = (f,g)^T$ compactly supported in Ω, but with the new boundary conditions

$$(v + \mathcal{S}_1 u)(0,y) = 0, \quad (u + \mathcal{S}_2 v)(1,y) = 0, \qquad y \in R, \tag{11}$$

where the operators $\mathcal{S}_l$, $l = 1,2$ are general, pseudo-differential operators acting in the y direction.

Lemma 1. *If the operators $\mathcal{S}_l$, $l = 1, 2$, have the Fourier symbol*

$$\sigma_l := \mathcal{F}(\mathcal{S}_l) = \frac{ik}{\sqrt{\eta} + \sqrt{\eta + k^2}}, \quad l = 1, 2, \tag{12}$$

then the solution of the Cauchy-Riemann equations (1) on the domain $\Omega = (0, 1) \times \mathbb{R}$ with boundary conditions (11) coincides with the restriction to the domain Ω of the solution of the Cauchy-Riemann equations (1) posed on $\mathbb{R}^2$.

Proof. It suffices to show that the difference between the solution of the global problem and the solution of the restricted problem vanishes. This difference, denoted by $\mathbf{e}$, satisfies the homogeneous counterpart of (1) with boundary conditions (11), and its Fourier transform is

$$\hat{\mathbf{e}}(x, k) = \alpha e^{\sqrt{\eta + k^2} x} \begin{pmatrix} \sqrt{\eta + k^2} + \sqrt{\eta} \\ -ik \end{pmatrix} + \beta e^{-\sqrt{\eta + k^2} x} \begin{pmatrix} -ik \\ \sqrt{\eta + k^2} + \sqrt{\eta} \end{pmatrix}. \tag{13}$$

Now the first boundary condition in (11) implies $\beta \sqrt{\eta + k^2} = 0$, and hence $\beta = 0$, and the second one implies $\alpha \sqrt{\eta + k^2} e^{-\sqrt{\eta + k^2}} = 0$, which gives $\alpha = 0$, and hence $\hat{\mathbf{e}} \equiv 0$.

Remark 1. The symbols (12) can be written in several mathematically equivalent forms,

$$\sigma_l = \frac{ik}{\sqrt{\eta} + \sqrt{\eta + k^2}} = \frac{\sqrt{\eta} - \sqrt{\eta + k^2}}{ik} = \sqrt{\frac{\sqrt{\eta} - \sqrt{\eta + k^2}}{\sqrt{\eta} + \sqrt{\eta + k^2}}}. \tag{14}$$

The first form contains a local and a non-local term in k, since multiplication with ik corresponds to derivation in y, which is a local operation (as the application of any polynomial in ik would be), whereas the term containing the square-root of k^2 is a non-local operation. The second form contains two non-local operations, since the division by ik corresponds to an integration. This integration can however be passed to the other variable in (11) by multiplication with ik. The last form contains only non-local terms. These different forms motivate different local approximations of the transparent boundary conditions, and thus lead to different optimized Schwarz methods, as we will show in the sequel.

We now consider the associated elliptic equations (3) on the domain $\Omega = (0, 1) \times \mathbb{R}$, with f compactly supported in Ω, but with the new boundary conditions

$$(\partial_x - \tilde{\mathcal{S}}_1) u(0, y) = 0, \quad (\partial_x + \tilde{\mathcal{S}}_2) u(1, y) = 0, \qquad y \in R, \tag{15}$$

where the operators $\tilde{\mathcal{S}}_l$, $l = 1, 2$ are general, pseudo-differential operators acting in the y direction.

Lemma 2. *If the operators $\tilde{\mathcal{S}}_l$, $l = 1, 2$, have the Fourier symbol*

$$\tilde{\sigma}_l := \mathcal{F}(\tilde{\mathcal{S}}_l) = \sqrt{\eta + k^2}, \quad l = 1, 2, \tag{16}$$

then the solution of (3) on the domain $\Omega = (0, 1) \times \mathbb{R}$ with boundary conditions (15) coincides with the restriction to Ω of the solution of (3) on $\mathbb{R}^2$.

Proof. The proof follows as in Lemma 1 using Fourier analysis.

Proposition 2. *The velocity component u of the solution of the Cauchy-Riemann equations (1) with boundary conditions (11), (12) coincides with the solution $\tilde{u}$ of the elliptic problem (3) with boundary conditions (15), (16) for all $x, y \in \Omega = (0,1) \times \mathbb{R}$.*

Proof. We have already seen in Proposition 1 that the equations inside the domain coincide. It therefore suffices to show that the boundary conditions are also equivalent. By using the first Fourier transformed equation inside the domain, i.e. $ik\hat{v} = (\partial_x - \sqrt{\eta})\hat{u}$, the boundary condition at $x = 1$, i.e. $(\sqrt{\eta} + \sqrt{\eta + k^2})\hat{u} + ik\hat{v} = 0$, becomes $(\partial_x + \sqrt{\eta + k^2})\hat{u} = 0$, which is the transparent boundary condition for the elliptic equation. The same argument applies to the other boundary condition: using the first Fourier transformed equation, the boundary condition at $x = 0$ becomes $(\sqrt{\eta} + \sqrt{\eta + k^2})(\partial_x - \sqrt{\eta})\hat{u} - k^2\hat{u} = 0$. Taking into account that $k^2 = (\sqrt{\eta + k^2} + \sqrt{\eta})(\sqrt{\eta + k^2} - \sqrt{\eta})$, we further obtain $(\sqrt{\eta + k^2} + \sqrt{\eta})(\sqrt{\eta + k^2} - \partial_x)\hat{u} = 0$, which is equivalent to the transparent boundary condition for the scalar equation at $x = 0$.

We generalize now the classical Schwarz algorithm (5) by changing the transmission conditions at the interfaces,

$$\begin{aligned} \mathcal{L}\mathbf{u}_1^n &= 0, & &\text{in } \Omega_1, \\ u_1^n(L, y) + \mathcal{S}_1 v_1^n(L, y) &= u_2^{n-1}(L, y) + \mathcal{S}_1 v_2^{n-1}(L, y), & & \\ \mathcal{L}\mathbf{u}_2^n &= 0, & &\text{in } \Omega_2, \\ v_2^n(0, y) + \mathcal{S}_2 u_2^n(0, y) &= v_1^{n-1}(0, y) + \mathcal{S}_2 u_1^{n-1}(0, y). & & \end{aligned} \tag{17}$$

Proceeding as in Theorem 2, the convergence factor for a double iteration is

$$\rho_{opt}(\eta, L, k, \sigma_1, \sigma_2) = \left| \frac{-ik + \sigma_1(\sqrt{\eta + k^2} + \sqrt{\eta})}{\sqrt{\eta + k^2} + \sqrt{\eta} - ik\sigma_1} \, \frac{-ik + \sigma_2(\sqrt{\eta + k^2} + \sqrt{\eta})}{\sqrt{\eta + k^2} + \sqrt{\eta} - ik\sigma_2} e^{-2\sqrt{\eta + k^2}L} \right|^{\frac{1}{2}}. \tag{18}$$

A good choice of σ_l, $l = 1, 2$ is a choice that makes the convergence factor ρ_{opt} small for all values of k, and from (18), we see that the choice (12) is optimal, since then $\rho_{opt} \equiv 0$ for all k. But a good choice should also lead to transmission conditions which are as easy and inexpensive to use as the classical characteristic Dirichlet conditions. Guided by the equivalence with the scalar case, we will compare the following cases:

Case 1: $\sigma_1 = \sigma_2 = 0$, the classical algorithm (5) with convergence factor (9).

Case 2: $\sigma_1 = \frac{ik}{\sqrt{\eta} + p}$, $\sigma_2 = \frac{\sqrt{\eta} - p}{ik}$, $p > 0$, a mixed case, where the first form of the exact symbol in (14) is used to approximate σ_1 and the second form is used to approximate σ_2. This corresponds to first order transmission conditions, since ik corresponds to a derivative in y and the division by ik can be avoided by multiplying the entire transmission condition by ik. The convergence factor is

$$\rho_2(\eta, L, k, p) = \left| \left(\frac{\sqrt{\eta + k^2} - p}{\sqrt{\eta + k^2} + p} \right)^2 e^{-2\sqrt{\eta + k^2}L} \right|^{\frac{1}{2}}, \tag{19}$$

which is equivalent to the algorithm in the elliptic case with Robin transmission conditions $\partial_x \pm p$, see [6].

Case 3: $\sigma_1 = \sigma_2 = \sigma = \frac{ik}{\sqrt{\eta} + p}$, $p > 0$, where only the first form of the exact symbol (14) has been used to approximate both σ_1 and σ_2. The resulting convergence factor is

$$\rho_3(\eta, L, k, p) = \left| \frac{\sqrt{\eta + k^2} - \sqrt{\eta}}{\sqrt{\eta + k^2} + \sqrt{\eta}} \right|^{\frac{1}{2}} \rho_2(\eta, L, k, p) < \rho_2(\eta, L, k, p), \qquad (20)$$

and thus the convergence factor is smaller than in Case 2 by the same factor that was gained in Case 1 over the classical elliptic case.

Choosing the second form of the symbol (14) to approximate both σ_1 and σ_2 is not a good idea, since it inverts the additional low frequency factor, which is less than one in (20).

Case 4: $\sigma_1 = \frac{ik}{\sqrt{\eta}+p_1}$, $\sigma_2 = \frac{\sqrt{\eta}-p_2}{ik}$, $p_{1,2} > 0$, a choice motivated by Remark 1, which leads to the convergence factor

$$\rho_4(\eta, L, k, p_1, p_2) = \left| \frac{\sqrt{\eta + k^2} - p_1}{\sqrt{\eta + k^2} + p_1} \cdot \frac{\sqrt{\eta + k^2} - p_2}{\sqrt{\eta + k^2} + p_2} e^{-2\sqrt{\eta+k^2}L} \right|^{\frac{1}{2}}. \qquad (21)$$

This corresponds to the two-sided Robin transmission conditions in the elliptic case in [6], which are of the form $\partial_x - p_1$ for the first subdomain and $\partial_x + p_2$ for the second one.

Case 5: $\sigma_1 = \frac{ik}{\sqrt{\eta}+p_1}$, $\sigma_2 = \frac{ik}{\sqrt{\eta}+p_2}$, $p_{1,2} > 0$, which gives the even better convergence factor

$$\rho_5(\eta, L, k, p_1, p_2) = \left| \frac{\sqrt{\eta + k^2} - \sqrt{\eta}}{\sqrt{\eta + k^2} + \sqrt{\eta}} \right|^{\frac{1}{2}} \rho_4(\eta, L, k, p_1, p_2) < \rho_4(\eta, L, k, p_1, p_2).$$

In the cases with parameters, the best choice for the parameters is in general the one that minimizes the convergence factor for all $k \in K$, where K denotes the set of relevant numerical frequencies, for example $K = [k_{\min}, k_{\max}]$. One therefore needs to solve the min-max problems

$$\min_{p>0} \max_{k \in K} \rho_j(\eta, L, k, p),\ j = 2, 3, \quad \min_{p_1,p_2>0} \max_{k \in K} \rho_j(\eta, L, k, p_1, p_2),\ j = 4, 5. \qquad (22)$$

In Case 2 and 4, the solution of the problem is already given in [6] for the equivalent elliptic case, and can therefore directly be used for the Cauchy-Riemann equations. The other cases are specific to the Cauchy-Riemann equations and an asymptotic analysis similar to the one shown in [6] leads to the results given in Table 1, where the estimate $k_{\max} = \frac{C}{h}$, C a positive constant, was used (a reasonable value would be $C = \pi$).

One can clearly see in this table that there are much better transmission conditions than the characteristic ones for the Cauchy-Riemann equations: for a Schwarz algorithm with overlap of the order of the mesh parameter, $L = h$, the characteristic transmission conditions lead to a convergence factor $1 - O(\sqrt{h})$, which depends strongly on h, whereas with better transmission conditions, one can achieve the convergence factor $1 - O(h^{\frac{1}{6}})$, which now depends only very weakly on h, at the same cost per iteration. Similar results also hold for the Schwarz algorithm without overlap, as shown in Table 1.

5 Numerical Experiments

We now show numerical experiments for the Cauchy-Riemann equations solved on the unit square $\Omega = (0, 1) \times (0, 1)$. We decompose the unit square into two subdomains $\Omega_1 = (0, b) \times (0, 1)$ and $\Omega_2 = (a, 1) \times (0, 1)$, where $0 < a \leq b < 1$, and

Table 1. Asymptotic convergence rate and optimal choice of the parameters in the transmission conditions for the five variants of the optimized Schwarz method applied to the Cauchy-Riemann equations, when the overlap L or the mesh parameter h is small, and the maximum numerical frequency is estimated by $k_{\max} = \frac{C}{h}$.

	with overlap, $L > 0$		without overlap, $L = 0$	
Case	ρ	parameters	ρ	parameters
1	$1 - 2\eta^{\frac{1}{4}}\sqrt{L}$	none	$1 - \frac{\sqrt{\eta}}{C}h$	none
2	$1-2^{\frac{13}{6}}\eta^{\frac{1}{6}}L^{\frac{1}{3}}$	$p = \frac{2^{-\frac{1}{3}}\eta^{\frac{1}{3}}}{L^{\frac{1}{3}}}$	$1 - \frac{4\eta^{\frac{1}{4}}\sqrt{h}}{\sqrt{C}}$	$p = \frac{\sqrt{C}\eta^{\frac{1}{4}}}{\sqrt{h}}$
3	$1 - 2^{\frac{3}{2}}\eta^{\frac{1}{8}}L^{\frac{1}{4}}$	$p = \frac{\eta^{\frac{1}{4}}}{\sqrt{L}}$	$1 - \frac{2^{\frac{4}{3}}\eta^{\frac{1}{6}}}{C^{\frac{1}{3}}}h^{\frac{1}{3}}$	$p = \frac{2^{\frac{1}{3}}C^{\frac{2}{3}}\eta^{\frac{1}{6}}}{h^{\frac{2}{3}}}$
4	$1-2^{\frac{4}{5}}\eta^{\frac{1}{10}}L^{\frac{1}{5}}$	$p_1=\frac{\eta^{\frac{1}{5}}}{2^{\frac{2}{5}}L^{\frac{3}{5}}}$, $p_2=\frac{\eta^{\frac{2}{5}}}{16^{\frac{1}{5}}L^{\frac{1}{5}}}$	$1 - \frac{\sqrt{2}\eta^{\frac{1}{8}}}{C^{\frac{1}{4}}}h^{\frac{1}{4}}$	$p_1 = \frac{\sqrt{2}C^{\frac{3}{4}}\eta^{\frac{1}{8}}}{h^{\frac{3}{4}}}$, $p_2 = \frac{C^{\frac{1}{4}}\eta^{\frac{3}{8}}}{\sqrt{2}h^{\frac{1}{4}}}$
5	$1 - 2\eta^{\frac{1}{12}}L^{\frac{1}{6}}$	$p_1 = \frac{\eta^{\frac{1}{3}}}{L^{\frac{1}{3}}}$, $p_2 = \frac{\eta^{\frac{1}{6}}}{L^{\frac{2}{3}}}$	$1 - \frac{2^{\frac{4}{5}}\eta^{\frac{1}{10}}}{C^{\frac{1}{5}}}h^{\frac{1}{5}}$	$p_1=\frac{(2C)^{\frac{4}{5}}\eta^{\frac{1}{10}}}{h^{\frac{4}{5}}}$, $p_2=\frac{(2C)^{\frac{2}{5}}\eta^{\frac{3}{10}}}{h^{\frac{2}{5}}}$

therefore the overlap is $L = b - a$, and we consider both decompositions with and without overlap. We discretize the equations using the finite volume method on a uniform mesh with mesh parameter h. In all comparisons that follow, we simulate directly the error equations, $f = 0$, and we use a random initial guess to ensure that all the frequency components are present in the iteration.

Table 2 shows the iteration count for all Schwarz algorithms considered, in the overlapping and non-overlapping case, and when the mesh is refined.

Table 2. Number of iterations to attain convergence for different interface conditions and different mesh sizes in the overlapping and non-overlapping case. The tolerance is fixed at $\varepsilon = 10^{-6}$.

	with overlap, $L = 3h$				without overlap, $L = 0$			
h	1/32	1/64	1/128	1/256	1/32	1/64	1/128	1/256
Case 1	16	24	34	48	131	203	355	593
Case 2	11	14	17	22	51	78	107	157
Case 3	10	12	14	16	18	25	41	131
Case 4	11	13	14	17	27	30	35	43
Case 5	9	10	11	13	17	19	23	31

These results are in good agreement with the theoretical results in Table 1: the classical algorithm has the strongest dependence on the mesh parameter, and the other algorithms become less and less dependent.

6 Conclusions

We have shown for the Cauchy-Riemann equations that the classical Schwarz algorithm with characteristic Dirichlet transmission conditions can be convergent even without overlap. This is because it corresponds to a simple optimized Schwarz method for an equivalent elliptic problem, and optimized Schwarz methods are convergent without overlap. We then showed that there are more effective transmission conditions than the characteristic Dirichlet conditions, and we analyzed an entire hierarchy of transmission conditions with better and better performance.

Since the Cauchy-Riemann equations can be interpreted as a time discretization of a hyperbolic system of equations, our analysis indicates that more effective transmission conditions than the characteristic ones can be found for hyperbolic problems, and for their time discretized counterparts. Convergence almost independent of the mesh parameter can be achieved with and without overlap. We have extended these ideas to Maxwell's equations, see [3], and also obtained a similar hierarchy of methods with better and better performance.

References

[1] M. Bjørhus. Semi-discrete subdomain iteration for hyperbolic systems. Technical Report 4, NTNU, 1995.

[2] S. Clerc. Non-overlapping Schwarz method for systems of first order equations. *Contemp. Math.*, 218:408–416, 1998.

[3] V. Dolean, L. Gerardo-Giorda, and M.J. Gander. Optimized Schwarz methods for Maxwell's equations, 2007. Submitted.

[4] V. Dolean, S. Lanteri, and F. Nataf. Construction of interface conditions for solving compressible Euler equations by non-overlapping domain decomposition methods. *Internat. J. Numer. Methods Fluids*, 40:1485–1492, 2002.

[5] V. Dolean, S. Lanteri, and F. Nataf. Convergence analysis of a Schwarz type domain decomposition method for the solution of the Euler equations. *Appl. Numer. Math.*, 49:153–186, 2004.

[6] M.J. Gander. Optimized Schwarz methods for Helmholtz problems. In *Thirteenth International Conference on Domain Decomposition*, pages 245–252, 2001.

[7] M.J. Gander. Optimized Schwarz methods. *SIAM J. Numer. Anal.*, 44(2):699–731, 2006.

[8] A. Quarteroni. Domain decomposition methods for systems of conservation laws: spectral collocation approximation. *SIAM J. Sci. Stat. Comput.*, 11:1029–1052, 1990.

[9] A. Quarteroni and L. Stolcis. Homogeneous and heterogeneous domain decomposition methods for compressible flow at high Reynolds numbers. Technical Report 33, CRS4, 1996.

How to Use the Smith Factorization for Domain Decomposition Methods Applied to the Stokes Equations

Victorita Dolean[1], Frédéric Nataf[2], and Gerd Rapin[3]

[1] Univ. de Nice Sophia-Antipolis, Laboratoire J.-A. Dieudonné, Nice, France. dolean@math.unice.fr
[2] Laboratoire J. L. Lions, Université Pierre et Marie Curie, 75252 Paris Cedex 05, France. nataf@ann.jussieu.fr
[3] Math. Dep., NAM, University of Göttingen, D-37083, Germany. grapin@math.uni-goettingen.de

Summary. In this paper we demonstrate that the Smith factorization is a powerful tool to derive new domain decomposition methods for vector valued problems. Here, the factorization is applied to the two-dimensional Stokes system. The key idea is the transformation of the Stokes problem into a scalar bi-harmonic problem. We show how a proposed domain decomposition method for the bi-harmonic problem leads to an algorithm for the Stokes equations which inherits the convergence behavior of the scalar problem.

1 Introduction

The last decade has shown that Neumann-Neumann type algorithms, FETI, and BDDC methods are very efficient domain decomposition methods for scalar symmetric positive definite second order problems. Then, these methods have been extended to other problems, like advection-diffusion equations, plate or shell problems. Also for the Stokes equations several iterative substructuring methods have been discussed in the literature, like Neumann-Neumann precondtioners (cf. [6, 2]), FETI (cf. [3]) or BDDC methods (cf. [4]).

Our work is motivated by the fact that many domain decomposition methods for vector valued problems are less optimal than domain decomposition methods for scalar problems. Indeed, in the case of two subdomains consisting of the two half planes it is well known that Neumann-Neumann preconditioners are exact (the preconditioned operator simplifies to the identity) preconditioners for the Schur complement equation for scalar equations like the Laplace problem. Unfortunately, this is not valid for the Stokes problem as we have shown in [5] for standard Neumann-Neumann preconditioners. The goal of this paper is the derivation of an algorithm which preserves this property, cf. [1] for detailed proofs.

Using the Smith factorization we show the equivalence between the Stokes equations and a bi-harmonic problem in Section 2. The Smith factorization is a classical algebraic tool for matrices with polynomial entries. Then, in Section 3 we introduce an exact domain decomposition method for the bi-harmonic equation and transform it to the Stokes equations. Section 4 is dedicated to numerical results. Finally, we give some concluding remarks.

2 Equivalence Between the Stokes Equations and Bi-harmonic Problems

We will show the equivalence between the two-dimensional Stokes system

$$-\nu \Delta \boldsymbol{u} + \nabla p + c\boldsymbol{u} = \boldsymbol{f}, \quad \nabla \cdot \boldsymbol{u} = 0 \quad \text{in } \Omega$$

and a fourth order scalar problem (the bi-harmonic problem) by means of the Smith factorization. This is motivated by the fact that scalar problems are easier to manipulate and the construction of new algorithms is more intuitive. The approach is not limited to the two-dimensional case. The three-dimensional case is discussed in [1].

The data is given by $\boldsymbol{f} = (f_1, f_2)^T \in [L^2(\Omega)]^2$, $\nu > 0$, and $c \geq 0$. Very often c stems from an implicit time discretization and then c is given by the inverse of the time step size. We denote the two-dimensional Stokes operator by $\mathcal{S}_2(\boldsymbol{v}, q) := -\nu\Delta\boldsymbol{v} + c\boldsymbol{v} + \nabla q$. We recall the Smith factorization of a matrix with polynomial entries ([7], Theorem 1.4):

Theorem 1. *Let A be a $n \times n$ matrix with polynomial entries with respect to the variable λ: $A = (a_{ij}(\lambda))_{1\leq i,j\leq n}$. Then, there exist matrices E, D and F with polynomial entries satisfying the following properties:*

- *$det(E)$ and $det(F)$ are constants,*
- *D is a diagonal matrix uniquely determined up to a multiplicative constant,*
- *$A = EDF$.*

The Smith factorization is applied to the two-dimensional model problem $\mathcal{S}_2(\boldsymbol{u}, p) = \boldsymbol{g}$ in $\mathbb{R}^2$ with right hand side $\boldsymbol{g} = (f_1, f_2, 0)^T$ where we suppose that all variables vanish at infinity. Moreover, it is assumed that the coefficients c, ν are constants. The spatial coordinates are denoted by x and y. In order to apply the factorization to the Stokes system we first take formally the Fourier transform of $\mathcal{S}_2(\boldsymbol{u}, p) = \boldsymbol{g}$ with respect to y. The dual variable is denoted by k. The Fourier transform of a function f is written as $\hat{f}$ or $\mathcal{F}_y f$. Thus, we get

$$\hat{\mathcal{S}}_2(\hat{\boldsymbol{u}}, \hat{p}) = \begin{pmatrix} -\nu(\partial_{xx} - k^2) + c & 0 & \partial_x \\ 0 & -\nu(\partial_{xx} - k^2) + c & ik \\ \partial_x & ik & 0 \end{pmatrix} \begin{pmatrix} \hat{u} \\ \hat{v} \\ \hat{p} \end{pmatrix}. \tag{1}$$

Considering $\hat{\mathcal{S}}_2(\hat{\boldsymbol{u}}, \hat{p})$ as a matrix with polynomial entries with respect to ∂_x we perform for $k \neq 0$ the Smith factorization. We obtain

$$\hat{\mathcal{S}}_2 = \hat{E}_2 \hat{D}_2 \hat{F}_2 \tag{2}$$

with a diagonal matrix $\hat{D}_2 = \text{diag}(1, 1, (\partial_{xx} - k^2)\hat{\mathcal{L}}_2)$ and

$$\hat{F}_2 = \begin{pmatrix} \nu k^2 + c & \nu i k \partial_x & \partial_x \\ 0 & \hat{\mathcal{L}}_2 & ik \\ 0 & 1 & 0 \end{pmatrix}, \quad \hat{E}_2 = \hat{T}_2^{-1} \begin{pmatrix} ik\hat{\mathcal{L}}_2 & \nu\partial_{xxx} & -\nu\partial_x \\ 0 & \hat{T}_2 & 0 \\ ik\partial_x & -\partial_{xx} & 1 \end{pmatrix}$$

where T_2 is a differential operator in y-direction whose symbol is $ik(\nu k^2 + c)$. Moreover, $\hat{\mathcal{L}}_2 := \nu(-\partial_{xx} + k^2) + c$ is the Fourier transform of $\mathcal{L}_2 := -\nu\Delta + c$.

Remark 1. Thus, the Stokes problem $\mathcal{S}_2(\boldsymbol{u}, p) = \boldsymbol{g}$ in $\mathbb{R}^2$ can be written as

$$\hat{D}_2 \hat{\boldsymbol{w}} = \hat{E}_2^{-1}\hat{\boldsymbol{g}}, \qquad \hat{\boldsymbol{w}} := (\hat{w}_1, \hat{w}_2, \hat{w}_3)^T := \hat{F}_2(\hat{\boldsymbol{u}}, \hat{p})^T. \tag{3}$$

From (3) we get $\hat{w}_1 = (\hat{E}_2^{-1}\hat{\boldsymbol{g}})_1$ and $\hat{w}_2 = (\hat{E}_2^{-1}\hat{\boldsymbol{g}})_2$. Noticing that $\hat{w}_3 = \left(\hat{F}_2(\hat{\boldsymbol{u}}, \hat{p})^T\right)_3$ $= \hat{v}$ the previous equation yields after applying an inverse Fourier transform

$$\Delta(-\nu\Delta + c)v = \mathcal{F}_y^{-1}\left((\hat{E}_2^{-1}\hat{\boldsymbol{g}})_3\right). \tag{4}$$

Since the determinants of the matrices $\hat{E}_2$ and $\hat{F}_2$ are non-zero numbers (i.e. a polynomial of order zero) the entries of their inverses are still polynomial in ∂_x. Thus, applying $\hat{E}_2^{-1}$ to the right hand side $\hat{\boldsymbol{g}}$ amounts to taking derivatives of $\hat{\boldsymbol{g}}$ and making linear combinations of them. If the plane $\mathbb{R}^2$ is split into subdomains $\mathbb{R}^- \times \mathbb{R}$ and $\mathbb{R}^+ \times \mathbb{R}$ the application of $\hat{E}_2^{-1}$ and $\hat{F}_2^{-1}$ to a vector can be done for each subdomain independently. No communication between the subdomains is necessary. The local problems are only coupled by the biharmonic problem (4). Thus, we can obtain a domain decomposition method for the Stokes problem by defining a domain decomposition method for (4) and recasting it to the Stokes problem using the Smith factorization.

3 A New Algorithm for the Stokes Equations

We construct an algorithm for $\mathcal{B} := \Delta\mathcal{L}_2 = \Delta(-\nu\Delta + c)$ on the whole plane divided into two half-planes, which converges in two iterations. Then, via the Smith factorization, we recast it in a new algorithm for the Stokes system.

We consider the following problem: Find $\phi : \mathbb{R}^2 \to \mathbb{R}$ such that

$$\mathcal{B}(\phi) = g \text{ in } \mathbb{R}^2, \qquad |\phi(\boldsymbol{x})| \to 0 \text{ for } |\boldsymbol{x}| \to \infty \tag{5}$$

where g is a given right hand side. The domain $\Omega = \mathbb{R}^2$ is decomposed into two half planes $\Omega_1 = \mathbb{R}^- \times \mathbb{R}$ and $\Omega_2 = \mathbb{R}^+ \times \mathbb{R}$ with interface $\Gamma := \{0\} \times \mathbb{R}$. Let $(\boldsymbol{n}_i)_{i=1,2}$ be the outward normal of $(\Omega_i)_{i=1,2}$. In contrast to the overlapping additive Schwarz algorithm in [8] we propose an iterative-substructuring algorithm.

ALGORITHM 1 *For any initial values ϕ_1^0 and ϕ_2^0 with $\phi_1^0 = \phi_2^0$ and $\mathcal{L}_2\phi_1^0 = \mathcal{L}_2\phi_2^0$ on Γ we obtain $(\phi_i^{n+1})_{i=1,2}$ from $(\phi_i^n)_{i=1,2}$ by the following procedure:*

Correction step. *We compute the corrections $(\tilde{\phi}_i^{n+1})_{i=1,2}$:*

$$\begin{cases} \mathcal{B}\tilde{\phi}_1^{n+1} & = 0 \text{ in } \Omega_1 \\ \lim\limits_{|\mathbf{x}|\to\infty} |\tilde{\phi}_1^{n+1}| & = 0 \\ \dfrac{\partial\tilde{\phi}_1^{n+1}}{\partial\boldsymbol{n}_1} & = \gamma_1^n \text{ on } \Gamma \\ \dfrac{\partial\mathcal{L}_2\tilde{\phi}_1^{n+1}}{\partial\boldsymbol{n}_1} & = \gamma_2^n \text{ on } \Gamma \end{cases} \qquad \begin{cases} \mathcal{B}\tilde{\phi}_2^{n+1} & = 0 \text{ in } \Omega_2 \\ \lim\limits_{|\mathbf{x}|\to\infty} |\tilde{\phi}_2^{n+1}| & = 0 \\ \dfrac{\partial\tilde{\phi}_2^{n+1}}{\partial\boldsymbol{n}_2} & = \gamma_1^n \text{ on } \Gamma \\ \dfrac{\partial\mathcal{L}_2\tilde{\phi}_2^{n+1}}{\partial\boldsymbol{n}_2} & = \gamma_2^n \text{ on } \Gamma \end{cases} \tag{6}$$

where $\gamma_1^n = -\frac{1}{2}\left(\frac{\partial \phi_1^n}{\partial \boldsymbol{n}_1} + \frac{\partial \phi_2^n}{\partial \boldsymbol{n}_2}\right)$ *and* $\gamma_2^n = -\frac{1}{2}\left(\frac{\partial \mathcal{L}_2 \phi_1^n}{\partial \boldsymbol{n}_1} + \frac{\partial \mathcal{L}_2 \phi_2^n}{\partial \boldsymbol{n}_2}\right)$.

Updating step. *We update* $(\phi_i^{n+1})_{i=1,2}$ *by solving the local problems:*

$$\begin{cases} \mathcal{B}\phi_1^{n+1} & = g \text{ in } \Omega_1 \\ \lim\limits_{|\mathbf{x}|\to\infty} |\phi_1^{n+1}| = 0 & \\ \phi_1^{n+1} & = \phi_1^n + \delta_1^{n+1} \text{ on } \Gamma \\ \mathcal{L}_2\phi_1^{n+1} & = \mathcal{L}_2\phi_1^n + \delta_2^{n+1} \text{ on } \Gamma \end{cases} \qquad \begin{cases} \mathcal{B}\phi_2^{n+1} & = g \text{ in } \Omega_2, \\ \lim\limits_{|\mathbf{x}|\to\infty} |\phi_2^{n+1}| = 0 & \\ \phi_2^{n+1} & = \phi_2^n + \delta_1^{n+1} \text{ on } \Gamma \\ \mathcal{L}_2\phi_2^{n+1} & = \mathcal{L}_2\phi_2^n + \delta_2^{n+1} \text{ on } \Gamma \end{cases} \tag{7}$$

where $\delta_1^{n+1} = \frac{1}{2}(\tilde{\phi}_1^{n+1} + \tilde{\phi}_2^{n+1})$ *and* $\delta_2^{n+1} = \frac{1}{2}(\mathcal{L}_2\tilde{\phi}_1^{n+1} + \mathcal{L}_2\tilde{\phi}_2^{n+1})$.

Using the Fourier transform we can prove the following result.

Proposition 1. *Algorithm 1 converges in two iterations.*

After having found an optimal algorithm which converges in two steps for the fourth order operator $\mathcal{B}$ problem we focus on the Stokes system. It suffices to replace the operator $\mathcal{B}$ by the Stokes system and ϕ by the last component $(F_2(\boldsymbol{u}, p)^T)_3$ of the vector $F_2(\boldsymbol{u}, p)^T$ in the boundary conditions.

ALGORITHM 2 *We choose* $(\boldsymbol{u}_1^0, p_1^0)$ *and* $(\boldsymbol{u}_2^0, p_2^0)$ *such that* $(F_2(\boldsymbol{u}_1^0, p_1^0)^T)_3 = (F_2(\boldsymbol{u}_2^0, p_2^0)^T)_3$ *and* $\mathcal{L}_2(F_2(\boldsymbol{u}_1^0, p_1^0)^T)_3 = \mathcal{L}_2(F_2(\boldsymbol{u}_2^0, p_2^0)^T)_3$ *on* Γ.
We compute $((\boldsymbol{u}_i^{n+1}, p_i^{n+1}))_{i=1,2}$ *from* $((\boldsymbol{u}_i^n, p_i^n))_{i=1,2}$ *by the following iterative procedure:*

Correction step. *We compute the corrections* $((\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}))_{i=1,2}$*:*

$$\begin{cases} \mathcal{S}_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1}) & = 0 \text{ in } \Omega_1 \\ \lim\limits_{|\boldsymbol{x}|\to\infty} |\tilde{\boldsymbol{u}}_1^{n+1}| & = 0 \\ \dfrac{\partial(F_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1})^T)_3}{\partial \boldsymbol{n}_1} & = \gamma_1^n \text{ on } \Gamma \\ \dfrac{\partial\mathcal{L}_2(F_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1})^T)_3}{\partial \boldsymbol{n}_1} & = \gamma_2^n \text{ on } \Gamma \end{cases} \quad \begin{cases} \mathcal{S}_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1}) & = 0 \text{ in } \Omega_2 \\ \lim\limits_{|\boldsymbol{x}|\to\infty} |\tilde{\boldsymbol{u}}_2^{n+1}| & = 0 \\ \dfrac{\partial(F_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1})^T)_3}{\partial \boldsymbol{n}_2} & = \gamma_1^n \text{ on } \Gamma \\ \dfrac{\partial\mathcal{L}_2(F_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1})^T)_3}{\partial \boldsymbol{n}_2} & = \gamma_2^n \text{ on } \Gamma \end{cases} \tag{8}$$

where

$$\begin{aligned} \gamma_1^n &= -\frac{1}{2}\left(\frac{\partial(F_2(\boldsymbol{u}_1^n, p_1^n)^T)_3}{\partial \boldsymbol{n}_1} + \frac{\partial(F_2(\boldsymbol{u}_2^n, p_2^n)^T)_3}{\partial \boldsymbol{n}_2}\right) \\ \gamma_2^n &= -\frac{1}{2}\left(\frac{\partial\mathcal{L}_2(F_2(\boldsymbol{u}_1^n, p_1^n)^T)_3}{\partial \boldsymbol{n}_1} + \frac{\partial\mathcal{L}_2(F_2(\boldsymbol{u}_2^n, p_2^n)^T)_3}{\partial \boldsymbol{n}_2}\right). \end{aligned}$$

Updating step. *We update* $((\boldsymbol{u}_i^{n+1}, p_i^{n+1}))_{i=1,2}$ *by solving the local problems:*

$$\begin{cases} \mathcal{S}_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) & = \boldsymbol{g} \quad \text{in } \Omega_i \\ \lim\limits_{|\boldsymbol{x}|\to\infty} |\boldsymbol{u}_i^{n+1}| & = 0 \\ (F_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1})^T)_3 & = (F_2(\boldsymbol{u}_i^n, p_i^n)^T)_3 + \delta_1^{n+1} \text{ on } \Gamma \\ \mathcal{L}_2(F_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1})^T)_3 & = \mathcal{L}_2(F_2(\boldsymbol{u}_i^n, p_i^n)^T)_3 + \delta_2^{n+1} \text{ on } \Gamma \end{cases} \tag{9}$$

where

$$\begin{aligned} \delta_1^{n+1} &= \frac{1}{2}[(F_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1})^T)_3 + (F_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1})^T)_3], \\ \delta_2^{n+1} &= \frac{1}{2}[\mathcal{L}_2(F_2(\tilde{\boldsymbol{u}}_1^{n+1}, \tilde{p}_1^{n+1})^T)_3 + \mathcal{L}_2(F_2(\tilde{\boldsymbol{u}}_2^{n+1}, \tilde{p}_2^{n+1})^T)_3]. \end{aligned}$$

This algorithm seems quite complex since it involves third order derivatives of the unknowns in the boundary conditions on $(F_2(\tilde{\boldsymbol{u}}_i, \tilde{p}_i)^T)_3$. Writing $\boldsymbol{u}_i = (u_i, v_i)$ and using $(F_2(\tilde{\boldsymbol{u}}_i, \tilde{p}_i)^T)_3 = \tilde{v}_i$ it is possible to simplify it. By using the Stokes equations in the subdomains we can lower the degree of the derivatives in the boundary conditions. We further introduce the stress

$$\boldsymbol{\sigma}^i(\boldsymbol{u}, p) := \nu \partial_{\boldsymbol{n}_i} \boldsymbol{u} - p \boldsymbol{n}_i$$

on the boundary $\partial \Omega_i$ for a velocity $\boldsymbol{u} = (u, v)$, a pressure p and the normal vector $\boldsymbol{n}_i$. For any vector $\boldsymbol{u}$ its normal (resp. tangential) component on the interface is $u_{\boldsymbol{n}_i} = \boldsymbol{u} \cdot \boldsymbol{n_i}$ (resp. $u_{\boldsymbol{\tau}_i} = (I - \boldsymbol{n}_i \otimes \boldsymbol{n}_i)\boldsymbol{u}$). We denote $\sigma^i_{\boldsymbol{n}_i} := \sigma^i_{\boldsymbol{n}_i}(\boldsymbol{u}_i, p_i) \cdot \boldsymbol{n}_i$ and $\sigma^i_{\boldsymbol{\tau}_i} := (I - \boldsymbol{n}_i \otimes \boldsymbol{n}_i)\sigma^i$ as the normal and tangential parts of $\boldsymbol{\sigma}^i$, respectively. We can thus write the new algorithm for the Stokes equations for general decomposition into non overlapping subdomains: $\bar{\Omega} = \cup_{i=1}^N \bar{\Omega}_i$ and denote by Γ_{ij} the interface between subdomains Ω_i and Ω_j, $i \neq j$. The new algorithm for the Stokes system reads:

ALGORITHM 3 *Starting with an initial guess* $((\boldsymbol{u}_i^0, p_i^0))_{i=0}^N$ *satisfying* $\boldsymbol{u}^0_{i,\boldsymbol{\tau}_i} = \boldsymbol{u}^0_{j,\boldsymbol{\tau}_j}$ *and* $\sigma^i_{\boldsymbol{n}_i}(\boldsymbol{u}_i^0, p_i^0) = \sigma^j_{\boldsymbol{n}_j}(\boldsymbol{u}_j^0, p_j^0)$ *on* Γ_{ij}, $\forall i, j$, $i \neq j$, *the* **correction step** *is expressed as follows for* $1 \leq i \leq N$*:*

$$\begin{cases} \mathcal{S}_2(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) & = 0 \quad \text{in } \Omega_i \\ \tilde{u}_{i,\boldsymbol{n}_i}^{n+1} & = -\dfrac{1}{2}(u^n_{i,\boldsymbol{n}_i} + u^n_{j,\boldsymbol{n}_j}) \quad \text{on } \Gamma_{ij} \\ \boldsymbol{\sigma}^i_{\boldsymbol{\tau}_i}(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) & = -\dfrac{1}{2}(\boldsymbol{\sigma}^i_{\boldsymbol{\tau}_i}(\boldsymbol{u}_i^n, \tilde{p}_i^n) + \boldsymbol{\sigma}^j_{\boldsymbol{\tau}_j}(\boldsymbol{u}_j^n, \tilde{p}_j^n)) \quad \text{on } \Gamma_{ij} \end{cases} \tag{10}$$

followed by an **updating step** *for* $1 \leq i \leq N$*:*

$$\begin{cases} \mathcal{S}_2(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) & = \boldsymbol{g} \quad \text{in } \Omega_i \\ \boldsymbol{u}_{i,\boldsymbol{\tau}_i}^{n+1} & = \boldsymbol{u}^n_{i,\boldsymbol{\tau}_i} + \dfrac{1}{2}(\tilde{\boldsymbol{u}}^{n+1}_{i,\boldsymbol{\tau}_i} + \tilde{\boldsymbol{u}}^{n+1}_{j,\boldsymbol{\tau}_j}) \quad \text{on } \Gamma_{ij} \\ \sigma^i_{\boldsymbol{n}_i}(\boldsymbol{u}_i^{n+1}, p_i^{n+1}) & = \sigma^i_{\boldsymbol{n}_i}(\boldsymbol{u}_i^n, p_i^n) \\ & \quad + \dfrac{1}{2}(\sigma^i_{\boldsymbol{n}_i}(\tilde{\boldsymbol{u}}_i^{n+1}, \tilde{p}_i^{n+1}) + \sigma^j_{\boldsymbol{n}_j}(\tilde{\boldsymbol{u}}_j^{n+1}, \tilde{p}_j^{n+1})) \text{ on } \Gamma_{ij}. \end{cases} \tag{11}$$

Since Algorithm 3 is only a reformulation of Algorithm 1 we obtain:

Proposition 2. *For a domain* $\Omega = \mathbb{R}^2$ *divided into two non overlapping half planes, Algorithms 2 and 3 are equivalent and converge in two iterations.*

In each iteration step of Algorithm 3 two local boundary value problems have to be solved in each subdomain. Therefore the cost of an iteration step is the same as for the NN algorithm.

4 Numerical Results

For the discretization of the two-dimensional case we choose a second order centered Finite Volume approach with a staggered grid. We consider two different types of domain decomposition methods: the discrete version of Algorithm 3 and an accelerated version using the GMRES method.

In the sequel we compare the performance of the new algorithm with the standard Schur complement approach using a Neumann-Neumann preconditioner (without coarse space), cf. [2]. We consider the domain $\Omega = [0.2, 1.2] \times [0.1, 1.1]$. We choose

$\nu = 1$ and the right hand side $\boldsymbol{f}$ such that the exact solution $\boldsymbol{u} = (u, v)$ is given by $u(x, y) = \sin(\pi x)^3 \sin(\pi y)^2 \cos(\pi y)$, $v(x, y) = -\sin(\pi x)^2 \sin(\pi y)^3 \cos(\pi x)$ and $p(x, y) = x^2 + y^2$.

First, the interface system is solved by a purely iterative method (denoted respectively by it_{new} and it_{NN} for the new algorithm and the Neumann-Neumann preconditioner) and then accelerated by GMRES (denoted respectively by ac_{New} and ac_{NN}). In all tables we count the smallest number of iterations, which is needed to reduce the euclidian norm of the residual by $TOL = 10^{-8}$. In brackets the number of steps is printed, which is needed to achieve an error with respect to one-domain solution which is less than 10^{-6}. The case that the method is not converged within 100 steps is denoted by $-$.

We first consider a decomposition into two subdomains of same width and study the influence of the reaction parameter and of the mesh size on the convergence. We can see in Table 1 (left) that the convergence of the new algorithm is optimal. For the iterative version convergence is reached in two iterations. Since in this case the preconditioned operator for the corresponding Krylov method reduces in theory to the identity, the Krylov method converges in one step. This is also valid numerically. Moreover, both algorithm are completely insensitive with respect to the reaction parameter. The advantage in comparison to the Neumann-Neumann algorithm is obvious.

In Table 1 (right) we fix the reaction parameter $c = 10^{-5}$ and vary the mesh size: Both algorithms converge independently of the mesh size and, again, we observe a clearly better convergence behavior of the new algorithm. The same kind of results are valid for different values of c (not presented here).

Table 1. Influence of the reaction parameter on the convergence ($h = \frac{1}{96}$) (left), influence of the mesh size for $c = 10^{-5}$ (right).

c	it_{New}	it_{NN}	ac_{New}	ac_{NN}
10^2	2 (2)	16 (15)	1 (1)	6 (6)
10^0	2 (2)	17 (15)	1 (1)	6 (6)
10^{-3}	2 (2)	17 (15)	1 (1)	6 (6)
10^{-5}	2 (2)	17 (15)	1 (1)	6 (6)

h	it_{New}	it_{NN}	ac_{New}	ac_{NN}
1/24	2 (2)	16 (14)	1 (1)	6 (6)
1/48	2 (2)	17 (15)	1 (1)	6 (6)
1/96	2 (2)	17 (15)	1 (1)	6 (6)

Now, the case of a strip-wise decomposition into more than two subdomains is considered. The mesh size is fixed ($h = 1/96$) and for different values of c we vary the number of subdomains. In the case of a strip-wise decomposition into N subdomains, the iteration number is increasing very quickly for very small c and in Table 2 (left) we can see only a small advantage of the new algorithm over the more classical approach. For larger c (Table 2 (right)) the behavior of the two domain case is conserved. The number of iteration steps is almost reduced by a factor of two. Moreover, for all cases the convergence is still independent of the mesh size.

The final test cases treat general decompositions into $N \times N$ subdomains. Two different values for the reaction coefficient c are analyzed. The iterative variants do not converge in the multi-domain case with cross points within 100 steps (except one case), cf. Table 3. Applying the accelerated variants we observe in the case 2×2

Table 2. Influence of the number of subdomains ($h = \frac{1}{96}$): $c = 10^{-5}$ (left), $c = 10^2$ (right).

N	it_{New}	it_{NN}	ac_{New}	ac_{NN}
2	2 (2)	17 (15)	1 (1)	6 (6)
4	- (-)	- (-)	6 (8)	7 (-)
6	- (-)	- (-)	10 (15)	13 (-)
8	- (-)	- (-)	13 (21)	19 (-)

N	it_{New}	it_{NN}	ac_{New}	ac_{NN}
2	2 (2)	16 (15)	1 (1)	6 (6)
4	45 (34)	- (-)	5 (5)	10 (9)
6	- (-)	- (-)	8 (7)	15 (15)
8	- (-)	- (-)	11 (10)	21 (21)

Table 3. Influence of the number of subdomains ($h = \frac{1}{96}$): $c = 1$ (left), $c = 10^2$ (right).

$N \times N$	it_{New}	it_{NN}	ac_{New}	ac_{NN}
2x2	- (-)	- (-)	9 (9)	13 (13)
3x3	- (-)	- (-)	27 (30)	26 (28)
4x4	- (-)	- (-)	35 (39)	36 (39)

$N \times N$	it_{New}	it_{NN}	ac_{New}	ac_{NN}
2x2	66 (61)	- (-)	8 (7)	11 (11)
3x3	- (-)	- (-)	21 (22)	21 (21)
4x4	- (-)	- (-)	25 (27)	27 (27)

a faster convergence of the new algorithm. For more subdomains both algorithms need almost the same number of iteration steps. This behavior can be explained by the presence of floating subdomains, which causes additional problems. Here, a suitable coarse space will decrease the number of needed iteration steps.

5 Conclusion

We have shown that the Smith factorization is a powerful tool in order to derive new domain decomposition methods for vector valued partial differential equations. The proposed algorithm for the Stokes system shows very fast convergence and is robust with respect to mesh sizes and reaction coefficients. Of course, the convergence is not satisfactory in the multi-domain case with cross points. But the number of needed iteration steps can be dramatically decreased by using an appropriate coarse space. A suitable choice of a coarse space for our new approach is subject of further research.

References

[1] V. Dolean, F. Nataf, and G. Rapin. Deriving a new domain decomposition method for the Stokes equations using the Smith factorization. Submitted, 2006.

[2] P. Le Tallec and A. Patra. Non-overlapping domain decomposition methods for adaptive *hp* approximations of the Stokes problem with discontinuous pressure fields. *Comput. Methods Appl. Mech. Engrg.*, 145:361–379, 1997.

[3] J. Li. A Dual-Primal FETI method for incompressible Stokes equations. *Numer. Math.*, 102:257–275, 2005.

[4] J. Li and O.B. Widlund. BDDC algorithms for incompressible Stokes equations. *SIAM J. Numer. Anal.*, 44:2432–2455, 2006.

[5] F. Nataf and G. Rapin. Construction of a new domain decomposition method for the Stokes equations. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lecture Notes Comput. Sci. Engrg.*, pages 247–254. Springer, Berlin, 2007.

[6] L.F. Pavarino and O.B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55:302–335, 2002.

[7] J.T. Wloka, B. Rowley, and B. Lawruk. *Boundary Value Problems for Elliptic Systems.* Cambridge University Press, Cambridge, 1995.

[8] X. Zhang. Domain decomposition algorithms for the biharmonic Dirichlet problem. In *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 119–126. SIAM, 1992.

p-Multigrid for Fekete Spectral Element Method

Victorita Dolean, Richard Pasquetti, and Francesca Rapetti

Lab. J.A. Dieudonné, UMR CNRS 6621, UNSA, Parc Valrose, 06108 Nice, France.
{victorita.dolean,richard.pasquetti,francesca.rapetti}@unice.fr

Summary. Spectral element approximations based on triangular elements and on the so-called Fekete points of the triangle have been recently developed. p-multigrid methods offer an interesting way to resolve efficiently the resulting ill-conditioned algebraic systems. For elliptic problems, it is shown that a well chosen restriction operator and a good set up of the coarse grid matrices may lead to valuable results, even with a standard Gauss-Seidel smoother.

1 Introduction

As well known, high-order approximations are highly accurate as soon as the solution is smooth and, usually, require less grid-points than low-order methods. Unfortunately, the resulting algebraic system is severely ill-conditioned. Thus, for a two-dimensional (2D) second order Partial Differential Equation (PDE), a high order Finite Element Method (FEM) usually yields a condition number proportional to N^4, where $N \equiv p$ is the (total) degree of the polynomial approximation on each triangular element. Efficient solvers are then required.

Different approaches have been investigated in our previous works. Especially, for Fekete triangular spectral elements we have focused on Overlapping Schwarz methods [7] and on Schur complement methods [9]. In both cases, the idea was to consider each element as a different subdomain and then to apply classical domain decomposition preconditioners. Similarly, here we investigate a p-multigrid method so that the roughest approximation may be the one obtained with the standard $\mathbb{P}_1$ FEM. For the usual SEM (Spectral Element Method), a multigrid spectral element approach was first proposed in [10] and more recently investigated in [3]. For standard spectral methods one can cite [13, 4] and, among others, [6] for hp-FEM.

The outline of the paper is the following. To be self contained, in Section 2 the Fekete-Gauss TSEM (Triangles based SEM) is briefly described. In Section 3 we propose different restriction algorithms and strategies for setting up the coarse-grid algebraic systems, test these different approaches and then optimize the smoother for one triangular spectral element. In Section 4, the best approach is implemented in a TSEM solver, applied to an elliptic model problem and a convergence study is carried out. We conclude and offer some perspectives in Section 5.

2 The Fekete-Gauss TSEM

The (quadrilateral-based) SEM makes use of the Gauss-Lobatto-Legendre (GLL) points, for both the approximation and the quadrature points. GLL points have indeed nice approximation *and* integration properties. Unfortunately, such a single set of points does not exist for the triangle. Thus, in its initial version the Fekete points based TSEM [11] may fail to show the "spectral accuracy" property [8].

The Fekete-Gauss TSEM makes use of two sets of points:

- The Fekete points, $\{x_i\}_{i=1}^n$, as approximation points:

$$u(x) \approx \sum_{i=0}^{n} u(x_i)\, \varphi_i(x), \quad x \in T$$

where the φ_i are the Lagrange polynomials, given by $\varphi_i(x_j) = \delta_{ij}$.

- Gauss points, $\{y_i\}_{i=1}^m$, as quadrature points:

$$\int_T uv\, dT \approx \sum_{i=0}^{m} \rho_i u(y_i) v(y_i)$$

where the ρ_i are the Gauss quadrature weights.

Let $T = \{(r,s) :\ -1 \leq r, s,\ r+s \leq 0\}$ and $\mathcal{P}_N(T)$ be the set of polynomials on T of total degree $\leq N$. Let $n = (N+1)(N+2)/2$ and $\{\psi_j\}_{j=1}^n$ be any basis of $\mathcal{P}_N(T)$. The Fekete points $\{x_i\}_{i=1}^n$ are those which maximize over T the determinant of the Vandermonde matrix V, given by $V_{ij} = \psi_j(x_i)$, $1 \leq i, j \leq n$.

In Fig. 1 (top) we compare the GLL points of the quadrilateral and the Fekete points of the triangle [12], for $N = 12$ (maximum degree in each variable for the quadrilateral and total degree for the triangle). In Fig. 1 (bottom) we give the Gauss points of the triangle for $M = 19$ (maximum polynomial degree for which the quadrature is exact) and those obtained from the Gauss points, with a mapping of the quadrilateral onto the triangle. The latter set of points may be of interest for values of M for which symmetrically distributed Gauss points are unknown. As advocated in [5], GLL points mapped onto the triangle may be used for both approximation and quadrature points, but at the price of an useless accumulation of points in one vertex.

The Fekete points of the triangle show some nice properties [12, 1]: (i) Fekete points are GLL points for the cube; (ii) Fekete points of the triangle are GLL points on the sides; (iii) The Lagrange polynomials based on Fekete points are maximum at these points.

3 Multigrid Strategy for the Triangle

We assume to have two grids, a coarse grid (grid 1) and a fine grid (grid 2) and denote the polynomial approximation degree by N_j, the set of Fekete points by $\{x_i^j\}$ and the Lagrange polynomials based on these points by $\{\varphi_i^j\}$, for grid j, $1 \leq j \leq 2$.

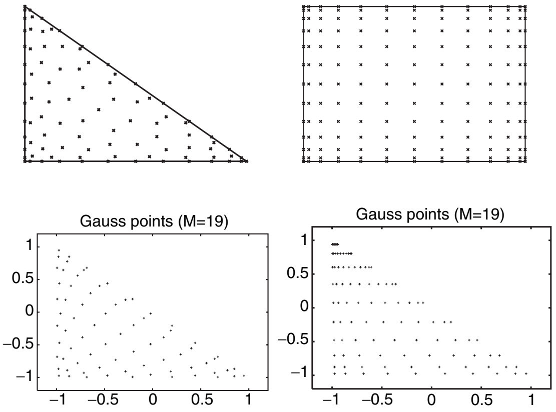


Fig. 1. Top: Triangle-Fekete and quadrilateral-GLL points ($N = 12$), Bottom: Triangle-Gauss and quadrilateral-Gauss mapped points ($M = 19$)

3.1 Prolongation / Restriction Operators and Coarse Grid System

Defining a prolongation operator is natural in the frame of spectral methods. Since the numerical approximation is everywhere defined, one has simply to express the coarse grid approximation at the Fekete points of the fine grid, to obtain:

$$u_2(x_i^2) = u_1(x_i^2) = \sum_j u_1(x_j^1)\varphi_j^1(x_i^2)$$

where $u_j \equiv u_{N_j}$ denotes the numerical approximation on grid j. In matrix form, with obvious notations:

$$\mathbf{u}_2 = P\,\mathbf{u}_1\,, \quad [P]_{ij} = \varphi_j^1(x_i^2)\,.$$

Defining a restriction operator is less straightforward. We have investigated the following approaches:

- Interpolation: similarly to what is done for the prolongation operator, one can use the fine grid approximation to set up the restriction operator:

$$u_1(x_i^1) = \sum_j u_2(x_j^2)\varphi_j^2(x_i^1)\,, \quad \mathbf{u}_1 = R\mathbf{u}_2\,, \quad [R]_{ij} = \varphi_j^2(x_i^1)\,.$$

Such an approach is essentially justified for collocation methods, i.e., when the Right Hand Side (RHS) is a function and not an integral simply associated to a particular point through the corresponding Lagrange polynomial.

- Transposition (variational methods): if one takes into account the particular structure of the RHS, then

$$(f, \varphi_i^1) = (f, \sum_j \varphi_i^1(x_j^2)\varphi_j^2) = \sum_j \varphi_i^1(x_j^2)(f, \varphi_j^2) \quad \text{so that} \quad R = P^t$$

- Projection: let $\{\psi_i\}_{i=1}^{\infty}$ be an orthogonal hierarchical basis, e.g., the Koornwinder-Dubiner basis [2], then :

$$u_2(x_i^2) = \sum_{k \le n_2} \hat{u}_k \psi_k(x_i^2) \quad \text{and} \quad u_1(x_i^1) = \sum_{k \le n_1} \hat{u}_k \psi_k(x_i^1)$$

so that: $R = V_1[Id, 0]V_2{}^{-1}$ (Id, Identity matrix). Again this approach is better adapted to collocation methods.

It remains to set up the coarse grid algebraic system. On the coarse grid one has to solve $A_1\mathbf{e}_1 = \mathbf{r}_1$, with $\mathbf{r}_1 = R\,\mathbf{r}_2$ ($\mathbf{r}_2$, residual at the fine grid level; $\mathbf{e}_1$, error at the coarse grid level). One has at least the two following possibilities:

- Matrix A_1 may be set up directly, i.e., like A_2. This approach is the one used in [10].

- Matrix A_1 may be set up from: $A_1 = RA_2P$, i.e., by "aggregation" of A_2. In this case one can easily check that if $R = P^t$, then $\mathbf{e}_1$ such that $A_1\mathbf{e}_1 = R\mathbf{r}_2$ solves the constrained optimization problem: minimize

$$\begin{aligned} &\phi(\mathbf{u}^*) = 0.5(A_2\mathbf{u}^*, \mathbf{u}^*) - (\mathbf{b}, \mathbf{u}^*) \quad \text{constrained by} \\ &\quad \mathbf{u}^* = \mathbf{u}_2 + P\mathbf{e}_1 \,. \end{aligned}$$

Numerical tests have been carried out for $-\Delta u + u = f$ in T, with the exact solution: $u_{exact} = \sin(2x+y)\,\sin(x+1)\,\sin(1-y)$ and the corresponding source term and Dirichlet boundary conditions.

Table 1. Number of iterations at the fine grid level / number of V-cycles. Comparison with Gauss-Seidel (GS).

N-Grids	I-D	T-D	P-D	T-A	GS
(6,12)	48/6	88/11	48/6	40/5	78
(3,6,12)	48/6	92/12	48/6	40/5	78
(6,12,18)	92/12	356/45	84/11	72/9	203
(3,6,12,18)	92/12	364/46	84/11	72/9	203

Depending on (i) the restriction strategy: Interpolation, Transposition or Projection and (ii) the setting up of the coarse matrix: Direct or Aggregation, four cases are considered: I-D, T-D, P-D and T-A. In these numerical tests, the number of grids is not restricted to 2, we use a V-cycle and at the smoothing grid levels 4 Gauss-Seidel iterations.

The number of iterations at the fine grid and the number of V-cycles required to get a residual less than 10^{-6} are given in Table 1. The multigrid results are compared to those obtained with the Gauss-Seidel (GS) method. Clearly, the transposition-aggregation (T-A) strategy gives the best results. Moreover, one observes that the number of iterations at the fine grid level is nearly independent of the number of grids involved in the V-cycle.

3.2 Analysis of the Smoother

On the basis of the following Successive Over Relaxation (SOR) decomposition of the matrix A_2, associated to the fine grid

$$A_2 = \frac{1}{\omega}(D + \omega L) - \frac{1}{\omega}[(1-\omega)D - \omega U] \equiv N - M$$

with D, L, U: the Diagonal, strictly Lower and Upper triangular parts of A_2, we want to optimize the relaxation coefficient ω and the number of iterations m of each SOR smoothing. Note that the GS smoothing is recovered for $\omega = 1$ and that, to obtain a stable algorithm, $0 < \omega < 2$.

We follow here an approach similar to the one proposed in [10]. Let n be the iteration index, defined as the sum of the number of iterations on grid 2 and the number of coarse grid corrections, $\mathbf{e}^n$ the error and $\mathbf{r}^n$ the residual.

- Pre-smoothing: after m iterations:

$$\mathbf{e}^{n+m} = (N^{-1}M)^m \mathbf{e}^n \qquad \mathbf{r}^{n+m} = A_2\mathbf{e}^{n+m} = A_2(N^{-1}M)^m A_2^{-1}\mathbf{r}^n$$

- After the coarse grid correction:

$$\begin{aligned}\mathbf{e}^{n+m+1} &= \mathbf{e}^{n+m} - PA_1^{-1}R\,\mathbf{r}^{n+m}\\ \mathbf{r}^{n+m+1} &= (Id - A_2PA_1^{-1}R)\,\mathbf{r}^{n+m}\end{aligned}$$

- Post-smoothing: after m iterations:

$$\begin{aligned}\mathbf{r}^{n+2m+1} &= A_2(N^{-1}M)^m A_2^{-1}\,\mathbf{r}^{n+m+1} = T\,\mathbf{r}^n\\ T &= A_2(N^{-1}M)^m A_2^{-1}(Id - A_2PA_1^{-1}R)A_2(N^{-1}M)^m A_2^{-1}\,.\end{aligned}$$

Then:

$$\|\mathbf{r}^{2m+1+n}\| = \|T\mathbf{r}^n\| \le \|T\|\|\mathbf{r}^n\| \equiv \rho^{2m+1}\|\mathbf{r}^n\|, \quad \rho(\omega,m) = \|T\|^{1/(2m+1)}\,.$$

The parameter $\rho(\omega, m)$ that we have introduced may constitute a good indicator of the smoothing efficiency and so it allows an optimization of the relaxation parameter and of the number of iterations.

From the conclusion of Section 3.1, the restriction is achieved by transposition and the coarse grid matrix A_1 is set up by aggregation of the fine grid matrix A_2. Figure 2 shows isolines of ρ in the (m, ω) plane. Clearly, choosing $\omega = 1$ appears satisfactory and increasing the number of iterations beyond 4 appears useless, since this does not allow to really decrease the value of ρ.

4 Application to a Model Problem

The present multigrid method has been implemented in a TSEM solver using the T-A strategy and an arbitrary number (≥ 2) of grids. The matrix A_i, associated to the level i, is computed from :

$$A_i = {\sum_{k=1}^{K}}' R_i\, A_{k,i+1}\, P_i \qquad R_i = P_i^t$$

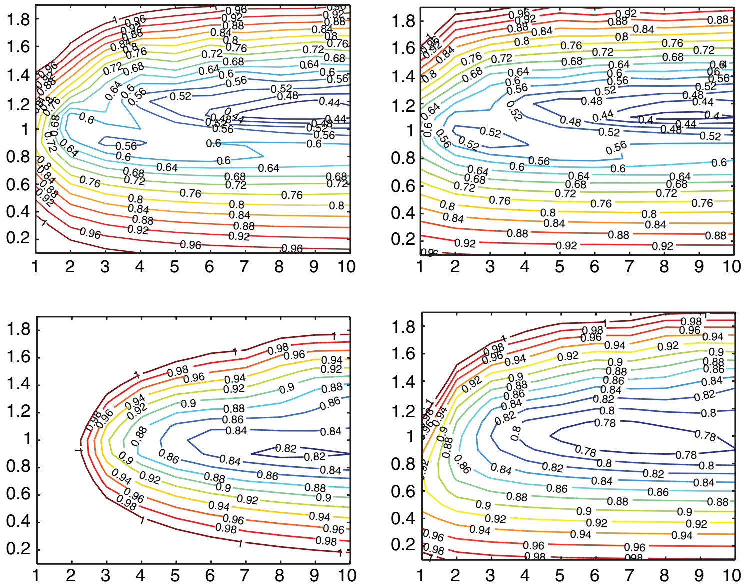


Fig. 2. $\rho(\omega, m)$ for the $\|\cdot\|_\infty$ (left) and $\|\cdot\|_2$ norms (right); $N_1 = 3$, $N_2 = 6$ (top) and $N_1 = 6$, $N_2 = 12$ (bottom)

where R_i, P_i are the restriction and prolongation operators between grids i (coarse) and $i{+}1$ (fine), $\sum'$ is the stiffness sum and $A_{k,i+1}$ is the element matrix associated to the element $k \leq K$ at the grid level $(i{+}1)$. Note that the restriction and prolongation operators are set up on the reference element, where the polynomial approximation holds, and so they do not depend on the element index k.

Convergence tests have been made for the elliptic PDE, $-\Delta u + u = f$ in $\Omega = (-1, 1)^2$, with the exact solution $u_{exact} = \sin(\pi x)\sin(\pi y)$ and corresponding Dirichlet boundary conditions and source term.

The computational domain $\Omega = (-1, 1)^2$ has been discretized using $K = 10 \times 10 \times 2 = 200$ triangular elements and $N = 12$. One has then 14641 degrees of freedom and the condition number of the system matrix equals 55345.

In Fig. 3 are shown convergence results for different configurations involving from 2 to 4 grids and comparisons are provided with the Conjugate Gradient and the Gauss-Seidel algorithms. Clearly the multigrid technique appears very efficient. Moreover, just like for one triangular element, the results obtained with $N = 12$ for the fine grid show that the convergence rate is nearly independent of the number of grids, so that the exact solve is only required on a very coarse grid. The convergence result given for the finer grid $N = 18$ shows that the convergence rate only slightly deteriorates, consistently with the results obtained for one element.

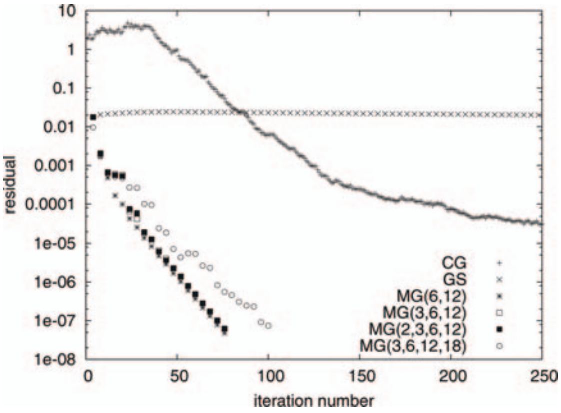


Fig. 3. MG convergence for $N = (6, 12)$, $N = (3, 6, 12)$, $N = (2, 3, 6, 12)$ (10 cycles) and $N = (3, 6, 12, 18)$ (13 cycles). Comparisons with CG and GS for $N = 12$.

5 Conclusion and Perspectives

A multigrid approach has been investigated for the TSEM approximation of elliptic problems. In particular,

- For one triangular Fekete-Gauss spectral element, different formulations of the restriction operator and of the coarse grid matrix have been compared. The best results are obtained when the restriction operator is defined by transposition and the coarse matrices by aggregation.

- An analysis of the influence of the control parameters (ω and m) of the SOR-smoother has been carried out. Good properties are obtained for $\omega = 1$ and $m = 4$.

- This multigrid approach has been implemented in a TSEM solver and tests have been carried out for a model problem.

Many points have not yet been investigated, e.g., (i) influence of a deformation of the mesh, (ii) comparisons with standard (quadrilateral based) SEM and (iii) improvement of the smoother.

Beyond that, it would be interesting to provide extensions to $3D$ geometries and also to more realistic problems, like fluid flows in complex geometries.

References

[1] L. Bos, M.A. Taylor, and B.A. Wingate. Tensor product Gauss-Lobatto points are Fekete points for the cube. *Math. Comp.*, 70:1543–1547, 2001.

[2] M. Dubiner. Spectral methods on triangles and other domains. *J. Sci. Comput.*, 6:345–390, 1991.

[3] P.F. Fischer and J.W. Lottes. Hybrid Schwarz-multigrid methods for the spectral element method: Extensions to Navier-Stokes. *J. Sci. Comput.*, 6:345–390, 2005.

[4] H. Heinrichs. Spectral multigrid methods for the reformulated Stokes equations. *J. Comput. Phys.*, 107:213–224, 1993.

[5] G.E. Karniadakis and S.J. Sherwin. *Spectral hp Element Methods for CFD.* Oxford University Press, London, 1999.

[6] C.R. Nastase and D.J. Mavriplis. High-order discontinuous Galerkin methods using a spectral multigrid approach. In *AIAA 2005-1268 Paper*, 2005.

[7] R. Pasquetti, L.F. Pavarino, F. Rapetti, and E. Zampieri. Overlapping Schwarz preconditioners for Fekete spectral elements. In *Domain Decomposition Methods in Science and Engineering XVI*, volume 55 of *Lect. Notes Comput. Sci. Eng.*, pages 715–722. Springer, Berlin, 2007.

[8] R. Pasquetti and F. Rapetti. Spectral element methods on unstructured meshes: comparisons and recent advances. *J. Sci. Comp.*, 27:377–387, 2006.

[9] R. Pasquetti, F. Rapetti, L.F. Pavarino, and E. Zampieri. Neumann-Neumann-Schur complement methods for Fekete spectral elements. *J. Engrg. Math.*, 56(3):323–335, 2006.

[10] E.M. Rønquist and A.T. Patera. Spectral element multigrid. 1- formulation and numerical results. *J. Sci. Comput.*, 2:389–405, 1987.

[11] M.A. Taylor and B.A. Wingate. A generalized diagonal mass matrix spectral element method for non-quadrilateral elements. *Appl. Num. Math.*, 33:259–265, 2000.

[12] M.A. Taylor, B.A. Wingate, and R.E. Vincent. An algorithm for computing Fekete points in the triangle. *SIAM J. Numer. Anal.*, 38:1707–1720, 2000.

[13] T.A. Zang, Y.S. Wong, and M.Y. Hussaini. Spectral multigrid methods for elliptic equations. *J. Comput. Phys.*, 48:485–501, 1982.

p-FEM Quadrature Error Analysis on Tetrahedra

Tino Eibner[1] and Jens M. Melenk[2]

[1] Technische Universität Chemnitz, teibner@mathematik.tu-chemnitz.de
[2] Technische Universität Wien, melenk@tuwien.ac.at

1 Introduction

In the p-FEM and the closely related spectral method, the solution of an elliptic boundary value problems is approximated by piecewise (mapped) polynomials of degree p on a fixed mesh $\mathcal{T}$. In practice, the entries of the p-FEM stiffness matrix cannot be evaluated exactly due to variable coefficients and/or non-affine element maps and one has to resort to numerical quadrature to obtain a fully discrete method. Computationally, choosing shape functions that are related to the quadrature formula employed can significantly improve the computational complexity. For example, for tensor product elements (i.e., quadrilaterals, hexahedra) choosing tensor product Gauss-Lobatto quadrature with $q + 1 = p + 1$ points in each spatial direction and taking as shape functions the Lagrange interpolation polynomials (of degree p) in the Gauss-Lobatto points effectively leads to a spectral method. The quadrature error analysis for the p-FEM/spectral method is available even for this case of minimal quadrature (see, e.g., [5, 6] and reference there). Key to the error analysis is a one-dimensional discrete stability result for the Gauss-Lobatto quadrature due to [2] (corresponding to $\alpha = 0$ in Lemma 2 below) that can readily be extended to quadrilaterals/hexahedra by tensor product arguments.

In the present paper, we show an analog of the error analysis of the above minimal quadrature for the p-FEM on tetrahedral meshes (the easier case of triangles can be treated completely analogously). Quadrature on a tetrahedron can be done by a mapping to a hexahedron via the Duffy transformation D of (3). We show in Theorem 1 that for tensor product Gauss-Lobatto-Jacobi quadrature formulas with $q + 1 = p + 1$ points in each direction, one again has discrete stability for the fully discrete p-FEM. A complete quadrature error analysis (Theorem 2, Corollary 1) then follows from Strang's lemma and shows that the convergence rates of the Galerkin p-FEM (where all integrals are evaluated exactly) is retained by the fully discrete p-FEM. The present error analysis complements the work [3] for the p-FEM on triangles/tetrahedra where it is shown that by adapting the shape functions to the quadrature formula, the stiffness matrix can be set up in optimal complexity. However, we mention that the approximation spaces employed in [3] are no longer the classical spaces $S^{p,1}(\mathcal{T})$ of piecewise polynomials but the spaces $S^{p,1}(\mathcal{T})$ augmented by bubble shape functions for each element, which makes the static condensation more expensive.

To fix ideas, we consider

$$-\nabla \cdot (A(x)\nabla u) = f \qquad \text{on } \Omega \subset \mathbb{R}^3, \qquad u|_{\partial\Omega} = 0, \tag{1}$$

where $A \in C(\overline{\Omega}, \mathbb{R}^{3\times 3})$ is pointwise symmetric positive definite. We require A and f to be analytic on $\overline{\Omega}$ and the standard ellipticity condition

$$0 < \lambda_{min} \leq A(x) \leq \lambda_{max}, \qquad \forall x \in \Omega.$$

2 Quadrature Error Analysis

Notation

The reference tetrahedron $\widehat{K}$ and the reference cube $\mathcal{Q}$ are defined as

$$\widehat{K} = \{(x,y,z) \mid -1 < x,y,z \ \wedge \ x+y+z < -1\}, \qquad\qquad \mathcal{Q} := (-1,1)^3. \tag{2}$$

The Duffy transformation $D : \mathcal{Q} \to \widehat{K}$ is given by

$$D(\eta_1,\eta_2,\eta_3) := \left(\frac{(1+\eta_1)(1-\eta_2)(1-\eta_3)}{4} - 1, \frac{(1+\eta_2)(1-\eta_3)}{2} - 1, \eta_3 \right). \tag{3}$$

Lemma 1. *The Duffy transformation is a bijection between the (open) cube $\mathcal{Q}$ and the (open) tetrahedron $\widehat{K}$. Additionally,*

$$D'(\eta_1,\eta_2,\eta_3) := \left[\frac{\partial \xi_i}{\partial \eta_j}\right]^3_{i,j=1} = \begin{bmatrix} \frac{1}{4}(1-\eta_2)(1-\eta_3) & 0 & 0 \\ -\frac{1}{4}(1+\eta_1)(1-\eta_3) & \frac{1}{2}(1-\eta_3) & 0 \\ -\frac{1}{4}(1+\eta_1)(1-\eta_2) & -\frac{1}{2}(1+\eta_2) & 1 \end{bmatrix}^\top,$$

$$\left(D'(\eta_1,\eta_2,\eta_3)\right)^{-1} = \frac{1}{(1-\eta_2)(1-\eta_3)} \begin{bmatrix} 4 & 2(1+\eta_1) & 2(1+\eta_1) \\ 0 & 2(1-\eta_2) & 1-\eta_2^2 \\ 0 & 0 & (1-\eta_2)(1-\eta_3) \end{bmatrix},$$

$$\det D' = \left(\frac{1-\eta_2}{2}\right)\left(\frac{1-\eta_3}{2}\right)^2. \tag{4}$$

Proof. See, for example, [4].

We employ standard notation by writing $\mathcal{P}_p(\widehat{K})$ for the space of polynomials of degree p on $\widehat{K}$, and by denoting $\mathcal{Q}_p(\mathcal{Q})$ the tensor-product space of polynomials of degree p in each variable, [7]; additionally we set

$$\widetilde{\mathcal{Q}}_p := \{u \in \mathcal{Q}_p(\mathcal{Q}) \,|\, \partial_1 u = \partial_2 u = \partial_3 u = 0 \text{ on } \eta_3 = 1 \text{ and } \partial_1 u = 0 \text{ on } \eta_2 = 1\}.$$

Remark 1. The Duffy transformation D maps the face $\eta_3 = 1$ to the point $(-1,-1,1)$ and the face $\eta_2 = 1$ to a line. An important property of $\widetilde{\mathcal{Q}}_p$ is that $u \in \mathcal{P}_p(\widehat{K})$ implies $u \circ D \in \widetilde{\mathcal{Q}}_p$.

2.1 Gauss-Lobatto-Jacobi Quadrature

Gauss-Lobatto-Jacobi Quadrature in 1D

For $\alpha > -1$, $n \in \mathbb{N}$, the Gauss-Lobatto-Jacobi quadrature formula is given by

$$\mathrm{GLJ}_{(\alpha,n)}(f) := \sum_{i=0}^{n} \omega_i^{(\alpha,n)} f(x_i^{(\alpha,n)}) \approx \int_{-1}^{1} (1-x)^\alpha f(x)\mathrm{d}x; \tag{5}$$

(see, e.g., [4, App. B]): the quadrature nodes $x_i^{(\alpha,n)}$, $i = 0, \ldots, n$, are the zeros of the polynomial $x \mapsto (1-x^2)P_n^{(\alpha+1,1)}(x)$, where $P_n^{(\alpha,\beta)}$ denotes the Jacobi polynomial of degree n with respect to the weight function $(1-x)^\alpha(1+x)^\beta$. The quadrature weights $\omega_i^{(\alpha,n)}$, $i = 0, \ldots, n$, are *positive* and explicit formulas can be found, for example, in [4, App. B]. We have:

Lemma 2. *Let $\mathcal{P}_n$ be the space of polynomials of degree n. Then for $\alpha > -1$:*

1. *For all $f \in \mathcal{P}_{2n-1}$ there holds $\mathrm{GLJ}_{(\alpha,n)}(f) = \int_{-1}^{1} f(x)(1-x)^\alpha dx$.*
2. *For all $f \in \mathcal{P}_n$ there holds*

$$\int_{-1}^{1} f^2(x)(1-x)^\alpha \mathrm{d}x \le \mathrm{GLJ}_{\alpha,n}(f^2) \le \left(2 + \frac{\alpha+1}{n}\right) \int_{-1}^{1} f^2(x)(1-x)^\alpha \mathrm{d}x.$$

Proof. The first assertion is well-known. The second assertion follows by the same arguments as in the case $\alpha = 0$, which can be found, for example, in [2] or [1, Corollary 1.13].

Gauss-Lobatto-Jacobi Quadrature on $\widehat{K}$

Using the change of variables formula $\int_{\widehat{K}} g\mathrm{d}x = \int_{\mathcal{Q}} (g \circ D)|\det D'|\mathrm{d}x$, we can introduce a quadrature formulas such that

$$\mathrm{GLJ}_{\mathcal{Q},n}(f) \approx \int_{\mathcal{Q}} f(\eta)|\det D'(\eta)|\mathrm{d}\eta, \qquad \mathrm{GLJ}_{\hat{K},n}(g) \approx \int_{\hat{K}} g(\xi)\,\mathrm{d}\xi$$

by setting

$$\mathrm{GLJ}_{\mathcal{Q},n}(f) := 1/8 \sum_{i_1,i_2,i_3=0}^{n} \omega_{i_1}^{(0,n)} \omega_{i_2}^{(1,n)} \omega_{i_3}^{(2,n)} f\left(x_{i_1}^{(0,n)}, x_{i_2}^{(1,n)}, x_{i_3}^{(2,n)}\right), \tag{6}$$

$$\mathrm{GLJ}_{\widehat{K},n}(g) := \mathrm{GLJ}_{\mathcal{Q},n}(g \circ D). \tag{7}$$

Using standard tensor product arguments one can deduce from the properties of the quadrature rules $\mathrm{GLJ}_{\alpha,n}$ and the formula (4) the following result:

Lemma 3. *Let $1 \le p \le q$ and let $\widehat{u} \in \mathcal{Q}_p(\mathcal{Q})$, $\widehat{v} \in \mathcal{Q}_{2q-1}(\mathcal{Q})$. Set $u := \widehat{u} \circ D^{-1}$, $v := \widehat{v} \circ D^{-1}$. Then the equalities $\mathrm{GLJ}_{\mathcal{Q},q}(\widehat{v}) = \int_{\mathcal{Q}} \widehat{v}|\det D'|\mathrm{d}\Omega$ and $\mathrm{GLJ}_{\widehat{K},q}(v) = \int_{\hat{K}} v\mathrm{d}\Omega$ are true and, for $\underline{C} := (2+1/p)(2+2/p)(2+3/p) \le 60$,*

$$\int_{\mathcal{Q}} |\widehat{u}|^2|\det D'|d\Omega \le \mathrm{GLJ}_{\mathcal{Q},q}(\widehat{u}^2) \le \underline{C} \int_{\mathcal{Q}} |\widehat{u}|^2|\det D'|d\Omega,$$
$$\|u\|^2_{L^2(\widehat{K})} \le \mathrm{GLJ}_{\hat{K},q}(u^2) \le \underline{C}\|u\|^2_{L^2(\hat{K})}.$$

2.2 Discrete Stability

The following discrete stability result is the heart of the quadrature error analysis; its proof is deferred to Section 3.

Theorem 1. *Let $A \in C(\overline{\widehat{K}}, \mathbb{R}^{3\times 3})$ be pointwise symmetric positive definite, $c \in C(\overline{\widehat{K}})$. Assume the existence of λ_{min}, λ_{max}, $c_{min} > 0$ with*

$$\lambda_{min} \leq A(x) \leq \lambda_{max}, \qquad c_{min} \leq c(x) \forall x \in \widehat{K}.$$

Then for $q \geq p$ there holds for all $u \in \{u \,|\, u \circ D \in \widetilde{\mathcal{Q}}_p\}$

$$\mathrm{GLJ}_{\widehat{K},q}(\nabla u \cdot A \nabla u) \geq \frac{\lambda_{min}}{10404} \|\nabla u\|^2_{L^2(\widehat{K})} \geq \frac{\lambda_{min}}{10404 \lambda_{max}} \int_{\widehat{K},q} \nabla u \cdot A \nabla u \mathrm{d}\Omega, \tag{8}$$

$$\mathrm{GLJ}_{\widehat{K},q}(cu^2) \geq c_{min} \|u\|^2_{L^2(\widehat{K})}. \tag{9}$$

2.3 Convergence Analysis of Fully Discrete *p*-FEM

For the model problem (1) and given mesh $\mathcal{T}$ consisting of (curvilinear) tetrahedra with element maps $F_K : \widehat{K} \to K$, we define the discrete bilinear form a^q and right-hand side F^q by

$$a^q(u,v) := \sum_{K \in \mathcal{T}} \mathrm{GLJ}_{\hat{K},q} \left(((\nabla u \cdot A \nabla v)|_K \circ F_K) |\det F_K'| \right),$$

$$F^q(u) := \sum_{K \in \mathcal{T}} \mathrm{GLJ}_{\hat{K},q} \left(((fu)|_K \circ F_K) |\det F_K'| \right).$$

We let $S_0^{p,1}(\mathcal{T}) := \{u \in H_0^1(\Omega) \,|\, u|_K \circ F_K \in \mathcal{P}_p(\widehat{K}) \quad \forall K \in \mathcal{T}\}$ and consider finite dimensional spaces V_N satisfying

$$S_0^{p,1}(\mathcal{T}) \subset V_N \subset \widetilde{S}_0^{p,1}(\mathcal{T}) := \{u \in H_0^1(\Omega) \,|\, u|_K \circ F_K \circ D \in \widetilde{Q}_p \quad \forall K \in \mathcal{T}\}. \tag{10}$$

Remark 2. By Remark 1, choosing $V_N = S_0^{p,1}(\mathcal{T})$ is admissible. Taking V_N larger than $S_0^{p,1}(\mathcal{T})$ permits adapting the shape functions to the quadrature points and permits efficient ways to generate the stiffness matrix, [3].

The fully discrete problem is then:

$$\text{Find } u_N \in V_N \text{ s.t.} \qquad a^q(u_N, v) = F^q(v) \qquad \forall v \in V_N. \tag{11}$$

The discrete stability result Theorem 1 for a single element is readily extended to meshes with several elements and existence and uniqueness of solutions to (11) follows. An application of Strang's Lemma then gives error estimates:

Theorem 2. *Let the mesh $\mathcal{T}$ be fixed and the element maps F_K be analytic on $\overline{\widehat{K}}$. Assume (10) and $q \geq p$. Let u solve (1) and u_N solve (11). Then there exist C, $b > 0$ depending only on Ω, the analytic data A, f of (1), and the analytic element maps F_K such that*

$$\|u - u_N\|_{H^1(\Omega)} \leq C \left(\inf_{v \in S_0^r(\mathcal{T})} \|u - v\|_{H^1(\Omega)} + Cr^3 e^{-b(2q+p-r)} \right)$$

for arbitrary $1 \leq r \leq \min\{p, 2(q-1) - p\}$.

Proof. The proof follows along the lines of [6, Secs. 4.2, 4.3]: Theorem 1 enables us to use a Strang lemma, and the resulting consistency terms can be made exponentially small by the analyticity of A, f, and the F_K.

Remark 3. It is worth stressing that analyticity of $\partial\Omega$ is not required in Theorem 2—only analyticity of the element maps is necessary. Hence, also piecewise analytic geometries are covered by Theorem 2. The requirement that A, f be analytic can be relaxed to the condition that $A|_K$, $f|_K$ be analytic on $\overline{K}$ for all elements.

We note that choosing $r = \lfloor p/2 \rfloor$ in Theorem 2 implies that the *rate of convergence* of the fully discrete p-FEM is typically the same as the Galerkin p-FEM in which all quadratures are performed exactly:

Corollary 1. *Assume the hypotheses of Theorem 2. Then:*

1. *If* $\inf_{v \in S_0^{p,1}(\mathcal{T})} \|u - v\|_{H^1(\Omega)} = O(p^{-\alpha})$*, then* $\|u - u_N\|_{H^1(\Omega)} = O(p^{-\alpha})$.
2. *If* $\inf_{v \in S_0^{p,1}(\mathcal{T})} \|u - v\|_{H^1(\Omega)} = O(e^{-bp})$ *for some* $b > 0$*, then there exists* $b' > 0$ *such that* $\|u - u_N\|_{H^1(\Omega)} = O(e^{-b'p})$.

3 Proof of Theorem 1

The heart of the proof of Theorem 1 consists in the assertion that for the Duffy transformation D, the matrix $(D')^{-1}(D')^{-\top}$ is equivalent to its diagonal. To that end, we recall for square matrices A, $B \in \mathbb{R}^{n\times n}$ the standard notation $A \leq B$ which expresses $v^\top A v \leq v^\top B v$ for all $v \in \mathbb{R}^n$. We have:

Lemma 4. *Let* $E(\eta) := (D'^{-1}D'^{-\top})(\eta)$ *and denote by* $\operatorname{diag} E(\eta) \in \mathbb{R}^{3\times 3}$ *the diagonal of* $E(\eta)$*. Then*

$$\frac{1}{3468} \operatorname{diag} E(\eta) \leq E(\eta) \leq 3 \operatorname{diag} E(\eta) \qquad \forall \eta \in \mathcal{Q}. \tag{12}$$

Proof. One easily shows for any invertible matrix $G \in \mathbb{R}^{n\times n}$

$$B \leq A \quad \Longleftrightarrow \quad G^\top B G \leq G^\top A G. \tag{13}$$

In order to prove (12), we define the diagonal matrix

$$B(\eta) := \operatorname{diag}\left[(1-\eta_2)(1-\eta_3), (1-\eta_3), 1\right]$$

and in view of (13) we are led to showing

$$\frac{1}{3468}(B^\top (\operatorname{diag} E) B)(\eta) \leq (B^\top E B)(\eta) \leq 3 (B^\top (\operatorname{diag} E) B)(\eta) \quad \forall \eta \in \mathcal{Q}. \tag{14}$$

Explicitly computing

$$(B^\top E B)(\eta) = \begin{pmatrix} 8(1+\eta_1)^2 + 16 & (1+\eta_1)\{4 + 2(1+\eta_2)\} & 2(1+\eta_1) \\ \text{sym.} & 4 + (1+\eta_2)^2 & (1+\eta_2) \\ \text{sym.} & \text{sym.} & 1 \end{pmatrix}$$

and applying the three estimates

$$2(1+\eta_1)\{4+2(1+\eta_2)\}v_1v_2 \le 8(1+\eta_1)^2v_1^2 + [4+(1+\eta_2)^2]v_2^2,$$
$$4(1+\eta_1)v_1v_3 \le 4(1+\eta_1)^2v_1^2 + v_3^2, \qquad 2(1+\eta_2)v_2v_3 \le (1+\eta_2)^2v_2^2 + v_3^2$$

for all $\eta \in \mathcal{Q}, v_1, v_2, v_3 \in \mathbb{R}$, we conclude for any vector $v = (v_1, v_2, v_3)^\top \in \mathbb{R}^3$

$$v^\top (B^\top EB)(\eta)v \le v^\top \operatorname{diag}\left[20(1+\eta_1)^2 + 16, 8 + 3(1+\eta_2)^2, 3\right] v.$$

In view of $(B^\top(\operatorname{diag} E)B)(\eta) = \operatorname{diag}\left[8(1+\eta_1)^2 + 16, 4 + (1+\eta_2)^2, 1\right]$ we arrive at $(B^\top EB)(\eta) \le 3(B^\top(\operatorname{diag} E)B)(\eta)$. In order to prove the lower bound of (14) we observe that $(B^\top EB)(\eta)$ is symmetric positive definite for all $\eta \in \mathcal{Q}$; denoting by $0 < \lambda_1 \le \lambda_2 \le \lambda_3$ the three eigenvalues of $(B^\top EB)(\eta)$, we conclude from the Gershgorin circle theorem $0 < \lambda_1 \le \lambda_2 \le \lambda_3 \le 68$ for all $\eta \in \mathcal{Q}$. Moreover, a direct calculation shows $\det(B^\top EB)(\eta) = 64$. Thus, $\lambda_1 \ge \det(B^\top EB)/\lambda_2^2 \ge 4/289$ for all $\eta \in \mathcal{Q}$. Hence for all $\eta \in \mathcal{Q}$

$$\begin{aligned}(B^\top EB)(\eta) &\ge \frac{4}{289} I \ge \frac{4}{289} \operatorname{diag}\left[\frac{8(1+\eta_1)^2+16}{48}, \frac{4+(1+\eta_2)^2}{8}, 1\right] \\ &\ge \frac{1}{3468}(B^\top(\operatorname{diag} E)B)(\eta).\end{aligned}$$

P*roof of Theorem 1.* We will only show (8) as (9) follows easily from Lemma 3. Let u be such that $\widehat{u} := u \circ D \in \widetilde{\mathcal{Q}}_p$. In view of the positivity of the quadrature weights and Lemma 4 we get for $\widetilde{E} := \operatorname{diag}((D')^{-1}(D')^{-\top})$

$$\begin{aligned}&\mathrm{GLJ}_{\widehat{K},q}(\nabla u \cdot A \nabla u) \ge \lambda_{min}\, \mathrm{GLJ}_{\widehat{K},q}(|\nabla u|^2) \\ &= \lambda_{min}\, \mathrm{GLJ}_{\mathcal{Q},q}(\nabla \widehat{u} \cdot (D')^{-1}(D')^{-\top}\nabla\widehat{u}) \ge \frac{\lambda_{min}}{3468}\, \mathrm{GLJ}_{\mathcal{Q},q}(\nabla\widehat{u} \cdot \widetilde{E}\nabla\widehat{u}).\end{aligned}$$

A calculation reveals $\widetilde{E} = \left(E^{(1)}\right)^2 + \left(E^{(2)}\right)^2$ if we introduce

$$\begin{aligned}E^{(1)} &:= \operatorname{diag}\left\{\frac{\sqrt{8}(1+\eta_1)}{(1-\eta_2)(1-\eta_3)}, \frac{1+\eta_2}{1-\eta_3}, 1\right\}, \\ E^{(2)} &:= \operatorname{diag}\left\{\frac{4}{(1-\eta_2)(1-\eta_3)}, \frac{2}{1-\eta_3}, 0\right\}.\end{aligned}$$

The assumption $\widehat{u} \in \widetilde{\mathcal{Q}}_p$ implies that the components of $E^{(1)}\nabla\widehat{u}$ and $E^{(2)}\nabla\widehat{u}$ are in $\mathcal{Q}_p(\mathcal{Q})$; hence, from Lemma 3

$$\begin{aligned}\mathrm{GLJ}_{\mathcal{Q},q}(\nabla\widehat{u} \cdot \widetilde{E}\nabla\widehat{u}) &= \mathrm{GLJ}_{\mathcal{Q},q}(|E^{(1)}\nabla\widehat{u}|^2) + \mathrm{GLJ}_{\mathcal{Q},q}(|E^{(2)}\nabla\widehat{u}|^2) \\ &\ge \int_{\mathcal{Q}} |E^{(1)}\nabla\widehat{u}|^2 |\det D'| \mathrm{d}\Omega + \int_{\mathcal{Q}} |E^{(2)}\nabla\widehat{u}|^2 |\det D'| \mathrm{d}\Omega \\ &= \int_{\mathcal{Q}} (\nabla\widehat{u})^\top \widetilde{E} \nabla\widehat{u}\, |\det D'| \mathrm{d}\Omega \\ &\ge \frac{1}{3}\int_{\mathcal{Q}} (\nabla\widehat{u})^\top (D')^{-1}(D')^{-\top}\nabla\widehat{u}\, |\det D'|\mathrm{d}\Omega = \frac{1}{3}\int_{\widehat{K}} |\nabla u|^2 \mathrm{d}\Omega,\end{aligned}$$

where we also appealed to Lemma 4. Collecting our findings, we arrive at

$$\mathrm{GLJ}_{\widehat{K},q}(\nabla u \cdot A\nabla u) \ge \frac{\lambda_{min}}{3468}\frac{1}{3}\|\nabla u\|^2_{L^2(\widehat{K})} \ge \frac{\lambda_{min}}{10404\lambda_{max}}\int_{\widehat{K}} \nabla u \cdot A \nabla u \mathrm{d}\Omega.$$

4 Numerical Example

Corollary 1 states that the fully discrete p-FEM converges at the same rate as a Galerkin p-FEM where all integrals are evaluated exactly. We illustrate this behavior for the following example:

$$-\nabla \cdot (A \nabla u) = 1 \quad \text{on } \Omega := \widehat{K} \quad \text{and} \quad u = 0 \quad \text{on } \partial\Omega, \tag{15}$$

$$A(x_1, x_2, x_3) := \operatorname{diag}\left[\frac{1}{r^2+1}, \exp\left(r^2\right), \cos\left(\frac{1}{r^2+1}\right)\right], \tag{16}$$

where $r^2 = x_1^2 + x_2^2 + x_3^2$. We base the p-FEM on a single element on two different sets of shape functions: Φ^{KS} is the set of shape functions proposed by Karniadakis and Sherwin [4] and spans $\mathcal{P}_p(\widehat{K}) \cap H_0^1(\widehat{K})$; the set Φ^{Lag} is, roughly, speaking, the set of Lagrange interpolation points in the quadrature points (on $\mathcal{Q}$); it spans a space that contains $\mathcal{P}_p(\widehat{K}) \cap H_0^1(\widehat{K})$ and we refer to [3] for details. In both cases the stiffness matrix is set up using the minimal quadrature, i.e., $q = p$. Figure 1 shows the relative energy norm error $(\frac{E_{exact} - a^q(u_N, u_N)}{E_{exact}})^{1/2}$ for both cases, where $E_{exact} = \int_\Omega \nabla u \cdot A \nabla \mathrm{d}\Omega$. To illustrate that the optimal rate of convergence is not affected by the quadrature, we include in Fig. 1 a calculation (based on Φ^{KS}) that corresponds to (15) with $A = I$; in this case the linear system of equations can be set up without quadrature errors. We observe indeed that the rate of convergence is the same as in the case of quadrature.

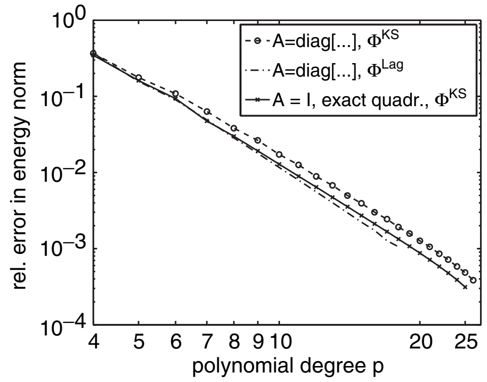


Fig. 1. Relative energy norm error

We close by pointing out that the shape functions in Φ^{Lag} are adapted to the quadrature rule. While the number of functions in Φ^{Lag} is (asymptotically for large p) 6 times that of Φ^{KS}, setting up the stiffness matrix is not slower than setting up the stiffness matrix based on Φ^{KS}. We refer to [3] for a detailed study.

References

[1] C. Bernardi and Y. Maday. *Approximations Spectrales de Problèmes aux Limites Elliptiques.* Springer Verlag, 1992.

[2] C. Canuto and A. Quarteroni. Approximation results for orthogonal polynomials in Sobolev spaces. *Math. Comput.*, 38(257):67–86, 1982.

[3] T. Eibner and J.M. Melenk. Fast algorithms for setting up the stiffness matrix in *hp*-fem: a comparison. In Elias A. Lipitakis, editor, *Computer Mathematics and its Applications-Advances and Developments (1994-2005)*, pages 575–596. LEA Publisher, 2006.

[4] G.E. Karniadakis and S.J. Sherwin. *Spectral/hp Element Methods for CFD.* Oxford University Press, 1999.

[5] Y. Maday and E.M. Rønquist. Optimal error analysis of spectral methods with emphasis on non-constant coefficients and deformed geometries. *Comput. Methods Appl. Mech. Engrg.*, 80:91–115, 1990.

[6] J.M. Melenk and C. Schwab. *hp* FEM for reaction-diffusion equations I: Robust exponentional convergence. *SIAM J. Numer. Anal.*, 35:1520–1557, 1998.

[7] C. Schwab. *p- and hp-Finite Element Methods.* Oxford University Press, 1998.

A Direct Solver for the Heat Equation with Domain Decomposition in Space and Time

Marc Garbey

University of Houston, Computer Science (http://www.cs.uh.edu/~garbey/)

Summary. In this paper we generalize the Aitken-like acceleration method of the additive Schwarz algorithm for elliptic problems to the additive Schwarz waveform relaxation for the heat equation. The domain decomposition is in space and time. The standard Schwarz waveform relaxation algorithm has a linear rate of convergence and low numerical efficiency. This algorithm is, however, friendly to cache use and scales with the memory in parallel environments. We show that our new acceleration procedure of the waveform acceleration algorithm results in a fast direct solver.

1 Introduction

Currently, standard processors are becoming multi-cores and there is a strong incentive to make use of all these parallel resources while avoiding conflict in memory access. We also have an overwhelming abundance of parallel computers available when using grids. The Additive Schwarz (AS) method for elliptic problems or the Additive Schwarz Waveform Relaxation (ASWR) method for parabolic problems can be implemented easily in distributed computing environments and have very simple and systematic communication schemes. This algorithm is friendly to memory cache use and scales with the memory in parallel environments. ASWR in particular minimizes the number of messages sent in a parallel implementation and is very insensitive to delays due to a high latency network. The main drawback of the method is that it is one or several orders of magnitude slower than modern solvers such as multigrids. In the meantime, multigrids have poor parallel efficiency with high latency networks.

There have been two main classes of methods to speed up AS and ASWR. One is to introduce a coarse grid preconditioner. But a coarse grid operator reduces drastically the parallel efficiency on a slow network. A second option is to optimize the transmission conditions. This general avenue of work has been followed with success by numerous workers - see for example [3, 8, 9, 10] and their references. We have introduced in [7] a different and somehow complementary approach that consists of accelerating the sequence of trace on the interface generated by the AS method. The advantage of our postprocessing algorithm, besides its simplicity, is that it has quasi-optimum arithmetic complexity for the Poisson equation discretized on Cartesian grid while offering unique parallel efficiency on the grid. This is the only example, to our knowledge, of a numerically efficient Poisson solver that performs well on a grid of computers [2]. Our method offers also a general framework to speed up elliptic and non-linear elliptic solvers in a broad variety of conditions [1, 2, 6, 7].

Our main objective in this paper is to present an extension of this technique to the heat equation with Domain Decomposition (DD) in space *and* time. A generalization to Parabolic operators and its application to grid computing will be reported elsewhere [5].

2 Aitken-Schwarz Method for Linear Operators in One Space Dimension

The basic Aitken-Additive-Schwarz (AAS) method for linear elliptic problems can be found for example in [7]. Let us describe our AASWR algorithm for a domain decomposition in space *and* time with the following Initial Boundary Value Problem (IBVP):

$$\frac{\partial u}{\partial t} \; = \; L[u] \; + \; f(x,t), \; (x,t) \in \Omega = (0,1) \times (0,T), \tag{1}$$

$$u(x,0) \; = \; u_o(x), \; x \in (0,1), \tag{2}$$

$$u(0,t) = a(t), \; u(1,t) = b(t), \; t \in (0,T), \tag{3}$$

L is a second order linear elliptic operator. We assume that L coefficients are time independent and that the problem is well posed and has a unique solution.

We introduce the following discretization in space and time

$$0{=}x_0 < x_1 < ... < x_{N-1} < x_N{=}1, \; h_j{=}x_j - x_{j-1}, t_k{=}k \; dt, \; k{=}0 \dots M, \; dt{=}\frac{T}{M}.$$

Let us denote by X the column vector $X = (x_1, \dots, x_{N-1})^t$. A first order Euler implicit scheme in time writes

$$\frac{U^{k+1} - U^k}{dt} \; = \; D \, U^{k+1} \; + \; f(X, t^{k+1}), \; k = 0, .., M-1, \tag{4}$$

$$U^0 \; = \; u_o(X), U_0^{k+1} = a(t^{k+1}), \; U_N^{k+1} = b(t^{k+1}), \; k = 0, .., M-1, \tag{5}$$

where U^k is the column vector $U^k = (U_1^k, \dots, U_{N-1}^k)^t$. We also introduce the notation U_j for the row vector $U_j = (U_j^1, \dots, U_j^M)$.

D is a square matrix that comes from a finite difference or a finite element approximation for example. We do not need to specify this approximation. Our purpose is to compute efficiently the numerical solution of the discrete problem (4)-(5). At each time step one solves the linear system

$$(Id \; - \; dt \; D)U^{k+1} = F(U^k), \tag{6}$$

where Id is the matrix of the identity operator.

We assume that the matrix $A \; = \; Id \; - \; dt \; D$ of the linear system (6) is regular.

Introducing the matrices $U = (U^1, ..., U^M)$ and $F = (F(U^1), ..., F(U^M))$, we have

$$A\,U \;=\; F,\; U_0 = (a(t^1), \dots, a(t^M)),\; U_N = (b(t^1), ..., b(t^M)). \tag{7}$$

Let $\Omega_i = (y_i^l, y_i^r),\ i = 1..q$, be a partition of Ω with

$$x_0 = y_1^l < y_2^l < y_1^r < y_3^l < y_2^r, \dots, y_q^l < y_{q-1}^r < y_q^r = x_N.$$

One iteration of the ASWR algorithm writes

```
$for\ i = 1..q, do$
        $A_i\, V_i^{n+1} \;=\; F_i,\ in\ \Omega_i \times (0,T),$
        $V_i^{n+1}(y_i^l) = V_{i-1}^n(y_i^l),\ V_i^{n+1}(y_i^r) = V_{i+1}^n(y_i^r),$
$enddo$
```

where A_i is the appropriate sub-block of A corresponding to the discretization of the IBVP problem in $\Omega_i \times (0,T)$. This algorithm generates a sequence of vectors $W^{k_s} = (V_2^{l,k_s}, V_1^{r,k_s}, V_3^{l,k_s}, V_2^{r,k_s}, \dots, V_q^{l,k_s})$ corresponding to the boundary values on the set

$$\mathcal{S} = (y_2^l, y_1^r, y_3^l, y_2^r, \dots, y_q^l, y_{q-1}^r) \times (t^1, ..., t^M)$$

of the V_i for each iterate k.

The proof of convergence of the additive Schwarz waveform relaxation on the continuous problem (1) with the heat equation given in [4] is based on the maximum principle. The convergence of the ASWR algorithm at the discrete level follows from a discrete maximum principle as well and apply for example to the classical three points finite difference scheme with the heat equation problem. Because the parabolic problem (1) is linear, the trace transfer operator

$$W^{k_s+1} - W^{\infty} \rightarrow W^{k_s} - W^{\infty}$$

is linear. Its matrix P has the following pentadiagonal structure:

$$\left|\begin{array}{ccccccccc}
0 & P_1^r & 0 & 0 & \dots & & & & \\
P_2^{l,l} & 0 & 0 & P_2^{l,r} & \dots & & & & \\
P_2^{r,l} & 0 & 0 & P_2^{r,r} & \dots & & & & \\
 & & & & & & & & \\
 & & & & \dots & P_{q-1}^{l,l} & 0 & 0 & P_{q-1}^{l,r} \\
 & & & & \dots & P_{q-1}^{r,l} & 0 & 0 & P_{q-1}^{r,r} \\
 & & & & \dots & 0 & 0 & P_q^l & 0
\end{array}\right| .$$

The block $P_i^{l,l}, P_i^{l,r}, P_i^{r,l}, P_i^{r,r}$ are square matrices of size $(M-1)^2$. If the matrix P is known and the matrix $Id - P$ is regular, one step of the ASWR provides enough information to reconstruct the exact interface values by solving the linear system

$$(Id - P)W^{\infty} \;=\; W^1 - P\,W^0. \tag{8}$$

We can then define Algorithm (I):

Step 1: compute the first iterate of ASWR.

Step 2: solve the linear problem (8).

Step 3: compute the second iterate using the exact boundary value W^{∞}.

We observe that this algorithm is a direct solver provided that $Id - P$ is regular, no matter the overlap, or the fact that ASWR converges or not. This method is a generalization of the Aitken-Schwarz algorithm described in [7] for the case of linear elliptic operators. We call *algorithm (I)* the Aitken-Additive Schwarz waveform relaxation algorithm. We have the following result [5].

Theorem 1. *If the ASWR algorithm converges, then AASWR is a direct solver.*

The construction of P is done using the following basis of functions

$$\delta_j^k = 1, \text{ if } j = k, \ 0 \text{ otherwise}, \ j, k \in \{1, .., M\}$$

to represent the trace of the solution on the interfaces

$$y_i^{l/r} \ \times \ \{t_1, ..., t_M\}, \ i = 1..q.$$

Let us consider the family of subproblems in $\Omega_i \times (0, T)$,

$$\frac{V_{i,j}^{k+1} - V_{i,j}^k}{dt} \ = \ D_i[V_{i,j}^{k+1}], \ k = 0, \ldots, M-1, \tag{9}$$

$$V_{i,j}^0 \ = \ 0, V_{i,j}^{k+1}(y_i^l) \ = \ 0, \ V_{i,j}^{k+1}(y_i^r) \ = \ \delta_j^{k+1}, \ k = 0, \ldots, M-1. \tag{10}$$

Let $V_{i,j}$ denote the matrix that is the solution of the discrete problem (9)-(10). The j column vector of $P^{r,r}$, respectively $P^{r,l}$, is the trace of $V_{i,j}$ on y_{i+1}^l, respectively y_{i-1}^r. $P_i^{r,r}$ and $P_i^{r,l}$ are consequently lower triangular matrices.

We notice that all $V_{i,j}$ are obtained from $V_{i,1}$ by a translation in time, i.e.,

$$V_{i,j}(X_i, t) \ = \ V_{i,1}(X_i, t - t_{j-1}), \ t \in \{t_j, \ldots, t_M\}, \tag{11}$$

and

$$V_{i,j}(X_i, t) \ = \ 0, \ t \in \{t_0, t_{j-1}\}. \tag{12}$$

The first column vector of $P^{r,r}$, respectively $P^{r,l}$, is the trace of $V_{i,1}$ on y_{i+1}^l, respectively y_{i-1}^r. From (11) we see that all columns of $P_i^{r,r}$, respectively $P_i^{r,l}$, are obtained from the first column of matrix $P_i^{r,r}$, respectively $P_i^{r,l}$, with no additional computation. To conclude, the construction of the matrix P of the trace transfer operator is achieved if one computes once and for all the solution of the two following sub-problems in $\Omega_i \times (0, T)$,

$$\frac{V_{i,j}^{k+1} - V_{i,j}^k}{dt} \ = \ D_i[V_{i,j}^{k+1}], \ k = 0, \ldots, M-1, \tag{13}$$

$$V_{i,j}^0 \ = \ 0, \ V_{i,j}^{k+1}(y_i^l) \ = \ \delta_1^{k+1}, \ V_{i,j}^{k+1}(y_i^r) \ = \ 0, \ k = 0, \ldots, M-1, \tag{14}$$

and

$$\frac{V_{i,j}^{k+1} - V_{i,j}^k}{dt} \ = \ D_i[V_{i,j}^{k+1}], \ k = 0, \ldots, M-1, \tag{15}$$

$$V_{i,j}^0 \ = \ 0, \ V_{i,j}^{k+1}(y_i^l) \ = \ 0, \ V_{i,j}^{k+1}(y_i^r) \ = \ \delta_1^{k+1}, \ k = 0, \ldots, M-1. \tag{16}$$

Remark 1. All sub-problems listed above needed for the construction of the trace transfer operator matrix can be solved with embarrassing parallelism.

We are going now to illustrate the method with the classical finite difference approximation for the one dimensional heat equation. The domain of computation is $(0, 1) \times (0, T)$. The grid has constant space step h and time step $dt = h$. We keep the number of grid points per sub-domain fixed with $N_b = 20$. Further the overlap is kept minimum, that is a one mesh interval. The Standard Method (SM) applies a direct tridiagonal solver to integrate each time step. The LU decomposition of the tridiagonal system can be computed once, since the same linear system is solved at every time step. The arithmetic complexity of the SM is then $n_1 = C_1\ N\ M$, where C_1 is an integer. $C_1 = 5$ for Gaussian elimination. The arithmetic complexity of one iterate of the ASWR algorithm is $n_q\ =\ C_1\ M\ (N + q - 1)$ which is asymptotically equivalent to n_1.

All subdomains correspond to the same finite difference operator. Consequently, the construction of the matrix P requires to solve one sub-domain problem (13)-(14) or (15)-(16). The arithmetic complexity of the construction of P is then $C_1\ M\ \frac{N+q-1}{q}$ and can be neglected against n_q. The acceleration step requires to solve the sparse linear system (8) uses asymptotically $n_{interface} = C_2 M[(q-1)^2 + O(q)]$ floating point operations (flops). $n_{interface}$ is small compare to n_q as long as $q << \sqrt{N}$.

Overall the number of flops for the AASWR procedure is about twice the number of flops for the standard SM with no DD. However modern computer architectures do not perform linearly with the number of flops. To illustrate this concept, we have performed the computation with both algorithm SM and AASWR on a PC running Matlab with a Pentium 4 2.66GHz. This PC has 1GB of main memory. With moderate number of time steps and large problem size, the advantage of the AASWR algorithms over the SM is clear. Figure 1 provides some comparison between both algorithm with ten time steps, i.e $M = 10$, $N_b = 20$ and a number of subdomains that varies from 2 to 20. The elapsed time is given in seconds and averages the measurement provided by one hundred runs. We remind here that the size of the problems grows linearly with the number of domains according to $N\ =\ N_b\ +\ (q-1)\ (N_b - 1)$. Overall the construction of P and the acceleration step has negligible elapse time. In AASWR the elapse time grows linearly with the number of subdomains. AASWR performs better than SM for $q > 6$. We believe that the cache size is responsible for the two peaks in the curve giving the performance of the SM. On the contrary the AASWR seems to be insensitive to the cache size for the dimension of the sub-domain that has been chosen here.

Figure 2 shows that the condition number of the matrix $(Id - P)$ used in the acceleration step grows linearly with the number of subdomains, which is proportional to the problem size in space N. However from our numerical experiments we have concluded that the acceleration procedure does not seems to impact significantly the accuracy of our exact solver.

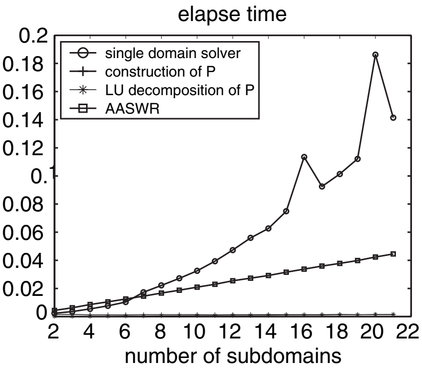


Fig. 1. Convergence of ASWR and AASWR for the heat equation.

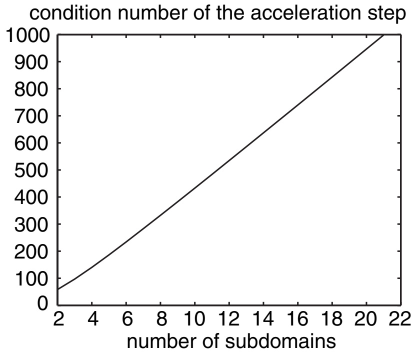


Fig. 2. Condition number of the linear system (8).

Most of the results obtained in this section can be extended to multi-dimensional parabolic problems provided L is separable or a weak perturbation of a separable operator [5].

3 Aitken-Schwarz Method for Linear Operators in the Multidimensional Case

To simplify the notations we will restrict ourselves to two space dimensions. We further assume that the domain Ω is a square discretized by a rectangular Cartesian grid with arbitrary space steps in each direction. Let us consider the IBVP:

$$\frac{\partial u}{\partial t} \;=\; L[u] \;+\; f(x,y,t),\ (x,y,t) \in \Omega = (0,1)^2 \times (0,T), \tag{17}$$

$$u(x,y,0) \;=\; u_o(x,y),\ (x,y) \in (0,1)^2, \tag{18}$$

$$u(0,y,t) = a(y,t),\ u(1,y,t) = b(y,t),\ y \in (0,1),\ t \in (0,T), \tag{19}$$

$$u(x,0,t) = c(x,t),\ u(x,1,t) = d(x,t),\ x \in (0,1),\ t \in (0,T), \tag{20}$$

where L is a second order linear elliptic operator. We assume that the problem is well posed and has a unique solution. Using an appropriate shift in space we can restrict ourselves to homogeneous Dirichlet boundary conditions.

The domain $\Omega = (0,1)^2$ is decomposed into q overlapping strips $\Omega_i = (y_i^l, y_i^r) \times (0,1)$.

We first present the general algorithm when L is a separable linear operator and refer to the theoretical framework established in [1] for elliptic operator:

$$L = L_1 + L_2,\ L_1 = e_1\partial_{xx} + f_1\partial_x + g_1,\ \ L_2 = e_2\partial_{yy} + f_2\partial_y + g_2.$$

e_1, f_1, g_1 are functions of x only, and e_2, f_2, g_2 are functions of y only. We write the discretized problem as follows

$$\frac{U^{k+1} - U^k}{dt} \;=\; D_{xx}[U^{k+1}] \;+\; D_{yy}[U^{k+1}] \;+\; f(X,Y,t^{k+1}),\ k = 0,\ldots,M-1, \tag{21}$$

with appropriate boundary conditions corresponding to (18)-(20).

Our main objective is to rewrite the discretized problem in such a way that we can reuse the results of Section 2 that is for the one space dimension case. Let us assume that D_{yy} has a family of $(N_y - 1)$ independent eigenvectors $\Phi_j,\ j = 1,..,N_y$ in $\mathbb{R}^{N_y-1}$ with corresponding eigenvalues μ_j.

The Φ_j are implicitly the numerical approximation in $(0,1)$ of the solutions of the following continuous eigenvector problems:

$$L_2[v(y)] \;=\; \mu\, v(y),\ v(0) = v(1) = 0. \tag{22}$$

Let us introduce the decompositions

$$U^k(x,y,t) = \sum_{j=1}^{N_y-1} \Lambda_j^k(x,t)\Phi_j(Y), \qquad u_o(x,y) = \sum_{j=1}^{N_y-1} \lambda_j^k(x)\Phi_j(y),$$
$$f(x,y,t^k) = \sum_{j=1}^{N_y-1} f_j^k(x,t^k)\Phi_j(y), \qquad a(y,t^k) = \sum_{j=1}^{N_y-1} a_j(t^k)\Phi_j(y),$$
$$b(y,t^k) = \sum_{j=1}^{N_y-1} b_j(t^k)\Phi_j(y).$$

The discrete solution of (21) satisfies the following set of $(Ny-1)$ uncoupled problems

$$\frac{\Lambda_j^{k+1}-\Lambda_j^k}{dt} = D_{xx}[\Lambda_j^{k+1}] + \mu_j\ \Lambda^{k+1} + f_j(X,t^{k+1}),\ k=0,\ldots,M-1, \quad (23)$$
$$\Lambda_j^0 = \lambda_j(X),\ \Lambda^{k+1}(x_0) = a_j(t^{k+1}),\ \Lambda^{k+1}(x_{N_x}) = b_j(t^{k+1}),\ k=0,\ldots,M-1. \quad (24)$$

The trace transfer operator can be decomposed into (N_y-1) independent trace transfer operators

$$W_j^{k_s} - W_j^{\infty} \to W_j^{k_s+1} - W_j^{\infty},$$

that apply to each component of the trace of the solution expanded in the eigenvector basis $E = \{\Phi_j,\ j=1,\ldots,(N_y-1)\}$. Let Q_j be the matrix of this linear operator. The matrix P has now a (N_y-1) diagonal block structure, where each block is the matrix Q_j. The acceleration procedure of *Algorithm (I) Step 2* writes now

- Expand the trace of the solution in the eigenvector basis E and solve component wise

$$(Id-Q_j)W_j^{\infty} = W_j^1 - Q_j\ W_j^0,\ \forall j \in \{1,\ldots,(Ny-1)\}. \quad (25)$$

Assemble the boundary condition $W^{\infty} = \sum_{j=1,\ldots,Ny-1} W_j^{\infty}\ \Phi_j$.

Let us emphasize that the sub-domain problems in $\Omega_j \times (0,T)$ can be integrated by any existing efficient numerical solver. It is only the acceleration step 2 that requires the decomposition of the *trace* of the solution into the eigenvector basis E. Because all eigenvector components of the solution are independents, we have then as in the one dimension space case:

Theorem 2. *If the ASWR algorithm converges, then AASWR is a direct solver.*

The construction of the Q_j can be done exactly as in the one space dimension case and can be computed with embarrassing parallelism.

This algorithm applies to the standard heat equation problem discretized in space on a five point stencil with central finite differences on a regular Cartesian mesh. Following the same steps as in Section 2.4, one can show that AASWR requires roughly two times as many floating point operations. But as stated before the AASWR algorithm is a parallel algorithm fairly tolerant to high latency networks. We have verified also that AASWR performs better than SM on a scalar processor with small number of time steps and large problem size.

We have verified also that the accuracy of our AASWR solver is satisfactory for three dimensional problem with singular source terms.

Remark 2. Our result can be easily generalized to tensorial products of a one dimensional grid with adaptive space stepping. The key hypothesis is the separability of the discrete operator $D_{xx} + D_{yy}$ on the tensorial product of grid. Because h_y is not a constant, the eigenvectors Φ_j are not known analytically and should be computed numerically as in [1].

Details of the parallel implementation of our method that are specific to space *and* time decomposition are reported in [5].

4 Conclusion

In this paper we have shown how to generalize the Aitken-like acceleration method of the additive Schwarz algorithm for elliptic problems to the additive Schwarz waveform relaxation for the heat equation. This new DD algorithm is in space *and* time. Since the concept of our acceleration technique is general and might be applied in principle to any block-wise relaxation scheme, we expect that it can be combined with some optimized transmission conditions for the same PDE problem. A further step in the development of our methodology would be to consider unstructured meshes, and approximate the trace transfer operator with for example, the coarse grid interface approximation presented in [6].

Acknowledgement. This work was supported in part by NSF Grant ACI-0305405.

References

[1] J. Baranger, M. Garbey, and F. Oudin-Dardun. On Aitken like acceleration of Schwarz domain decomposition method using generalized Fourier. *Domain Decomposition Methods in Science and Engineering*, pages 341 – 348, 2003.

[2] N. Barberou, M. Garbey, M. Hess, M. Resch, T. Rossi, J. Toivanen, and D. Tromeur-Dervout. On the efficient meta-computing of linear and nonlinear elliptic problems. *J. Parallel Distrib. Comput. (special issue on grid computing)*, 63:564 – 577, 2003.

[3] B. Després. Décomposition de domaine et problème de Helmholtz. *C. R. Acad. Sci. Paris Sér. I Math.*, 311(6):313–316, 1990.

[4] M. Gander and H. Zhao. Overlapping Schwarz wavefrom relaxation for the heat equation in n-dimensions. *BIT*, 40(4):001 – 004, 2000.

[5] M. Garbey. Acceleration of a Schwarz waveform relaxation method for parabolic problems. Preprint UH-CS-06-11. Submitted.

[6] M. Garbey. Acceleration of the Schwarz method for elliptic problem. *SIAM J. Sci. Comput.*, 26(6):1871 – 1893, 2005.

[7] M. Garbey and D. T. Dervout. On some Aitken-like acceleration of the Schwarz method. *Internat. J. Numer. Methods Fluids*, 40(12):1493 – 1513, 2002.

[8] C. Japhet, F. Nataf, and F. Rogier. The optimized order 2 method. Application to convection-diffusion problems. *Future Generation Computer Systems FUTURE*, 18, 2001.

[9] V. Martin. An optimized Schwarz waveform relaxation method for unsteady convection-diffusion equation. *Appl. Numer. Math.*, 52(4):401 – 428, 2005.

[10] F. Nataf and F. Rogier. Factorization of the convection-diffusion operator and a (possibly) non overlapping Schwarz method. *M3AS*, 5(1):67 – 93, 1995.

Toward a Real Time, Image Based CFD

Marc Garbey and Bilel Hadri

Computer Science Department, University of Houston.
{garbey,hadri}@cs.uh.edu

Summary. We present a method to combine fluid dynamics and image analysis into a single fast simulation environment. Our target applications are hemodynamic studies. Our method combines an NS solver that relies on the L_2 penalty approach pioneered by Caltagirone and co-workers, and a level set method based on the Mumford-Shah energy model. Working in Cartesian coordinates regardless of grid no matter the complexity of the geometry, one can use fast parallel domain decomposition solvers in a fairly robust and consistent way. The input of the simulation tool is a set of JPEG images, and the output can be various flow components as well as shear stress indicators on the vessel or domain wall. In two space dimensions the code runs close to real time.

1 Introduction and Motivation

The objective of this work is to use angiogram medical images to produce flow simulations by a very robust and fast method. The emphasis is not on high accuracy, since there are many sources of errors or variability in medical data. From medical imaging, we extract the geometry of large vessels. Our algorithm provides a first order approximation of some main quantities of interest in cardiovascular disease: the shear stress and the pressure on the wall, as well as the flow components in the artery.

We present a fast, versatile and robust NS solver that relies heavily on the L_2 penalty approach pioneered by Caltagirone and co-workers [2] and combines nicely with a level set method based on the Mumford-Shah energy model [4].

The wall boundary condition is immersed in the Cartesian mesh thanks to the penalty term added to the momentum equation. We use the domain decomposition (DD) algorithm of [3] that has high numerical efficiency and scales well with parallel computers in order to take full advantage of the regular data structure of the problem. This DD is coupled with a sub-domain solver that is tuned to provide the fastest result on the computer available for the run. In this paper we present simulations in two space dimensions while results in three space dimensions will be reported elsewhere.

2 Navier-Stokes Flow Solver

Since we concentrate our study on large vessels, we use an incompressible NS fluid flow model [8, 11].

In this paper we will use the penalty method introduced by Caltagirone and co-workers [2] since it is simpler to implement than our previous boundary fitted methods [7] and applies naturally to flow in complex domains with moving walls [10].

The flow of incompressible fluid in a rectangular domain $\Omega = (0, L_x) \times (0, L_y)$ with prescribed values of the velocity on $\partial\Omega$ obeys the NS equations:

$$\partial_t U + (U.\nabla)U + \nabla p - \nu \nabla.(\nabla U) = f, \text{ in } \Omega$$

$$div(U) \;=\; 0, \text{ in } \Omega, U = g \text{ on } \partial\Omega,$$

We denote by $U(x, y, t)$ the velocity with components (u_1, u_2) and by $p(x, y, t)$ the normalized pressure of the fluid. ν is a kinematic viscosity.

With an immersed boundary approach the domain Ω is decomposed into a fluid subdomain Ω_f and a wall subdomain Ω_w. In the L_2 penalty method the right hand side f is a forcing term that contains a mask function Λ_{Ω_w}

$$\Lambda_{\Omega_w}(x, y) = 1, \text{ if } (x, y) \in \Omega_w, \text{ 0 elsewhere},$$

and is defined as follows

$$f = -\frac{1}{\eta} \Lambda_{\Omega_w} \, \{U - U_w(t)\}.$$

U_w is the velocity of the moving wall and η is a small positive parameter that goes to zero.

A formal asymptotic analysis helps us to understand how the penalty method matches the no slip boundary condition on the interface $S_w^f = \bar{\Omega}_f \bigcap \bar{\Omega}_w$ as $\eta \to 0$. Let us define the following expansion:

$$U = U_0 \;+\; \eta \, U_1, \; p \; = p_0 \;+\; \eta \, p_1.$$

Formally, in first order, we obtain,

$$\frac{1}{\eta} \Lambda_{\Omega_w} \, \{U_0 - U_w(t)\} = 0,$$

that is

$$U_0 = U_w, \text{ for } (x, y) \in \Omega_w.$$

The leading order terms U_0 and p_0 in the fluid domain Ω_f satisfy the standard set of NS equations:

$$\partial_t U_0 + (U_0.\nabla)U_0 + \nabla p_0 - \nu \nabla.(\nabla U_0) = 0, \text{ in } \Omega_f$$

$$div(U_0) \;=\; 0, \text{ in } \Omega.$$

At the next order we have in Ω_w,

$$\nabla p_0 \;+\; U_1 \;+ Q_w = 0, \tag{1}$$

where

$$Q_w \;=\; \partial_t U_w + (U_w.\nabla)U_w - \nu\nabla.(\nabla U_w).$$

Further the wall motion U_w must be divergence free. Continuing to the next order we have in Ω_f,

$$\partial_t U_1 + (U_0.\nabla)U_1 + (U_1.\nabla)U_0 + \nabla p_1 - \nu\nabla.(\nabla U_1) = 0,$$

with

$$div(U_1) = 0.$$

In the simplest situation where $U_w \equiv 0$, we observe that the motion of the flow is driven by the pressure following a classical Darcy law. η stands for a small permeability. To summarize as $\eta \to 0$, the flow evolution is dominated by the NS equations in the artery, and by the Darcy law with very small permeability in the wall. This actually corresponds to a standard multiscale model of blood flow in the main arteries. From the analytical point of view it was shown in [1] for a fixed wall, i.e. $U_w \equiv 0$, that the convergence order of the penalty method is of order $\eta^{\frac{3}{4}}$, in the fluid domain, and $\eta^{\frac{1}{4}}$ in the wall.

The mask function Λ_{Ω_w} is obtained with an image segmentation technique that is a level set method. Since the contours of the image are not necessarily sharp, it is interesting to use the level set method presented in [4] and based on the Mumford-Shah Model.

Regarding the resolution of the equation, we use a projection method for the time step as follows :

- Step 1: prediction of the velocity $\hat{u}^{k+1}$ by solving either :

$$\frac{\hat{u}^{k+1} - u^{k,*}}{\Delta t} - \nu\Delta u^k = f^{k+1} - \nabla p^k \; or; \frac{\hat{u}^{k+1} - u^{k,*}}{\Delta t} - \nu\Delta u^{k+1} = f^{k+1} - \nabla p^k$$

  in $(0, L_x) \times (0, L_y)$ with the boundary condition $\hat{u}^{k+1} = g$ on $\partial\Omega$. We denote that $u^{k,*}$ is obtained thanks to the method of characteristics.
- Step 2: projection of the predicted velocity to the space of divergence free functions.

$$-div\nabla\delta p = -\frac{1}{\Delta t} div\hat{u}^{k+1}; u^{k+1} = \hat{u}^{k+1} - \Delta t\delta p$$

$$p^{k+1} = p^k + \delta p\,.$$

The NS calculation decomposes into three steps: the prediction of the flow speed components, the solution of a Poisson problem for the pressure, and eventually the computation of the shear stress along the wall. The momentum equations can be solved quickly, while the performance of the code is dominated by the Poisson solver for the pressure. We detail this part of the algorithm in the next section.

3 Multi-Algorithm for the Pressure Solver

The pressure equation can be integrated with a number of existing fast Poisson solvers since the discretization grid is regular. It is convenient for example to use a (full) multigrid solver here. The arithmetic complexity of this solver is optimum. Further the iterative solver converges extremely fast for those grid points that are in the solid wall. However the pressure equation has to be solved at every time step and what turns out to be the faster solver may depend on the computer architecture. In the following we use the framework of the Aitken- Additive Schwarz method to fine tune the subdomain solver [5, 6]. The main reason being that on a standard Beowulf system with fast ethernet switch, the high latency of the network significantly lowers the performance of the multigrid solver. On the contrary the Aitken-Schwarz algorithm may double the number of flops compared to an optimal solver but is highly insensitive to the latency of the network.

Interface software [6] has been written to reuse a broad variety of existing linear algebra software for each subdomain such as LU factorization, a large number of Krylov methods with incomplete LU preconditioner, and geometric or algebraic multigrid solvers.

Only experiments can provide the fastest method for the resolution of a linear system.

Let us restrict ourselves to three software systems: Linpack for LU, Sparskit for iterative solver, Hypre for algebraic multigrid. We build a surface response model [9] based upon the least square quadratic polynomial approximation of the elapsed time as a function of the grid size (n_x, n_y):

$$T(n_x, n_y) = \beta_0 + \beta_1 n_x + \beta_2 n_y + \beta_3 n_x n_y + \beta_4 n_x^2 + \beta_5 n_y^2 \,.$$

The performance modeling can be done in principle with any linear algebra software. This model to predict the elapsed time for the resolution of a linear solver with LU or a Krylov solver with a relative prediction error of the prediction less than a few percent. To build the model we need on the order of 10 test runs with various grid configurations that cover the region of prediction. For Hypre, this model does not give good predictions in general. One observes that the elapsed time is very sensitive to the size.

Based on the surface response model, one can then decide what is the optimum solver for a given subdomain dimension. For illustration purpose let us restrict ourselves to the 2D Poisson problem that corresponds to the pressure solver. One can notice that LU performs well for small sizes, while, iterative solvers such as BiCGStab (Krylov method) and AMG-GMRES (multigrid method) give the fastest results for large grid sizes. More details on this study can be found in [6].

Figures 1 and 2 give the performances on different processor architectures, a dual processor AMD 1800+ with 2GB of RAM and a dual processor 900 MHz Itanium2 with 3 GB of RAM, respectively. We clearly see the difference of the region of the area where LU is faster than the iterative method. The automatic tuning through the model of the solver helps us to choose wisely the fastest solver for each subdomain solver.

It should be observed from Figures 3 and 4 that the optimum choice of the subdomain solver depends weakly on the number of processors. In the framework of the AS algorithm, we have observed that message passing favors an iterative solver versus a direct solver when the difference on performance between the two solvers is moderate.

Let us now illustrate our parallel algorithm on the incompressible Navier-Stokes flow that uses the pressure solver of this section.

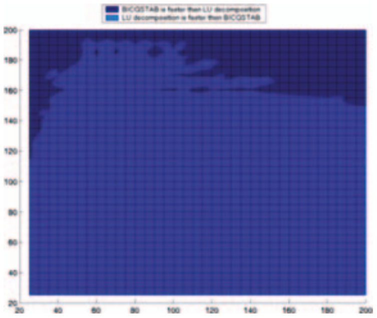


Fig. 1. Comparison between BiCGStab and LU decomposition on a 32bit AMD processor

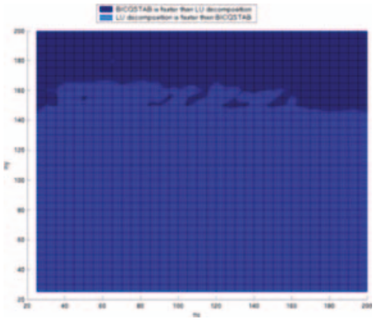


Fig. 2. Comparison between BiCGStab and LU decomposition on a 64 bit Itanium

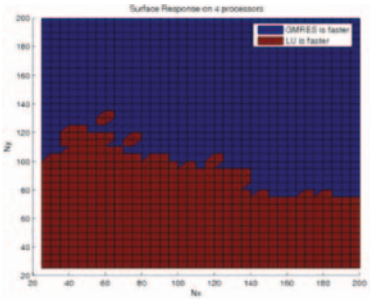


Fig. 3. Surface response with 4 processors

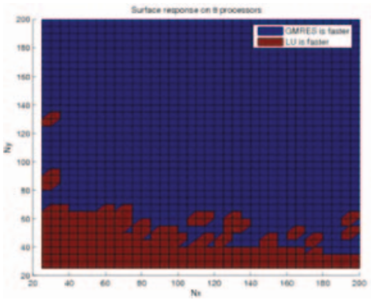


Fig. 4. Surface response with 8 processors

4 Parallel Performances of the NS Code

Let us first present a two dimensional simulation obtained from an x-ray. Figure 5 shows the frontal projection of a carotid in the brain area during an angiogram procedure. Figure 6 shows an example for a steady flow calculation in the region of interest with a Reynolds number of order 330. The flow comes from the left side. The size of the grid in this simulation is about 210 × 170.

The simulation of one cardiac cycle with an 8-way Opteron machine, for the grid sizes, 300 × 100 and 450 × 150, takes around 3 and 9 seconds, respectively. The time for the image segmentation is less. These simulation are then close to real time.

Figure 7 shows the speedup of this code with different grid sizes. Until four processors, the code has a linear speedup, and then the speed up deteriorates.

Actually this is mainly due to the design of the crossbar architecture of the low cost Opteron system which does not scale from 4 to 8 processors.

The results on scalability of our code are better.

In figure 8 we keep the aspect ratio of the grid $\frac{h_x}{h_y}$ the same: three tests have been performed respectively with a problem of size 141 × 567 for two processors, 200 × 801

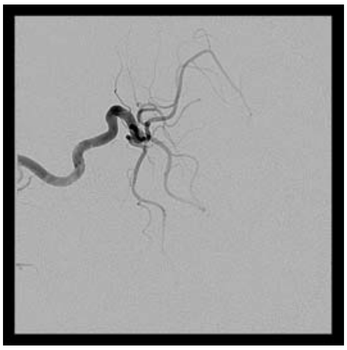


Fig. 5. Benchmark problem

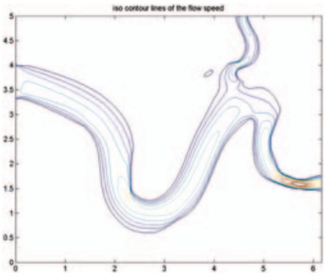


Fig. 6. Contour u

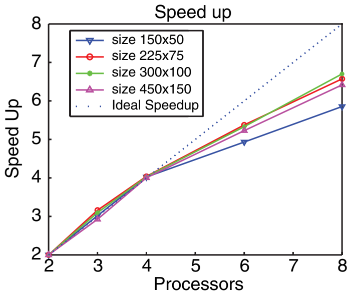


Fig. 7. Speedup of the Navier-Stokes code.

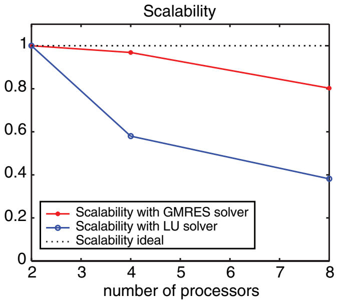


Fig. 8. Scalability of the Navier-Stokes code

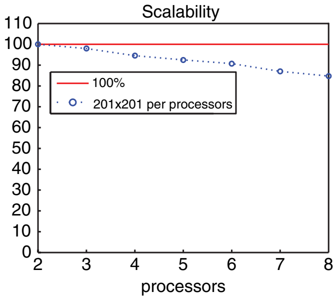


Fig. 9. Scalability of the Navier-Stokes code with LU solver

on 4 processors, and 283 × 1129 on 8 processors. GMRES gives a better scalability result compared to LU because the growth of the bandwidth of the matrix as the number of subdomain increases penalize the LU solver. On the contrary if we keep the size of the subdomain fixed, here 201 × 201 in figure 9, we obtain a very good scalability of the code no matter the subdomain solver.

5 Conclusion

We have presented an image based CFD algorithm designed for hemodynamic simulation. In two space dimension we obtain a code that can be easily optimized for a specific parallel computer architecture. Robustness and simplicity of the solver are key elements to make the simulation applicable to clinical conditions. We have developed recently a three dimension version of the method that will be reported elsewhere. Our technique may not be appropriate for turbulent flow for high Reynolds numbers, but there are a number of cardiovascular problems that corresponds to unsteady situations with relatively modest Reynolds numbers [8, 11].

Acknowledgement. This research was partially supported by Awards 0305405 and 0521527 from the National Science Foundation.

References

[1] P. Angot, C.H. Bruneau, and P. Fabrie. A penalisation method to take into account obstacles in viscous flows. *Numer. Math.*, 81(4):497–520, 1999.

[2] E. Arquis and J.P. Caltagirone. Sur les conditions hydrodynamiques au voisinage d'une interface milieu fluide-milieux poreux: Application à la convection naturelle. *C. R. Acad. Sci. Paris Sér. II*, 299:1–4, 1984.

[3] N. Barberou, M. Garbey, M. Hess, M. Resch, T. Rossi, J. Toivanen, and D. Tromeur Dervout. Efficient metacomputing of elliptic linear and non-linear problems. *J. Parallel Distrib. Comput.*, 63:564–577, 2003.

[4] T.F. Chan and L.A. Vese. Active contours without edges. *IEE Transaction on Image Processing*, 10 (2):266–277, 2001.

[5] M. Garbey and D. Tromeur Dervout. On some Aitken like acceleration of the Schwarz method. *Internat. J. Numer. Methods Fluids*, 40:1493–1513, 2002.

[6] M. Garbey, W. Shyy, B. Hadri, and E. Rougetet. Efficient solution techniques for CFD and heat transfer. In *ASME Heat Transfer/Fluids Engineering Summer Conference*, 2004.

[7] M. Garbey and Yu.V. Vassilevski. A parallel solver for unsteady incompressible 3D Navier-Stokes equations. *Parallel Comput.*, 27(4):363–389, 2001.

[8] D.A. McDonald. *Bloof Flow in Arteries.* Edward Arnold, 3rd edition, 1990.

[9] D.C. Montgomery and R.H. Myers. *Response Surface Methodology: Process and Product Optimization Using Designed Experiments.* Wiley, 2nd edition, 2002.

[10] K. Schneider and M. Farge. Numerical simulation of the transient flow behaviour in tube bundles using a volume penalisation method. *J. of Fluids and Structures*, 20(4):555–566, 2005.

[11] X.Y. Xu, M.W. Collins, and C.J.H. Jones. Flow studies in canine aortas. *ASME J. Biomech. Engrg.*, 114(11):505–511, 1992.

A Multilevel Method for Solution Verification

Marc Garbey and Christophe Picard

University of Houston, Computer Science (`http://www.cs.uh.edu/~garbey/`)

Summary. This paper addresses the challenge of solution verification and accuracy assessment for computing complex Partial Differential Equation (PDE) model. Our main target applications are bio-heat transfer and blood flow simulation problems. However our long term goal is to provide a postprocessing package that can be attached to any existing numerical simulation package, for example widely used commercial codes such as ADINA, Ansys, Fluent, Star-CD etc., and provide an a posteriori error estimate to their simulation.

1 Introduction and Motivation

The problem of accuracy assessment is a necessary step that follows the code verification step and precedes the code validation step, completing the global task of providing a reliable virtual experiment tool [5].

Our major goal in this paper is to pursue our work on the design of a new multilevel method that offer a general framework to do Solution Verification (SV) efficiently. The standard approach in applied mathematics to handle the problem of SV is to work on the approximation theory of the PDE. For each specific PDE problem, the right Finite Element (FE) approximation may provide the correct a posteriori error estimate. Unfortunately this approach may require a complete rewriting of an existing CFD code based on Finite Volume (FV) for example and lack generality.

Our method relies on four main ideas that are (1) the embedding of the problem of error estimation into an optimum design framework that can extract the best information from a set of two or three existing numerical results, (2) the resolution of the problem as much as possible as a (non)linear set of discrete equations to produce a general tool, and renounce on using the specific approximation theory used the compute the PDE solution. Since we usually have no access to the detailed knowledge of the internal structure of the code that produces the numerical solution, (3) the formulation of a framework that can reuse any a posteriori estimator if they are available (4) the use of distributed computing (or grid computing) to get a cost effective SV.

2 Method

From the applied mathematics point of view, a posteriori estimates have been around for many years [1, 8]. There is a vast literature on this subject. The main challenge is still *to estimate numerical accuracy on under-resolved grids* [5]. As a matter of fact, in complex modeling, as described in the ASCI project, best grid solutions provided by our best computing resources are fairly under-resolved at least locally.

We present in this paper an entirely different framework to construct reliable a posteriori estimates for *general* PDEs or system of PDEs. Let us first describe the general concept of our method [2, 3, 7].

2.1 General Concept

We consider a boundary value problem (Ω is a polygonal domain and $n = 2$ or 3) :

$$L[u(x)] = f(x), \; x \in \Omega \subset \mathbb{R}^n, \; u = g \text{ on } \partial\Omega. \tag{1}$$

We assume that the PDE problem is well posed and has a unique smooth solution. We consider a finite volume approximation of (1) on a family of meshes $M(h)$ parametrized by $h > 0$ a small parameter. The smaller h the finer should be the discretization. We denote symbolically the corresponding family of linear systems

$$A_h U_h = F_h. \tag{2}$$

Let p_h denotes the projection of the continuous solution u onto the mesh $M(h)$. We assume a priori that ($||.||$ is a given discrete norm):

$$||U_h - p_h(u)|| \; \rightarrow \; 0, \text{ as } h \; \rightarrow \; 0, \tag{3}$$

Let $M(h_1)$ and $M(h_2)$ be two different meshes used to build two approximations U_1 and U_2 of the PDE problem (1). A consistent linear extrapolation formula should have the form

$$\alpha U_1 + (1 - \alpha) U_2,$$

where α is a weight function. In classical Richardson Extrapolation (RE) the α function is a constant. In our optimized extrapolation method α is an unknown space dependent function solution of the following optimization problem, where G is an objective function to be defined:

P_α*: Find* $\alpha \in \Lambda(\Omega) \subset L_\infty$ *such that* $G(\alpha\, U_1 \; + \; (1-\alpha)\, U_2)$ *is minimum.*

The Optimized Extrapolated Solution (OES) if it exists, is denoted $V_e = \alpha U_1 + (1 - \alpha) U_2$. For computational efficiency, $\Lambda(\Omega)$ should be a finite vector space of very small dimension compared to the size of matrix A_h defined in (2). The objective function G might be derived from any existing a posteriori error estimators if possible. For a number of fluid dynamic methods used in bioengineering such as the immersed boundary technique, or the chimera technique there is no solid theoretical framework that can provides such rigorous a posteriori estimators. For complex bioengineering problems, the fact that there exist a functional space framework to derive a posteriori estimate is more the exception than the generality. Our ambition is to provide a numerical estimate on $||U_j - U_\infty||$, $j = 1, 2$, without computing U_∞ effectively. The solution U_j can then be verified assuming (3). The fine mesh $M(h_\infty)$ should be set such that it captures all the scales of the continuous solution with the level of accuracy required by the application. We have a priori $h_\infty \ll h_1,\ h_2$. Both coarse grid solutions U_1 and U_2 must be projected onto $M(h_\infty)$. We will denote $\tilde{U}_1$ and $\tilde{U}_2$ the corresponding functions. We choose then to minimize the consistency error for the numerical approximation of (1) on a fine mesh $M(h_\infty)$. The objective function is then

$$G(U^\alpha) = ||A_{h_\infty}\, U^\alpha - F_{h_\infty}||, \text{ where } U^\alpha = \alpha\, \tilde{U}_1\ +\ (1-\alpha)\, \tilde{U}_2. \tag{4}$$

The choice of the discrete norm should depend on the property of the solution. In the Least Square Extrapolation (LSE) method [2, 3] we chose the discrete L_2 norm. The choice of the L_1 or the L_∞ norm provides some useful additional information, for example, for stiff elliptic problems.

One of the difficulties encountered with a two-level extrapolation method is that there exists subsets of $M(h_\infty)$ where $\tilde{U}_1$ and $\tilde{U}_2$ are much closer to each other than what the expected order of accuracy based on local error analysis should provide. In such areas, the sensitivity of the extrapolation to the variation of α is very weak and the problem is ill posed. These subsets should be treated as outliers of the optimization computation procedure. A potentially more robust procedure consists of using three levels of grid solution. The optimization problem writes then

$P_{\alpha,\beta}$*: Find* $\alpha, \beta \in \Lambda(\Omega)\ \subset\ L_\infty$ *such that* $G(\alpha\, U_1\ +\ \beta\, U_2\ +\ (1-\alpha-\beta)\, U_3)$ *is minimum.*

We notice that if all $U_j,\ j = 1,\ldots,3$, coincide at the same space location there is either no local convergence or all solutions U_j are exact. In such a situation, one cannot expect improved local accuracy from any OES. The robustness of OES should come from the fact that we do not suppose a priori any asymptotic on the convergence rate of the numerical method as opposed to RE.

Let us assume that the optimization problem P_α or $P_{\alpha,\beta}$ has been solved and that we have computed an optimum solution V_e either from the two levels or three levels method. We are going to discuss now its application to provide a posteriori error estimators.

Let us denote U_j to be one of the coarse grid approximations at our disposal. A global a posteriori estimate of the error $||U_j - p_h(u)||$ may come in two different ways. For the sake of simplicity we will assume that G is the L_2 norm of the residual (4).

• First is the recovery method based on the idea that the optimized extrapolated solution is more accurate than the coarse grid solution. Let us denote $\tilde{U}_j$ the coarse grid solution projected onto the fine grid $M(\infty)$ via a suitable interpolation procedure. Let us assume that the extrapolated solution is decisively more accurate than that based on interpolation from the coarse grid solution, namely,

$$||V_e - p_h(u)||_2 \ll ||\tilde{U}_j - p_h(u)||_2. \tag{5}$$

Then $||\tilde{U}_j - V_e||_2 \sim ||\tilde{U}_j - p_h(u)||_2$ and $||V_e - \tilde{U}_2||$ is a good error indicator to assess the accuracy on U_2.

We have seen in our numerical experiments with steady incompressible Navier-Stokes (NS) solutions that this method may give a good *lower* bound error estimate. But we do not know in general if the hypothesis (5) is correct. There is no guarantee that a smaller residual for V_e than for U_2 on the fine grid $M(h_\infty)$ leads to a smaller error.

• Second is a global *upper* bound that follows from a stability estimate with the discrete operator. We have

$$||V_e - U^0|| \ < \ \mu \, G(V_e), \text{ where } \mu \geq ||(A_{h_\infty})^{-1}|| \,,$$

where U^0 is the fine grid solution.

We conclude then

$$||\tilde{U}_2 - U^0||_2 \ < \ \mu \, G(V_e) + ||V_e - \tilde{U}_2||_2. \tag{6}$$

The procedure to derive an estimate for μ uses a combination of standard eigenvalue computation procedures applied to $A_{h_j}, \ j = 1, \ldots, 3$ and some extrapolation technique designed for scalar functions.

(6) is a good global a posteriori error estimator provided that

$$||U^0 - p_h(u)||_2 \ \ll \ ||U^0 - \tilde{U}_2||_2. \tag{7}$$

One way to test this hypothesis (7) is to measure the sensitivity of the upper bound (6) with respect to the choice of the fine grid $M(h_\infty)$. This is a feasible test because the fine grid solution is never computed in OES. Our verification procedure checks that $||U^0 - U_2||_2$ increases toward an asymptotic limit as $M(h_\infty)$ gets finer.

The algorithm procedure to construct V_e solution of P_α or $P_{\alpha,\beta}$ is straightforward when the operator is linear and the objective function is the discrete L_2 norm of the residual. Let $e_i, \ i = 1, \ldots, m$ be a set of basis function of $\Lambda(\Omega)$. The solution process can be decomposed into three steps.

• First, interpolation of the coarse grid solution from $M(h_j), \ j = 1, \ldots, p$ to $M(h_\infty)$, with $p = 2$ for the two level method, respectively 3 for the three level method.

• Second, evaluate the residual $R[e_i \ (\tilde{U}_j - \tilde{U}_{j+1})], \ i = 1, \ldots, m, \ j = 1, \ldots, p-1$, and $R[\tilde{U}_p]$ on the fine grid $M(h_\infty)$.

• Third, the solution of the least square linear algebra problem that has m unknowns for each weight coefficient α and β used in the extrapolation procedure. In practice, m is much lower than the number of grid points on any coarse grid used.

We have generalized the LSE method to non-linear elliptic problems via a Newton like loop [2, 3]. We have also obtained preliminary results for unsteady parabolic problems [7]. Most of this work has been done on solutions produced by our own code on a fairly large variety of linear and nonlinear PDE problems on structured grids. To apply these techniques on solution produced by commercial code that have thousands of lines, and work with unstructured grids requires a more general and abstract approach, that we present in the next section.

2.2 Solution Verification of Off-the-Shelf CFD Code

We propose to generalize our method here to steady, CFD solutions produced by existing code. The challenge is that in most commercial codes, one cannot rely on the exact knowledge of the discretization method, neither have access to any information on the internal structure of the code. What we propose is fundamentally different than existing methods. We describe in the following the main ideas without seeking an exact formal mathematical description of a given specific PDE problem.

Let $(E, ||.||_E)$ and $(F, ||.||_F)$ be two normed linear space, $G \in L(E, F)$ be the operator corresponding to the CFD problem. Further let us denote $S \in F$ the input data of the CFD code and $U \in E$ the solution we are looking for.

In practice we look for an approximation of the accuracy of the solution U_h on the mesh $M(h)$ produced by the code $\mathcal{C}$ that operates on the data S_h: $\mathcal{C} : \ S_h \ \rightarrow U_h$. The objective is still to get an error estimate versus a very fine grid solution U_∞ that is never computed, because the cost is prohibitive. We will skip the index h when it is not essential. The space $E, \ F$ have (very large) finite dimensions indeed when they are for the discrete solutions on $M(h_\infty)$, and discrete data S_{h_∞}.

We assume that the code $\mathcal{C}$ has a procedure that provides the residual, i.e $V \rightarrow \rho = G(U_h) - G(V)$, where $V \in E, \ \rho \in F$. We note that this hypothesis is realistic, since most of the commercial code offer this feature or either provides a (first order explicit) time stepping procedure:

$$\frac{U_h^{n+1} - U_h^n}{dt} = G(U_h^n) - S \,. \tag{8}$$

The residual is then $\rho = \frac{U_h^1 - U_h}{dt}$. We assume that the following problem

$$G(u) \ = \ s, \ \forall s \in B(S, d)$$

is well posed for $s \in B(S, d)$, where B is a ball of center S and diameter d in $(F, ||.||_F)$. There should exist a unique solution for all data in $B(S, d)$ and the dependency of the solution on these data is supposed to be smooth enough to use a second order Taylor expansion.

Let us suppose that $G(U_h) \in B(S, d)$, that is

$$||\rho||_F = ||G(U_h) - S||_F < d. \tag{9}$$

We would like to get an error estimate on $e = U_h - U_\infty = G^{-1}(U_h) - G^{-1}(U_\infty)$. A Taylor expansion writes

$$G^{-1}(S) = G^{-1}(S + \rho) - (\rho \cdot \nabla_s) G^{-1}(S + \rho) + \frac{1}{2}\rho \cdot [\rho \cdot R(S)] \tag{10}$$

$$\text{where } ||R(S)||_E \leq K = \sup_{s \in B(S,d)} ||\nabla_s^2 G^{-1}(s)||_E. \tag{11}$$

Therefore

$$||e||_E \ \leq \ ||\rho||_F \ (||\nabla_s G^{-1}(S + \rho)||_E \ + \ \frac{K}{2}||\rho||_F). \tag{12}$$

This completely general error estimate point out to two different tasks:

- Task 1: get an accurate upper bound on $||\nabla_S G^{-1}(S + \rho)||$
- Task 2: obtain a solution $U_\infty + e$ that gives a residual $||\rho||$ small enough to make the estimate useful, i.e. compatible with (9).

Task 2 is the purpose of the OES method, while Task 1 can be achieved by a perturbation method that can reuse the code.

2.3 Task 1: Stability Estimate

Let $\{b_i^E,\ i=1,\ldots,N\}$, (resp., $\{b_i^F,\ i=1,\ldots,N\}$) be a basis of E_h, (resp., F_h) and $\varepsilon \in \mathbb{R}$ such that $\varepsilon = o(1)$. Let $(V_i^{\mp})_{i=1,\ldots,N}$, be the family of solutions of the following problems: $G(U_h \mp \varepsilon V_i) = S + \rho \mp \varepsilon b_i$. We get from finite differences the approximation

$$C_{h_\infty} = ||\nabla_S G^{-1}(S+\rho)|| \approx ||(\frac{1}{2}(V_j^+ - V_j^-))_{j=1,\ldots,N}|| + O(\varepsilon^2).$$

We can get in as similar manner an approximation of the norm of the Hessian $\nabla_s^2 G^{-1}(S+\rho)$. For ρ small enough, we can verify that the upper bound is given essentially by:

$$||e||_E \ \preceq\ C_{h_\infty}||\rho||_F. \tag{13}$$

The column vectors $V_j^{\mp}$ can be computed with embarrassing parallelism. It is however unrealistic to compute these solutions that lies on the fine grid $M(h_\infty)$.

To make this task manageable, we have to reduce the dimension of the problem. We use the following two observations. While the solution of the CFD problem can be very much grid dependent, the conditioning number of the problem is in general much less sensitive to the grid. The idea is then to compute an approximation of C_{h_∞} by extrapolation from an estimate of two or three coarse grid computation of C_{h_j}. Further, let us assume that the fine grid $M(h_\infty)$ is a regular Cartesian grid. The number of terms to represent accurately the projected solution $\tilde{U}_j,\ j=1,\ldots,3$ with a spectral expansion or a wavelet approximation at a given accuracy is much less than the dimension of the coarse grid used in a Finite Element/Finite Volume computation. We propose to use preferably a grid M_{h_∞} that has enough regularity to allow a representation of the solution U_∞ with some form of compact representation, using either trigonometric expansion or wavelets.

The grid M_{h_∞} may have many more grid points than necessary, and therefore might not be computationally efficient for a true fine grid computation. But we do not have to do this computation anyway.

Let us denote $\hat{E}$ and $\hat{F}$ the spaces corresponding to one of these compact representation of the solution and residual. Let $(\hat{b}_j^{E/F}, j=1,\ldots,\hat{N})$, be the corresponding base with $\hat{N} \ll N$.Let $q_{E/F}$ be a mapping $E/F \rightarrow \hat{E}/\hat{F}$, respectively $q_{\hat{E}/\hat{F}}$ be a mapping $\hat{E}/\hat{F} \rightarrow E/F$ and let $\hat{C}:\ \hat{S}_h \rightarrow \hat{U}_h$. To summarize the procedure for Task 1, The estimate on C_{h_∞} will be applied to verify the code $\hat{C}$ based on the computation of $(\hat{V}_j^{\mp}, j=1,\ldots,\hat{N})$ vectors on the coarse grids $M(h_j),\ j=1,\ldots,3$ done by the code $\hat{C}$. We notice that the computation of the vector $\hat{V}_j^{\mp}$ can be done with embarrassing parallelism. Further because ε is small the code $\hat{C}$ can use as an initial guess in its iterative process the solution U_h that is hopefully very close to the unknown $\hat{U}_h \pm \hat{V}_j^{\mp}$.

2.4 Task 2: Optimized Extrapolation

We use here an optimized extrapolation method. To reduce the dimension of this problem we search for the unknown weight functions in a small space that can be described either by trigonometric expansion, or wavelet expansion, or possibly spectral elements. If Ω is the physical domain for the CFD solution, the unknown weight function can be search in a square domain $(0,1)^2$ modulo a change of variables. As a matter of fact no boundary conditions are required on the unknown weight functions. Let $\{\theta_j,\ j=1,\ldots,m\}$ be the set of basis function of $\Lambda(\Omega)$.

We look for the solution of the optimization problem in the two level case

$$\text{Find } (\alpha_j) \in \mathbb{R}^m, \text{such that}$$
$$||G([\sum_{j=1,\ldots,m} \alpha_j \Theta_j]\tilde{U_1} + [1 - \sum_{j=1,\ldots,m} \alpha_j \Theta_j]\tilde{U_2})||_F \text{ is minimum.} \qquad (14)$$

We have a similar formulation for the three level OES. Following the same argument than before we will rather look for this minimum in $\hat{F}$. As shown in [2, 3], we need a filtering process of the solution to have this minimization process numerically efficient. The postprocessing q_F is then useful. We can obtain easily the result when the weight function is a scalar function. To make this computation robust we use a response surface methodology [4] that is rather trivial in the scalar case. This procedure consist to compute a lower order polynomial best fit of the function $||G(\alpha\tilde{U_1} + (1-\alpha)\tilde{U_2})||$ by sampling α according to the expected convergence order range of the code. The minimization on α is then done with this polynomial approximation by a standard method. The sampling process is a cumbersome embarrassing parallel process that can take advantage of a computational grid [6].

3 A Numerical Example

To illustrate the pertinence of our methodology, let us present a Navier-Stokes back step flow example. The computation is done with ADINA. The ADINA system is a comprehensive finite element software that enable analysis of structures, fluid simulations, and fluid flows simulations with structural interactions.

Figure 1 shows an example of an unstructured mesh calculation of the back flow step problem at Reynolds number 500.

In this simulation, the number of elements are respectively 10347 on the fine grid G^{∞}, 1260 on the coarse grid G_1, and 2630 on the coarse grid G_2.

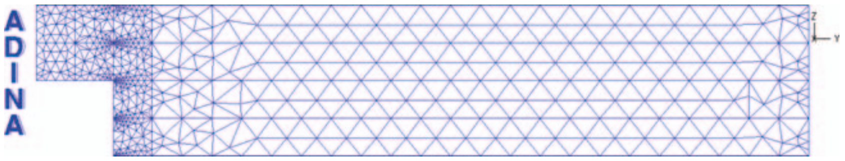


Fig. 1. Coarse mesh for the backstep

The steady solutions are obtained using a transient scheme for the incompressible Navier-Stokes equation.

For this test case OES outperforms the accuracy of the RE method by one order of magnitude - see Figure 2. An accurate error estimate is obtained for a representation of the solution on a 20 × 20 trigonometric expansion - see Figure 3. Let us conclude this paper with the design of the software that we are developing as a solution verification system independent of the CFD code.

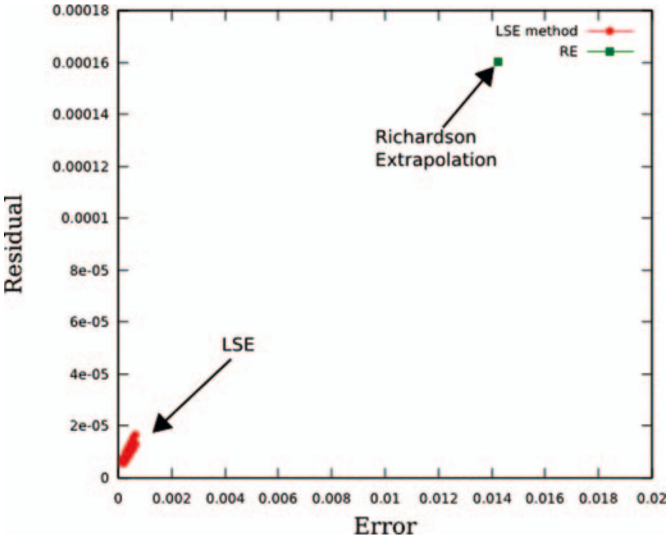


Fig. 2. Performance of LSE and Richardson Extrapolation.

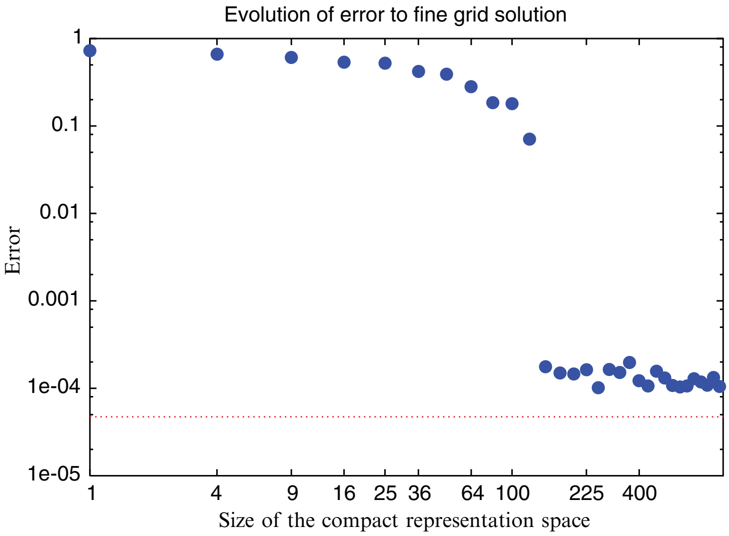


Fig. 3. Error estimate

4 Scientific Software Design and Conclusion

Our algorithm gives rise to a large set of cumbersome computations that can be done in parallel with a minimum of synchronization. This is a key feature to make our SV cost effective. We are developing a network oriented interface that allow our SV method to be executed remotely on several processing units, using the following methodology:

(i) a three-tier client server model architecture: it allows the system to be transparent, the user should not have to worry about technical details, to be open, each subsystem is open to interaction with the others, and to be scalable, the system should be easy to modify as the number of resources, users, softwares evolved.

(ii) Portability: to be able to run on UNIX/Linux/Windows platform

(iii) Security in data transfer, because industrial applications as well as computation on clinical data require that the data be protected.

(iv) Friendly user interface.

Some preliminary result on the performance of our distributed computing system for SV are reported in [6].

References

[1] M. Ainsworth. Identification and a posteriori estimation of transported errors in finite element analysis. *Comput. Methods Appl. Mech. Engrg.*, 176(1):3 – 18, 1999.

[2] M. Garbey and W. Shyy. A least square extrapolation method for improving solution accuracy of PDE computations. *J. Comput. Phys.*, 186:1 – 23, 2003.

[3] M. Garbey and W. Shyy. A least square extrapolation method for the a posteriori error estimate of the incompressible Navier-Stokes problem. *Internat. J. Numer. Methods Fluids*, 48:43 – 59, 2005.

[4] R.H. Myers and D.C. Montgomery. *Response Surface Methodology : Process and Product Optimization Using Designed Experiments.* Wiley, 2002.

[5] W.L. Oberkampf and T.G. Trucano. Verification and validation in computational fluid dynamics. Technical Report 2002-0529, Sandia, March 2002.

[6] C. Picard, M. Garbey, and V. Subramanian. Mapping LSE on a grid: Software archicteture and performance gains. In *Proceedings of the International Conference on Parallel Computational Fluid Dynamic*, 2005.

[7] C. Soize, editor. *A Least Square Extrapolation Method for the A Posteriori Error Estimate of CFD and Heat Transfer Problem*, 2005.

[8] R. Verfurth. *A Review of A Posteriori Estimation and Adaptive Mesh Refinement Techniques.* Wiley-Teubner, 1996.

A Robust Preconditioner for the Hessian System in Elliptic Optimal Control Problems

Etereldes Gonçalves[1], Tarek P. Mathew[1], Markus Sarkis[1,2], and Christian E. Schaerer[1]

[1] Instituto de Matemática Pura e Aplicada - IMPA, Estrada Dona Castorina 110, Rio de Janeiro, 22460-320, Brazil.

[2] Department of Mathematical Sciences, Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609, USA.

Summary. We consider an elliptic optimal control problem in two dimensions, in which the control variable corresponds to the Neumann data on a boundary segment, and where the performance functional is *regularized* to ensure that the problem is well posed. A finite element discretization of this control problem yields a saddle point linear system, which can be reduced to a symmetric positive definite Hessian system for determining the *control* variables. We formulate a robust preconditioner for this reduced Hessian system, as a matrix product involving the discrete Neumann to Dirichlet map and a mass matrix, and show that it yields a condition number bound which is uniform with respect to the mesh size and regularization parameters. On a uniform grid, this preconditioner can be implemented using a fast sine transform. Numerical tests verify the theoretical bounds.

1 Introduction

Elliptic control problems arise in various engineering applications [4]. We consider a problem in which the "control" variable $u(.)$ corresponds to the Neumann data on a boundary segment, and it must be chosen so that the solution $y(.)$ to the elliptic equation with Neumann data $u(.)$ closely matches a specified "target" function $\hat{y}(.)$. To determine the "optimal" control, we employ a performance functional which measures a square norm error between $\hat{y}(.)$ and the actual solution $y(.)$, and the control variable is sought so that it minimizes the performance functional [1, 3, 5, 4]. However, this results in an *ill-posed* constrained minimization problem, which can be *regularized* by adding a small Tikhonov regularization term to the performance functional. We discretize the regularized optimal control problem using a finite element method, and this yields a saddle point system [1, 2, 7].

In this paper, we formulate a robust preconditioner for the symmetric positive definite Hessian system for the control variables, obtained by block elimination of the saddle point system. In § 2, we formulate the elliptic optimal control problem and its discretization. In § 3, we derive the Hessian system and formulate our preconditioner as a symmetric matrix product involving the discrete Neumann to Dirichlet map and a mass matrix. We show that it yields a condition number bound that is independent of the mesh size and the regularization parameters. On a uniform grid, we describe a fast sine transform (FST) implementation of it. Numerical results are presented in § 4.

2 Optimal Control Problem

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain and let Γ be an edge of its boundary $\partial\Omega$. We consider the problem of determining a Neumann control data $u(\cdot)$ on Γ such that the solution $y(\cdot)$ to the following problem with forcing term $f(\cdot)$:

$$\begin{cases} -\Delta y(x) = f(x), & \text{in } \Omega \\ \frac{\partial y(x)}{\partial n} = u(x), & \text{on } \Gamma \\ y(x) = 0, & \text{on } \partial\Omega\backslash\Gamma \end{cases} \tag{1}$$

minimizes the following performance functional $J(y, u)$:

$$J(y,u) \equiv \frac{1}{2}\left(\|y-\hat{y}\|^2_{L^2(\Omega)} + \alpha_1\,\|u\|^2_{L^2(\Gamma)} + \alpha_2\,\|u\|^2_{H^{-1/2}(\Gamma)}\right), \tag{2}$$

where $\hat{y}(\cdot) \in L^2(\Omega)$ is a given target, and α_1, $\alpha_2 \geq 0$ denote *regularization* parameters. Later in the paper we also consider the case where $\|y-\hat{y}\|_{L^2(\Omega)}$ in (2) is replaced by $\|y-\hat{y}\|_{L^2(\Gamma)}$. The term $\|u\|_{H^{-1/2}(\Gamma)}$ denotes the dual Sobolev norm associated with $H^{1/2}_{00}(\Gamma)$. We let $H^1_D(\Omega)$ denote the subspace of $H^1(\Omega)$ consisting of functions vanishing on $D \equiv (\partial\Omega \setminus \Gamma)$.

To obtain a weak formulation of the minimization of (2) within set (1), we employ the function space $H^1_D(\Omega)$ for $y(\cdot)$ and $H^{-1/2}(\Gamma)$ for $u(\cdot)$. Given $f \in L^2(\Omega)$, define the constraint set $\mathcal{V}_f \subset \mathcal{V} \equiv H^1_D(\Omega) \times H^{-1/2}(\Gamma)$:

$$\mathcal{V}_f \equiv \left\{(y,u) \in \mathcal{V} \,:\, \mathcal{A}(y,w) = (f,w) + <u,w>, \;\; \forall w \in H^1_D(\Omega)\right\}, \tag{3}$$

where the forms are defined by:

$$\begin{cases} \mathcal{A}(y,w) \equiv \int_\Omega \nabla y \cdot \nabla w \, dx, & \text{for } y, w \in H^1_D(\Omega) \\ (f,w) \equiv \int_\Omega f(x)\, w(x)\, dx, & \text{for } w \in H^1_D(\Omega) \\ <u,w> \equiv \int_\Gamma u(x)\, w(x)\, ds_x, & \text{for } u \in H^{-1/2}(\Gamma),\; w \in H^{1/2}_{00}(\Gamma). \end{cases} \tag{4}$$

The constrained minimization problem then seeks $(y_*, u_*) \in \mathcal{V}_f$ satisfying:

$$J(y_*, u_*) = \min_{(y,u) \in \mathcal{V}_f} J(y,u). \tag{5}$$

To obtain a saddle point formulation of (5), introduce $p(\cdot) \in H^1_D(\Omega)$ as a Lagrange multiplier function to enforce the constraints. Define the following Lagrangian functional $\mathcal{L}(\cdot,\cdot,\cdot)$:

$$\mathcal{L}(y,u,p) \equiv J(y,u) + \left(\mathcal{A}(y,p) - (f,p) - <u,p>\right), \tag{6}$$

for $(y, u, p) \in H^1_D(\Omega) \times H^{-1/2}(\Gamma) \times H^1_D(\Omega)$. Then, the constrained minimum (y_*, u_*) of $J(.,.)$ can be obtained from the saddle point (y_*, u_*, p_*) of $\mathcal{L}(\cdot,\cdot,\cdot)$, where $(y_*, u_*, p_*) \in H^1_D(\Omega) \times H^{-1/2}(\Gamma) \times H^1_D(\Omega)$ satisfies:

$$\sup_q \mathcal{L}(y_*, u_*, q) = \mathcal{L}(y_*, u_*, p_*) = \inf_{(y,u)} \mathcal{L}(y, u, p_*). \tag{7}$$

For a discussion on the well-posedness of problem (7), see [5, 4].

To obtain a finite element discretization of (5), choose a quasi-uniform triangulation $\tau_h(\Omega)$ of Ω. Let $V_h(\Omega) \subset H^1_D(\Omega)$ denote the $\mathbb{P}_1$-conforming finite element space associated with the triangulation $\tau_h(\Omega)$, and let $V_h(\Gamma) \subset L^2(\Gamma)$ denote its restriction to Γ. A finite element discretization of (5) will seek $(y^*_h, u^*_h) \in V_h(\Omega) \times V_h(\Gamma)$ such that:

$$J(y^*_h, u^*_h) = \min_{(y_h, u_h) \in \mathcal{V}_{h,f}} J(y_h, u_h) \tag{8}$$

where the discrete constraint space $\mathcal{V}_{h,f} \subset \mathcal{V}_h \equiv V_h(\Omega) \times V_h(\Gamma)$ is defined by:

$$\mathcal{V}_{h,f} = \{(y_h, u_h) \in \mathcal{V}_h \;:\; \mathcal{A}(y_h, w_h) = (f, w_h) + < u_h, w_h >, \;\; \forall w_h \in V_h(\Omega)\}.$$

Let $p_h \in V_h(\Omega)$ denote discrete Lagrange multiplier variables, and let $\{\phi_1(x), \ldots, \phi_n(x)\}$ and $\{\psi_1(x), \ldots, \psi_m(x)\}$ denote the standard nodal basis functions for $V_h(\Omega)$ and $V_h(\Gamma)$, respectively. Expanding y_h, u_h and p_h with respect to its finite element basis, yields:

$$y_h(x) = \sum_{i=1}^n \mathbf{y}_i \, \phi_i(x), \quad u_h(x) = \sum_{j=1}^m \mathbf{u}_i \, \psi_i(x), \quad p_h(x) = \sum_{l=1}^n \mathbf{p}_l \, \phi_l(x), \tag{9}$$

and seeking the discrete saddle point of $\mathcal{L}(\cdot,\cdot,\cdot)$, yields the linear system:

$$\begin{bmatrix} M_\Omega & 0 & A^T \\ 0 & G & B^T \\ A & B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{u} \\ \mathbf{p} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{f}_3 \end{bmatrix}, \tag{10}$$

where the sub-matrices M_Ω, A and Q (to be used later), are defined by:

$$\begin{cases} (M_\Omega)_{ij} \equiv \int_\Omega \phi_i(x)\, \phi_j(x)\, dx, & \text{for } 1 \le i, j \le n \\ (A)_{ij} \equiv \int_\Omega \nabla\phi_i(x) \cdot \nabla\phi_j(x)\, dx, & \text{for } 1 \le i, j \le n \\ (Q)_{ij} \equiv \int_\Gamma \psi_i(x)\, \psi_j(x)\, ds_x, & \text{for } 1 \le i, j \le m, \end{cases} \tag{11}$$

and the forcing vectors are defined by $(\mathbf{f}_1)_i = \int_\Omega \hat{y}(x)\, \phi_i(x)\, dx$, for $1 \le i \le n$ with $\mathbf{f}_2 = \mathbf{0}$, and $(\mathbf{f}_3)_i = \int_\Omega f(x)\, \phi_i(x)\, dx$ for $1 \le i \le n$. Matrix M_Ω of dimension n corresponds to a mass matrix on Ω, and matrix A to the stiffness matrix. Matrix Q of dimension m corresponds to a lower dimensional mass matrix on Γ. Matrix B will be defined in terms of Q, based on an ordering of nodal unknowns in $\mathbf{y}$ and $\mathbf{p}$ with nodes in the *interior* of Ω ordered prior to the nodes on Γ. Denote such block partitioned vectors as $\mathbf{y} = \left(\mathbf{y}_I^T, \mathbf{y}_B^T\right)^T$ and $\mathbf{p} = \left(\mathbf{p}_I^T, \mathbf{p}_B^T\right)^T$, and define B of dimension $n \times m$ as $B^T = \begin{bmatrix} 0 & Q^T \end{bmatrix}$, and define matrix G of dimension m, representing the regularizing terms as:

$$G \equiv \alpha_1 Q + \alpha_2 \left(B^T A^{-1} B\right). \tag{12}$$

3 Preconditioned Hessian System

The algorithm we shall consider for solving (10) will be based on the solution of the following *Hessian system* for the discrete control $\mathbf{u}$. It is the Schur complement system obtained by block elimination of $\mathbf{y}$ and $\mathbf{p}$ in system (10):

$$\left(G + B^T A^{-T} M_\Omega A^{-1} B\right) \mathbf{u} = \mathbf{f}_2 - B^T A^{-T} \mathbf{f}_1 + B^T A^{-T} M_\Omega A^{-1} \mathbf{f}_3. \tag{13}$$

The Hessian matrix $H \equiv \left(G + B^T A^{-T} M_\Omega A^{-1} B\right)$ is symmetric and positive definite of dimension m, and system (13) can be solved using a PCG algorithm. Each matrix vector product with $G + B^T A^{-T} M_\Omega A^{-1} B$ will require the action of A^{-1} twice per iteration (this can be computed iteratively, resulting in a *double iteration*). Once $\mathbf{u}$ has been determined, we obtain $\mathbf{y} = A^{-1}\left(\mathbf{f}_3 - B\mathbf{u}\right)$ and $\mathbf{p} = A^{-T}\left(\mathbf{f}_1 - M_\Omega A^{-1}\mathbf{f}_3 + M_\Omega A^{-1} B\mathbf{u}\right)$.

The task of finding an effective preconditioner for the Hessian matrix H is complicated by the presence of the parameters $\alpha_1 \geq 0$ and $\alpha_2 \geq 0$. As noted in [5], when α_1 or α_2 is large (or equivalently, when $\lambda_{\min}(G)$ is sufficiently large), then G is spectrally equivalent to H and therefore G will be an effective preconditioner for H, while when both α_1 and α_2 are small (or equivalently, when $\lambda_{\max}(G)$ is sufficiently small), then the matrix $(B^T A^{-T} M_\Omega A^{-1} B)$ will be an effective preconditioner for H. For intermediate values of α_i, however, neither limiting approximation may be effective. In the special case when we replace $\|y - \hat{y}\|_{L^2(\Omega)}$ in (2) by $\|y - \hat{y}\|_{L^2(\Gamma)}$, then matrix M_Ω is replaced by $M_\Gamma \equiv \text{blockdiag}(0, Q)$ and we shall indicate a preconditioner for H, uniformly effective with respect to $\alpha_1 > 0$ or $\alpha_2 > 0$.

The preconditioner we shall formulate for H will be based on spectrally equivalent representations of G and $(B^T A^{-T} M A^{-1} B)$, for special choices of the matrix M. Lemma 1 below describes uniform spectral equivalences between G, $(B^T A^{-1} B)$, $(B^T A^{-T} M_\Omega A^{-1} B)$ and one or more of the matrices Q and S^{-1}, where $S = \left(A_{\Gamma\Gamma} - A_{I\Gamma}^T A_{II}^{-1} A_{I\Gamma}\right)$ denotes the discrete Dirichlet to Neumann map. Properties of S have been studied extensively in the domain decomposition literature [8].

Lemma 1. *Let $\Omega \subset \mathbb{R}^2$ be a convex domain. Then, the following equivalences:*

$$\begin{aligned} (B^T A^{-1} B) &= Q\, S^{-1}\, Q \\ (B^T A^{-T} M A^{-1} B) &= Q\, S^{-1}\, Q\, S^{-1}\, Q \qquad \text{when } M = M_\Gamma \\ (B^T A^{-T} M A^{-1} B) &\asymp Q\, S^{-1}\, Q\, S^{-1}\, Q\, S^{-1}\, Q \text{ when } M = M_\Omega, \end{aligned} \tag{14}$$

will hold with constants independent of h, where $S = (A_{\Gamma\Gamma} - A_{I\Gamma}^T A_{II}^{-1} A_{I\Gamma})$, $M_\Gamma =$ blockdiag$(0, Q)$ and M_Ω is the mass matrix on Ω.

Proof. The first statement is a trivial calculation. To prove the second, use:

$$A^{-1} = \begin{bmatrix} A_{II}^{-1} + A_{II}^{-1} A_{I\Gamma} S^{-1} A_{I\Gamma}^T A_{II}^{-1} & -A_{II}^{-1} A_{I\Gamma} S^{-1} \\ -S^{-1} A_{I\Gamma}^T A_{II}^{-1} & S^{-1} \end{bmatrix}.$$

Employing this and using the block matrix structure of B yields:

$$A^{-1} B\mathbf{u} = \begin{bmatrix} -A_{II}^{-1} A_{I\Gamma} S^{-1} Q\mathbf{u} \\ S^{-1} Q\mathbf{u} \end{bmatrix}.$$

Substituting this expression yields that $B^T A^{-T} M_\Gamma A^{-1} B = QS^{-1}QS^{-1}Q$. To prove the third equivalence, let u_h denote a finite element control function defined on Γ with associated nodal vector $\mathbf{u}$. Let v_h denote the Dirichlet data associated with the Neumann data u_h, i.e. with associated nodal vector $\mathbf{v} = S^{-1}Q\mathbf{u}$. When $M = M_\Omega$, then $\mathbf{u}^T(B^T A^{-T} M A^{-1} B)\mathbf{u}$ will be equivalent to $\|Ev_h\|^2_{L^2(\Omega)}$, where Ev_h denotes the *discrete harmonic* extension of the Dirichlet boundary data v_h into Ω with associated nodal vector $A^{-1}B\mathbf{u}$. When Ω is convex, $H^2(\Omega)$ elliptic regularity will hold for (1) and a result from [6] shows that $\|Ev_h\|^2_{L^2(\Omega)}$ is spectrally equivalent to $\|v_h\|^2_{H^{-1/2}(\Gamma)}$. In matrix terms, the nodal vector associated with the discrete Dirichlet data v_h will be $\mathbf{v} = S^{-1}Q\mathbf{u}$, given by the discrete Neumann to Dirichlet map. For $v_h \in H^{-1/2}(\Gamma)$, it will hold that $\|v_h\|^2_{H^{-1/2}(\Gamma)}$ is spectrally equivalent to $\mathbf{v}^T Q^T S^{-1} Q \mathbf{v}$, and in turn equivalent to $\mathbf{u}^T Q^T S^{-1} Q^T S^{-1} Q S^{-1} Q \mathbf{u}$ and the third equivalence follows, since $Q^T = Q$ and $S^{-T} = S^{-1}$.

As a consequence, we obtain the following uniform spectral equivalences.

Lemma 2. *Let $\Omega \subset R^2$ be a convex domain. Then, the following equivalences will hold for the Hessian matrix $H \equiv \left(G + B^T A^{-T} M A^{-1} B\right)$:*

$$\begin{array}{ll} H = H_0 \equiv \alpha_1 Q + \alpha_2 Q S^{-1} Q + Q S^{-1} Q S^{-1} Q, & \text{when } M = M_\Gamma \\ H \asymp H_0 \equiv \alpha_1 Q + \alpha_2 Q S^{-1} Q + Q S^{-1} Q S^{-1} Q S^{-1} Q, & \text{when } M = M_\Omega, \end{array} \tag{15}$$

with constants independent of h, α_1 and α_2.

H_0 will be our model preconditioner for H. To obtain an efficient solver for H_0, in applications we shall replace Q and S by $Q_0 \asymp Q$ and $S_0 \asymp S$. However, since a product of matrices is involved, caution must be exercised in the choice of Q_0 and S_0. Bounds independent of h and α_i will be retained only under additional regularity assumptions or the commutativity of Q, S, Q_0 and S_0.

3.1 An FST Based Preconditioner $\tilde{H} \asymp H_0$ for H

If $\Omega \subset \mathbb{R}^2$ is rectangular and the grid is uniform, and Γ is one of the four edges forming $\partial\Omega$, then the Dirichlet to Neumann map S (hence S^{-1}) and the mass matrix Q will be diagonalized by the discrete Sine Transform F, where:

$$(F)_{ij} = \sqrt{\frac{2}{m+1}}\,\sin(\frac{i\,j\,\pi}{m+1}) \quad \text{for} \quad 1 \le i,\, j \le m,$$

see [8]. Regularity theory shows that the Dirichlet to Neumann map S satisfies $S \asymp S_0 \equiv Q^{1/2}\left(Q^{-1/2} L Q^{-1/2}\right)^{1/2} Q^{1/2} \asymp \|\cdot\|^2_{H^{1/2}_{00}(\Gamma)}$, where L denotes a discretization of the *Laplace-Beltrami* operator $L_B = -\frac{d^2}{ds_x^2}$ on Γ with homogeneous Dirichlet conditions, see [8]. For a uniform grid, the Laplace-Beltrami matrix is $L = h^{-1}\,\mathrm{tridiag}(-1, 2, -1)$, and it is diagonalized by the sine transform F with $L = F\Lambda_L F^T$, where the diagonal matrix Λ_L has entries $\Lambda_L(ii) = 4\,(m+1)\,\sin^2(\frac{i\,\pi}{2\,(m+1)})$. For a uniform grid, the mass matrix satisfies $Q = Q_0 \equiv \frac{h}{6}\,\mathrm{tridiag}(1, 4, 1)$ and it is also diagonalized by F, satisfying $Q_0 = F\Lambda_{Q_0}F^T$ for $\Lambda_{Q_0}(ii) = \frac{1}{3\,(m+1)}\,(3 - 2\,\sin^2(\frac{i\,\pi}{2\,(m+1)}))$. Thus, we obtain:

$$\begin{aligned} S \asymp S_0 &\equiv F\Lambda_{S_0}F^T = F\left(\Lambda_{Q_0}^{1/4}\Lambda_L^{1/2}\Lambda_{Q_0}^{1/4}\right)F^T \\ Q = Q_0 &= F\Lambda_{Q_0}F^T . \end{aligned}$$

Since matrices S, Q, S_0 and Q_0 are diagonalized by F on a uniform grid, these matrices *commute*. As a result, it can be verified that $\tilde{H} \asymp H_0 \asymp H$:

$$\begin{aligned} \tilde{H} &\asymp F\left(\alpha_1\,\Lambda_{Q_0} + \alpha_2\,\Lambda_{Q_0}^2\Lambda_S^{-1} + \Lambda_{Q_0}^3\,\Lambda_S^{-2}\right)F^T, \text{ when } M = M_\Gamma \\ \tilde{H} &\asymp F\left(\alpha_1\,\Lambda_{Q_0} + \alpha_2\,\Lambda_{Q_0}^2\,\Lambda_S^{-1} + \Lambda_{Q_0}^4\,\Lambda_S^{-3}\right)F^T, \text{ when } M = M_\Omega, \end{aligned} \tag{16}$$

with bounds independent of h and α_i. The eigenvalues of $\tilde{H}^{-1}$ can be found analytically, and the action of $\tilde{H}^{-1}$ can be computed at low cost using FST's.

4 Numerical Experiments

We present numerical tests of control problem (2) on the two-dimensional unit square $(0,1)\times(0,1)$. Neumann conditions are imposed on $\Gamma = (0,1)\times\{0\}$, and homogeneous Dirichlet conditions are imposed on the remaining sides of $\partial\Omega$, with forcing term $f(x,y) = 0$ in Ω. We consider a structured triangulation on Ω with mesh parameter $h = 2^{-N}$, where N is an integer denoting the number of refinements. We test different values for the relaxation parameters α_1 and α_2, for the mesh size h, and for mass matrix M. In all numerical experiments, we run PCG until the preconditioned l_2 initial residual is reduced by a factor of 10^{-9}. We use the FST based preconditioner described in (16).

Table 1. Number of PCG iterations and (condition) for $\alpha_2 = 0$ and $M = M_\Omega$.

$N \setminus \alpha_1$	1	$(0.1)^2$	$(0.1)^4$	$(0.1)^6$	0
3	3 (1.02)	5 (1.65)	7 (1.60)	6 (1.44)	7 (1.54)
4	3 (1.02)	5 (1.63)	9 (1.95)	6 (1.29)	7 (1.56)
5	3 (1.02)	5 (1.63)	8 (2.00)	7 (1.50)	7 (1.56)
6	3 (1.02)	5 (1.64)	8 (2.01)	6 (1.86)	6 (1.55)
7	3 (1.02)	5 (1.64)	8 (2.00)	6 (1.96)	5 (1.51)

Tables 1 and 2 list results on runs with $M = M_\Omega$ and target function $\hat{y}(x,y) = 1$ on $[1/4, 3/4] \times [0, 3/4]$ and equal to zero otherwise. We list the number of PCG iterations and in parenthesis the condition number estimate for the preconditioned system. As expected from the analysis, the number of iterations and the condition number remain bounded, and when no preconditioning is used, the problem becomes very ill-conditioned for small regularization α_i; see Table 3. In Tables 4 and 5 we report the results for $M = M_\Gamma$ with target function $\hat{y}(x,0) = 1$ on $[1/4, 3/4] \times \{0\}$, and equal to zero otherwise. As before, the number of iterations and the condition number remain bounded.

Table 2. Number of PCG iterations and (condition) for $\alpha_1 = 0$ and $M = M_\Omega$.

$N \setminus \alpha_2$	1	$(0.1)^2$	$(0.1)^4$	$(0.1)^6$	0
3	7 (2.15)	6 (1.45)	7 (1.50)	7 (1.53)	7 (1.54)
4	8 (2.26)	7 (1.71)	7 (1.45)	7 (1.56)	7 (1.56)
5	7 (2.24)	7 (1.84)	6 (1.32)	7 (1.56)	7 (1.56)
6	5 (2.03)	7 (1.95)	5 (1.33)	6 (1.52)	6 (1.55)
7	4 (1.82)	6 (1.76)	5 (1.40)	5 (1.44)	5 (1.51)

Table 3. Number of CG iterations and (condition) for $\alpha_2 = 0$ and $M = M_\Omega$.

$N \setminus \alpha_1$	1	$(0.1)^2$	$(0.1)^4$	$(0.1)^6$	0
3	7 (2.75)	7 (6.80)	8 (351)	8 (2.2+3)	8 (2.3+3)
4	9 (2.97)	9 (7.47)	15 (448)	23 (1.6+4)	23 (2.4+4)
5	7 (3.03)	8 (7.64)	16 (468)	35 (3.8+4)	53 (2.0+5)
6	6 (3.04)	6 (7.69)	12 (472)	39 (4.6+4)	106 (1.6+6)
7	4 (3.05)	5 (7.70)	11 (473)	34 (4.7+4)	162 (1.3+7)

Table 4. Number of PCG iterations and (condition) for $\alpha_2 = 0$ and $M = M_\Gamma$.

$N \setminus \alpha_1$	1	$(0.1)^2$	$(0.1)^4$	$(0.1)^6$	0
3	3 (1.01)	4 (1.17)	4 (3.96)	4 (5.08)	4 (5.09)
4	2 (1.00)	4 (1.07)	7 (2.73)	8 (5.64)	8 (5.72)
5	2 (1.00)	3 (1.02)	7 (1.76)	11 (5.44)	11 (5.75)
6	2 (1.00)	3 (1.00)	5 (1.29)	12 (4.69)	13 (5.78)
7	2 (1.00)	3 (1.01)	4 (1.10)	8 (3.14)	10 (5.65)

Table 5. Number of PCG iterations and (condition) for $\alpha_1 = 0$ and $M = M_\Gamma$.

$N \setminus \alpha_2$	1	$(0.1)^2$	$(0.1)^4$	$(0.1)^6$	0
3	4 (2.29)	4 (3.99)	4 (5.08)	4 (5.09)	4 (5.09)
4	8 (2.41)	8 (3.81)	8 (5.68)	8 (5.72)	8 (5.72)
5	8 (2.37)	9 (3.25)	11 (5.66)	11 (5.75)	11 (5.75)
6	7 (2.33)	8 (2.84)	12 (5.57)	13 (5.78)	13 (5.78)
7	5 (2.09)	6 (2.45)	9 (5.24)	10 (5.64)	10 (5.65)

5 Conclusions

We have introduced a robust preconditioner for the Hessian matrix in a class of elliptic optimal control problems. We have shown that the Hessian matrix is spectrally equivalent to a composition of the discrete *Laplace-Beltrami* and mass matrices.

For a uniform grid, these matrices are simultaneously diagonalized by a fast sine transform. The resulting preconditioner is optimal with respect to the mesh size and relaxation parameters. Numerical results confirm the robustness of the preconditioner.

References

[1] G. Biros and O. Gattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. I. The Krylov-Schur solver. *SIAM J. Sci. Comput.*, 27(2):687–713, 2005.

[2] E. Haber and U.M. Ascher. Preconditioned all-at-once methods for large, sparse parameter estimation problems. *Inverse Problems*, 17(6):1847–1864, 2001.

[3] M. Heinkenschloss and H. Nguyen. Neumann-Neumann domain decomposition preconditioners for linear-quadratic elliptic optimal control problems. *SIAM J. Sci. Comput.*, 28(3):1001–1028, 2006.

[4] H.L. Lions. *Some Methods in the Mathematical Analysis of Systems and Their Control.* Gordon and Breach Science, New York, 1981.

[5] T. Mathew, M. Sarkis, and C.E. Schaerer. Block matrix preconditioners foir elliptic optimal control problems. *Numer. Linear Algebra Appl.*, 2006. To appear.

[6] P. Peisker. On the numerical solution of the first biharmonic equation. *RAIRO Modél. Math. Anal. Numér.*, 22(4):655–676, 1988.

[7] E. Prudencio, R. Byrd, and X. Cai. Parallel full space SQP Lagrange-Newton-Krylov-Schwarz algorithms for PDE-constrained optimization problems. *SIAM J. Sci. Comput.*, 27(4):1305–1328, 2006.

[8] A. Toselli and O.B. Widlund. *Domain Decomposition Methods—Algorithms and Theory.* Spinger-Verlag, 2005.

A Schur Complement Method for DAE/ODE Systems in Multi-Domain Mechanical Design

David Guibert[1] and Damien Tromeur-Dervout[1,2*]

[1] CDCSP - ICJ UMR5208/CNRS Université Lyon 1 F-69622 Villeurbanne Cédex
[2] INRIA/IRISA/Sage Campus de Beaulieu F-35042 Rennes Cédex
{dguibert,dtromeur}@cdcsp.univ-lyon1.fr

Summary. The large increase of unknowns in multi-domain mechanical modeling leads to investigate new parallel implementation of ODE and DAE systems. Unlike space domain decomposition, no geometrical information is given to decompose the system. The connection between unknowns have to be built to decompose the system in subsystems. A Schur Complement DDM can then be applied. During some time steps, the Jacobian matrix can be frozen allowing to speed-up the Krylov solvers convergence by projecting onto the Krylov subspace. This kind of DAE are stiff and the numerical procedure needs special care.

1 Introduction

Problems coming from the multi-domain mechanical design lead to solve systems of Ordinary Differential Equations (ODE) or Differential Algebraic Equations (DAE). Designers of mechanical systems want to achieve realistic modeling, taking into account more and more physics. Mathematically speaking, these features lead to have DAE/ODE systems with a large number of unknowns. Moreover these systems are usually stiff and eventually exhibits some discontinuities. So robust solvers with adaptive time stepping strategy must be designed.

Up to now, the main approach to obtain an ODE system parallel solver uses the parallelizing "across the method" of the time integrator. Runge-Kutta methods involve several stages. The aim of this kind of parallelization is to compute each stage of the method on a dedicated processor ([2, 1, 9, 3]).

This kind of parallelization is very limited. The number of processors involved can only be equal to the number of stages of the method.

We have shown in ([3]) that a speed-up nearby 2.5 can be obtained on 3 processors using Radau IIa method (a 3-stage RK method, see [4, 5] for more details).

In this paper, we propose a new approach based on the Schur Complement Method in order to decompose the system into smaller systems. In the Partial Differential Equation framework, the decomposition is given by the geometrical data

[*] Funded: National Research Agency: technologie logicielle 2006-2009 PARADE Project Mathgrid and cluster ISLES project HPC Région Rhône-Alpes

and the order of discretization scheme. Conversely in the DAE/ODE framework, no a priori knowledge of the coupled variables is available. This is the main issue to be solved.

We will show in section 2 how a Schur complement method can be implemented in the resolution of an ODE system. A brief description of the LSODA integrator will be given. Then in section 3, a strategy is applied to extract automatically the dependencies between the variables. These dependencies are viewed as an adjacency matrix and then, as in spatial domain decomposition, classical partitioning tools can be used. The algorithm is explained in section 4 and some numerical results are shown in section 5.

2 Differential Integrators

An initial value problem is considered,

for an ODE

$$(P_{ODE}) \begin{cases} \dfrac{dy}{dt} = f(t, y(t)), \\ y(t_0) = y_0, \end{cases} \tag{1}$$

or for a DAE

$$(P_{DAE}) \begin{cases} F(t, y(t), y'(t)) = 0, \\ y(t_0) = y_0, \\ y'(t_0) = y'_0. \end{cases} \tag{2}$$

The problem is assumed to be stiff. To solve the problem (P), a "predictor-corrector" scheme may be used. The main idea of such solver is to build a prediction $y_{n(0)}$ of the solution from a polynomial fit. Then the prediction is corrected by solving a nonlinear system

$$G_{ODE}(y_n) = y_n - \beta_0 h_n f(t_n, y_n) - \sum_{i>0}^{k} \alpha_{n,i} y_{n-i} = 0, \tag{3}$$

$$G_{DAE}(y_n) = F\left(t_n, y_n, h_n^{-1} \sum_{i=0}^{q} \alpha_{n,i} y_{n-i}\right) = 0, \tag{4}$$

where β_0 is a constant given by the integration scheme and h_n the current time step and $\alpha_{n,i}$ are the parameters of the method in use.

This means that the solution y_n at the time t_n is computed as follows.

- A predicted value $y_{n(0)}$ is computed to be used as initial guess for the nonlinear iteration (where $\alpha_{n,i}^p$ and β_0^p are parameters of the prediction formula):

$$y_{n(0)} = \sum_{i=1}^{k} \alpha_i^p y_{n-i} + \beta_0^p h_n \frac{dy_{n-1}}{dt}. \tag{5}$$

- Correction step (by Newton iterations)

$$\begin{cases} \left(\frac{\partial G}{\partial y}(y_{n(m)})\right) \delta y_n = -G(y_{n(m)}), \\ y_{n(m+1)} = y_{n(m)} + \delta y_n \end{cases} \tag{6}$$

with

$$\frac{\partial G_{ODE}}{\partial y} = I - \gamma \frac{\partial f}{\partial y} = I - \gamma J, \tag{7}$$

$$\frac{\partial G_{DAE}}{\partial y} = \frac{\partial F}{\partial y} + \alpha \frac{\partial F}{\partial y'} = J, \tag{8}$$

where J is the Jacobian matrix, $\gamma = \beta_0 h_n$ and $\alpha = \alpha_{n,0} h_n^{-1}$. At the end of the iteration process $y_n = y_{n(m+1)}$.

We want to apply the Schur complement method to this linear system. In domain decomposition in space, regular data dependencies are inherent to the spatial discretization scheme, which enables a relatively easy introduction of the Schur complement. The interface nodes are on the physical junction between the subdomains. In DAE/ODE, there is no regular data dependencies (even by renumbering locally the unknowns). In the considered problems, the coupling are embedded in the whole function f which is not —necessarily— explicitly known. Indeed, the function f is composed by several sub-models which are sometimes hidden (too complex, or used as black boxes). Hence a decomposition of the matrix is far from trivial to implement.

3 Automatic Partitioning of DAE/ODE Systems

In this section, we propose a method to partition *automatically* the unknowns of the dynamic system. We assume that the function f is composed by some subfunctions f_i seen as black boxes. This means that for a function f_i only its outputs and inputs are known. Let us illustrate this on the following example

$$\begin{cases} \frac{dy_1}{dt} = f_1(y_1, y_2, y_4), \\ \frac{dy_2}{dt} = f_2(y_1, y_3), \\ \frac{dy_3}{dt} = f_3(y_3, y_4), \\ \frac{dy_4}{dt} = f_4(y_3, y_4). \end{cases} \tag{9}$$

A modification of the variables y_1 and/or y_3 may change the value of the output of f_2, the time derivative of y_2.

The coupling between the variables and their derivatives can be summarized into an incidence matrix (see the graph theory for example in [6])

$$\begin{pmatrix} 1\,1\,0\,1 \\ 1\,0\,1\,0 \\ 0\,0\,1\,1 \\ 0\,0\,1\,1 \end{pmatrix}. \tag{10}$$

The value 1 can be viewed as a dependence between two nodes in the computation of one column of the Jacobian matrix.

Having this pattern, we know that in graph theory formulation, the reduction of the coupling between the nodes of a graph is done by minimizing of the number of edges cut in the graph. A graph partitioning tool such as [7] is used.

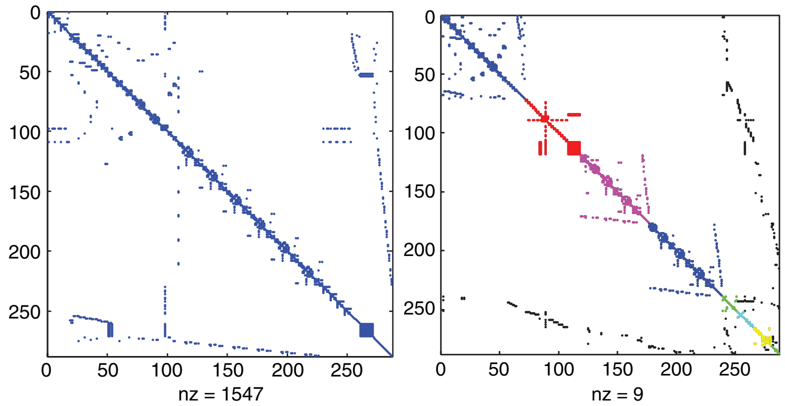


Fig. 1. Example of the Jacobian matrix of a V10 engine pump problem. Initial pattern on top and the pattern using 4 partitions

4 Algorithm

Now we concentrate on the algorithm to solve the linear system. The first step was the construction of the pattern of the Jacobian matrix (i.e. the incidence matrix corresponding to the interaction between the variables). The use of a graph partitioning tool decouples the system in sub-systems, separating the variables in internal variables and interface variables (those that need the values of variables belonging to another subdomain).

Given a partition, consider a doubly bordered block diagonal form of a matrix $A = I - \gamma J$ or $A = J$ (provided by the integrator)

$$A = \begin{pmatrix} B_1 & & & F_1 & \cdots & 0 \\ & \ddots & & \vdots & \ddots & \vdots \\ & & B_N & 0 & \cdots & F_N \\ E_1 & & & C_{11} & \cdots & C_{1N} \\ & \ddots & & \vdots & \ddots & \vdots \\ & & E_N & C_{N1} & \cdots & C_{NN} \end{pmatrix} = \begin{pmatrix} B & F \\ E & C \end{pmatrix}. \quad (11)$$

Locally on each subdomain one has to solve:

$$\begin{pmatrix} B_i & F_i \\ E_i & C_{ii} \end{pmatrix} \begin{pmatrix} x_i \\ y_i \end{pmatrix} + \begin{pmatrix} 0 \\ \sum_{j \neq i} C_{ij} y_j \end{pmatrix} = \begin{pmatrix} f_i \\ g_i \end{pmatrix}. \quad (12)$$

We assume that B_i is not singular. Then

$$x_i = B_i^{-1}(f_i - F_i y_i). \quad (13)$$

Upon substituting a reduced system is obtained:

$$S_i y_i + \sum_{j \neq i} C_{ij} y_j = g_i - E_i B_i^{-1} f_i \text{ with } S_i = C_{ii} - E_i B_i^{-1} F_i. \quad (14)$$

Multiplying by S_i^{-1}, one obtain the following preconditioned reduced system for the interface

$$\begin{pmatrix} I & S_1^{-1}C_{12} & \cdots & S_1^{-1}C_{1N} \\ S_2^{-1}C_{21} & I & \cdots & S_2^{-1}C_{2N} \\ \vdots & & \ddots & \vdots \\ S_N^{-1}C_{N1} & \cdots & S_N^{-1}C_{NN-1} & I \end{pmatrix} \begin{pmatrix} y_1 \\ \vdots \\ y_N \end{pmatrix} = \begin{pmatrix} \hat{g_1} \\ \vdots \\ \hat{g_N} \end{pmatrix}. \tag{15}$$

A solution method involves four steps:

- Obtain the right hand side of the preconditioned reduced system

$$\hat{g_i} = S_i^{-1} \left(g_i - E_i B_i^{-1} f_i \right). \tag{16}$$

- "Form" the Schur complement matrix.
  - A LU decomposition of the matrix without pivoting gives the LU decomposition of the matrix S_i

$$\begin{pmatrix} B_i & F_i \\ E_i & C_{ii} \end{pmatrix} = \begin{pmatrix} L_{B_i} & 0 \\ E_i U_{B_i}^{-1} & L_{S_i} \end{pmatrix} \begin{pmatrix} U_{B_i} & L_{B_i}^{-1} F_i \\ 0 & U_{S_i} \end{pmatrix}. \tag{17}$$

- Solve the preconditioned reduced system.
- Back-substitute to obtain the other unknowns (fully parallel step).

4.1 Resolution of the Reduced System

The reduced system is solved by an iterative solver. The iterative solver only needs to define the matrix-vector action.

For the iterative scheme we use the generalized conjugate residual (GCR) method (as in [8]). The GCR method is described for a system of the form $Ax = b$.

```
1. Compute $r_0 = b - Ax_0$. set $p_0 = r_0$.
2. For $j = 0, 1, 2, ...,$ until convergence do:
   a) $\alpha_j = \frac{(r_j, Ap_j)}{(Ap_i, Ap_j)}$
   b) $x_{j+1} = x_j + \alpha_j p_j$
   c) $r_{j+1} = r_j - \alpha_j Ap_j$
   d) compute $\beta_{ij} = -\frac{(Ar_{j+1}, Ap_j)}{(Ap_i, Ap_i)}$, for $i = 0, 1, ..., j$
   e) $p_{j+1} = r_{j+1} + \sum_{i=0}^{j} \beta_{ij} p_i$
   end do
```

Additionally to the vectors $\{p_j\}_{j=1}^k$, which are $A^T A$-orthogonal, extra vectors $\{Ap_j\}_{j=1}^k$ have to be stored. Since the projection of b onto the space $A\mathcal{K}$ with $\mathcal{K} = \text{span}\{p_j\}_{j=1}^k$ is equal to

$$\sum_{j=1}^{k} \frac{(b, Ap_j)}{(Ap_j, ap_j)} Ap_j, \tag{18}$$

the projection of the solution $x = A^{-1}b$ onto $\mathcal{K}$ is

$$\hat{x} = \sum_{j=1}^{k} \frac{(b, Ap_j)}{(Ap_j, ap_j)} Ap_j. \tag{19}$$

This observation implies that the projection onto the accumulated Krylov subspace may compute the unknown solution quite easily (involving k scalar products). Table 1 exhibits that the numerical speed-up is nearby of 15%.

Table 1. Numerical speed-up of the GCR method using the projection onto the accumulated Krylov subspace on the V10 engine pump problem

Krylov projection	#proc	CPU time	numerical speed-up
no	4	1750	1
yes	4	1515	1.15

5 Some Numerical Results

The speed-up obtained is quite good as shown in Table 2 by the elapsed times for solving the previous V10 engine pump problem. The partition number is increased from 1 to 4. One processor is used to solve one subdomain problem.

Table 2 exhibits a speed-up higher on 3 processors using this Schur DDM approach than using the parallelizing across the method. But with the Schur DDM that has been proposed here, the number of processors (which is equal to the partition number) is only limited by parallel performance considerations. For the small case considered here with 387 unknowns, the optimum partition number is 4 (see 3).

Table 2. Speed-up obtained on the V10 engine pump problem

#proc	CPU time	speed-up	#Jac	#discont	#steps
1	6845	1	65355	1089	311115
2	4369	1.56	66131	1061	315357
3	1820	3.76	65787	1059	313064
4	1513	4.52	65662	1043	313158

Table 3. Percentage of interface unknowns with respect to then number of processors $\frac{ne}{ne+\frac{n-ne}{np}}$ on the V10 engine pump problem ($n = 287$ unknowns).

#proc (np)	1	2	3	4	8	16
#interface unknowns (ne)	0	21	31	47	80	126
ratio of interface (%)	0	13	26	43	75	92

This limitation comes from the ratio between the number of interface unknowns and the computing complexity to solve subdomain problems. For the test case under consideration, the speed-up is supra-linear, because of two effects. The first one is a smaller full LU decomposition locally (i.e. on each processor). The second one is the parallelizing of the resolution on each subdomain that fits in the cache memory. Nevertheless, we expect only a linear speed-up when a sparse LU decomposition will be applied.

6 Conclusion

A Schur domain decomposition method has been investigated to solve systems of ordinary/algebraic differential equations. Because the data dependencies are not regular, an automatic process has been developed to separate the unknown variables in interface unknown variables and subdomain internal unknown variables. This approach was absolutely needed because the function $f(t, y)$ is given as a black box with only knowledge on the input variables and on the components of f to be affected. The condition number of the linear systems involved in the time integrator required a preconditioned linear Krylov solver. Some techniques to reuse computed information to speed-up the convergence have been investigated and save some elapsed-time. Next works will investigate some numerical tools to reuse computed information when some parts of the system become non-active or active during the cycle of simulation. Some questions are still open: what can be the numerical criterion to reuse the Krylov subspace when some dynamical systems situation reappears? May it be possible to use reduced systems obtained by proper orthogonal decomposition to model the interactions of other sub-systems to one given sub-system in the Schur DDM. This is the kind of question that will be addressed in the framework of the "PARallel Algebraic Differential Equations" ANR project.

References

[1] K. Burrage and H. Suhartanto. Parallel iterated method based on multistep Runge-Kutta of Radau type for stiff problems. *Adv. Comput. Math.*, 7(1-2):59–77, 1997. Parallel methods for ODEs.

[2] K. Burrage and H. Suhartanto. Parallel iterated methods based on multistep Runge-Kutta methods of Radau type. *Adv. Comput. Math.*, 7(1-2):37–57, 1997. Parallel methods for ODEs.

[3] D. Guibert and D. Tromeur-Dervout. Parallel adaptive time domain decomposition for stiff systems of ODE/DAE. *Computers & Structures*, 85(9):553–562, 2007.

[4] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations. I—Nonstiff Problems*, volume 8 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1993.

[5] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations. II—Stiff and Differential-Algebraic Problems*, volume 14 of *Springer Series in Computational Mathematics*. Springer-Verlag, Berlin, second edition, 1996.

[6] P. Hansen and D. de Werra, editors. *Regards sur la Théorie des Graphes*, Lausanne, 1980. Presses Polytechniques Romandes.

[7] Metis. http://glaros.dtc.umn.edu/gkhome/views/metis. Karypis Lab.

[8] Y. Saad. *Iterative Methods for Sparse Linear Systems.* Society for Industrial and Applied Mathematics, Philadelphia, PA, second edition, 2003.

[9] P. J. van der Houwen and B. P. Sommeijer. Parallel iteration of high-order Runge-Kutta methods with stepsize control. *J. Comput. Appl. Math.*, 29(1):111–127, 1990.

PDE-based Parameter Reconstruction through Schur and Schwarz Decompositions

Yuan He and David E. Keyes*

Department of Applied Physics & Applied Mathematics, Columbia University, New York NY, 10027 {yh2030,kd2112}@columbia.edu

1 Introduction

We consider in this work a distributed parameter identification problem for the FitzHugh-Nagumo system of equations of electrocardiology [8]. Specifically, we consider the two-component reaction-diffusion system

$$\begin{array}{ll} \partial_t u = \mu \Delta u + u(u-\alpha)(1-u) - v, & \text{in } Q, \\ \partial_t v = \kappa \Delta v + \epsilon(\vartheta u - \gamma v), & \text{in } Q, \\ u(\boldsymbol{x},0) = 0, \qquad v(\boldsymbol{x},0) = 0 & \text{in } \Omega, \\ \mathbf{n} \cdot \nabla u(\boldsymbol{x},t) = I(\boldsymbol{x},t), \quad \mathbf{n} \cdot \nabla v(\boldsymbol{x},t) = 0, & \text{on } \partial Q, \end{array} \tag{1}$$

where $\Omega \subset \mathbb{R}^n$, with $n = 2$ for the results in section 4. Q and ∂Q are defined as $Q \equiv \Omega \times (0,T)$ and $\partial Q \equiv \partial\Omega \times (0,T)$, respectively. See [8] for details on system parameters. The objective of the parameter identification is to reconstruct the reactive coefficient $\alpha(\boldsymbol{x})$ in the first equation from boundary measurements of the electrical potential u.

Our aim here is to present a numerical algorithm that can solve the reconstruction problem in large-scale (parallel) environments. The algorithm is of Newton-Krylov-Schur-Schwarz type; it combines Newton's method for numerical optimization with Krylov subspace solvers for the resulting reduced Karush-Kuhn-Tucker (KKT) systems. Schwarz preconditioning is used to solve the partial differential equations that are involved in the inversion procedure.

2 PDE-constrained Optimization

The parameter identification problem for the FitzHugh-Nagumo system is to reconstruct the physical coefficient $\alpha(\boldsymbol{x})$ from the knowledge of $h = u(\boldsymbol{x},t)$ on the boundary of the domain. Since one can measure the boundary potential for various applied current stimuli I, one thus has access to the time-dependent partial

* This work is supported in part by the U. S. National Science Foundation under CCF-03-52334

Neumann-to-Dirichlet map: $\Lambda : I(\boldsymbol{x}, t) \rightarrow h$. The parameter identification problem employs knowledge of this map Λ to recover function $\alpha(\boldsymbol{x})$.

We solve the inverse problem by formulating it as a PDE-constrained optimization problem [1, 4, 5, 6]:

$$\min_{\alpha,u} \mathcal{F}(\alpha, u)$$

$$\text{subject to} \qquad \mathcal{C}_s(I_s, \alpha, u_s, v_s) = 0, \quad s = 1, 2, ..., N_s. \tag{2}$$

where $\mathcal{C}_s(I_s, \alpha, u_s, v_s) = 0$ is abstract notation for (1) with source I_s. N_s is the number of source scenarios producing detectable measurements. The functional to be minimized is defined as

$$\mathcal{F}(\alpha, u) := \frac{1}{2} \sum_{s=1}^{N_s} \sum_{j=1}^{N_d} \int_0^T \int_{\partial\Omega} (u_s - h_s)^2 \delta(\boldsymbol{x} - \boldsymbol{x}_j) \; d\sigma(\boldsymbol{x}) dt + \rho \mathcal{R}(\alpha), \tag{3}$$

with h_s the measurement corresponding to source I_s. $\boldsymbol{x}_j$, $j = 1, ..., N_d$, are detector positions. To simplify notation, we write $u = (u_1, ..., u_s, ..., u_{N_s})$. $d\sigma$ denotes the surface measure on $\partial\Omega$. $\mathcal{R}(\alpha)$ is a regularization functional, and the regularization parameter ρ controls the strength of regularization.

The Lagrangian functional for the above minimization problem is

$$\mathcal{L}(u, v, \alpha, \lambda, \eta) = \mathcal{F}(\alpha, u) + \sum_{s=1}^{N_s} \langle (\lambda_s, \eta_s), \mathcal{C}_s \rangle \,, \tag{4}$$

where again sets of variables corresponding to the full set of different sources, such as $v = (v_1, ..., v_s, ..., v_{N_s})$, are implied. λ_s and η_s denote the Lagrangian multipliers (adjoint variables) corresponding to u_s and v_s, respectively. The solution to the constrained minimization problem satisfies the first-order optimality conditions of the Lagrangian functional, which are

$$\begin{array}{ll} \mathcal{L}_\lambda(u, v, \alpha, \lambda, \eta) = 0, & \mathcal{L}_\eta(u, v, \alpha, \lambda, \eta) = 0, \\ \mathcal{L}_u(u, v, \alpha, \lambda, \eta) = 0, & \mathcal{L}_v(u, v, \alpha, \lambda, \eta) = 0, \\ \mathcal{L}_\alpha(u, v, \alpha, \lambda, \eta) = 0. & \end{array} \tag{5}$$

Denoting $(u, v, \alpha, \lambda, \eta)$ by $\mathbf{u}$, we can recast (5) as the root-finding problem:

$$\mathcal{L}_{\mathbf{u}}(\mathbf{u}) = 0. \tag{6}$$

3 The Newton-Krylov-Schur-Schwarz Algorithm

In order to solve the optimality equations, a hybrid set of algebraic equations and quasilinear partial differential equations of reaction-diffusion type, we need a discretization of the PDEs and an algebraic solver for the resulting large nonlinear algebraic system. The Newton-Krylov family of methods provides an efficient way to solve such PDE systems [9].

3.1 The Newton-Krylov Method

Newton methods for solving (6) follow the iteration

$$\mathbf{u}_{k+1} = \mathbf{u}_k + l_k \delta \mathbf{u}_k, \tag{7}$$

with some initial guess $\mathbf{u}_0$, until convergence criteria are satisfied. The update direction $\delta\mathbf{u}_k$ at Newton iteration k is given by solving the saddle point problem

$$\mathcal{L}_{\mathbf{uu}}(\mathbf{u}_k)\delta\mathbf{u}_k = -\mathcal{L}_{\mathbf{u}}(\mathbf{u}_k). \tag{8}$$

Here the step length l_k is given by a line search or other globalization technique. The nested iteration is called Newton-Krylov when Krylov subspace methods are used to solve the inner KKT system. The method has received wide attention from practitioners in recent years; see the references cited in [9].

For the FitzHugh-Nagumo model we consider here, the KKT system has the form

$$\begin{pmatrix} \mathcal{L}_{uu} & 0 & \mathcal{L}_{u\alpha} & \mathcal{L}_{u\lambda} & \mathcal{L}_{u\eta} \\ 0 & 0 & 0 & \mathcal{L}_{v\lambda} & \mathcal{L}_{v\eta} \\ \mathcal{L}_{\alpha u} & 0 & \mathcal{L}_{\alpha\alpha} & \mathcal{L}_{\alpha\lambda} & 0 \\ \mathcal{L}_{\lambda u} & \mathcal{L}_{\lambda v} & \mathcal{L}_{\lambda\alpha} & 0 & 0 \\ \mathcal{L}_{\eta u} & \mathcal{L}_{\eta v} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta v \\ \delta\alpha \\ \delta\lambda \\ \delta\eta \end{pmatrix} = - \begin{pmatrix} \mathcal{L}_u \\ \mathcal{L}_v \\ \mathcal{L}_\alpha \\ \mathcal{L}_\lambda \\ \mathcal{L}_\eta \end{pmatrix}, \tag{9}$$

where $\delta u = [\delta u_1, \delta u_2, \ldots, \delta u_{N_s}]^T$, and δv, $\delta\lambda$, $\delta\eta$ and $\mathcal{L}_u$, $\mathcal{L}_v$, $\mathcal{L}_\lambda$, $\mathcal{L}_\eta$ are similarly defined. Because the forward problems for different sources are decoupled, the operator $\mathcal{L}_{uu}$ has diagonal structure:

$$\mathcal{L}_{uu} = \mathrm{diag}\{\mathcal{L}_{u_1u_1}, \mathcal{L}_{u_2u_2}, \ldots, \mathcal{L}_{u_{N_s}u_{N_s}}\} \tag{10}$$

and similarly for operators $\mathcal{L}_{u\lambda}$, $\mathcal{L}_{u\eta}$, $\mathcal{L}_{v\lambda}$, and $\mathcal{L}_{v\eta}$ and their adjoint operators, $\mathcal{L}_{\lambda u}$, $\mathcal{L}_{\eta u}$, $\mathcal{L}_{\lambda v}$, and $\mathcal{L}_{\eta v}$. Operators $\mathcal{L}_{u\alpha}$ and $\mathcal{L}_{\lambda\alpha}$ have the structure that $\mathcal{L}_{u\alpha} = [\mathcal{L}_{u_1\alpha}, \mathcal{L}_{u_2\alpha}, \ldots, \mathcal{L}_{u_{N_s}\alpha}]^T$ and $\mathcal{L}_{\lambda\alpha} = [\mathcal{L}_{\lambda_1\alpha}, \mathcal{L}_{\lambda_2\alpha}, \ldots, \mathcal{L}_{\lambda_{N_s}\alpha}]^T$. $\mathcal{L}_{\alpha u}$ and $\mathcal{L}_{\alpha\lambda}$ are their adjoint operators, respectively.

3.2 The Schur Complement Reduced Space Method

To avoid a huge storage requirement, we do not solve the KKT system (9) directly. Instead, in each Newton iteration, for a given α, we first solve the FitzHugh-Nagumo system (2), which turns out to be the first two equations in (5). We then solve the adjoint problem, (the fourth and fifth equations in (5)), thereupon, the terms $\mathcal{L}_u$, $\mathcal{L}_v$, $\mathcal{L}_\lambda$ and $\mathcal{L}_\eta$ vanish in the KKT system (9). The KKT system thus becomes

$$\begin{pmatrix} \mathcal{L}_{uu} & 0 & \mathcal{L}_{u\alpha} & \mathcal{L}_{u\lambda} & \mathcal{L}_{u\eta} \\ 0 & 0 & 0 & \mathcal{L}_{v\lambda} & \mathcal{L}_{v\eta} \\ \mathcal{L}_{\alpha u} & 0 & \mathcal{L}_{\alpha\alpha} & \mathcal{L}_{\alpha\lambda} & 0 \\ \mathcal{L}_{\lambda u} & \mathcal{L}_{\lambda v} & \mathcal{L}_{\lambda\alpha} & 0 & 0 \\ \mathcal{L}_{\eta u} & \mathcal{L}_{\eta v} & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta u \\ \delta v \\ \delta\alpha \\ \delta\lambda \\ \delta\eta \end{pmatrix} = - \begin{pmatrix} 0 \\ 0 \\ \mathcal{L}_\alpha \\ 0 \\ 0 \end{pmatrix}. \tag{11}$$

We can now build the Schur complement of (11) by eliminating δu, δv, $\delta\lambda$ and $\delta\eta$. We then obtain

$$\mathcal{H}_{red}\delta\alpha = -\mathcal{L}_\alpha, \tag{12}$$

where the reduced gradient $\mathcal{L}_\alpha$ is given by

$$\mathcal{L}_\alpha = \sum_{s=1}^{N_s} \int_0^T \lambda_s u_s (1 - u_s) dt + \rho \mathcal{R}'(\alpha) \tag{13}$$

and the reduced Hessian $\mathcal{H}_{red}$ (Schur complement of the KKT) is given by

$$\mathcal{H}_{red} = \mathcal{L}_{\alpha\alpha} - \mathcal{L}_{\alpha u} W - W^* \mathcal{L}_{u\alpha} + W^* \mathcal{L}_{uu} W, \tag{14}$$

with W defined as $W = [\mathcal{L}_{\lambda u}^{-1} + \mathcal{L}_{\lambda u}^{-1} (\mathcal{L}_{\eta v} - \mathcal{L}_{\eta u} \mathcal{L}_{\lambda u}^{-1})^{-1} \mathcal{L}_{\eta u} \mathcal{L}_{\lambda u}^{-1}] \mathcal{L}_{\lambda\alpha}$. Here W^* denotes the adjoint of W. The reduced Hessian $\mathcal{H}_{red}$ has a much smaller size (and is much denser) than the original Hessian $\mathcal{L}_{\mathbf{uu}}$. It can be verified that $\mathcal{H}_{red} = \mathcal{H}_{red}^*$, that is, $\mathcal{H}_{red}$ is self-adjoint.

One can obtain the Gauss-Newton approximation by dropping second derivative information in the $\mathcal{L}_{\alpha u}$ and $\mathcal{L}_{u\alpha}$ terms [7, 10], resulting in the reduced Hessian:

$$\mathcal{H}_{red}^{GN} = \mathcal{L}_{\alpha\alpha} + W^* \mathcal{L}_{uu} W. \tag{15}$$

Table 1. The reduced-space Newton algorithm

```
Algorithm 1: Reduced-space Newton algorithm
set $k_{max}$, $\varepsilon_1$, $\varepsilon_2$
guess $\alpha_0(\boldsymbol{x})$; set $k = 0$
evaluate $\mathcal{F}(\alpha_0)$
while ($k < k_{max}$ & $\frac{\|\mathcal{L}_{\alpha_k}\|}{\|1+\mathcal{F}(\alpha_k)\|} > \varepsilon_1$, $\frac{\mathcal{F}(\alpha_k)}{\mathcal{F}(\alpha_0)} > \varepsilon_2$)
    evaluate $\mathcal{L}_{\alpha_k}$ by (13)
    compute $\delta\alpha_k$ by (12)
    compute $l_k$ by a line search
    $\alpha_{k+1} = \alpha_k + l_k \delta\alpha_k$
    evaluate $\mathcal{F}(\alpha_{k+1})$
    $k = k + 1$
end while
```

We thus obtain the following Newton-Krylov-Schur (reduced-space) method as described in Table 1. For full space methods of similar type, see [5, 6].

3.3 The Schwarz Decomposition PDE Solver

In the aforementioned Newton-Krylov-Schur inversion procedure, at each Newton step, many time-dependent PDEs need to be solved. Some of those PDEs are quasilinear (the FitzHugh-Nagumo system), others are linear (the adjoint equations). The efficiency of the inversion algorithm depends strongly on the efficiency of the algebraic solvers that are used. Our strategy for building an efficient parallel solver is based on the parallel solver toolkit PETSc from Argonne National Laboratory [2]. All the PDEs are passed to the SNES solver in PETSc after being discretized in time by implicit Euler.

4 Numerical Simulations

We present in this section some performance analysis for the algorithm presented above. For detailed analysis on the quality of reconstructions and its relationship with various algorithmic parameters, we refer interested readers to [8]. All the results shown in this section are obtained on the Mac cluster *System X* at the Virginia Polytechnic Institute and State University.

4.1 Performance of Different Solver-preconditioner Combinations

In the first study, we compare the performance of different algebraic solvers and preconditioning methods on our forward model problem.

The algebraic solvers considered here are all Krylov subspace methods (KSP), including the generalized minimal residual (GMRES), modified GMRES, flexible GMRES, conjugate gradient (CG), bi-conjugate gradient (BiCG), and the stabilized version of bi-conjugate gradient squared (BCGS). We refer to [3] for details of those methods. The preconditioning methods we considered include the Jacobi, block Jacobi and the additive Schwarz method. We present in Table 2 the execution time of different combinations. Since many linear systems we encounter in the solution of the forward and inverse problems are indefinite, we use the classical GMRES method with additive Schwarz as the preconditioner in the following sections although there are other combinations that can achieve similar performance as indicated in Table 2.

Table 2. Execution time for the forward model using different KSP accelerators with different preconditioners

	none	Jacobi	bJacobi	ASM (basic)
GMRES (classical GS)	89.5	90.0	81.3	67.9
GMRES (modified GS)	94.7	74.2	84.5	87.2
f GMRES	91.0	77.0	68.0	87.7
CG	96.1	66.3	63.8	66.9
BiCG	88.8	67.3	80.5	88.8
BCGS	83.6	66.3	66.4	63.0

4.2 Scalability Results on the Forward Solver

We now consider parallel performance of the algorithm we have developed on the forward problem. We show in Fig. 1 some fixed-size scaling results obtained by increasing the number of processors with a fixed grid size. The strong speedup and efficiency results based on execution time for two different spatial grid size, 128×128 and 256×256 are presented. As expected, speedup and efficiency improve with problem size.

In Table 3 we show the results on execution time and implementation efficiency ε [5] which is based on the average Mflop/s. We find that the implementation efficiency

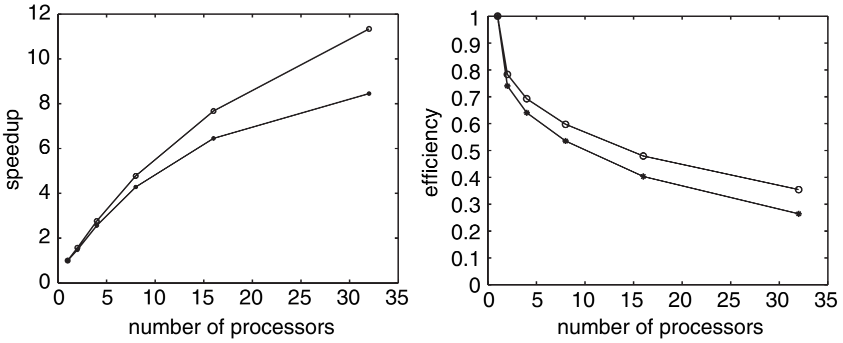


Fig. 1. Strong speedup (left) and efficiency (right) for the forward solver on 128×128 (⋆) and 256 × 256 (∘) spatial grids, respectively.

Table 3. Performance analysis of the forward solver. NP denotes the number of processors; ε is the implementation efficiency.

	128 × 128		256×256	
NP	execution time	ε	execution time	ε
1	312.1	1.00	1881.3	1.00
2	219.8	0.89	1241.0	0.79
4	121.8	0.84	679.4	0.73
8	72.9	0.75	393.6	0.62
16	48.4	0.58	245.2	0.50
32	36.9	0.54	166.0	0.36

of small size problem is slightly better than the implementation efficiency of the problem of large size.

4.3 Scalability Results on the Inversion Algorithm

We present in Figure 2 the strong speedup and efficiency results for the inversion algorithm for up to 32 processors. We observe by comparing Fig. 2 and Fig. 1 that the scalability of the inversion algorithm is slightly better than that of the forward solver. One explanation of this phenomenon is the forward solver deals with only nonlinear problems (the FitzHugh-Nagumo model), while the inverse solver deals with both nonlinear (forward problem) and linear (adjoint problem). The performance from the linear problem part is better than that from the nonlinear problem part.

In Table 4 we show some results on execution time and the implementation efficiency ε for the inversion algorithm. The implementation efficiency of small problem size is fairly independent of problem size. Again, by comparing with Table 3, the implementation efficiency of the inversion algorithm is slightly better than the implementation efficiency of forward model.

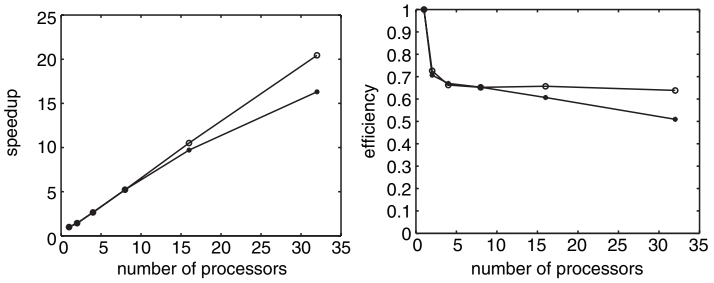


Fig. 2. Strong speedup (left) and efficiency (right) for the inversion algorithm on 128 × 128 ($\star$) and 256 × 256 ($\circ$) spatial grids, respectively.

Table 4. Performance analysis of the inverse solver. NP denotes the number of processors; ε is the implementation efficiency.

	128 × 128		256×256	
NP	execution time	ε	execution time	ε
1	9914.0	1.00	72728.9	1.00
2	7015.5	0.79	50101.0	0.70
4	3701.6	0.85	27743.1	0.73
8	1897.7	0.89	13937.0	0.75
16	1021.2	0.85	6916.2	0.79
32	608.2	0.75	3558.1	0.76

5 Conclusion

We have presented in limited space a parallel numerical algorithm for a PDE-based distributed parameter reconstruction problem. This Newton-Krylov algorithm combines Newton's method for numerical optimization with Krylov subspace solvers for the resulting KKT system. We have also discussed the performance of both the forward solver and the inversion algorithm. Physical results of the inversion are available in [8].

Future research will focus on accelerating the current code, extending it to three dimensions on much larger numbers of processors, and comparing simulations on more realistic geometries with experimental measurements.

References

[1] V. Akcelik, G. Biros, O. Ghattas, J. Hill, D. Keyes, and B. van Bloemen Waanders. Parallel algorithms for PDE-constrained optimization. In M. Heroux, P. Raghaven, and H. Simon, editors, *Frontiers of Parallel Computing*. SIAM, 2006.

[2] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc Homepage, 2007. http://www.mcs.anl.gov/petsc.

[3] R. Barrett, M. Berry, T. F. Chan, J. Demmel, J. Donato, J. Dongarra, V. Eijkhout, R. Pozo, C. Romine, and H. van der Vorst. *Templates for the Solution of Linear Systems: Building Blocks for Iterative Methods*. SIAM, Philadelphia, 2nd edition, 1994.

[4] L. Biegler, O. Ghattas, M. Heinkenschloss, and B. van Bloemen Waanders, editors. *Large-Scale PDE-Constrained Optimization*. Lecture Notes in Computational Science and Engineering. Springer-Verlag, Berlin, 2003.

[5] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part I: The Krylov-Schur solver. *SIAM J. Sci. Comput.*, 27:687–713, 2005.

[6] G. Biros and O. Ghattas. Parallel Lagrange-Newton-Krylov-Schur methods for PDE-constrained optimization. Part II: The Lagrange-Newton solver and its application to optimal control of steady viscous flows. *SIAM J. Sci. Comput.*, 27:714–739, 2005.

[7] E. Haber, U. Ascher, and D. Oldenburg. On optimization techniques for solving nonlinear inverse problems. *Inverse Problems*, 16:1263–1280, 2000.

[8] Y. He and D. E. Keyes. Reconstructing parameters of the FitzHugh-Nagumo system from boundary potential measurements. *J. Comput. Neurosci.*, 23(2), 2007.

[9] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. *J. Comput. Phys.*, 193:357–397, 2004.

[10] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer-Verlag, New York, 1999.

Numerical Method for Wave Propagation Problem by FDTD Method with PML

Takashi Kako and Yoshiharu Ohi

The University of Electro-Communications, Chofu, Japan.
{kako,ohi}@sazae.im.uec.ac.jp

1 Introduction

It is necessary to set a computational domain appropriately for the numerical simulation of wave propagation phenomena in unbounded region. There are several approaches for this problem. In 1994, J.-P. Bérenger introduced the technique of Perfectly Matched Layer (PML). It is said that PML technique gives the best performance for Finite Difference Time Domain (FDTD) method in unbounded region. Some researchers expanded this idea into the linearized Euler equation and acoustic wave equation. In this paper, we consider some mathematical and numerical problem of PML technique, and propose a new discretization scheme that is better than the original scheme.

2 PML Method

2.1 Formulation of PML

The Maxwell equation is written as:

$$\frac{\partial \mathbf{E}}{\partial t} = -\frac{\sigma}{\epsilon}\mathbf{E} + \frac{1}{\epsilon}\nabla \times \mathbf{H}, \qquad \frac{\partial \mathbf{H}}{\partial t} = -\frac{1}{\mu}\nabla \times \mathbf{E}. \tag{1}$$

where, $\mathbf{E}$ is electric field, $\mathbf{H}$ is magnetic field, and ϵ, μ and σ are permittivity, magnetic permeability and electrical conductivity, respectively.

To treat the problem in unbounded region, we introduce PML technique which surrounds interior region by an absorption medium introduced in [1]. In the PML region, the electromagnetic wave propagates without reflection and decreases amplitude exponentially, and there is no reflection on the boundary between the interior and PML regions. The solution in interior region is not polluted. This behavior is realized by introducing dissipation term into the Maxwell equation (1), and imposing the impedance matching condition $\sigma/\epsilon_0 = \sigma^*/\mu_0$:

$$\frac{\partial \mathbf{E}}{\partial t} = -\frac{\sigma}{\epsilon_0}\mathbf{E} + \frac{1}{\epsilon_0}\nabla \times \mathbf{H}, \qquad \frac{\partial \mathbf{H}}{\partial t} = -\frac{\sigma^*}{\mu_0}\mathbf{H} - \frac{1}{\mu_0}\nabla \times \mathbf{E}, \tag{2}$$

where, σ^* is magnetic conductivity.

2.2 Exact Solution in PML Region

In this section, we investigate some properties of PML technique. First, we consider one dimensional continuous problem. In case that the solutions of (2) depend only on t and x, the equation is rewritten as:

$$\epsilon_0 \frac{\partial E_y}{\partial t} + \sigma E_y = -\frac{\partial H_z}{\partial x}, \qquad \mu_0 \frac{\partial H_z}{\partial t} + \sigma^* H_z = -\frac{\partial E_y}{\partial x}. \tag{3}$$

We take a unit such that $\epsilon_0 = \mu_0 = 1$, then the impedance matching condition becomes $\sigma = \sigma^*$. Also, we put $E_y = u$ and $H_z = v$, then (3) becomes the wave equation for u and v:

$$\frac{\partial u}{\partial t} + \sigma u = -\frac{\partial v}{\partial x}, \qquad \frac{\partial v}{\partial t} + \sigma v = -\frac{\partial u}{\partial x}. \tag{4}$$

The exact solutions of (4) with initial values $u(0,x)$ and $v(0,x)$ at $t = 0$ are given as:

$$\begin{aligned} u(t,x) &= \frac{1}{2}\left(e^{-\int_0^x \sigma(s)ds} f(x-t) + e^{\int_0^x \sigma(s)ds} g(x+t)\right), \\ v(t,x) &= \frac{1}{2}\left(e^{-\int_0^x \sigma(s)ds} f(x-t) - e^{\int_0^x \sigma(s)ds} g(x+t)\right), \end{aligned}$$

where,

$$\begin{aligned} f(x) &= e^{\int_0^x \sigma(s)ds}(u(0,x) + v(0,x)), \\ g(x) &= e^{-\int_0^x \sigma(s)ds}(u(0,x) - v(0,x)). \end{aligned}$$

3 FDTD Method and PML

3.1 Discretization of Dissipation Term in FDTD Method

In 1966, K.S. Yee [2] introduced FDTD method to treat electromagnetic wave problem. In this section, we consider a discretization scheme for (4). We set $\Delta t = \Delta x \equiv \tau$ and $\sigma_s \equiv \sigma(s\Delta x)$, $s = m \,\mathrm{or}\, m + 1/2$, and make use of the approximation:

$$\sigma(s\Delta x)u(s\Delta x) \approx \sigma_s \frac{1}{2}(u^n_{s+\frac{1}{2}} + u^n_{s-\frac{1}{2}}).$$

Then, the difference approximation of (4) becomes

$$u^{n+1}_m = a_m u^n_m - b_m(v^{n+\frac{1}{2}}_{m+\frac{1}{2}} - v^{n+\frac{1}{2}}_{m-\frac{1}{2}}), \tag{5}$$

$$v^{n+\frac{1}{2}}_{m+\frac{1}{2}} = a_{m+\frac{1}{2}} u^{n-\frac{1}{2}}_{m+\frac{1}{2}} - b_{m+\frac{1}{2}}(v^n_{m+1} - v^n_m), \tag{6}$$

with

$$a_s = \frac{1 - \frac{\tau}{2}\sigma_s}{1 + \frac{\tau}{2}\sigma_s}, \; b_s = \frac{1}{1 + \frac{\tau}{2}\sigma_s}, \; s = m \,\mathrm{or}\, m + \frac{1}{2}. \tag{7}$$

We call (5) - (7) a plain scheme.

3.2 Artificial Reflection Caused by Discretization

In this section, we consider the artificial reflection caused by discretization. We set $\sigma = 0$ in $x \leq 0$ and $\sigma > 0$ in $x > 0$. Then the solution given as:

$$\{u_m^n, v_{m-\frac{1}{2}}^{n-\frac{1}{2}}\},\ u_m^n = \delta_{0,n-m},\ v_{m-\frac{1}{2}}^{n-\frac{1}{2}} = \delta_{0,n-m}$$

which propagates towards the positive direction of x. When $t = 0\,(n = 0)$, the solution is one at $x = 0$ which is boundary between interior and PML regions and zero elsewhere. When $n = 1/2$, we have from (6):

$$v_{m+\frac{1}{2}}^{\frac{1}{2}} = \begin{cases} b_{1/2} & : m = 0, \\ 0 & : \text{otherwise.} \end{cases}$$

When $n = 1$, we have from (5):

$$u_m^1 = \begin{cases} a_0 - b_0 b_{1/2} & : m = 0, \\ b_1 b_{1/2} & : m = 1, \\ 0 & : \text{otherwise.} \end{cases}$$

Hence, u_0^1 propagates towards the negative direction of x. We can express u_0^1 concretely

$$\begin{aligned} u_0^1 &= a_0 - b_0 b_{1/2} \\ &= \frac{1 - \frac{\tau\sigma_0}{2}}{1 + \frac{\tau\sigma_0}{2}} - \frac{1}{1 + \frac{\tau\sigma_0}{2}} \frac{1}{1 + \frac{\tau\sigma_{1/2}}{2}} = \frac{\frac{\tau\sigma_{1/2}}{2}}{1 + \frac{\tau\sigma_{1/2}}{2}}. \end{aligned}$$

Then, if $\sigma > 0$, an artificial reflection occurs. Therefore, when we set non-trivial PML, an artificial reflection occurs inevitably. We assume that $\sigma(x)$ can be expanded in the Tayler series in $[0, +\infty)$ as:

$$\sigma(x) = \sigma_0 + \sum_{k=1}^{N} \frac{1}{k!} \frac{d^k}{dx^k} \sigma(0) x^k + O(x^{N+1}).$$

Then, the artificial reflection coefficient R is given as:

$$R = \sigma_o \tau + \frac{\sigma'(0)}{2} \tau^2 + O(\tau^3).$$

In particular, the artificial reflection is almost proportional to the product of jump of σ and τ. In case the jump of σ is zero, it is proportional to the product of derivative of σ and τ^2 by neglecting $O(\tau^3)$ term. Furthermore, if σ is differentiable at the boundary, the artificial reflection is at most order τ^3.

3.3 A New Scheme with Lower Reflection

From the analysis in the previous section, even if the dissipation is constant, an artificial reflection occurs in PML region. To eliminate this spurious reflection at PML region where σ is constant, we propose the new scheme. The new scheme is defined as:

$$u_m^{n+1} = a_m^{new} u_m^n - b_m^{new}(v_{m+\frac{1}{2}}^{n+\frac{1}{2}} - v_{m-\frac{1}{2}}^{n+\frac{1}{2}}), \tag{8}$$

$$v_{m+\frac{1}{2}}^{n+\frac{1}{2}} = a_{m+\frac{1}{2}}^{new} u_{m+\frac{1}{2}}^{n-\frac{1}{2}} - b_{m+\frac{1}{2}}^{new}(v_{m+1}^n - v_m^n), \tag{9}$$

with

$$a_s^{new} = e^{-\tau\sigma_s},\ b_s^{new} = e^{-\tau\sigma_s/2},\ s = m\,\mathrm{or}\,m + \frac{1}{2}. \tag{10}$$

σ_s is constant with respect s, we can show easily that $a_s - b_s b_{s+1/2} = 0$. This concludes that there is no spurious reflection in PML region where σ is constant.

4 Some Numerical Examples

4.1 Comparison among Various Schemes in 1D Case

In this section, we give some numerical examples to confirm our analysis. In the first example, we compare the spurious reflections among various schemes in 1D case. The whole region $[0, 2]$ is set to be PML with constant dissipation: $\sigma(x) \equiv \log 10 = 2.302585\cdots$, $x \in [0, 2]$. We set the initial values u and v to be

$$u(0,x) = \begin{cases} \cos^2\Big(20\pi(x - 1.0)\Big), & 0.95 < x < 1.05, \\ \hfill 0, & \text{otherwise}, \end{cases} \qquad v(0,x) \equiv 0.$$

We assume the homogeneous Dirichlet condition on both ends of $[0, 2]$. In this case, the analytical reflection coefficient for this PML is $e^{-2\int_0^2 \sigma(x)dx} = 10^{-4}$. Namely the incident wave from the left end has a primal reflection with the magnitude 10^{-4}.

Figure 1 - 3 show the comparison of reflection waves computed by Bérenger's original scheme, plain scheme and our new scheme. We take a common mesh size $\tau = 1/160$ for space and time. The horizontal coordinate represents the time t and the vertical coordinate shows the value of $u(t, x)$ at time $t = 0.0,\ 0.2,\ 0.4,\ 0.6$ respectively. Plain scheme and Bérenger's scheme give spurious reflective trail behind the wave front whereas our new scheme is pollution free. The magnitude of the spurious waves is proportional to $\sigma^2\tau^2$, it could be controlled to be small enough in practical applications.

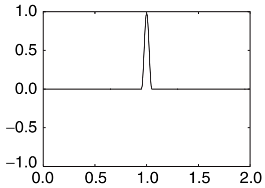


Fig. 1. The initial shape of $u(0, x)$.

In the second example, we compare the reflection waves from PML for three different shapes of function $\sigma(x)$ in our new scheme. Figure 4 shows the shapes of function $\sigma(x)$. The vacuum region is $[0.0, 1.0]$ and PML one is $[1.0, 1.2]$. In the first case, $\sigma(x)$ increases discontinuously at the boundary between interior and PML

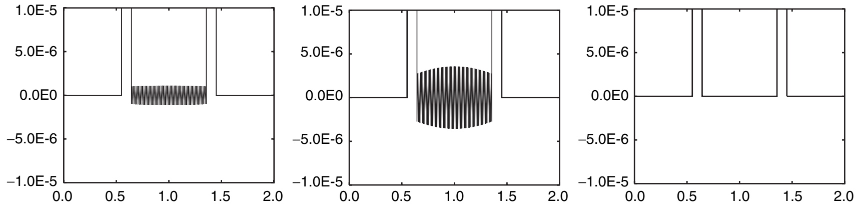


Fig. 2. Comparison of reflection waves at $t = 0.4$ for Bérenger (left), plain (middle) and new scheme (right).

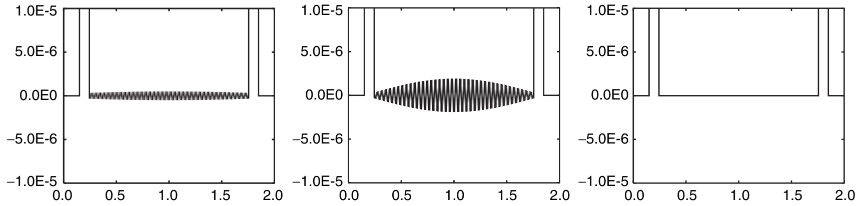


Fig. 3. Comparison of reflection waves at $t = 0.8$ for Bérenger (left), plain (middle) and new scheme (right).

regions with magnitude $\sigma_0 = 10 \log 10 = 23.025 \cdots$. In the second case, $\sigma(x)$ increases linearly on $[1.0, 1.1]$. In the last case, $\sigma(x)$ increases as the 3rd order spline on $[1.0, 1.1]$. In all cases, the integrals of $\sigma(x)$ on $[1.0, 1.2]$ are the same. Next, we measure the reflection at $x = 0.5$. In figure 5 - 6, the horizontal coordinate is time and the vertical one is the value of $u(t, 0.5)$ at the observation point. The wave form during the time between 1.9-2.0 propagate from the interior vacuum region to the PML region, and reflects back at an edge of a PML region, and comes back to the interior region again. We call this wave the real reflection wave. The wave in the neighborhood of $t = 1.6$ is spurious one. In the first, the second and the last cases, the spurious waves are proportional to τ, τ^2, τ^4 respectively.

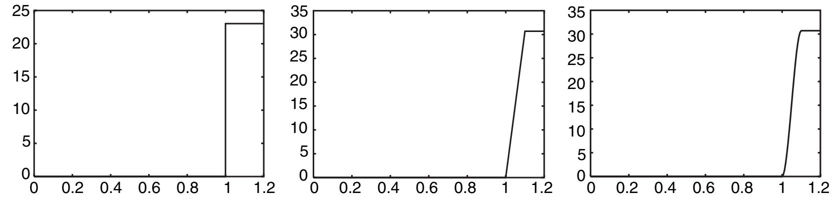


Fig. 4. Shapes of function $\sigma(x)$ for three different cases: discontinuous (left), linear (middle) and 3rd order spline (right).

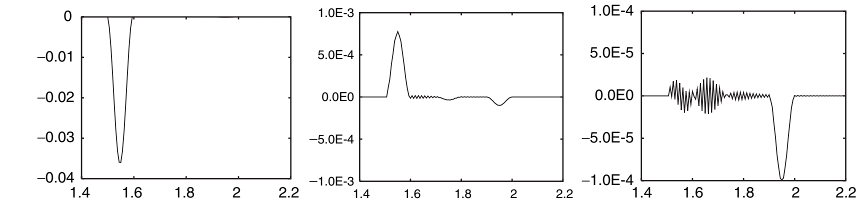


Fig. 5. Comparison of reflection wave to depend on three different shapes of $\sigma(x)$: $\tau = 1/160$.

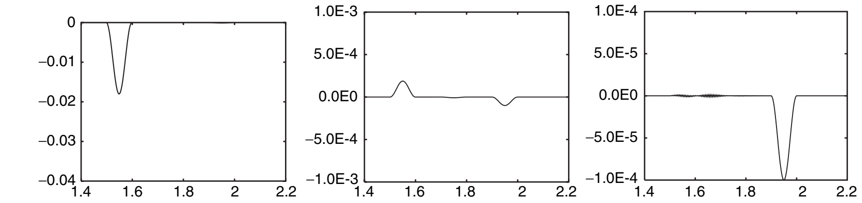


Fig. 6. Comparison of reflection wave to depend on three different shapes of $\sigma(x)$: $\tau = 1/320$.

4.2 Application to Two-Dimensional Electromagnetic Problem

We extend our scheme to the two-dimensional Maxwell equation for TE mode, and give some numerical examples. The concrete algorithm satisfies the CFL stability condition and $\Delta x = \Delta y = \Delta l = 1/160$ and $\Delta t = \Delta l/\sqrt{2}$. Bérenger's scheme is

$$H_{zx}(i,j) = e^{-\sigma_x(i)\Delta t} H_{zx}(i,j) - \frac{1-e^{-\sigma(i)\Delta t}}{\sigma_x(i)\Delta l}\{E_y(i+1,j) - E_y^n(i,j)\},$$

and our new scheme is

$$H_{zx}(i,j) = e^{-\sigma_x(i)\Delta t} H_{zx}(i,j) - \frac{\Delta t}{\Delta l} e^{-\sigma_x(i)\frac{\Delta t}{2}}\{E_y(i+1,j) - E_y^n(i,j)\}.$$

We set the computational domain to be a square $[-0.7, 0.7] \times [-0.7, 0.7]$ and the vacuum region is a square $[-0.5, 0.5] \times [-0.5, 0.5]$. The shapes of the dissipation functions $\sigma(x)$ and $\sigma(y)$ are the 3rd order spline like in the 1D case. The initial value is set to be

$$H_z(0,x,y) = e^{-(x-2+y^2)/16}, \quad E_x(0,x,y) = 0, \quad E_y(0,x,y)) = 0.$$

Figure 7 - 9 show the time history of the wave. The horizontal coordinate is x and the vertical one is y, and the value of $u(t,x,y)$ is represented by gradation of brightness. The results show good numerical performance with little reflection from the PML region.

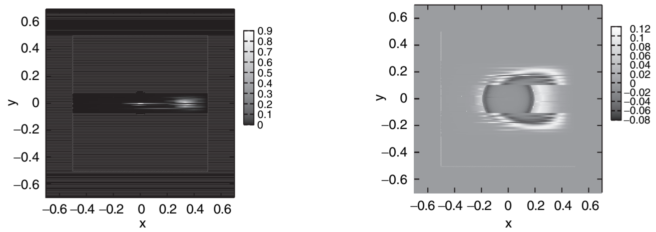


Fig. 7. Two-dimensional results, $t = 0.0$ (left), $t = 0.2$ (right).

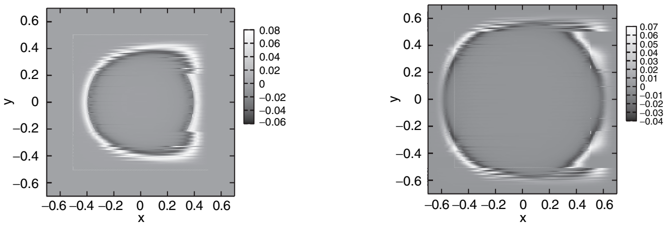


Fig. 8. Two-dimensional results, $t = 0.4$ (left), $t = 0.6$ (right).

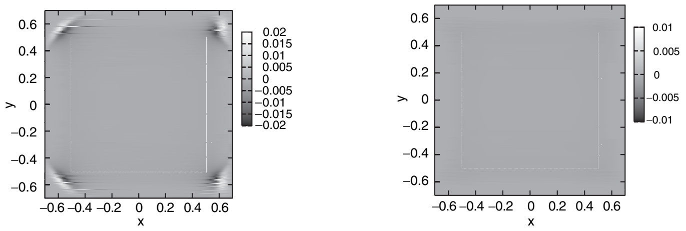


Fig. 9. Two-dimensional results, $t = 0.8$ (left), $t = 1.0$ (right).

5 Conclusion and Future Works

We explained the origin of the artificial reflection based on the mathematical analysis for 1D problem, and proposed a new scheme for which the artificial reflection does not occur in the region where $\sigma(x)$ is constant. By some numerical examples, we confirmed our mathematical analysis and effectiveness of our new scheme. Moreover, we extended the new scheme to 2D problem and got good results. As the result of these numerical performance, we conclude that the new PML is efficient in 1D and 2D computation of wave propagation problems.

The theoretical analysis for 2D problem and the proposal of stable 3D numerical method are future works. We will then proceed to the application in the real world problem such as the transient phenomena in various wave propagation problems including the voice generation simulation and the electromagnetic wave simulation in MRI problem.

References

[1] J.-P. Berenger. A perfectly matched layer for the absorption of electromagnetic waves. *J. Comput. Phys.*, 114(2):185–200, 1994.

[2] K. Yee. Numerical solution of initial boundary value problems involving Maxwell's equation in isotropic media. *IEEE Trans. Antennas and Propagation*, 14(3):302–307, 1966.

Fast Domain Decomposition Algorithms for Discretizations of 3-*d* Elliptic Equations by Spectral Elements

Vadim Korneev[1] and A. Rytov[2]

[1] St. Petersburg State University, Russia. korneev@pobox.spbu.ru
[2] St. Petersburg State Polytechnical University, Russia. Alryt@mail.ru

1 Introduction

In DD (domain decomposition) methods, the main contribution to the computational work is due to the two major components – solvers for local Dirichlet problems on subdomains of decomposition and local problems on their faces. Without loss of generality, we assume that the domains of FE's (finite elements) serve as subdomains of decomposition. At that, under the conditions of shape regularity, optimization of these components is reduced to obtaining fast preconditioners-solvers for the stiffness matrix of the p reference element and the Schur complement, related to its boundary.

Competitors for spectral FE's are *hierarchical* FE's, which have the tensor products of the integrated Legendre's polynomials for the form functions. As a starting point for optimization of major solvers for these two types of FE discretizations, primarily served the finite-difference preconditioners, suggested by [5], see also [8] for hierarchical and by [14] for spectral reference elements stiffness matrices. For internal stiffness matrices of hierarchical elements, a number of fast preconditioners-solvers have been justified theoretically by [6, 7, 2, 3] and thoroughly tested numerically. For spectral elements, to the best of the authors knowledge, there is known, the multilevel solver of [16], which efficiency was well approved numerically.

Hierarchical and spectral elements look differently. However, [11, 12] established an interrelation between them, showing that in computations they can be treated with a great measure of similarity. In particular, they considered optimal multilevel and DD types preconditioners-solvers for 2-d spectral elements, similar to those designed earlier for hierarchical elements. In this paper, first of all, we obtain fast multiresolution wavelet preconditioners-solvers for the internal FE and face subproblems, arising in DD algorithms for 3-d discretizations by spectral elements. The former realizes a technique alike the one implemented by [3] for hierarchical elements. The preconditioner of the same kind can be derived for the mass matrix, allowing in turn to obtain the face solver by K-interpolation. Inefficient prolongations from the interface boundary can also compromise optimality of DD algorithm. We approve the computationally fast prolongations by means of the inexact iterative solver for inner problems on FE's. With the mentioned three main fast DD components in hands, it is left to find a good preconditioner for the wire basket subproblem, having relatively small dimension $\mathcal{O}(\mathcal{R}p)$, where $\mathcal{R}$ is the number of finite elements. We use the one considered by [4] and other authors (see this paper for references), assuming that in a whole it is sufficiently fast. Our main conclusion is that the DD preconditioner-solver, with the pointed out components, has the relative condition number $\mathcal{O}((1+\log p)^2)$, while solving the system of algebraic equations with the DD preconditioner for the matrix requires $\mathcal{O}(N(1+\log p))$ arithmetic operations, where $N \simeq \mathcal{R}p^3$ is the order of the FE system.

We use notations: $\mathcal{Q}_{p,\mathbf{x}}$ – the space of polynomials of the order $p \geq 1$ in each variable of $\mathbf{x} = (x_1, x_2, .., x_d)$, d is the dimension; GLL and GLC nodes are the nodes of the Gauss-Lobatto-Legendre and Gauss-Lobatto-Chebyshev quadratures, respectively; signs $\prec$, $\succ$, $\asymp$ are used for the inequalities and equalities hold up to positive absolute constants; $\mathbf{A}^+$ – pseudo-inverse to a matrix $\mathbf{A}$; $\mathbf{A} \prec \mathbf{B}$ with nonnegative matrices $\mathbf{A}, \mathbf{B}$ implies $\mathbf{v}^\top \mathbf{A}\mathbf{v} \prec \mathbf{v}^\top \mathbf{B}\mathbf{v}$ for any vector $\mathbf{v}$ and similarly for signs $\succ$, $\asymp$; $\tau_0 = (-1,1)^d$ is the reference cube. Notations $|\cdot|_{k,\Omega}$, $\|\cdot\|_{k,\Omega}$ stand for the semi-norm and the norm in Sobolev's space $H^k(\Omega)$, $\overset{\circ}{H}{}^1(\Omega) = (v \in H^1(\Omega) : v|_{\partial\Omega} = 0)$. Since their similarity in our context, the both Lagrange elements with the GLL and GLC nodes are called *spectral*.

2 Finite-Difference and Factorized Preconditioners for Stiffness Matrices of Spectral *p* Elements

The GLL nodes $\eta_i \in [-1,1]$ satisfy $(1-\eta_i^2)P_p'(\eta_i) = 0$, whereas the GLC nodes are extremal points of the Chebyshev polynomials: $\eta_i = \cos\left(\frac{\pi(p-i)}{p}\right)$, $i = 0, 1, .., p$. For $i \leq N$, the steps $\hbar_i := \eta_i - \eta_{i-1}$ of the both meshes have the asymptotic behavior $\hbar_i \asymp i/p^2$. The both orthogonal tensor product meshes with the nodes $\mathbf{x} = \boldsymbol{\eta}_{\boldsymbol{\alpha}} = (\eta_{\alpha_1}, \eta_{\alpha_2}, .., \eta_{\alpha_d})$, $\boldsymbol{\alpha} \in \omega := \{\boldsymbol{\alpha} = (\alpha_1, \alpha_2, .., \alpha_d) : 0 \leq \alpha_1, \alpha_2, .., \alpha_d \leq p\}$, are termed in the paper *Gaussian.* We consider the stiffness matrices $\mathbf{A}_{\text{sp}}$ of the respective Lagrange reference elements, induced by the Dirichlet integral

$$a_{\tau_0}(u,v) = \int_{\tau_0} \nabla u \cdot \nabla v \, d\mathbf{x} \, .$$

Let $\mathcal{H}(\tau_0)$ be the space of functions, continuous on $\overline{\tau}_0$ and belonging to $\mathcal{Q}_{1,\mathbf{x}}$ on each nest of the Gaussian mesh $x_k = \eta_i$, then $\boldsymbol{\mathcal{A}}_{\text{sp}}$ denotes the preconditioner, which is the FE matrix, corresponding to this space and Dirichlet integral a_{τ_0}. As a preconditioner for $\mathbf{A}_{\text{sp}}$ in 3-d, it can be used the simpler matrix

$$\mathbb{A}_\hbar = \boldsymbol{\Delta}_\hbar \otimes \mathbb{D}_\hbar \otimes \mathbb{D}_\hbar + \mathbb{D}_\hbar \otimes \boldsymbol{\Delta}_\hbar \otimes \mathbb{D}_\hbar + \mathbb{D}_\hbar \otimes \mathbb{D}_\hbar \otimes \boldsymbol{\Delta}_\hbar \, ,$$

where $\mathbb{D}_\hbar$ is the diagonal matrix $\mathbb{D}_\hbar = \operatorname{diag}[\widetilde{\hbar}_i = \frac{1}{2}(\hbar_i + \hbar_{i+1})]_{i=0}^p$, with $\widetilde{\hbar}_i = 0$ for $i = 0, p+1$, and $\boldsymbol{\Delta}_\hbar$ is the FE matrix of the bilinear form $(v', w')_{(-1,1)}$ on the space $\mathcal{H}(-1,1)$ of continuous and piece wise linear on the 1-d Gaussian mesh $x = \eta_i$.

We also introduce the mass matrix $\mathbf{M}_{\text{sp}}$ of the spectral element, its FE preconditioner $\boldsymbol{\mathcal{M}}_{\text{sp}}$, generated by the space $\mathcal{H}(\tau_0)$, and $\mathbb{M}_\hbar := \mathbb{D}_\hbar \otimes \mathbb{D}_\hbar \otimes \mathbb{D}_\hbar$.

Lemma 1. *Uniformly in p*

$$\mathbb{A}_\hbar, \boldsymbol{\mathcal{A}}_{\text{sp}} \prec \mathbf{A}_{\text{sp}} \prec \boldsymbol{\mathcal{A}}_{\text{sp}}, \mathbb{A}_\hbar \, , \qquad \mathbb{M}_\hbar, \boldsymbol{\mathcal{M}}_{\text{sp}} \prec \mathbf{M}_{\text{sp}} \prec \boldsymbol{\mathcal{M}}_{\text{sp}}, \mathbb{M}_\hbar \, .$$

Proof. The inequalities for $\boldsymbol{\mathcal{A}}_{\mathrm{sp}}$ in 1-d are due to [1], for the step to a greater dimension see, *e.g.,* [4]. With the inequalities for $\boldsymbol{\mathcal{A}}_{\mathrm{sp}}$ hold, the inequalities for $\mathbb{A}_{\hbar}$ are easy to obtain.

Now we will introduce factored preconditioners. The rest of this section and Section 3 deal with matrices related to the internal unknowns. Usually they are supplied with the lower index I, but in many instances we omit this index. Without loss of generality it is assumed $p = 2N$.

The change of variables $\widetilde{\mathbf{v}} = \mathbf{C}\mathbf{v}$ by the diagonal matrix $\mathbf{C} = p^{-4}\,\mathbb{D}_{\hbar}^{-1/2} \otimes \mathbb{D}_{\hbar}^{-1/2} \otimes \mathbb{D}_{\hbar}^{-1/2}$ (for 2-d $\mathbf{C} = p^{-2}\,\mathbb{D}_{\hbar}^{-1/2} \otimes \mathbb{D}_{\hbar}^{-1/2}$) transforms $\mathbb{A}_{I,\hbar}$ as the matrix of a quadratic form into the matrix $\widetilde{\mathbb{A}}_{I,\hbar} := \mathbf{C}^{-1}\mathbb{A}_{\hbar}\mathbf{C}^{-1}$. Let us introduce also $(p-1)\times(p-1)$ matrices $\boldsymbol{\Delta}_{\mathrm{sp}} = \mathrm{tridiag}\,[-1, 2, -1]$ and $\boldsymbol{\mathcal{D}}_{\mathrm{sp}} = \mathrm{diag}\,[1, 4, .., N^2, (N-1)^2, (N-2)^2, .., 4, 1]$, and the $(p-1)^3 \times (p-1)^3$ matrix

$$\boldsymbol{\Lambda}_{I,\mathrm{sp}} = \boldsymbol{\Delta}_{\mathrm{sp}} \otimes \boldsymbol{\mathcal{D}}_{\mathrm{sp}} \otimes \boldsymbol{\mathcal{D}}_{\mathrm{sp}} + \boldsymbol{\mathcal{D}}_{\mathrm{sp}} \otimes \boldsymbol{\Delta}_{\mathrm{sp}} \otimes \boldsymbol{\mathcal{D}}_{\mathrm{sp}} + \boldsymbol{\mathcal{D}}_{\mathrm{sp}} \otimes \boldsymbol{\mathcal{D}}_{\mathrm{sp}} \otimes \boldsymbol{\Delta}_{\mathrm{sp}}\,. \qquad (1)$$

Lemma 2. *Matrices $\widetilde{\mathbb{A}}_{I,\hbar}$, $\boldsymbol{\Lambda}_{I,\mathrm{sp}}$ and simultaneously the matrix $\boldsymbol{\Lambda}_{I,C} := \mathbf{C}\boldsymbol{\Lambda}_{I,\mathrm{sp}}\mathbf{C}$ and the stiffness matrix $\mathbf{A}_{I,\mathrm{sp}}$ are spectrally equivalent uniformly in p.*

See [11, 12] for the proof.

Since matrix $\mathbf{C}$ is diagonal, the arithmetical costs of solving systems with matrices $\boldsymbol{\Lambda}_{I,C}$ and $\boldsymbol{\Lambda}_{I,\mathrm{sp}}$ are the same in the order. Matrix $\boldsymbol{\Lambda}_{I,\mathrm{sp}}$ looks exactly as the 7-point finite-difference approximation on the *uniform square mesh* of size $\hbar = 2/p = 1/N$ of the differential operator

$$L_{\mathrm{sp}}u = -\left[\phi^2(x_2)\phi^2(x_3)u_{,1,1} + \phi^2(x_1)\phi^2(x_3)u_{,2,2} + \phi^2(x_1)\phi^2(x_2)u_{,3,3}\right]\,,$$

$u|_{\partial\tau_0} = 0$, where $\phi(x) = \min(x+1, x-1)$, $x \in [-1, 1]$. Indeed, for $\phi_i := \phi(-1 + i\hbar)$ and $\mathbf{u} = (u_{\mathbf{i}})_{i_1,i_2,i_3=1}^{p-1}$ expanded by zero to the boundary nodes

$$\boldsymbol{\Lambda}_{I,\mathrm{sp}}\mathbf{u}|_{\mathbf{i}} = -\frac{1}{\hbar^2}\sum_{k=1,2,3}\phi_{i_l}^2\phi_{i_j}^2\left[u_{\mathbf{i}-\mathbf{e}_k} - 2u_{\mathbf{i}} + u_{\mathbf{i}+\mathbf{e}_k}\right]\,, \quad 1 \le i_m \le (p-1)\,,$$

where $\mathbf{i} = (i_1, i_2, i_3)$, all numbers $k, l, j \in (1, 2, 3)$ are different, $\mathbf{e}_k = (\delta_{k,l})_{l=1}^3$, and $\delta_{k,l}$ are Kronecker's symbols.

We compare $\boldsymbol{\Lambda}_{I,\mathrm{sp}}$ with the finite-difference preconditioner for the hierarchical reference element, see, *e.g.,* $\boldsymbol{\Lambda}_{\mathrm{e}}$ in (2.5) of [11]. At $d = 2$, the related differential operators L_{sp} L, respectively, are similar, see for L (2.7), (2.8) in the same paper. In each quarter of τ_0, the differential expression for L_{sp} is the same as for L, defined on the square $(0, 1)^2$, up to the constant multiplier and rotation and translation of the axes. The same is true for finite-difference operators $\boldsymbol{\Lambda}_{I,\mathrm{sp}}$, $\boldsymbol{\Lambda}_{\mathrm{e}}$. At $d = 3$, the differential and finite-difference operators, related to the preconditioners for spectral and hierarchical elements, are different even in the order: L_{sp} is the differential operator of the 2-nd order, whereas L of the 4-th. However, multipliers $\boldsymbol{\mathcal{D}}_{\mathrm{sp}}$, $\boldsymbol{\Delta}_{\mathrm{sp}}$ and respectively $\boldsymbol{\Delta}$, $\boldsymbol{\mathcal{D}}$ in the representations of $\boldsymbol{\Lambda}_{I,\mathrm{sp}}$, $\boldsymbol{\Lambda}_{\mathrm{e}}$ by the sums of Kronecker's products are similar, see (1) above and (2.5) of [11]. Due to this, all known fast solvers for systems with the stiffness matrices of hierarchical reference elements can be adapted to systems with the stiffness matrices of spectral reference elements of any of the two types. We present two examples in the next section.

Instead of $\boldsymbol{\Lambda}_{I,\mathrm{sp}}$, one can as well use spectrally equivalent FE matrices, generated with the use of the 1-st order elements.

3 Fast Multilevel Wavelet Preconditioners-solvers for Interior of Reference Element and Face Problems

In order to obtain a fast preconditioner-solver for the internal stiffness matrix $\mathbf{A}_{I,\text{sp}}$ of a spectral element, it is sufficient to design a fast solver for the preconditioner $\boldsymbol{\Lambda}_{I,\text{sp}}$. For convenience, it is assumed $p = 2N,\ N = 2^{\ell_0 - 1}$.

For each $l = 1, 2, ..., \ell_0$, we introduce the uniform mesh $x_i^l,\ i = 0, 1, .., 2N_l$, $N_l = 2^{l-1}$, $x_0 = -1$, $x_{2N_l} = 1$ of the size $\hbar_l = 2^{1-l}$ and the space $\mathcal{V}_l(-1,1)$ of the continuous on $(-1,1)$ piece wise linear functions, vanishing at the ends of this interval. The dimension of $\mathcal{V}_l(-1,1)$ is $\mathcal{N}_l := p_l - 1 = 2^l - 1$ with $p_{\ell_0} = p$. Let $\phi_i^l \in \mathcal{V}_l(-1,1)$ be the the nodal basis function for the node x_i^l, so that $\phi_i^l(x_j^l) = \delta_{i,j}$ and $\mathcal{V}_l(-1,1) = \text{span}\,\{\phi_i^l\}_{i=1}^{p_l - 1}$. For the Gram matrices

$$\boldsymbol{\Delta}_l = \hbar_l \left(\langle (\phi_i^l)', (\phi_j^l)' \rangle_{\omega=1} \right)_{i,j=1}^{p_l - 1}, \quad \boldsymbol{\mathcal{M}}_l = \hbar_l^{-1} \left(\langle \phi_i^l, \phi_j^l \rangle_{\omega=\phi} \right)_{i,j=1}^{p_l - 1}$$

with ϕ introduced in Section 2 and $\langle v, u \rangle_\omega := \int_{-1}^{1} \omega^2 v\, u\, dx$, we establish

$$\boldsymbol{\Delta}_{\ell_0} = \boldsymbol{\Delta}_{\text{sp}}, \qquad \boldsymbol{\mathcal{M}} := \boldsymbol{\mathcal{M}}_{\ell_0} \asymp \boldsymbol{\mathcal{D}}_{\text{sp}}.$$

The representation $\mathcal{V}_l = \mathcal{V}_{l-1} \oplus \mathcal{W}_l$ results in the decomposition $\mathcal{V} = \mathcal{W}_1 \oplus \mathcal{W}_2 \oplus ... \oplus \mathcal{W}_{\ell_0}$ with the notations $\mathcal{V} = \mathcal{V}_{\ell_0}$ and $\mathcal{W}_1 = \mathcal{V}_1$. Let $\{\psi_k^l\}_{k,l=1}^{p_l - 1,\, \ell_0}$ denote the *multiscale wavelet basis*, composed of some *single scale bases* $\{\psi_k^l\}_{k=1}^{p_l - 1}$ in the spaces $\mathcal{W}_l = \text{span}\,\{\psi_k^l\}_{k=1}^{p_l - 1}$. It generates the matrices

$$\boldsymbol{\Delta}_{\text{wlet}} = \left(\langle (\psi_i^k)', (\psi_j^l)' \rangle_1 \right)_{i,j=1,\, k,l=1}^{p_l - 1, \ell_0}, \quad \boldsymbol{\mathcal{M}}_{\text{wlet}} = \left(\langle \psi_i^k, \psi_j^l \rangle_\phi \right)_{i,j=1,\, k,l=1}^{p_l - 1, \ell_0},$$

$$\mathbb{D}_1 = \text{diag}\,[\langle (\psi_i^l)', (\psi_i^l)', \rangle_1]_{i,l=1}^{p_l - 1, \ell_0}, \qquad \mathbb{D}_0 = \text{diag}\,[\langle \psi_i^l, \psi_i^l, \rangle_\phi]_{i,l=1}^{p_l - 1, \ell_0}.$$

The transformation matrix from the multiscale wavelet basis to the FE basis $\{\phi_k^{l_0}\}_{k=1}^{p-1}$ is denoted by $\mathbf{Q}$. Thus, if $\mathbf{v}$ and $\mathbf{v}_{\text{wlet}}$ are the vectors of the coefficients of a function from $\mathcal{V}(0,1)$, represented in these two bases, respectively, then $\mathbf{v} = \mathbf{Q}^\top \mathbf{v}_{\text{wlet}}$.

Theorem 1. *There exist multiscale wavelet bases, such that* $\boldsymbol{\Delta}_{\text{sp}}^{-1} \asymp \mathbf{Q}^\top \mathbb{D}_1^{-1} \mathbf{Q}$, $\boldsymbol{\mathcal{M}}_{\text{sp}}^{-1} \asymp \mathbf{Q}^\top \mathbb{D}_0^{-1} \mathbf{Q}$, *and matrix-vector multiplications* $\mathbf{Q}\,\mathbf{w}$, $\mathbf{Q}^\top \mathbf{w}$ *require* $\mathcal{O}(p)$ *arithmetic operations.*

Proof. The proof is simpler than the proof of similar results in [3], because the weight ϕ is symmetric on (-1,1). The cited authors justified existence of multiscale wavelet bases with the required properties in the case of the space $\mathcal{V}(0,1) := \{\, v \in \mathcal{V}(-1,1) \,|\, v(x) = 0 \text{ at } x \notin (0,1) \,\}$ and the weight $\phi = x$.

Theorem 2. *Let* $\boldsymbol{\mathcal{B}}_{I,\text{sp}} = \mathbf{C}\, \mathbb{B}_{I,\text{sp}} \mathbf{C}$ *and*

$$\mathbb{B}_{I,\text{sp}}^{-1} = \begin{cases} (\mathbf{Q}^\top \otimes \mathbf{Q}^\top)[\mathbb{D}_0 \otimes \mathbb{D}_1 + \mathbb{D}_1 \otimes \mathbb{D}_0]^{-1} (\mathbf{Q} \otimes \mathbf{Q}), & d = 2, \\ (\mathbf{Q}^\top \otimes \mathbf{Q}^\top \otimes \mathbf{Q}^\top)[\mathbb{D}_0 \otimes \mathbb{D}_0 \otimes \mathbb{D}_1 + \mathbb{D}_0 \otimes \mathbb{D}_1 \otimes \mathbb{D}_0 + & \\ \qquad \mathbb{D}_0 \otimes \mathbb{D}_0 \otimes \mathbb{D}_1]^{-1} (\mathbf{Q} \otimes \mathbf{Q} \otimes \mathbf{Q}), \mathcal{Q} & d = 3. \end{cases}$$

Then $\boldsymbol{\mathcal{B}}_{I,\text{sp}} \asymp \mathbf{A}_{I,\text{sp}}$ *and, therefore,* $\text{cond}\,[\boldsymbol{\mathcal{B}}_{I,\text{sp}}^{-1} \mathbf{A}_{I,\text{sp}}] \prec 1$. *The arithmetical cost of the operation* $\boldsymbol{\mathcal{B}}_{I,\text{sp}}^{-1} \mathbf{v}$ *for any vector* $\mathbf{v}$ *is* $\mathcal{O}(p^d)$.

Proof. In view of Lemmas 1 and 2, it is sufficient to prove the equivalence cond $[\mathbb{B}_{I,\mathrm{sp}}^{-1}\boldsymbol{\Lambda}_{I,\mathrm{sp}}] \asymp 1$. The last is the consequence of the mentioned above relationships $\boldsymbol{\Delta}_{\ell_0} = \boldsymbol{\Delta}_{\mathrm{sp}}$ and $\boldsymbol{\mathcal{M}} := \boldsymbol{\mathcal{M}}_{\ell_0} \asymp \boldsymbol{\mathcal{D}}_{\mathrm{sp}}$, Theorem 1, and the representations of the involved matrices by the corresponding sums of Kronecker products.

Another important problem for DD algorithms is development of fast solvers for internal problems on faces. As it is now known, see, *e.g.,* [15], at nonsignificant lost in the condition, it is reduced to the preconditioning of the matrix of the quadratic form ${}_{00}|\cdot|^2_{1/2,\tau_0}$, $\tau_0 = (-1,1) \times (-1,1)$, on the subspace of polynomials $\mathring{\mathcal{Q}}_{p,\mathbf{x}}$ of two variables $\mathbf{x} = (x_1, x_2)$, vanishing on the boundary $\partial\tau_0$. Here ${}_{00}|\cdot|_{1/2,\tau_0}$ is the norm in the space $H^{1/2}_{00}(\tau_0)$, with the square τ_0 representing a typical face of the 3-d reference cube.

Theorem 3. *Let $d_{0,i}, d_{1,i}$ be diagonal entries of the matrices $\mathbb{D}_0, \mathbb{D}_1$, respectively, and $\mathbb{D}_{1/2}$ be the diagonal matrix with the entries on the main diagonal*

$$d^{(1/2)}_{i,j} = d_{0,i}d_{0,j}\sqrt{d_{1,i}/d_{0,i} + d_{1,j}/d_{0,j}}.$$

Let also $\boldsymbol{\mathcal{S}}_0 = \mathbf{C}\,\mathbb{S}_0\,\mathbf{C}$ and $\mathbb{S}_0^{-1} = (\mathbf{Q}^\top \otimes \mathbf{Q}^\top)\,\mathbb{D}^{-1}_{1/2}\,(\mathbf{Q} \otimes \mathbf{Q})$. Then for all $v \in \mathring{\mathcal{Q}}_{p,\mathbf{x}}$ and vectors $\mathbf{v}$, representing v in the basis $\mathring{\mathcal{M}}_{2,p}$, the norms ${}_{00}|\,v\,|_{1/2,F_0}$ and $||\mathbf{v}||_{\boldsymbol{\mathcal{S}}_0}$ are equivalent uniformly in p.

Proof. For the square $\tau_0 = (0,1)^2$, we have the preconditioner $\boldsymbol{\mathcal{B}}_{I,\mathrm{sp}} = \mathbf{C}\,\mathbb{B}_{I,\mathrm{sp}}\mathbf{C}$ for the stiffness matrix $\mathbf{A}_{I,\mathrm{sp}}$. Similarly, we can define the preconditioner $\boldsymbol{\mathcal{M}}_{I,\mathrm{sp}} = \mathbf{C}\,\mathbb{M}_{I,\mathrm{sp}}\mathbf{C}$ for the internal mass matrix $\mathbf{M}_{I,\mathrm{sp}}$ with $\mathbb{M}^{-1}_{I,\mathrm{sp}} = (\mathbf{Q}^\top \otimes \mathbf{Q}^\top)[\mathbb{D}_0 \otimes \mathbb{D}_0]^{-1}(\mathbf{Q} \otimes \mathbf{Q})$. The further proof is produced by Peetre's K-interpolation method.

Presented fast solvers for the internal and face problems can be easily generalized on the "orthotropic" spectral elements with the shape polynomials having different orders along different axes.

4 Domain Decomposition Algorithm for Discretizations by Spectral Elements

Let we have to solve the problem

$$a_\Omega(u,v) := \int_\Omega \varrho(\mathbf{x})\nabla u \cdot \nabla v\, d\mathbf{x} = (f,v)_\Omega\,, \qquad \forall\, v \in \mathring{H}^1(\Omega)\,,$$

in the domain $\overline{\Omega} = \cup_{r=1}^{\mathcal{R}}\overline{\tau}_r$, which is an assemblage of compatible and in general curvilinear finite elements occupying domains τ_r. We assume that the finite elements are specified by means of non degenerate mappings $\mathbf{x} = \mathcal{X}^{(r)}(\mathbf{y}) : \overline{\tau}_0 \to \overline{\tau}_r$ satisfying the generalized conditions of the angular quasiuniformity, see, *e.g.*, [10]. The coefficient ϱ in the DD algorithm under consideration may be piece wise constant, but for brevity we imply $\varrho(\mathbf{x}) \equiv 1$. For the system $\mathbf{Ku} = \mathbf{f}$ of FE equations, we apply PCG (Preconditioned Conjugate Gradient Method) with the DD preconditioner

$$\boldsymbol{\mathcal{K}}^{-1} = \boldsymbol{\mathcal{K}}_I^{+} + \mathbf{P}_{V_B\to V}\boldsymbol{\mathcal{S}}_B^{-1}\mathbf{P}^\top_{V_B\to V}\,, \quad \boldsymbol{\mathcal{S}}_B^{-1} = \boldsymbol{\mathcal{S}}_F^{+} + \mathbf{P}_{V_W\to V_B}(\boldsymbol{\mathcal{S}}_W^B)^{-1}\mathbf{P}^\top_{V_W\to V_B}\,,$$

of the same structure as in [9, 10]. The involved in the preconditioner matrices are defined as follows.

i) $\boldsymbol{\mathcal{K}}_I = \mathrm{diag}\,[h_1\boldsymbol{\mathcal{B}}_{I,\mathrm{sp}}, h_2\boldsymbol{\mathcal{B}}_{I,\mathrm{sp}}, \dots, h_{\mathcal{R}}\boldsymbol{\mathcal{B}}_{I,\mathrm{sp}}]$ is the block diagonal preconditioner for the internal Dirichlet problems on FE's, where $\boldsymbol{\mathcal{B}}_{I,\mathrm{sp}}$ is the multiresolution wavelet preconditioner-solver found in Theorem 2 and h_r is the characteristic size of a finite element τ_r.

ii) $\boldsymbol{\mathcal{S}}_F = \mathrm{diag}\,[\kappa_1\boldsymbol{\mathcal{S}}_0, \kappa_2\boldsymbol{\mathcal{S}}_0, \dots, \kappa_Q\boldsymbol{\mathcal{S}}_0]$ is the block diagonal preconditioner for the internal problems on faces of finite elements, where $\boldsymbol{\mathcal{S}}_0$ is the multiresolution wavelet preconditioner for one face, defined in Theorem 3, Q is the number of different faces $F_k \subset \Omega$, and κ_k are multipliers. Let for a face F_k of the discretization, $r_1(k)$ and $r_2(k)$ are the numbers of two elements $\overline{\tau}_{r_1(k)}$ and $\overline{\tau}_{r_2(k)}$, sharing the face F_k. Then $\kappa_k = (h_{r_1(k)} + h_{r_2(k)})$.

iii) The preconditioner $\boldsymbol{\mathcal{S}}_W^B$ for the wire basket subproblem. We borrow it, as well as the prolongation $\mathbf{P}_{V_W \to V_B}$, from [4], see also [15]. Let us note that the solving procedure for the system with the matrix $\boldsymbol{\mathcal{S}}_W^B$ is described in these papers up to solution of the sparse subsystem of the order $\mathcal{O}(\mathcal{R}) \times \mathcal{O}(\mathcal{R})$. We assume that there is a solver for this subsystem with the arithmetical cost not greater $\mathcal{O}(\mathcal{R}p^3)$.

iv) The matrix $\mathbf{P}_{V_B \to V}$ performs prolongations from the interelement boundary on the computational domain $\overline{\Omega}$. Its restriction to each FE is the master prolongation $\mathbb{P}_0$ defined for the reference element. For $\forall\, \mathbf{v}_B$, living on $\partial\tau_0$, we set $\mathbb{P}_0\mathbf{v}_B := \mathbf{u}$ with the subvectors $\mathbf{u}_I, \mathbf{u}_B$, where $\mathbf{u}_B := \mathbf{v}_B$ and $\mathbf{u}_I := \overline{\mathbf{v}}_I + \mathbb{P}_{\mathrm{it}}(\mathbf{v}_B - \overline{\mathbf{v}}_B)$, where $\overline{\mathbf{v}}, \overline{\mathbf{v}}_I, \overline{\mathbf{v}}_B$ have for its entries the mean value on $\partial\tau_0$ of the polynomial $v \in \mathcal{O}_{p,\mathrm{x}}$, $v \leftrightarrow \mathbf{v}_B$. The matrix $\mathbb{P}_{\mathrm{it}}$ is implicitly defined by the fixed number $k_0 \asymp (1 + \log p)$ of the iterations $\mathbf{w}_I^{k+1} = \mathbf{w}_I^k - \sigma_{k+1}\boldsymbol{\mathcal{B}}_{I,\mathrm{sp}}^{-1}[\boldsymbol{\mathcal{A}}_{I,\mathrm{sp}}\mathbf{w}_I^k - \boldsymbol{\mathcal{A}}_{IB,\mathrm{sp}}(\mathbf{v}_B - \overline{\mathbf{v}}_B)]$, $\mathbf{w}_I^0 = \mathbf{0}$, with Chebyshev iteration parameters σ_k, so that $\mathbf{u}_I = \overline{\mathbf{v}}_I + \mathbf{w}^{k_0}$. Above, indices I, B are used for the subvectors, living on τ_0 and $\partial\tau_0$, respectively, and for the corresponding blocks of matrices, so that $\boldsymbol{\mathcal{A}}_{I,\mathrm{sp}}, \boldsymbol{\mathcal{A}}_{IB,\mathrm{sp}}$ are the blocks of $\boldsymbol{\mathcal{A}}_{I,\mathrm{sp}}$, which in the iteration process can be replaced by the blocks $\mathbb{A}_{\hbar,I}, \mathbb{A}_{\hbar,IB}$ of $\mathbb{A}_{\hbar}$.

Theorem 4. *The DD preconditioner-solver* $\boldsymbol{\mathcal{K}}$ *provides the condition number* $\mathrm{cond}\,[\boldsymbol{\mathcal{K}}^{-1}\mathbf{K}] \le c(1+\log p)^2$*, whereas for any* $\mathbf{f}$ *the arithmetical cost of the operation* $\boldsymbol{\mathcal{K}}^{-1}\mathbf{f}$ *is* $\mathcal{O}(p^3(1+\log p)\mathcal{R})$.

See [13] for the proof. Changes in the definition of $\boldsymbol{\mathcal{K}}$ allowing to retain Theorem 4 in the case of variable ϱ, $\varrho \asymp \overline{\varrho}$, where $\overline{\varrho} > 0$ is any element wise constant function, are obvious. Parallelization, robustness and h-adaptivity properties of the designed DD solver are exactly the same as for the DD solver in the case of hierarchical elements presented in [9], see also [13]. However, p-adaptivity is less flexible due to the Lagrange interpolation nature of spectral elements.

Acknowledgement. Research is supported by Russian Fund of Basic Research, grant 05-01-00779-a. The first author has partially been supported by the Austrian Science Fund (FWF) through the special research program (SFB) F013, subproject 6.

References

[1] C. Bernardi and Y. Maday. Polynomial interpolation results in Sobolev spaces. *J. Comput. Appl. Math.*, 43:53–80, 1992.

[2] S. Beuchler. Multigrid solver for the inner problem in domain decomposition methods for p-fem. *SIAM J. Numer. Anal.*, 40(4):928–944, 2002.

[3] S. Beuchler, R. Schneider, and Ch. Schwab. Multiresolution weighted norm equivalence and applications. *Numer. Math.*, 98(1):67–97, 2004.

[4] M. Casarin. Quasi-optimal Schwarz methods for the conforming spectral element discretization. *SIAM J. Numer. Anal.*, 34(6):2482–2502, 1997.

[5] S.A. Ivanov and V.G. Korneev. Preconditioning in the domain decomposition methods for the p-version with the hierarchical bases. *Matematicheskoie modelirovanie (Mathematical Modeling)*, 8(9):63–73, 1996.

[6] V.G. Korneev. Almost optimal method for solving Dirichlet problems on subdomains of decomposition of hierarchical hp–version. *Differetsial'nye uravnenia (Differential equations)*, 37(7):958–968, 2001. In Russian.

[7] V.G. Korneev. Local Dirichlet problems on subdomains of decomposition in hp discretizations, and optimal algorithms for their solution. *Matematicheskoie modelirovanie (Mathematical modelling)*, 14(5):51–74, 2002.

[8] V.G. Korneev and S. Jensen. Preconditioning of the p-version of the finite element method. *Comput. Methods Appl. Mech. Engrg.*, 150(1–4):215–238, 1997.

[9] V.G. Korneev, U. Langer, and L. Xanthis. Fast adaptive domain decomposition algorithms for hp-discretizations of 2-d and 3-d elliptic equations: recent advances. *Hermis-$\mu\pi$: An International Journal of Computer Mathematics and its Applications*, 4:27–44, 2003.

[10] V.G. Korneev, U. Langer, and L. Xanthis. On fast domain decomposition solving procedures for hp-discretizations of 3d elliptic problems. *Comput. Methods Appl. Math.*, 3(4):536–559, 2003.

[11] V.G. Korneev and A. Rytov. On existence of the essential interrelation between spectral and hierarchical elements. In *Mesh methods for boundary value problems and applications. Proceedings of 6-th Allrussian seminar (Materialy 6-ogo vserossiiskogo seminara)*, pages 141–150. Kazan State University, 2005.

[12] V.G. Korneev and A. Rytov. On the interrelation between fast solvers for spectral and hierarchical p elements. *Hermis-$\mu\pi$: An International Journal of Computer Mathematics and its Applications*, 6:99–113, 2005. See also http://www.aueb.gr/pympe/hermis/hermis-volume-6/.

[13] V.G. Korneev and A. Rytov. Domain decomposition algorithm for discretizations of 3-d elliptic equations by spectral elements. Technical Report RICAM Report No. 2006-21, Johann Radon Institute for Comput. and Appl. Math., Austrian Academy of Sciences, 2006.

[14] A. Orzag. Spectral methods for problems in complex geometries. *J. Comput. Phys.*, 37(4):70 – 92, 1980.

[15] L. Pavarino and O.B. Widlund. Polylogarithmic bound for an iterative substructuring method for spectral elements in three dimensions. *SIAM J. Num. Anal.*, 37(4):1303–1335, 1996.

[16] J. Shen, F. Wang, and J. Xu. A finite element multigrid preconditioner for Chebyshev-collocation methods. *Appl. Numer. Math.*, 33:471–477, 2000.

Reinforcement-Matrix Interaction Modeled by FETI Method

Jaroslav Kruis and Zdeněk Bittnar

Czech Technical University in Prague, Faculty of Civil Engineering, Department of Mechanics, Thákurova 7, 166 29 Prague, Czech Republic.
{jk,bittnar}@fsv.cvut.cz

1 Introduction

There are composite materials created from two constituents, composite matrix and reinforcement. The reinforcement is usually significantly stiffer than the composite matrix and proper orientation of the reinforcement leads to excellent overall properties of the composite materials. Interaction between the reinforcement and the composite matrix is very important. Perfect or imperfect bonding between the reinforcement and matrix may occur. The perfect bonding takes place only for lower level of applied loads. The perfect bonding occurs when there is no slip between interface points on fiber and points on composite matrix. In other words, interface points on fiber and matrix have the same displacements. Higher load levels cause debonding which decreases the overall stiffness of the composite. The debonding causes different displacements on the fiber and matrix. A special attention is devoted to the modeling of the interaction between the matrix and reinforcement because it can reduce properties of the composite.

The modeling of the interaction is based on pullout tests. The arrangement of such tests is the following. There is a composite matrix with one embedded fiber which is under tension. The growing force in the fiber causes debonding of matrix-fiber connection and fiber moves out from the matrix. Detailed description of pullout effects is relatively complicated and several simplified approaches are used. This contribution deals with the case with perfect bonding between reinforcement and matrix as well as debonding which is controlled by a linear relationship. The most general model with nonlinear debonding is not studied, but it is in the center of our attention.

This contribution deals with application of the FETI method to bonding or debonding problems. The perfect bonding can be directly described by the classical FETI method while the debonding can be modeled by slightly modified FETI method. The FETI method offers all necessary components for bonding/debonding problems.

2 Overview of the FETI Method

The FETI method was introduced by Farhat and Roux in 1991 in [2]. It is a non-overlapping domain decomposition method which enforces the continuity among subdomains by Lagrange multipliers. The FETI method or its variants have been applied to a broad class of two and three dimensional problems of the second and the fourth order. More details can be found e.g. in [6, 3, 4, 5, 1].

The FETI method will be shortly described on a problem of mechanical equilibrium of a solid body. The finite element method is used for the problem discretization. The equilibrium state minimizes the energy functional

$$\Pi(\mathbf{u}) = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f}, \tag{1}$$

where $\mathbf{u}$ denotes the vector of unknown displacements, $\mathbf{K}$ denotes the stiffness matrix and $\mathbf{f}$ denotes the vector of prescribed forces.

Let the original domain be decomposed to m subdomains. Unknown displacements defined on the j-th subdomain are located in the vector $\mathbf{u}^j$. All unknown displacements are located in the vector

$$\mathbf{u}^T = \left((\mathbf{u}^1)^T, (\mathbf{u}^2)^T, \dots, (\mathbf{u}^m)^T\right) . \tag{2}$$

The stiffness matrix of the j-th subdomain is denoted $\mathbf{K}^j$ and the stiffness matrix of the whole problem has the form

$$\mathbf{K} = \begin{pmatrix} \mathbf{K}^1 & & & \\ & \mathbf{K}^2 & & \\ & & \ddots & \\ & & & \mathbf{K}^m \end{pmatrix} . \tag{3}$$

The nodal loads of the j-th subdomain are located in the vector $\mathbf{f}^j$ and the load vector of the problem has the form

$$\mathbf{f}^T = \left((\mathbf{f}^1)^T, (\mathbf{f}^2)^T, \dots, (\mathbf{f}^m)^T\right) . \tag{4}$$

Continuity among subdomains has the form

$$\mathbf{B}\mathbf{u} = \mathbf{0} \tag{5}$$

where the boolean matrix $\mathbf{B}$ has the form

$$\mathbf{B} = \left(\mathbf{B}^1, \mathbf{B}^2, \dots, \mathbf{B}^m\right) . \tag{6}$$

The matrices $\mathbf{B}^j$ contain only entries equal to $1, -1, 0$. With the previously defined notation, the energy functional has the form

$$\Pi(\mathbf{u}, \boldsymbol{\lambda}) = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f} + \boldsymbol{\lambda}^T\mathbf{B}\mathbf{u} \tag{7}$$

where the vector $\boldsymbol{\lambda}$ contains Lagrange multipliers. Stationary conditions of the energy functional have the form

$$\frac{\partial \Pi}{\partial \mathbf{u}} = \mathbf{Ku} - \mathbf{f} + \mathbf{B}^T \boldsymbol{\lambda} = \mathbf{0} \tag{8}$$

$$\frac{\partial \Pi}{\partial \boldsymbol{\lambda}} = \mathbf{Bu} = \mathbf{0}\,. \tag{9}$$

Equation (8) expresses the equilibrium condition while (9) expresses the continuity condition. The known feature of the FETI method is application of a pseudoinverse matrix in relationship for unknown displacements

$$\mathbf{u} = \mathbf{K}^+ \left(\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}\right) + \mathbf{R}\boldsymbol{\alpha} \tag{10}$$

which stems from floating subdomains. The stiffness matrix of a floating subdomain is singular. The matrix **R** contains the rigid body modes of particular subdomains and the vector $\boldsymbol{\alpha}$ contains amplitudes that specify the contribution of the rigid body motions to the displacements. The pseudoinverse matrix and the rigid body motion matrix can be written in the form

$$\mathbf{K}^+ = \begin{pmatrix} (\mathbf{K}^1)^+ & & & \\ & (\mathbf{K}^2)^+ & & \\ & & \ddots & \\ & & & (\mathbf{K}^m)^+ \end{pmatrix}, \quad \mathbf{R} = \begin{pmatrix} \mathbf{R}^1 & & & \\ & \mathbf{R}^2 & & \\ & & \ddots & \\ & & & \mathbf{R}^m \end{pmatrix}. \tag{11}$$

Besides of utilization of the pseudoinverse matrix, a solvability condition in the form

$$\left(\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}\right) \perp \ker \mathbf{K} = \mathbf{R} \tag{12}$$

has to be taken into account. Substitution of unknown displacements to the continuity condition leads to the form

$$\mathbf{BK}^+\mathbf{B}^T \boldsymbol{\lambda} = \mathbf{BK}^+\mathbf{f} + \mathbf{BR}\boldsymbol{\alpha}\,. \tag{13}$$

The solvability condition can be written in the form

$$\mathbf{R}^T \left(\mathbf{f} - \mathbf{B}^T \boldsymbol{\lambda}\right) = \mathbf{0}\,. \tag{14}$$

Usual notation in the FETI method is the following

$$\mathbf{F} = \mathbf{BK}^+\mathbf{B}^T \tag{15}$$

$$\mathbf{G} = -\mathbf{BR} \tag{16}$$

$$\mathbf{d} = \mathbf{BK}^+\mathbf{f} \tag{17}$$

$$\mathbf{e} = -\mathbf{R}^T\mathbf{f}\,. \tag{18}$$

The continuity and solvability conditions can be rewritten with the defined notation in the form

$$\begin{pmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix}. \tag{19}$$

The system of equations (19) is called the coarse or interface problem.

3 Modification of the Method

The classical FETI method uses the continuity condition (5) which enforces the same displacements at the interface nodes. If there is a reason for different displacements between two neighbor subdomains, the continuity condition transforms itself to a slip condition. The slip condition can be written in the form

$$\mathbf{B}\mathbf{u} = \mathbf{s}\,. \tag{20}$$

The vector $\mathbf{s}$ stores slips between interface nodes. For this moment, the slip is assumed to be prescribed and constant.

Let the boundary unknowns be split to two disjunct parts. The boundary unknowns which satisfy the continuity condition are located in the vector $\mathbf{u}_c$, while the boundary unknowns which satisfy the slip condition are located in the vector $\mathbf{u}_s$. Similarly to the continuity condition in the FETI method, the vectors $\mathbf{u}_c$ and $\mathbf{u}_s$ can be written in the form

$$\mathbf{u}_c = \mathbf{B}_c\mathbf{u} \tag{21}$$

$$\mathbf{u}_s = \mathbf{B}_s\mathbf{u} \tag{22}$$

where $\mathbf{B}_c$ and $\mathbf{B}_s$ are the boolean matrices. Now, the continuity condition has the form

$$\mathbf{B}_c\mathbf{u} = \mathbf{0} \tag{23}$$

and the slip condition has the form

$$\mathbf{B}_s\mathbf{u} = \mathbf{s}\,. \tag{24}$$

The conditions (23) and (24) can be amalgamated to a new interface condition

$$\mathbf{B}\mathbf{u} = \begin{pmatrix} \mathbf{B}_c \\ \mathbf{B}_s \end{pmatrix} \mathbf{u} = \begin{pmatrix} \mathbf{0} \\ \mathbf{s} \end{pmatrix} = \mathbf{c}\,. \tag{25}$$

The energy functional can be rewritten to the form

$$\Pi = \frac{1}{2}\mathbf{u}^T\mathbf{K}\mathbf{u} - \mathbf{u}^T\mathbf{f} + \boldsymbol{\lambda}^T(\mathbf{B}\mathbf{u} - \mathbf{c})\,. \tag{26}$$

The stationary conditions have the form

$$\mathbf{K}\mathbf{u} - \mathbf{f} + \mathbf{B}^T\boldsymbol{\lambda} = \mathbf{0} \tag{27}$$

$$\mathbf{B}\mathbf{u} = \mathbf{c}\,. \tag{28}$$

As was mentioned before, the system of two stationary conditions is accompanied by the solvability condition (12). The expression of the vector $\mathbf{u}$ given in (10) remains the same and the interface condition has the form

$$\mathbf{B}\mathbf{K}^+\mathbf{B}^T\boldsymbol{\lambda} = \mathbf{B}\mathbf{K}^+\mathbf{f} + \mathbf{B}\mathbf{R}\boldsymbol{\alpha} - \mathbf{c} \tag{29}$$

and the solvability condition has the form

$$\mathbf{R}^T\left(\mathbf{f} - \mathbf{B}^T\boldsymbol{\lambda}\right) = \mathbf{0}\,. \tag{30}$$

The coarse problem can be written with the help of notation (15) - (18) in the form

$$\begin{pmatrix} \mathbf{F} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{d} - \mathbf{c} \\ \mathbf{e} \end{pmatrix} . \tag{31}$$

The modified coarse problem (31) differs from the original coarse problem (19) by the vector of prescribed slips **c** on the right hand side.

The prescribed slip between two subdomains is not a common case. On the other hand, the slip often depends on shear stress. Discretized form of equations used in the coarse problem requires a discretized law between slip as a difference of two neighbor displacements and nodal forces as integrals of stresses along element edges. One of the simplest laws is the linear relationship

$$\mathbf{c} = \mathbf{H}\boldsymbol{\lambda} \tag{32}$$

where **H** denotes the compliance matrix. Substitution of (32) to the coarse problem (31) leads to the form

$$\begin{pmatrix} \mathbf{F} + \mathbf{H} & \mathbf{G} \\ \mathbf{G}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \boldsymbol{\lambda} \\ \boldsymbol{\alpha} \end{pmatrix} = \begin{pmatrix} \mathbf{d} \\ \mathbf{e} \end{pmatrix} . \tag{33}$$

It should be noted that the coarse system of equations (33) is usually solved by the modified conjugate gradient method. Details can be found in [3] and [5]. The only difference with respect to the system (19) is the compliance matrix **H**. Only one step, the matrix-vector multiplication, of the modified conjugate gradient method should be changed. The compliance matrix may be a diagonal or nearly diagonal matrix.

4 Numerical Examples

Four cases of bonding/debonding behavior are computed by the classical and modified FETI method. There are always two subdomains. One subdomain represents the composite matrix and the other one represents the fiber. A perfect bonding is described directly by the classical FETI method. The usual continuity condition is used. The displacements of the fiber and composite matrix at a selected point are identical and the situation is depicted in Figure 1 (left).

An imperfect bonding is described by the modified FETI method with the constant compliance matrix **H**. The displacements of a fiber are greater than the displacements of the composite matrix. The greater force is applied, the greater slip occurs. The situation is depicted in Figure 1 (right).

A perfect bonding followed by an imperfect bonding is modeled by the modified FETI method. At the beginning, the compliance matrix is zero which expresses infinitely large stiffness between subdomains. At a certain load level, debonding effect is assumed and the compliance matrix is redefined and it is a constant matrix in the following steps. The displacements of the fiber and matrix are the same at the beginning but then they are different. The situation is depicted in Figure 2 (left).

The last example shows a similar problem to the previous one. The compliance matrix **H** is not assumed constant but the compliances are growing from zero values up to a certain level. It means, that the stiffness is decreasing from infinitely large value to some finite value. The greater force acts, the higher compliance is attained and the greater slip between the fiber and the composite matrix occurs. The situation is depicted in Figure 2 (right).

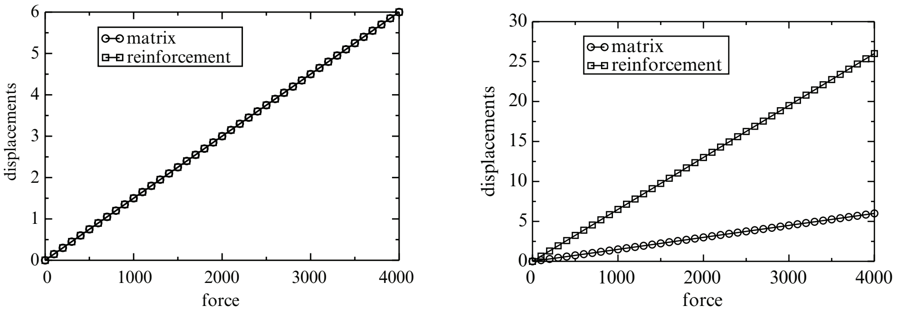


Fig. 1. Perfect bonding (left). Imperfect bonding (debonding) (right).

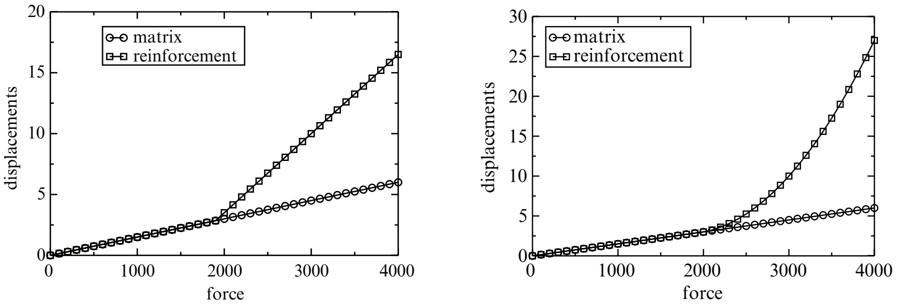


Fig. 2. Imperfect bonding: with delay (left), with changing compliance (right).

5 Conclusions

A slight modification of the FETI method is proposed for problems with the imperfect bonding between the composite matrix and reinforcement. The perfect bonding is modeled by the classical FETI method. Application of a constant compliance matrix leads to linear debonding while a variable compliance matrix can describe nonlinear debonding effects. The advantage of the proposed modification stems from the structure of the compliance matrix which can be nearly diagonal and therefore computationally cheap. The second advantage stems from possible parallelization. Each fiber, generally each piece of reinforcement, as well as the composite matrix can be assigned to one processor and thus large problems may be solved efficiently.

Acknowledgement. Financial support for this work was provided by the Ministry of Education, Youth and Sports of the Czech Republic, project VZ 03 CEZ MSM 6840770003. The financial support is gratefully acknowledged.

References

[1] M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen. Application of the FETI method to ASCI problems—scalability results on 1000 processors and discussion of highly heterogeneous problems. *Internat. J. Numer. Methods Engrg.*, 47:513–535, 2000.

[2] C. Farhat and F. X. Roux. A method of finite element tearing and interconnecting and its parallel solution algorithm. *Internat. J. Numer. Methods Engrg.*, 32:1205–1227, 1991.

[3] C. Farhat and F. X. Roux. Implicit parallel processing in structural mechanics. *Comput. Mech. Adv.*, 2:1–124, 1994.

[4] J. Kruis. *Domain Decomposition Methods for Distributed Computing.* Saxe-Coburg Publications, Kippen, Stirling, Scotland, 2006.

[5] D. J. Rixen, C. Farhat, R. Tezaur, and J. Mandel. Theoretical comparison of the FETI and algebraically partitioned FETI methods, and performance comparisons with a direct sparse solver. *Internat. J. Numer. Methods Engrg.*, 46:501–533, 1999.

[6] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory*, volume 34 of *Springer Series in Computational Mathematics.* Springer-Verlag, Berlin, Germany, 2005.

The Dilemma of Domain Decomposition Approaches in Fluid-Structure Interactions with Fully Enclosed Incompressible Fluids

Ulrich Küttler and Wolfgang A. Wall

Chair of Computational Mechanics
TU Munich
Boltzmannstr. 15, 85747 Garching, Germany
wall@lnm.mw.tum.de

Many popular non-overlapping domain decomposition approaches to fluid-structure interaction (FSI) problems fail to work for an interesting subset of FSI problems, the interaction of highly deformable structures with incompressible but fully enclosed fluids. This is particularly true for coupling approaches based on Dirichlet-Neumann substructuring, both for weak and strong coupling schemes. The breakdown of simulation can be attributed to a lack of knowledge transfer – e.g. of the incompressibility constraint to the structure – between the fields. Another explanation is the absence of any unconstrained outflow boundary at the fluid field, that is the fluid domain is entirely enclosed by Dirichlet boundary conditions. Inflating of a balloon with a prescribed inflow rate constitutes a simple problem of that kind. To overcome the dilemma inherent to partitioned or domain decomposition approaches in these cases a small augmentation is proposed that consists of introducing a volume constraint on the structural system of equations. Additionally the customary applied relaxation of the interface displacements has to be abandoned in favor of the relaxation of coupling forces. These modifications applied to a particular strongly-coupled Dirichlet-Neumann partitioning scheme result in an efficient and robust approach that exhibits only little additional numerical effort. A numerical example with large changes of fluid volume shows the capabilities of the proposed scheme.

1 The Domain Decomposition Approach to FSI Problems

Various solution approaches for FSI problems have been suggested. Most of them are based on a Dirichlet-Neumann partitioning of the coupled problem into fluid and structural part. This constitutes a non-overlapping domain decomposition with fluid field and structural field acting as separate domains. The wet structural surface acts as the coupling interface Γ_{FSI}. These solution schemes require an iterative treatment of the coupling conditions and therefore considerable computational resources, however stability and accuracy are not sacrificed. Additionally these schemes can be built based on available field solvers, which accounts for their constant popularity, see for instance [10, 4, 5, 7, 2, 1, 9, 8].

To sketch the FSI coupling algorithm the structural and fluid problems are abbreviated as follows

$$\mathbf{A}^S \mathbf{d}^S = \mathbf{f}^S \qquad \text{and} \qquad \mathbf{A}^F \mathbf{u}^F = \mathbf{f}^F \tag{1}$$

where both systems are understood to be nonlinear and the fluid system also needs to take the domain deformations into account.

In the following $(\cdot)_I$ and $(\cdot)_\Gamma$ denote variables or coefficients in the interior of a subdomain Ω^j and those coupled at the interface, respectively, while the absence of any subscript comprises degrees of freedom on the entire subdomain including interior and interface.

In every time step the following calculations have to be performed until convergence is reached. The variable i denotes the loop counter.

1. Transfer the latest structure displacements $\mathbf{d}^S_{\Gamma,i+1}$ to the fluid field, calculate the fluid domain deformation and determine the appropriate fluid velocities at the interface $\mathbf{u}^S_{\Gamma,i+1}$.
2. Solve the fluid equation for inner fluid velocities and all (inner and boundary) fluid pressures $\mathbf{u}^F_{I,i+1}$.
$$\mathbf{A}^F_{II} \mathbf{u}^F_{I,i+1} = \mathbf{f}^F_{I\,ext} - \mathbf{A}^F_{I\Gamma} \mathbf{u}^S_{\Gamma,i+1} \,. \tag{2}$$
3. Find the fluid forces $\mathbf{f}^F_{\Gamma,i+1}$ at the interface Γ_{FSI}.
$$\mathbf{f}^F_{\Gamma,i+1} = \mathbf{A}^F_{\Gamma I} \mathbf{u}^F_{I,i+1} + \mathbf{A}^F_{\Gamma\Gamma} \mathbf{u}^S_{\Gamma,i+1} \,. \tag{3}$$
4. Apply the fluid forces $\mathbf{f}^F_{\Gamma,i+1}$ to the structure. Solve the structure equations for the structural displacements.
$$\begin{bmatrix} \mathbf{A}^S_{\Gamma\Gamma} & \mathbf{A}^S_{\Gamma I} \\ \mathbf{A}^S_{I\Gamma} & \mathbf{A}^S_{II} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{d}}^S_{\Gamma,i+1} \\ \mathbf{d}^S_{I,i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^S_{\Gamma\,ext} - \mathbf{f}^F_{\Gamma,i} \\ \mathbf{f}^S_{I\,ext} \end{bmatrix} . \tag{4}$$
5. The calculation is finished when the difference between $\tilde{\mathbf{d}}^F_{\Gamma,i+1}$ and $\mathbf{d}^F_{\Gamma,i}$ is sufficiently small.
6. Relax the interface displacement using a suitable ω_i.
$$\mathbf{d}^S_{\Gamma,i+1} = \omega_i \tilde{\mathbf{d}}^S_{\Gamma,i+1} + (1-\omega_i)\mathbf{d}^S_{\Gamma,i} \,. \tag{5}$$
7. Update i and return to step 1.

Information on the appropriate choice of the relaxation coefficient ω_i can be found in [10, 5].

2 Dilemma with Fully Enclosed, i.e. Dirichlet-Constraint, Fluid Domains

The Dirichlet-Neumann algorithm described above fails if there are prescribed velocities on all boundaries of the fluid domain. A fully Dirichlet-bounded fluid domain can only be solved if (a) the prescribed velocities satisfy the mass balance of the incompressible fluid and (b) the pressure level is fixed by an additional constraint. Standard Dirichlet-Neumann algorithms fail on both conditions. Neither does the fluid domain deformation suggested by the structural solver match the fluid mass balance, nor are there means to transfer any pressure information form the structure to the fluid.

These two difficulties are closely related. The two fields are coupled much closer as compared to FSI problems with free outflow boundaries. Therefore any attempt to overcome the difficulties will result in an algorithm that is more expensive from a numerical point of view.

Several strategies might be pursued to arrive at a working coupling algorithm.

- The interface displacements, that is the structural solution, respect the incompressibility constraint of the fluid. Thus the introduction of a constraint to the structural equations is required. The fluid pressure level will need to be calculated from the structure solution. This approach is presented in detail in the following.
- Another point of departure is the pressure level coupling between structure and fluid. The natural way for the structure to determine the fluid pressure is to transfer interface forces from the structure to the fluid. It follows that the fluid has to prescribe the interface displacements on the structure, that is the Dirichlet-Neumann coupling is reversed to a Neumann-Dirichlet approach. The resulting algorithm, however, is numerically very sensitive and not suitable for general FSI problems. In addition it also runs into the old problem once one has to deal with incompressible solids, too. Details can be found in [3].
- Finally the whole problem is avoided if one can get rid of the incompressibility constraint, at least temporarily. However this also has been shown to be not a very robust or efficient approach. This idea has been pursued in [6] and will not be discussed here.

It is worth noting, that according to the insight discussed so far Dirichlet-Neumann approaches only work in standard examples because the fluid can temporarily escape through the Neumann boundary in staggered situations or during the field iterations in strong coupling schemes.

3 Augmented Dirichlet-Neumann Approach

3.1 Volume Constraint Applied to the Structural Equation

The augmentation of the structural solver to account for the mass balance of the enclosed fluid domain translates to a constraint of the interface displacements to enclose exactly the required volume. The required fluid volume V_c depends upon the Dirichlet boundary conditions of the fluid domain.

$$\begin{aligned} V_c = V^{n+1} &= V^n + \int_{\Gamma^F} \frac{1}{2} \Delta t \left(\mathbf{u}^{n+1} \cdot \mathbf{n} + \mathbf{u}^n \cdot \mathbf{n} \right) \, d\Gamma \\ &= V^n + \int_{\Gamma_{FSI}} \left(\mathbf{r}^{n+1} \cdot \mathbf{n} - \mathbf{r}^n \cdot \mathbf{n} \right) \, d\Gamma \qquad (6) \end{aligned}$$

$$+ \int_{\Gamma_{in} \cup \Gamma_{out}} \frac{1}{2} \Delta t \left(\mathbf{u}^{n+1} \cdot \mathbf{n} + \mathbf{u}^{n} \cdot \mathbf{n} \right) \, d\Gamma$$

where $\mathbf{r}^{n+1}$ and $\mathbf{r}^{n}$ are the interface positions at the time t^{n+1} and t^{n}. The constraint $V_c - V = 0$ is introduced into the structural equation of motion by means of a Lagrangian multiplier λ. This Lagrangian multiplier represents the pressure increment required additional to the fluid pressure in order to satisfy the volume constraint on the structure. Thus the multiplier specifies the physical fluid pressure level.

If the fluid forces at the interface $\mathbf{f}^{F}$ are sufficient to maintain the required volume V_c, the pressure increment λ will be zero. This can be achieved by a coupling algorithm that transfers λ to the fluid partition and adds it to the pressure boundary condition which is used to determine the pressure level. This way the Lagrangian multiplier λ will tend to zero in the course of the coupling iteration.

But changing the pressure boundary condition of the fluid during the coupling iteration means changing the overall problem definition. It is generally advisable to avoid it. Instead the fluid pressure level can be fixed to a constant value in the fluid domain. Of course the resulting pressure increment λ will not vanish in this case. Instead it is added to the fluid pressure values $\overline{p}^{F}$ after the fluid calculation to obtain the final pressure solution $\overline{p}$:

$$\overline{p} = \overline{p}^{F} + \lambda \, . \tag{7}$$

3.2 Modified Dirichlet-Neumann Coupling with Volume Constraint

The iterative coupling algorithm with the volume constraint in the structural equation is a slight modification of the algorithm in section 1. Because the structural solver has to account for the volume condition of the fluid domain, the displacement of the interface cannot be altered once the structural solution is done. In particular the relaxation of the displacements is no longer possible. Instead, because relaxation is needed to enforce and accelerate convergence, one has to relax the fluid forces at the interface.

By means of the symbolic structural and fluid system (1) in every time step the following calculations have to be performed.

1. Solve for the structural displacements loaded with the fluid forces $\mathbf{f}^{F}_{\Gamma,i}$, but respect the volume constraint required by the fluid

$$\begin{bmatrix} \mathbf{A}^{S}_{\Gamma\Gamma} & \mathbf{A}^{S}_{\Gamma I} & -V_{,\mathbf{d}^{S}_{\Gamma}} \\ \mathbf{A}^{S}_{I\Gamma} & \mathbf{A}^{S}_{II} & 0 \\ -V_{,\mathbf{d}^{S}_{\Gamma}} & 0 & 0 \end{bmatrix} \begin{bmatrix} \mathbf{d}^{S}_{\Gamma,i+1} \\ \mathbf{d}^{S}_{I,i+1} \\ \lambda_{i+1} \end{bmatrix} = \begin{bmatrix} \mathbf{f}^{S}_{\Gamma\, ext} - \mathbf{f}^{F}_{\Gamma,i} - V_{,\mathbf{d}^{S}_{\Gamma}} \lambda_i \\ \mathbf{f}^{S}_{I\, ext} \\ V_c - V_{,\mathbf{d}^{S}_{\Gamma}} \mathbf{d}^{S}_{\Gamma,i} \end{bmatrix} . \tag{8}$$

2. Transfer the interface displacements $\mathbf{d}^{S}_{\Gamma,i+1}$ to the fluid and determine the interface velocities $\mathbf{u}^{S}_{\Gamma,i+1}$. Solve for inner fluid velocities and all fluid pressures $\mathbf{u}^{F}_{I,i+1}$

$$\mathbf{A}^{F}_{II} \mathbf{u}^{F}_{I,i+1} = \mathbf{f}^{F}_{I\, ext} - \mathbf{A}^{F}_{I\Gamma} \mathbf{u}^{S}_{\Gamma,i+1} \, . \tag{9}$$

3. Find the fluid forces at the interface Γ_{FSI}

$$\tilde{\mathbf{f}}^{F}_{\Gamma,i+1} = \mathbf{A}^{F}_{\Gamma I} \mathbf{u}^{F}_{I,i+1} + \mathbf{A}^{F}_{\Gamma\Gamma} \mathbf{u}^{S}_{\Gamma,i+1} \, . \tag{10}$$

4. Relax the fluid forces

$$\mathbf{f}^F_{\Gamma,i+1} = \omega_i \tilde{\mathbf{f}}^F_{\Gamma,i+1} + (1-\omega_i)\mathbf{f}^F_{\Gamma,i}\,. \tag{11}$$

   The relaxation parameter ω_i can again be calculated by any of the methods suggested in [5].

The iteration finishes when the error of the fluid boundary force $\tilde{\mathbf{f}}^F_{\Gamma,i+1}$ is sufficiently small.

4 Example: Damped Structural Instability

As an example a bended fluid domain is calculated that is surrounded by two thin structures with neo-Hookean material and different stiffness. The system is shown in figure 1. The structures are fixed at their short edges, the long edges are free respectively interacting with the fluid.

At the fluid domain inflow velocities are prescribed with the left one a little less than the right in order to avoid perfect symmetry. The fluid is loaded with the body force $\mathbf{f}_y = -1N/m^2$ in y direction. The simulation is carried out utilizing the augmented Dirichlet-Neumann algorithm and a uniform time step size $\Delta t = 0.005s$.

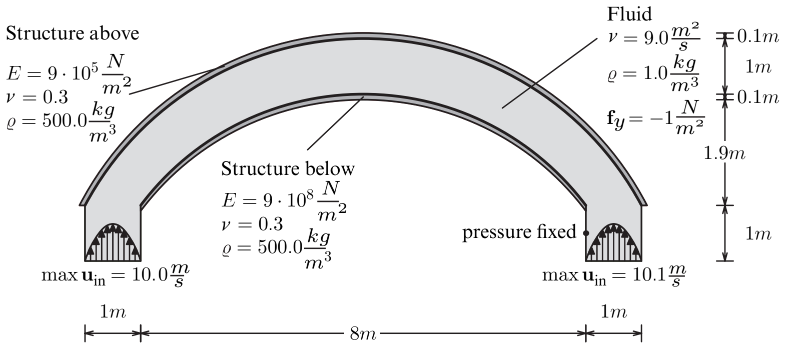

Structure above
$E = 9 \cdot 10^5 \frac{N}{m^2}$
$\nu = 0.3$
$\varrho = 500.0 \frac{kg}{m^3}$
Fluid
$\nu = 9.0 \frac{m^2}{s}$
$\varrho = 1.0 \frac{kg}{m^3}$
$\mathbf{f}_y = -1 \frac{N}{m^2}$
$0.1m$
$1m$
$0.1m$
$1.9m$
$1m$
Structure below
$E = 9 \cdot 10^8 \frac{N}{m^2}$
$\nu = 0.3$
$\varrho = 500.0 \frac{kg}{m^3}$
pressure fixed
$\max \mathbf{u}_{\text{in}} = 10.0 \frac{m}{s}$
$\max \mathbf{u}_{\text{in}} = 10.1 \frac{m}{s}$
$1m$
$8m$
$1m$

Fig. 1. A bended fluid domain with two inflow boundaries constraint by structures of different stiffness.

The constant inflow increases the fluid pressure so that first mainly the soft flexible structure above the fluid domain deforms to make room for the fluid. When a critical pressure value is reached the structure below the fluid collapses, however the instability is damped by the fluid volume constraint. That is why the deformation and the corresponding pressure decrease occur rather slowly. (Since this example is given just in order to demonstrate the augmented Dirichlet-Neumann approach, possible cavitation effects are not considered.) Afterward the system is in motion, the pressure varies rapidly in this phase. The pressure level development, that is the pressure increment λ calculated by the structural solver, is depicted in figure 2.

Figure 3 shows absolute velocities at different time steps.

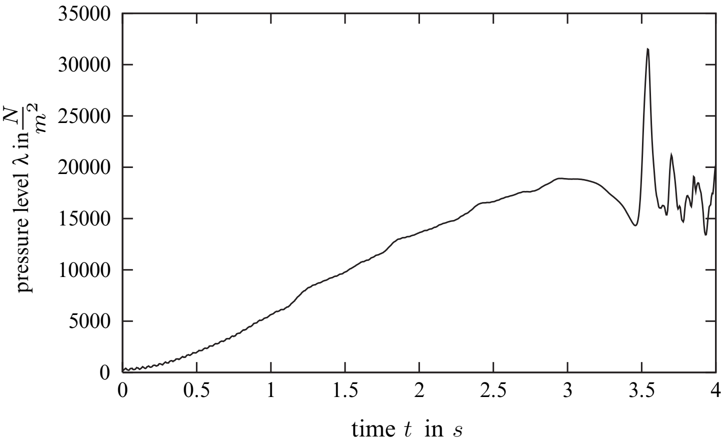


Fig. 2. Pressure level of the bended fluid domain.

5 Conclusion

The dilemma of non-overlapping domain decomposition approaches to FSI problems has been analyzed. Different solution strategies were considered. A small modification to an established iterative solution scheme has been proposed that consists of introducing the incompressibility constraint to the structural solver and results in a reliable and accurate algorithm.

This one condition seriously damages the bandwidth of the system matrix, it couples all displacements on the wet surface. Additionally the positive definiteness of the matrix is lost. Thus the approach is rather expensive from a numerical point of view. A common solver alternative in such a situation would be to use a staggered scheme on the structural side. However, the additional costs pertain the structural solver only. And because the fluid solution costs are dominating in most FSI calculations the proposed algorithm presents a viable approach for many FSI simulations which require Dirichlet constraints on all fluid boundaries.

References

[1] M.Á. Fernández and M. Moubachir. A Newton method using exact jacobians for solving fluid-structure coupling. *Computers & Structures*, 83(2–3):127–142, 2005.

[2] J.-F. Gerbeau, M. Vidrascu, and P. Frey. Fluid-structure interaction in blood flows on geometries coming from medical imaging. *Computers & Structures*, 83:155–165, 2005.

[3] U. Küttler, Ch. Förster, and W. A. Wall. A solution for the incompressibility dilemma in partitioned fluid-structure interaction with pure Dirichlet fluid domains. *Comput. Mech.*, 38(4):417–429, 2006.

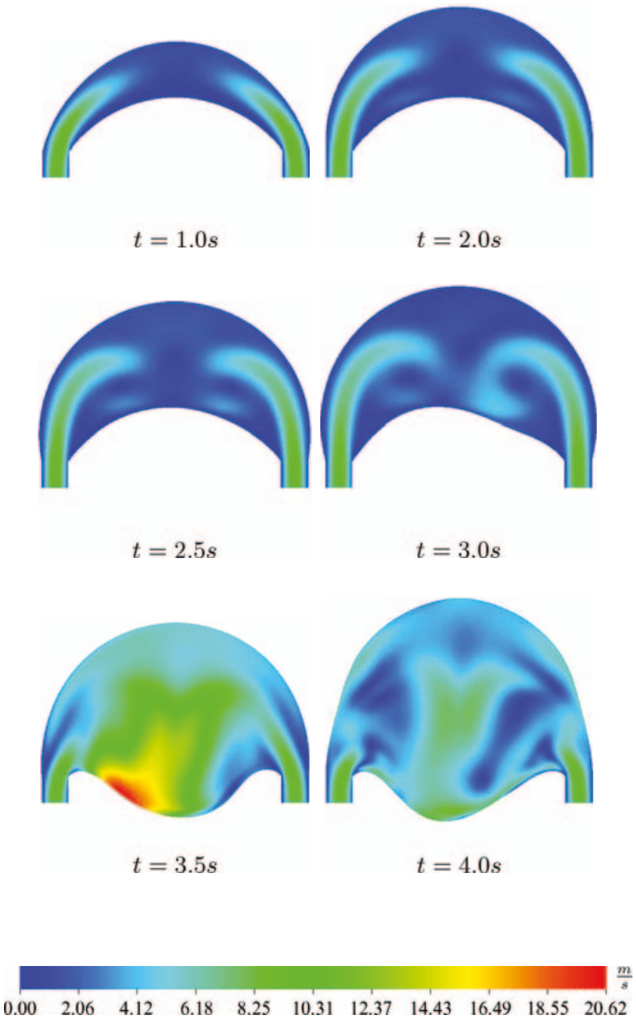


Fig. 3. Fluid velocity $|\mathbf{u}|$. The structural part is not shown.

[4] P. Le Tallec and J. Mouro. Fluid structure interaction with large structural displacements. *Comput. Methods Appl. Mech. Engrg.*, 190:3039–3067, 2001.

[5] D.P. Mok and W.A. Wall. Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures. In W.A. Wall, K.-U. Bletzinger, and K. Schweitzerhof, editors, *Trends in Computational Structural Mechanics*, 2001.

[6] P. Raback, J. Ruokolainen, M. Lyly, and E. Järvinen. Fluid-structure interaction boundary conditions by artificial compressibility. *ECCOMAS*, 2001.

[7] T.E. Tezduyar. Finite element methods for fluid dynamics with moving boundaries and interfaces. In E. Stein, R. De Borst, and T.J.R. Hughes, editors, *Encyclopedia of Computational Mechanics*, volume 3, chapter 17. John Wiley & Sons, 2004.

[8] T.E. Tezduyar and S. Sathe. Modelling of fluid-structure interactions with the space-time finite elements: solution techniques. *Internat. J. Numer. Methods Fluids*, 54(6-8):855–900, 2007.

[9] T.E. Tezduyar, S. Sathe, R. Keedy, and K. Stein. Space-time finite element techniques for computation of fluid-structure interactions. *Comput. Methods Appl. Mech. Engrg.*, 195:2002–2027, 2006.

[10] W.A. Wall, D.P. Mok, and E. Ramm. Partitioned analysis approach of the transient coupled response of viscous fluids and flexible structures. In W. Wunderlich, editor, *Solids, Structures and Coupled Problems in Engineering, Proceedings of the European Conference on Computational Mechanics ECCM '99, Munich*, 1999.

A FETI-DP Method for Mortar Finite Element Discretization of a Fourth Order Problem

Leszek Marcinkowski[1]* and Nina Dokeva[2]

[1] Department of Mathematics, Warsaw University, Banacha 2, 02–097 Warszawa, Poland, lmarcin@mimuw.edu.pl

[2] Department of Mathematics and Computer Science, Clarkson University, PO Box 5815, Potsdam, NY 13699–5815, USA, ndokeva@clarkson.edu

Summary. In this paper we present a FETI-DP type algorithm for solving the system of algebraic equations arising from the mortar finite element discretization of a fourth order problem on a nonconforming mesh. A conforming reduced Hsieh-Clough-Tocher macro element is used locally in the subdomains. We present new FETI-DP discrete problems and later introduce new parallel preconditioners for two cases: where there are no crosspoints in the coarse division of subdomains and in the general case.

1 Introduction

The mortar methods are effective methods for constructing approximations of PDE problems on nonconforming meshes. They impose weak integral coupling conditions across the interfaces on the discrete solutions, cf. [1].

In this paper we present a FETI-DP method (dual primal Finite Element Tearing and Interconnecting, see [6, 9, 8]) for solving discrete problems arising from a mortar discretization of a fourth order model problem. The original domain is divided into subdomains and a local conforming reduced HCT (Hsieh-Clough-Tocher) macro element discretization is introduced in each subdomain. The discrete space is constructed using mortar discretization, see [10]. Then the degrees of freedom corresponding to the interior nodal points are eliminated as usually in all substructuring methods. The remaining system of unknowns is solved by a FETI-DP method.

Many variants of FETI-DP methods for solving systems arising from the discretizations on a single conforming mesh of second and fourth order problems are fully analyzed, cf. [9, 8].

Recently there have been a few FETI-DP type algorithms for mortar discretization of second order problems, cf. [11, 5, 4, 3], and [7].

To our knowledge there are no FETI type algorithms for solving systems of equations arising from a mortar discretization of a fourth order problem in the literature.

* This work was partially supported by Polish Scientific Grant 2/P03A/005/24.

The remainder of the paper is organized as follows. In Section 2 we introduce our differential and discrete problems. When there are no crosspoints in the coarse division of the domain, the FETI operator takes a much simpler form and therefore this case is presented separately together with a parallel preconditioner in Section 3, while Section 4 is dedicated to a short description of the FETI-DP operator and a respective preconditioner in the general case.

2 Differential and Discrete Problems

Let Ω be a polygonal domain in $\mathbb{R}^2$. Then our model problem is to find $u^* \in H_0^2(\Omega)$ such that

$$a(u^*, v) = f(v) \qquad v \in H_0^2(\Omega), \tag{1}$$

where u^* is the displacement, $f \in L^2(\Omega)$ is the body force,

$$a(u,v) = \int_\Omega \left[\triangle u \triangle v + (1-\nu)\left(2u_{x_1x_2}v_{x_1x_2} - u_{x_1x_1}v_{x_2x_2} - u_{x_2x_2}v_{x_1x_1}\right)\right]\, dx.$$

Here

$$H_0^2(\Omega) = \{v \in H^2(\Omega):\ v = \partial_n v = 0 \ \text{ on } \ \partial\Omega\},$$

∂_n is the normal unit derivative outward to $\partial\Omega$, and $u_{x_ix_j} := \frac{\partial^2 u}{\partial x_i \partial x_j}$ for $i,j = 1,2$. We assume that the Poisson ratio ν satisfies $0 < \nu < 1/2$. From the Lax-Milgram theorem and the continuity and ellipticity of the bilinear form $a(\cdot\,,\cdot)$ it follows that there exists a unique solution of this problem.

Next we assume that Ω is a union of disjoint polygonal substructures Ω_i which form a coarse triangulation of Ω, i.e. the intersection of the boundaries of two different subdomains $\partial\Omega_k \cap \partial\Omega_l, k \neq l$, is either the empty set, a vertex or a common edge. We also assume that this triangulation is shape regular in the sense of Section 2, p. 5 in [2].

An important role is played by the interface Γ, defined as the union of all open edges of substructures, which are not on the boundary of Ω.

In each subdomain Ω_k we introduce a quasiuniform triangulation $T_h(\Omega_k)$ made of triangles. Let $h_k = \max_{\tau \in T_h(\Omega_k)} \operatorname{diam} \tau$ be the parameter of this triangulation.

In each Ω_k we introduce a local conforming reduced Hsieh-Clough-Tocher (RHCT) macro finite element space $X_h(\Omega_k)$ as follows, cf. Figure 1:

$$\begin{aligned} X_h(\Omega_k) = \{v \in C^1(\Omega_k):\ & v_{|\tau} \in P_3(\tau_i), \text{ for triangles } \tau_i,\ i = 1,2,3, \\ & \text{formed by connecting the vertices of } \tau \in T_h(\Omega_k) \text{ to} \\ & \text{its centroid, } \partial_n v \text{ is linear on each edge of } \partial\tau, \text{ and} \\ & v = \partial_n v = 0 \text{ on } \partial\Omega_k \cap \partial\Omega\}. \end{aligned} \tag{2}$$

The degrees of freedom of RHCT macro elements are given by

$$\{u(p_i), u_{x_1}(p_i), u_{x_2}(p_i)\}, \quad i = 1,2,3, \tag{3}$$

for the three vertices p_i of an element $\tau \in T_h(\Omega_k)$, cf. Figure 1.

We introduce next an auxiliary global space $X_h(\Omega) = \prod_{k=1}^N X_h(\Omega_k)$, and the so called broken bilinear form:

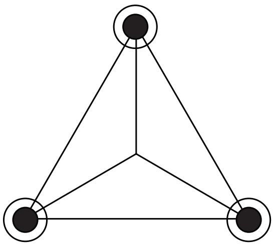


Fig. 1. Reduced HCT element

$$a_h(u,v) = \sum_{k=1}^{N} a_k(u,v),$$

where

$$a_k(u,v) = \int_{\Omega_k} \left[\triangle u \triangle v + (1-\nu)\left(2u_{x_1x_2}v_{x_1x_2} - u_{x_1x_1}v_{x_2x_2} - u_{x_2x_2}v_{x_1x_1}\right)\right] dx.$$

Then let $X(\Omega)$ be the subspace of $X_h(\Omega)$ consisting of all functions which have all degrees of freedom (dofs) of the RHCT elements continuous at the crosspoints – the vertices of the substructures.

The interface Γ_{kl} which is a common edge of two neighboring substructures Ω_k and Ω_l inherits two 1D independent triangulations: $T_{h,k}(\Gamma_{kl})$ – the h_k one from $T_h(\Omega_k)$ and $T_{h,l}(\Gamma_{kl})$ – the h_l one from $T_h(\Omega_l)$. Hence we can distinguish the sides (or meshes) of this interface. Let $\gamma_{m,k}$ be the side of Γ_{kl} associated with Ω_k and called master (mortar) and let $\delta_{m,l}$ be the side corresponding to Ω_l and called slave (nonmortar). Note that both the master and the slave occupy the same geometrical position of Γ_{kl}. The set of vertices of $T_{h,k}(\gamma_{m,k})$ on $\gamma_{m,k}$ is denoted by $\gamma_{m,k,h}$ and the set of nodes of $T_{h,l}(\delta_{m,l})$ on $\delta_{m,l}$ by $\delta_{m,l,h}$. In order to obtain our results we need a technical assumption of a uniform bound for the ratio $h_{\gamma_m}/h_{\delta_m}$ for any interface $\Gamma_{kl} = \gamma_{m,k} = \delta_{m,l}$.

An important role in our algorithm is played by four trace spaces onto the edges of the substructures. For each interface $\Gamma_{kl} = \partial\Omega_k \cap \partial\Omega_l$ let $W_{t,k}(\Gamma_{kl})$ be the space of C^1 continuous functions piecewise cubic on the 1D triangulation $T_{h,k}(\Gamma_{kl})$ and let $W_{n,k}(\Gamma_{kl})$ be the space of continuous piecewise linear functions on $T_{h,k}(\Gamma_{kl})$. The spaces $W_{t,l}(\Gamma_{kl})$ and $W_{n,l}(\Gamma_{kl})$ are defined analogously, but on the h_l triangulation $T_{h,l}(\Gamma_{kl})$ of Γ_{kl}.

Note that these four spaces are the tangential and normal trace spaces onto the interface $\Gamma_{kl} \subset \Gamma$ of functions from $X_h(\Omega_k)$ and $X_h(\Omega_l)$, respectively.

We also need to introduce two test function spaces for each slave $\delta_{m,l} = \Gamma_{kl}$. Let $M_t(\delta_{m,l})$ be the space of all C^1 continuous piecewise cubic on $T_{h,l}(\delta_{m,l})$ functions which are linear on the two end elements of $T_{h,l}(\delta_{m,l})$ and let $M_n(\delta_{m,l})$ be the space of all continuous piecewise linear on $T_{h,l}(\delta_{m,l})$ functions which are constant on the two end elements of $T_{h,l}(\delta_{m,l})$.

We now define the global space $M(\Gamma) = \prod_{\delta_{m,l} \subset \Gamma} M_t(\delta_{m,l}) \times M_n(\delta_{m,l})$ and the bilinear form $b(u,\psi)$ defined over $X(\Omega) \times M(\Gamma)$ as follows: let $u = (u_1, \ldots, u_N) \in X(\Omega)$ and $\psi = (\psi_m)_{\delta_m} = (\psi_{m,t}, \psi_{m,n})_{\delta_m} \in M(\Gamma)$, then let

$$b(u,\psi) = \sum_{\delta_m \subset \Gamma} b_{m,t}(u,\psi_{m,t}) + b_{m,n}(u,\psi_{m,n})$$

with

$$b_{m,t}(u,\psi_{m,t}) = \int_{\delta_m} (u_k - u_l)\psi_{m,y}\, ds \tag{4}$$

$$b_{m,n}(u,\psi_{m,n}) = \int_{\delta_m} (\partial_n u_k - \partial_n u_l)\psi_{m,n}\, ds. \tag{5}$$

Then our discrete problem is to find the pair $(u_h^*, \lambda^*) \in X(\Omega) \times M(\Gamma)$ such that

$$a_h(u_h^*, v) + b(v, \lambda^*) = f(v) \qquad \forall v \in X(\Omega) \tag{6}$$

$$b(u_h^*, \phi) = 0 \qquad \forall \phi \in M(\Gamma). \tag{7}$$

Note that if we introduce the discrete space

$$V^h = \{u \in X(\Omega) : b(u,\phi) = 0 \quad \forall \phi \in M(\Gamma)\}$$

then u_h^* is the unique function in V^h that satisfies

$$a_h(u_h^*, v) = f(v) \qquad \forall v \in V^h,$$

which is a standard mortar discrete problem formulation, cf. e.g. [10].

Note that we can split the matrix $K^{(l)}$ – the matrix representation of $a_l(u,v)$ in the standard nodal basis of $X_h(\Omega_l)$ as:

$$K^{(l)} := \begin{pmatrix} K_{ii}^{(l)} & K_{ic}^{(l)} & K_{ir}^{(l)} \\ K_{ci}^{(l)} & K_{cc}^{(l)} & K_{cr}^{(l)} \\ K_{ri}^{(l)} & K_{rc}^{(l)} & K_{rr}^{(l)} \end{pmatrix}, \tag{8}$$

where in the rows the indices i, c and r refer to the unknowns $u^{(i)}$ corresponding to the interior nodes, $u^{(c)}$ to the crosspoints, and $u^{(r)}$ to the remaining nodes, i.e. those related to the edges.

2.1 Matrix Form of the Mortar Conditions

Note that (7) is equivalent to two mortar conditions on each slave $\delta_{m,l} = \gamma_{m,k} = \Gamma_{kl}$:

$$b_{m,t}(u,\phi) = \int_{\delta_m} (u_k - u_l)\phi\, ds = 0 \qquad \forall \phi \in M_t(\delta_{m,l}) \tag{9}$$

$$b_{m,n}(u,\psi) = \int_{\delta_m} (\partial_n u_k - \partial_n u_l)\psi\, ds = 0 \qquad \forall \psi \in M_n(\delta_{m,l}). \tag{10}$$

Introducing the following splitting of two vectors representing the tangential and normal traces $u_{\delta_{m,l}}$ and $\partial_n u_{\delta_{m,l}}$ we get $u_{\delta_{m,l}} = u_{\delta_{m,l}}^{(r)} + u_{\delta_{m,l}}^{(c)}$ and $\partial_n u_{\delta_{m,l}} = \partial_n u_{\delta_{m,l}}^{(r)} + \partial_n u_{\delta_{m,l}}^{(c)}$ on a slave $\delta_{m,l} \subset \partial\Omega_l$, cf. (8). We can now rewrite (9) and (10) in a matrix form as

$$B_{t,\delta_{m,l}}^{(r)} u_{\delta_{m,l}}^{(r)} + B_{t,\delta_{m,l}}^{(c)} u_{\delta_{m,l}}^{(c)} = B_{t,\gamma_{m,k}}^{(r)} u_{\gamma_{m,k}}^{(r)} + B_{t,\gamma_{m,k}}^{(c)} u_{\gamma_{m,k}}^{(c)}, \tag{11}$$

$$B^{(r)}_{n,\delta_{m,l}}\partial_n u^{(r)}_{\delta_{m,l}} + B^{(c)}_{n,\delta_{m,l}}\partial_n u^{(c)}_{\delta_{m,l}} = B^{(r)}_{n,\gamma_{m,k}}\partial_n u^{(r)}_{\gamma_{m,k}} + B^{(c)}_{n,\gamma_{m,k}}\partial_n u^{(c)}_{\gamma_{m,k}},$$

where the matrices $B_{t,\delta_{m,l}} = (B^{(r)}_{t,\delta_{m,l}},\ B^{(c)}_{t,\delta_{m,l}})$ and $B_{n,\delta_{m,l}} = (B^{(r)}_{t,\gamma_{m,k}}, B^{(c)}_{t,\gamma_{m,k}})$ are mass matrices obtained by substituting the standard nodal basis functions of $W_{t,l}(\delta_{m,l}), W_{n,l}(\delta_{m,l})$ and $M_t(\delta_{m,l}), M_n(\delta_{m,l})$ into (9) and (10), respectively i.e.

$$B_{t,\delta_{m,l}} = \{(\phi_{x,s},\psi_{y,r})\}_{\substack{x,y\in\delta_{m,l,h}\\ s,r=0,1}} \qquad \phi_{x,s}\in W_t(\delta_{m,l}), \psi_{y,r}\in M_t(\delta_{m,l}), \tag{12}$$

$$B_{n,\delta_{m,l}} = \{(\phi_x,\psi_y)\}_{x,y\in\delta_{m,l,h}} \qquad \phi_x\in W_n(\delta_{m,l}), \psi_y\in M_n(\delta_{m,l}), \tag{13}$$

where $\phi_{x,s}$, $(\psi_{x,s}\ )$ is a nodal basis function of $W_t(\delta_{m,l})$, $(M_t(\delta_{m,l}))$ associated with a vertex x of $T_{h,l}(\delta_{m,l})$ and is either a value if $s = 0$ or a derivative if $s = 1$, and $\phi_x \in W_{n,l}(\delta_{m,l})$ and $\psi_x, \in M_n(\delta_{m,l})$ are nodal basis function of these respective spaces equal to one at the node x and zero at all remaining nodal points on $\overline{\delta}_{m,l}$. The matrices $B_{t,\gamma_{m,k}} = (B^{(r)}_{n,\delta_{m,l}},\ B^{(c)}_{n,\delta_{m,l}})$, and $B_{n,\gamma_{m,k}} = (B^{(r)}_{n,\gamma_{m,k}},\ B^{(c)}_{n,\gamma_{m,k}})$ are defined analogously.

Note that $B^{(r)}_{t,\delta_{m,l}}, B^{(r)}_{n,\delta_{m,l}}$ are positive definite square matrices, see e.g. [10], but the other matrices in (11) are in general rectangular.

We also need the block-diagonal matrices

$$B_{\delta_{m,l}} = \begin{pmatrix} B_{t,\delta_{m,l}} & 0 \\ 0 & B_{n,\delta_{m,l}} \end{pmatrix} \quad B_{\gamma_{k,l}} = \begin{pmatrix} B_{t,\gamma_{k,l}} & 0 \\ 0 & B_{n,\gamma_{k,l}} \end{pmatrix}. \tag{14}$$

3 FETI-DP Problem – No Crosspoints Case

In this section we present a FETI-DP formulation for the case with no crosspoints, i.e. two subdomains are either disjoint or have a common edge, cf. Figure 2. In this case both the FETI-DP problem and the preconditioner are fully parallel and simple to describe and implement.

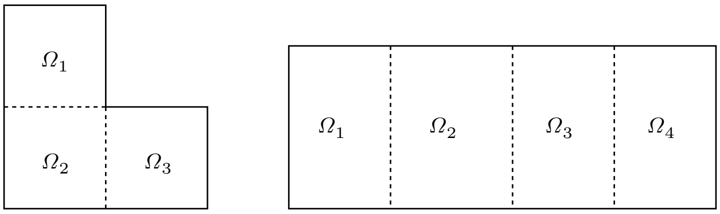


Fig. 2. Decompositions of Ω into subdomains with no crosspoints

3.1 Definition of the FETI Method

We now reformulate the system (6)–(7) as follows

$$K := \begin{pmatrix} K_{ii} & K_{ir} & 0 \\ K_{ri} & K_{rr} & B_r^T \\ 0 & B_r & 0 \end{pmatrix} \begin{pmatrix} u^{(i)} \\ u^{(r)} \\ \tilde{\lambda}^* \end{pmatrix} = \begin{pmatrix} f_i \\ f_r \\ 0 \end{pmatrix}, \tag{15}$$

where $B_r = \mathrm{diag}\{B_{r,\delta_{m,l}}\}_{\delta_m}$ with $B_{r,\delta_{m,l}} = \left(I_{\delta_{m,l}}, \quad -(B^{(r)}_{\delta_{m,l}})^{-1} B^{(r)}_{\gamma_{m,k}} \right)$. Here K_{rr} and K_{ii} are block diagonal matrices of $K^{(l)}_{rr}$ and $K^{(l)}_{ii}$, respectively, cf. (8), and $\tilde{\lambda}^* = \{(B^{(r)}_{\delta_{m,l}})^T\}\lambda^*$.

Next the unknowns related to interior nodes and crosspoints, i.e. $u^{(i)}$ in (15), are eliminated, which yields a new system

$$\begin{aligned} Su^{(r)} + B_r^T \tilde{\lambda}^* &= g_r, \\ B_r u^{(r)} &= 0, \end{aligned} \tag{16}$$

where $S = K_{rr} - K_{ri} \left(K_{ii}\right)^{-1} K_{ir}$ and $g_r = f_r - K_{ri} \left(K_{ii}\right)^{-1} f_i$. We now eliminate $u^{(r)}$ and we end up with the following FETI-DP problem – find $\tilde{\lambda}^* \in M(\Gamma)$ such that

$$F(\tilde{\lambda}^*) = d, \tag{17}$$

where $d = B_r S^{-1} g_r$ and $F = B_r S^{-1} B_r^T$. Note that both S and B are block diagonal matrices due to the assumption that there are no crosspoints.

Next we introduce the following parallel preconditioner

$$M^{-1} = B_r S B_r^T. \tag{18}$$

3.2 Convergence Estimates

We say that the coarse triangulation is in Neumann-Dirichlet ordering if every subdomain has either all edges as slaves or all as mortars. In the case of no crosspoints it is always possible to choose the master-slave sides so as to obtain an N-D ordering of subdomains.

We have the following theorem in which a condition bound is established:

Theorem 1. *For any $\lambda \in M(\Gamma)$ it holds that*

$$c\ (1 + \log(H/\underline{h})^p \left\langle M\lambda, \lambda \right\rangle \le \left\langle F\lambda, \lambda \right\rangle \le C\ \left\langle M\lambda, \lambda \right\rangle,$$

where c and C are positive constants independent of any mesh parameters, $H = \max_k H_k$ and $\underline{h} = \min_k h_k$, $p = 0$ in the case of Neumann-Dirichlet ordering and $p = 2$ in general case.

4 General Case

Here we present briefly the case with crosspoints: the matrix formulation of (6)–(7) is as follows:

$$K := \begin{pmatrix} K_{ii} & K_{ic} & K_{ir} & 0 \\ K_{ci} & \tilde{K}_{cc} & K_{cr} & B_c^T \\ K_{ri} & K_{rc} & K_{rr} & B_r^T \\ 0 & B_c & B_r & 0 \end{pmatrix} \begin{pmatrix} u^{(i)} \\ u^{(c)} \\ u^{(r)} \\ \tilde{\lambda}^* \end{pmatrix} = \begin{pmatrix} f_i \\ f_c \\ f_r \\ 0 \end{pmatrix}, \tag{19}$$

where the global block matrices $B_c = \mathrm{diag}\{B_{c,\delta_{m,l}}\}$ and $B_r = \mathrm{diag}\{B_{r,\delta_{m,l}}\}$ are split into local ones defined over the vector representation spaces of traces on the interface $\Gamma_{kl} = \gamma_{m,k} = \delta_{m,l}$:

$$B_{c,\delta_{m,l}} = \left((B^{(r)}_{\delta_{m,l}})^{-1}B^{(c)}_{\delta_{m,l}}, \quad -(B^{(r)}_{\delta_{m,l}})^{-1}B^{(c)}_{\gamma_{m,k}}\right), \tag{20}$$

and $B_{r,\delta_{m,l}}$ is defined in (15). Here $\tilde{K}_{cc}$ is a block built of $K^{(l)}_{cc}$ taking into account the continuity of dofs at crosspoints, $\tilde{\lambda}^* = \{(B^{(r)}_{\delta_{m,l}})^T\}\lambda^*$, and K_{rr} and K_{ii} are block diagonal matrices as in (15).

Next we eliminate the unknowns related to the interior nodes and crosspoints i.e. $u^{(i)}$, $u^{(c)}$ in (19) and we get

$$\begin{aligned} \hat{S}u^{(r)} + \hat{B}^T\tilde{\lambda}^* &= \hat{f}_r, \\ \hat{B}u^{(r)} + \hat{S}_{cc}\tilde{\lambda}^* &= \hat{f}_c, \end{aligned} \tag{21}$$

where the matrices are defined as follows: $\hat{S} = K_{rr} - (K_{ri}\ \ K_{rc})\tilde{K}^{-1}_{i\&c}\begin{pmatrix} K_{ir} \\ K_{cr} \end{pmatrix}$, $\hat{B} = B_r - (0\ B_c)\tilde{K}^{-1}_{i\&c}\begin{pmatrix} K_{ir} \\ K_{cr} \end{pmatrix}$, and $\hat{S}_{cc} = -(0\ B_c)\tilde{K}^{-1}_{i\&c}\begin{pmatrix} 0 \\ B_c^T \end{pmatrix}$ with $\tilde{K}_{i\&c} = \begin{pmatrix} K_{ii} & K_{ic} \\ K_{ci} & \tilde{K}_{cc} \end{pmatrix}$. We now eliminate $u^{(r)}$ and we end up with finding $\tilde{\lambda}^* \in M(\varGamma)$ such that

$$F(\tilde{\lambda}^*) = d, \tag{22}$$

where $d = f_c - \hat{B}\hat{S}^{-1}f_r$ and $F = \hat{S}_{cc} - \hat{B}\hat{S}^{-1}\hat{B}^T$.

Next we introduce the following parallel preconditioner: $M^{-1} = B_r S_{rr} B_r^T$ where $S_{rr} = \mathrm{diag}\{S^{(l)}_{rr}\}_{l=1}^N$ with $S^{(l)}_{rr} = (K^{(l)}_{rr} - K^{(l)}_{ri}(K^{(l)}_{ii})^{-1}K^{(l)}_{ir})$, i.e. $S^{(l)}_{rr}$ is the respective submatrix of the Schur matrix $S^{(l)}$ over $\varOmega_l$.

Then in the case of Neumann-Dirichlet ordering we have that the condition number $\kappa(M^{-1}F)$ is bounded by $(1 + \log(H/\underline{h})^2$ and in the general case by $(1 + \log(H/\underline{h})^4$.

References

[1] C. Bernardi, Y. Maday, and A.T. Patera. A new nonconforming approach to domain decomposition: the mortar element method. In *Nonlinear Partial Differential Equations and Their Applications. Collège de France Seminar, Vol. XI (Paris, 1989–1991)*, volume 299 of *Pitman Res. Notes Math. Ser.*, pages 13–51. Longman Sci. Tech., Harlow, 1994.

[2] S.C. Brenner. The condition number of the Schur complement in domain decomposition. *Numer. Math.*, 83(2):187–203, 1999.

[3] N. Dokeva, M. Dryja, and W. Proskurowski. A FETI-DP preconditioner with a special scaling for mortar discretization of elliptic problems with discontinuous coefficients. *SIAM J. Numer. Anal.*, 44(1):283–299, 2006.

[4] M. Dryja and W. Proskurowski. On preconditioners for mortar discretization of elliptic problems. *Numer. Linear Algebra Appl.*, 10(1-2):65–82, 2003.

[5] M. Dryja and O.B. Widlund. A FETI-DP method for a mortar discretization of elliptic problems. In *Recent developments in domain decomposition methods (Zürich, 2001)*, volume 23 of *Lect. Notes Comput. Sci. Eng.*, pages 41–52. Springer, Berlin, 2002.

[6] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Linear Algebra Appl.*, 7(7-8):687–714, 2000.

[7] H.H. Kim and C.-O. Lee. A preconditioner for the FETI-DP formulation with mortar methods in two dimensions. *SIAM J. Numer. Anal.*, 42(5):2159–2175, 2005.

[8] A. Klawonn, O.B. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. *SIAM J. Numer. Anal.*, 40(1):159–179, 2002.

[9] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88(3):543–558, 2001.

[10] L. Marcinkowski. A mortar element method for some discretizations of a plate problem. *Numer. Math.*, 93(2):361–386, 2002.

[11] D. Stefanica and A. Klawonn. The FETI method for mortar finite elements. In *Eleventh International Conference on Domain Decomposition Methods (London, 1998)*, pages 121–129. DDM.org, Augsburg, 1999.

Concepts for an Efficient Implementation of Domain Decomposition Approaches for Fluid-Structure Interactions

Miriam Mehl[1], Tobias Neckel[1], and Tobias Weinzierl[1]

Institut für Informatik, TU München
Boltzmannstr. 3, 85748 Garching,
{mehl,neckel,weinzier}@in.tum.de

1 Introduction

In the context of fluid-structure interactions, there are several natural starting points for domain decomposition. First, the decomposition into the fluid and the structure domain, second, the further decomposition of those domains into subdomains for parallelisation, and, third, the decomposition of the solver grid into finer and coarser levels for multilevel approaches. In this paper, we focus on the efficient implementation of the first two types of domain decomposition. The underlying concepts offer a kind of framework that can be 'filled' with variable content on the part of the discretization, the actual equation solver, and interpolations and projections for the inter-code communication. According to this pursuit of flexibility, modularity, and reusability, we restrict to partitioned approaches regarding the decomposition into fluid and structure domain, that is we use existing stand-alone fluid and structure solvers for the coupled fluid-structure simulation. The essential tools we use to achieve efficiency are space-partitioning and space-filling curves. Space-partitioning grids – quad- and octree grids are the most famous representations – are cartesian grids arising from a recursive local refinement procedure starting from a very coarse discretization of the domain. See Fig. 1 for examples. Such grids and the associated trees are widely used for example in computer graphics, grid generation, as computational grids etc. (compare [3]). They allow for the development of very efficient algorithms due to their strict structure and their local recursivity.

In Sect. 2, we describe our concept for the efficient and flexible realization of a partitioned approach for the simulation of fluid-structure interactions (or other multi-physics problems) and identify differences in comparison to MpCCI [9], the successor of GRISSLi [1]. MpCCI nowadays is the most widely used software for the coupling of several codes. Section 3 introduces the algorithmic and implementation aspects of our flow solver. It has to be fast and parallel (as many time steps are required), physically correct (forces acting on the structure), and easily integrable (frequent exchange of information over the coupling surface). In this paper, our focus will be on the two aspects parallelization and integrability. For further informations on physical correctness and general efficiency of our concept, we refer to [4, 3, 6].

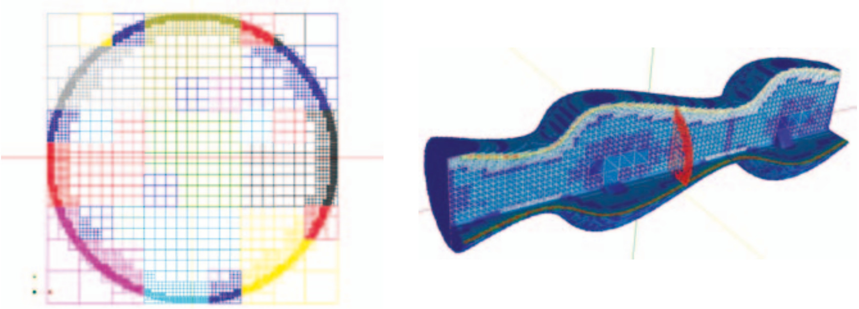


Fig. 1. Left: two-dimensional adaptive space-partitioning grid describing a spherical domain; right: three-dimensional adaptive space-partitioning grid describing a flow tube with asymmetrically oscillating diameter.

2 Decomposition I: Fluid – Structure

The main idea behind the concept of partitioned fluid-structure simulations is simplicity and flexibility with respect to the choice of solvers and the coupling strategy. However, it is not trivial to reach this task in practice. Here, the development of appropriate software components for the coupling of two different codes is decisive but, in contrast to the examination of numerical coupling strategies, a somewhat neglected field. Thus, we will concentrate on these neglected informatical aspects and propose a concept that can be 'filled' with the suitable mathematical content in a flexible way. The most widely used software MpCCI works quite well for a fixed pair of codes but has severe drawbacks as soon as one wants to exchange solvers and/or the coupling strategy frequently. To eliminate these drawbacks, we introduce a client-server approach with a coupling client containing the coupling strategy and acting as a really separating layer between the solvers, which strongly facilitates the exchange of components of the coupled simulation.

The coupling client has to address two components of coupling: first, the exchange of interface data (such as velocities, forces, . . .) between the solvers involved and, second, the control of the coupled simulation, that is the execution of the coupling strategy (explicit/implicit). Figure 2 displays the general concepts of MpCCI and of our approach [3].

2.1 Data Exchange

At the interface between fluid and structure, we have to introduce some mapping of data between two, in general non-matching grids. In the MpCCI concept, this mapping is done directly from one solver to the other with the help of either given library routines or specialized user-defined interpolations. This implies that each solver has to know the grid of the other solver and, thus, inhibits the exchange of one solver without changing the code of the other one. Our concept introduces a third separate component, the coupling component, which holds its own description of the interface between fluid and structure in the form of a surface triangulation (Fig. 3), which we will refer to as the central mesh in the following. The solvers have to map their data

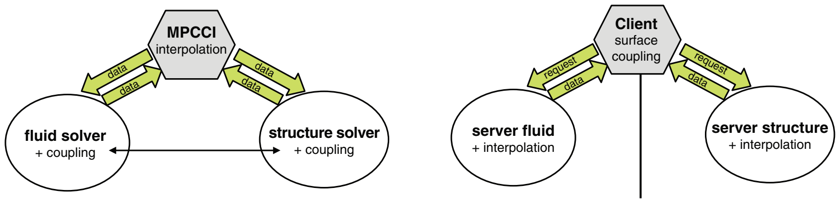


Fig. 2. Schematic view of the general coupling concepts used in `MpCCI` (left) and in our framework (right).

to or get their data from the central mesh. Thus, we can now exchange one solver without changing the data mapping in the other one. Of course, the concrete choice of the resolution and the grid points of the central mesh may depend on one or both solver grids. However, the general concepts and algorithms needed in each solver to perform the mapping of data between the solver grids and the central mesh are not affected by this concrete choice as they only depend on the principle structure of the central mesh.

The basis for all data mappings between solver grid and central mesh is – independent of the concrete choice of interpolation and projection operators – the identification of relations between data points of both grids. For the mapping of data from the solver to the central mesh, these relations are in general *inclusion* properties of surface nodes with respect to the solver grid cells. For the transport of data the way back from the central surface mesh to the solver grid, we need projections from the solver grid nodes to the surface triangles (see Fig. 3 for an example with a cartesian fluid grid). Whereas localizing a surface node in a solver grid cell is an easy task, finding projections on surface triangles requires more sophisticated methods. At this point, space-partitioning grids come into play as a key structure for the access of the (two-dimensional) surface triangulation in a spatial (three-dimensional) context.

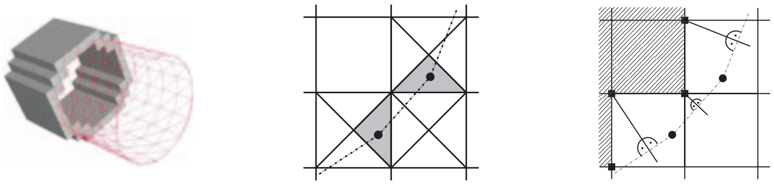
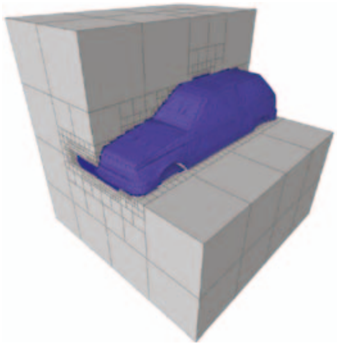


Fig. 3. Left: Surface triangulation of a cylinder and cartesian grid cells at the boundary of the cylinder; right: data transport between the surface triangulation and a cartesian fluid grid: interpolation of stresses at surface nodes from values at the nodes of the fluid grid cell containing the respective surface node (left), copying data from projection points of fluid boundary points on the surface triangulation to the respective boundary grid point (right, taken from [3]).

The algorithm to determine projection points is very closely related to the algorithm creating a new space-partitioning fluid grid (see Sect. 3) from the surface triangulation. In both cases, we have to identify intersections of solver grid cells and surface triangles. Here, the big potentials of space-partitioning grids are their inherent location awareness and their recursive structure, which permits the development of highly efficient algorithms exploiting the possibilities of handing down informations already gained in father cells to their son cells. Figure 4 gives an impression on how fast the resulting algorithms work for the example of grid generation.



octree depth	time (in sec.)	# octree nodes
7	0.872	84,609
9	3.375	1,376,753
11	24.563	22,104,017
13	293.875	353,685,761

Fig. 4. Runtimes for the generation of an octree grid from a surface triangulation of a car. Computed on a Pentium 4 with 2.4GHz and 512kB Cache and sufficient main memory to hold all data required (Intel C++ compiler 8.0).

2.2 Coupling Strategy

Except from data transfer, all other aspects of the coupling of two codes such as the definition and implementation of a coupling strategy and control of the whole coupled simulation are left to the user in `MpCCI`. That is, they have to be implemented in one or both of the involved solvers. As a consequence, an independent exchange of on component, either the coupling strategy or one of the solvers, is impossible. In contrast, our concept is based on a client-server approach (Fig. 2 and [2]). A so-called coupling-client controls the whole simulation. Fluid and structure solvers act as servers receiving requests from the client. The coupling strategy is implemented in the client and, thus, separated from the solvers.

3 Parallel Fluid Solver

This paragraph describes some properties and aspects of our fluid solver currently being developed [3]. Hereby, the focus is to provide a solver which at the same time offers the possibility to exploit the most efficient numerical methods such as multigrid and grid adaptivity and, at the same time, to be highly efficient in terms of hardware or, in particular, memory usage and parallelizability. In this paper, we restrict ourselves to the parallelization concepts which again are, the same as all other technical and implementation aspects, independent on the concrete mathematical content (FE-/FV-discretization, linear/nonlinear solvers, etc.). The only invariant is the choice of cartesian space-partitioning grids as computational grids. These grids offer several advantages. First, their strict structure minimizes memory requirements. Second, the recursive structure allows for a very cache-efficient implementation of data structure and data access (see [6]). Third, grid adaptivity can be arbitrarily local (no restriction to block-adaptivity). Fourth, the mapping of data between the solver grid and the central mesh of the coupling client is supported in an optimal way as the mapping algorithms are based on space-partitioning cartesian grids themselves (cf. Sect. 2.1 and Fig. 4). Fifth, finally, a balanced parallelization can be done in a natural way using the properties of space-filling curves.

Figure 5 shows a cut-off of the computed flow field together with the underlying grid for the two-dimensional flow around a cylinder at Reynolds number 20.

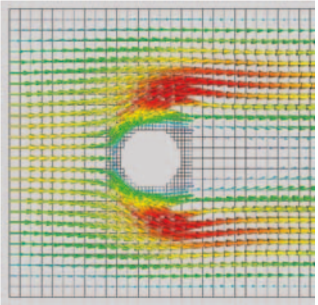


Fig. 5. Flow around a cylinder (Reynolds number 20): cut-off of the flow field and the underlying grid.

For the parallelization of our flow solver, we developed an implementation of a spatial domain decomposition method based on the combination of our space-partitioning computational grid with space-filling curves, which are known to be an efficient tool for the parallelization of algorithms on adaptively refined grids [5]. Hereby, the 'obvious' method is to string together the cells of the adaptive grid along the corresponding iterate of a space-filling curve and afterwards 'cut' the resulting cell string into equal pieces and assign one process to each of the pieces. For the domains of the single processes formed by these pieces of the cell string, a quasi-minimality of the domain surface resulting from the locality properties of the space-filling curve can be shown [5]. Thus, we get a balanced domain decomposition with quasi-minimal communication costs. The disadvantage of this simple approach is that we have to process the whole grid sequentially to build up the cell string.

In contrast to this, our domain decomposition algorithm works completely in parallel. It never has to handle the whole grid on a single master processor. Instead, we recursively distribute sub-tree of the cell-tree among the available processors simultaneously to the set-up phase of the adaptive grid. Each process refines its domain until it reaches a limit in the work load that requires the outsourcing of sub-trees to other processes. Thus, every process holds a complete space-partitioning grid and all the distributed space-partitioning grids together form the global grid or the associated tree, respectively (see Fig. 6). Note that this level-wise domain decomposition process implies a tree order of the computing nodes, too. In the end, the algorithm lists the nodes level by level depending on the workload and, thus, the algorithm is pretty flexible concerning dynamical self-adaptivity. To maintain a balanced load distribution even for extremely local adaptive grid refinements (e.g., at singularities) and grid coarsenings (e.g., as a reaction to changing geometries), the algorithm also supports the merging of the grid partitions of two processes.

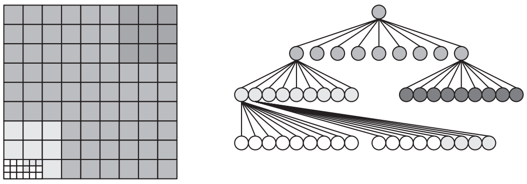


Fig. 6. Distribution of a space-partitioning grid (left) and the associated tree (right) to four processes (marked black, dark grey, grey and white).

Our algorithm uses the Peano space-filling curve to traverse the grid cells and, thus, also splits up the domain at certain points on this curve. As mentioned above, the resulting partition is known to be quasi-optimal and connected [5].

In addition to producing a quasi-minimal amount of communication, the Peano curve allows for a very efficient realization of the communication due to two properties (e.g., which we could not show for any Hilbert curve). First, the Peano curve fulfills a projection property [7], that is the d-dimensional mapping onto a lower-dimensional submanifold aligned with the coordinate axes results in a lower-dimensional Peano curve. Second, it has the so-called palindrome property [7], that is the processing order of cell faces on such a submanifold is not only a Peano curve again but even the *same* Peano curve at both sides of the submanifold only with inverted order. Thus, the locality properties of the Peano curve on a submanifold separating two partitions imply good data access locality in the interprocess communication and, in addition, due to the palindrome property, there is no need for reordering data sent to neighboring processes. This highly improves the communication efficiency.

In Table 1, we give some results achieved with an old version of our program still relying on a sequential domain decomposition algorithm and not yet working with asynchronous communication as our new code does. The left picture of Fig. 1 shows the domain decomposition for an adaptive two-dimensional grid for a sphere computed with our new code.

4 Conclusion

We proposed concepts for two substantially different types of domain decomposition occurring in the context of partitioned simulation of fluid-structure interactions. We

Table 1. Parallel speedup achieved for the solution of the three-dimensional Poisson equation on a spherical domain on an adaptive grid with $23,118,848$ degrees of freedom. The computations were performed on a myrinet cluster consisting of eight dual Pentium III processors with 2 GByte RAM per node [8].

processes	1	2	4	8	16
speedup	1.00	1.95	3.73	6.85	12.93

could show the great potential of space-partitioning grids in each case. For the link-up of fluid and structure solver via a coupling client we use them as a tool to develop fast algorithms to connect the different grids involved. In the fluid solver itself, they are used as a computational grid, enhancing (among others) the parallelization possibilities. With this work, we have shown the general functionality and efficiency of all components: octree algorithms in the coupling client, Navier-Stokes solver on adaptively refined Cartesian grids, balanced parallelization of a solver on adaptively refined Cartesian grids. From this basis, we will establish a unified framework for the simulation of fluid-structure interactions including a (commercial) structure solver, integrate different mathematical methods for code coupling, data mapping, discretization, and linear solvers and perform numerous simulations for various examples to further approve the payload and the flexible applicability of our basic concepts.

References

[1] U. Becker-Lemgau, M. Hackenberg, B. Steckel, and R. Tilch. Interpolation management in the grissli coupling-interface for multidisciplinary simulations. In K.D. Papailiou, D. Tsahalis, J. Périaux, C. Hirsch, and M. Pandolfi, editors, *Computational Fluid Dynamics '98, Proceedings of the 4th Eccomas Conference 1998*, pages 1266–1271, 1998.

[2] M. Brenk, H.-J. Bungartz, M. Mehl, R.-P. Mundani, A. Düster, and D. Scholz. Efficient interface treatment for fluid-structure interaction on cartesian grids. In *Proc. of the ECCOMAS Thematic Conf. on Comp. Methods for Coupled Problems in Science and Engineering*, 2005.

[3] M. Brenk, H.-J. Bungartz, M. Mehl, and T. Neckel. Fluid-structure interaction on cartesian grids: Flow simulation and coupling environment. In H.-J. Bungartz and M. Schäfer, editors, *Fluid-Structure Interaction*, volume 53 of *Lecture Notes in Computational Science and Engineering*, pages 233–269. Springer, 2006.

[4] H.-J. Bungartz and M. Mehl. Cartesian discretisation for fluid-structure interaction - efficient flow solver. In *Proceedings ECCOMAS CFD 2006, European Conference on Computational Fluid Dynamics, Egmond an Zee, September 5th-8th 2006*, 2006.

[5] M. Griebel and G.W. Zumbusch. Hash based adaptive parallel multilevel methods with space-filling curves. *NIC Series*, 9:479–492, 2002.

[6] F. Günther, M. Mehl, M. Pögl, and Ch. Zenger. A cache-aware algorithm for pdes on hierarchical data structures based on space-filling curves. *SIAM J. Sci. Comput.*, 28(5):1634–1650, 2006.

[7] A. Krahnke. *Adaptive Verfahren höherer Ordnung auf cache-optimalen Datenstrukturen für dreidimensionale Probleme*. PhD thesis, Institut für Informatik, TU München, 2005.

[8] M. Mehl. Cache-optimal data-structures for hierarchical methods on adaptively refined space-partitioning grids. In M. Gerndt and D. Kranzlmüller, editors, *International Conference on High Performance Computing and Communications 2006, HPCC06*, volume 4208 of *LNCS*, pages 138–147. Springer, 2006.

[9] Fraunhofer SCAI. Mpcci: Multidisciplinary simulations through code coupling, version 3.0.4 *MpCCI Manuals* [online], url: http://www.scai.fraunhofer.de/592.0.html [cited 18 oct. 2006], 2006.

An Overlapping Additive Schwarz-Richardson Method for Monotone Nonlinear Parabolic Problems

Marilena Munteanu and Luca F. Pavarino [*]

Department of Mathematics, University of Milan, Via C. Saldini 50, 20133 Milano, Italy. {munteanu,Luca.Pavarino}@mat.unimi.it

Summary. We construct a scalable overlapping Additive Schwarz-Richardson (ASR) algorithm for monotone nonlinear parabolic problems and we prove that the rate of convergence depends on the stable decomposition constant. Numerical experiments in the plane confirm the theoretical results.

1 Introduction

In the past decade, domain decomposition techniques have been increasingly employed to solve nonlinear problems. As a first approach, domain decomposition methods provide preconditioners for the Jacobian system in a Newton iteration. In this context, Schwarz-type preconditioners have been successfully used to solve problems from various applied fields, e.g. computational fluid dynamics [4, 7], full potential problems [3], cardiac electrical activity [9], unsteady nonlinear radiation diffusion [11]. Additive Schwarz-type methods have been used not only as inner iteration in a Newton-Krylov-Schwarz scheme, but also as outer iteration in nested solvers as ASPIN [5, 1] or nonlinear additive Schwarz [6]. We propose an iterative process based on the additive Schwarz algorithm applied to the nonlinear problem. The main idea of this paper can be traced back to [2], where a linear preconditioner for a nonlinear system arising from the discretization of a monotone elliptic problem is studied. Using the classical assumptions of the abstract theory of additive Schwarz methods (stable decomposition, strengthened Cauchy-Schwarz inequality and local stability, see [12]), we prove that the rate of convergence of the proposed algorithm depends on the stable decomposition constant C_0 and we construct a scalable Additive Schwarz-Richardson (ASR) method.

[*] This work was supported by Istituto Nazionale di Alta Matematica Francesco Severi, Roma and by grants of M.I.U.R (PRIN 2004014411).

2 Nonlinear Parabolic Problems

Let V be a Banach space and H a Hilbert space with a scalar product $(\cdot,\cdot)$ satisfying $V \subset H$ and V dense in H. Let V^* denote the dual space of V and $< \cdot,\cdot >$ the duality between V^* and V. The Riesz representation theorem and the density of V in H implies that for every $u \in H$ there exists an unique element in the dual space V^*, by convention still denoted by u, such that $(u,v) =< u,v > \ \forall\, v \in V$. Let Ω be a bounded domain of $\mathbb{R}^d, d = 2,3$ with the boundary $\partial\Omega$ polyhedral and Lipschitz continuous and $\mathcal{T}_h$ a triangulation of the domain Ω. In the following, we restrict to the case $V = \{v \in H^1(\Omega) : \gamma v = 0 \text{ on } \Gamma_1 \subset \partial\Omega,\ \mu(\Gamma_1) > 0\}$ and $H = L^2(\Omega)$. We consider the nonlinear form $b : H^1(\Omega) \times H^1(\Omega) \longrightarrow \mathbb{R}$ satisfying the following properties:

1. b is *Lipschitz continuous*:
   $\exists L > 0\ \forall v,\ w,\ z \in H^1(\Omega)\ |b(v,z) - b(w,z)| \leq L||v - w|| \cdot ||z||$
2. b is *bounded*:
   $\exists C > 0$ such that $|b(v,w)| \leq C(1 + ||v||)||w||,\ \forall v,\ w \in H^1(\Omega)$
3. b is *hemicontinuous*:
   $\forall u,\ v,\ w \in H^1(\Omega),\ f : [0,1] \longrightarrow \mathbb{R},\ f(\alpha) = b(u + \alpha v, w)$ is continuous
4. b is *strictly monotone*: $b(v, v - w) - b(w, v - w) \geq 0,\ \forall v, w \in H^1(\Omega)$
   and the equality holds only for $v = w$
5. $b\left(v, \sum\limits_{i=1}^{n} \alpha_i w_i\right) = \sum\limits_{i=1}^{n} \alpha_i b(v, w_i)\ \forall v, w_i \in H^1(\Omega),\ \forall \alpha_i \in \mathbb{R},\ i = 1,\dots,n$
6. $b(v,v) \geq c||v||^2_{H^1(\Omega)} - c_0||v||_1 - c_1||v||^2_{L^2(\Omega)} - c_2,\ \forall v \in H^1(\Omega)$,
   where $c > 0, c_0 > 0, c_1 \geq 0, c_2 \geq 0$ are constants.

We consider the following nonlinear parabolic problem: given $u_0 \in L^2(\Omega)$ and $f \in L^2((0,T);V^*)$ find $u \in W \equiv \{u \in L^2((0,T);V),\ u' \in L^2((0,T);V^*)\}$ such that

$$\begin{cases} < u'(t), w > + b(u(t), w) =< f(t), v >,\ \forall t \in (0,T) \setminus E_w,\ \forall w \in V \\ u(0) = u_0 \end{cases} \tag{1}$$

where $E_w \subset (0,T)$ is a set of measure zero that depends on the function w.
The continuous problem (1) is discretized in time by the backward Euler method and in space by the finite element method. Consequently, we obtain the fully discrete problem: given an arbitrary sequence $\{u_h^0\} \subset L^2(\Omega)$ of approximations of u^0 such that $\lim\limits_{h\to 0} ||u_h^0 - u^0|| = 0$, find $u_h^m \in V_h$ such that

$$\left(\frac{u_h^m - u_h^{m-1}}{\tau}, v\right) + b(u_h^m, v) =< f^m, v >,\ \forall v \in V_h \tag{2}$$

where $V_h = \{v|\ v = 0 \text{ on } \overline{\Gamma}_1, v \text{ is continuous on } \overline{\Omega},\ v|_T \text{ is linear } \forall T \in \mathcal{T}_h\}$ is the standard piecewise linear finite element space, $\tau = T/M$ and u_h^m is the value of the discrete function u_h at time $t^m = m\tau$.
Results on the existence and uniqueness of the solution of the discrete and continuous parabolic problems can be found e.g. in [13], Theorem 45.3 and Theorem 46.4, respectively. The convergence of the discrete solution to the continuous one is presented in [13], Theorem 46.4 and 47.1.

3 An Additive Schwarz-Richardson Algorithm

Given a finite element basis $\{\phi_j,\ j=1,\dots,n\}$ of V_h, for simplicity, we will drop the indexes h and m and still denote by u both the finite element approximation $u=\sum_{j=1}^{n} u_j\phi_j$ of the continuous solution and its vector representation $u=(u_1,\dots,u_n)^T$. Problem (2) is equivalent to the nonlinear algebraic system

$$B(u)=\hat{g}, \tag{3}$$

where $B(u)=(b_1,\dots,b_n)^T,\ b_j=(u,\phi_j)+\tau b(u,\phi_j),\ \hat{g}=(g_1,\dots,g_n)^T,\ g_j=\tau\cdot < f^m,\phi_j > +(u_h^{m-1},\phi_j)$. We consider a family of subspaces $V_i\subset V_h,\ i=0,\dots,N$ and the interpolation operators $R_i^T: V_i\longrightarrow V_h$.

We assume that V_h admits the following decomposition:

$$V_h=\sum_{i=0}^{N} R_i^T V_i.$$

In addition to the previous properties (1-6), we also assume the following property (verified in most reaction-diffusion problems in applications):

7. b can be written as a sum $b(u,v)=a(u,v)+\tilde{b}(u,v)$ of a bilinear, continuous and coercive form $a: V\times V\longrightarrow\mathbb{R}$ and a nonlinear form $\tilde{b}$ (that is monotone and Lipschitz continuous with constant $\tilde{L}$ due to 1. and 4.).

The bilinear form $a_\tau(u,v)=(u,v)+\tau a(u,v)$ defines a scalar product on V. We introduce the local symmetric, positive definite bilinear forms $\tilde{a}_{\tau,\ i}: V_i\times V_i\longrightarrow\mathbb{R}$ and, as in the abstract Schwarz theory [12], we make the following assumptions:

- *Stable Decomposition.* There exist a constant C_0, such that every $u\in V_h$ admits a decomposition $u=\sum_{i=0}^{N} R_i^T u_i,\ u_i\in V_i,\ i=0,\dots,N$ that satisfies
$$\sum_{i=0}^{N}\tilde{a}_{\tau,i}(u_i,u_i)\le C_0^2 a_\tau(u,u);$$
- *Strengthened Cauchy-Schwarz inequality.* $\exists\ \epsilon_{ij}\in[0,1]\ i,j=1,\dots,N$, s.t.
$$|a_\tau(R_i^T u_i,R_j^T u_j)|\le\epsilon_{ij}a_\tau(R_i^T u_i,R_i^T u_i)^{1/2}a_\tau(R_j^T u_j,R_j^T u_j)^{1/2},\ \forall u_i\in V_i,\ u_j\in V_j;$$
- *Local Stability.* There is $\omega>0$, such that
$$a_\tau(R_i^T u_i,R_i^T u_i)\le\omega\tilde{a}_{\tau,i}(u_i,u_i),\ \forall u_i\in V_i,\ 0\le i\le N.$$

We define the "projection"-like operators $\tilde{Q}_i: V_h\longrightarrow V_i$ by $\tilde{a}_{\tau,i}(\tilde{Q}_i(u),v_i)=(u,R_i^T v_i)+\tau b(u,R_i^T v_i),\ \forall v_i\in V_i, u\in V_h$, their extensions $Q_i: V_h\longrightarrow R_i^T V_i\subset V_h$ by $Q_i(u)=R_i^T\tilde{Q}_i(u)$ and $Q(u)=\sum_{i=0}^{N} Q_i(u)$.

Let $\tilde{A}_{\tau,i}\equiv(\tilde{a}_{\tau,i}(\phi_j,\phi_l))_{j,l}$ be the matrix representation of the local bilinear form $\tilde{a}_{\tau,i}$. The matrix form of $Q(u)$ is

$$Q(u)=\mathcal{M}^{-1}B(u), \tag{4}$$

where $\mathcal{M} = \left(\sum_{i=0}^{N} R_i^T \tilde{A}_{\tau,i}^{-1} R_i \right)^{-1}$. The matrix $\mathcal{M}$ is symmetric and positive definite and consequently it defines a norm, $||u||_{\mathcal{M}}^2 = u^T \mathcal{M} u$. Denoting $\breve{g} = \mathcal{M}^{-1}\hat{g}$, and using the matrix form of the nonlinear operator Q, it is straightforward to prove that the nonlinear system (3) is equivalent to the system

$$Q(u) = \breve{g}. \tag{5}$$

Additive Schwarz-Richardson (ASR) algorithm: for a fixed time t and for a properly chosen parameter λ, iterate for $k = 0, 1, \ldots$ until convergence

$$u^{k+1} = u^k + \lambda s^k, \tag{6}$$

where $s^k = -\mathcal{M}^{-1}\left(B(u^k) - \hat{g}\right) \Longleftrightarrow s^k = -\left(Q(u^k) - \breve{g}\right)$.
The operator Q satisfies the following Lemmas (for complete proofs see [8]).

Lemma 1. *There exists a positive constant* $\delta_0 = \frac{1}{C_0^2}$ *such that*

$$(Q(u+z) - Q(u), z)_{\mathcal{M}} \geq \delta_0 ||z||_{\mathcal{M}}^2 \ \forall u, v \in V_h.$$

Lemma 2. *There exists a positive constant* $\delta_1 = C\sqrt{\omega^3(1+\rho(\epsilon))^3(1+\tilde{L})^2 C_0^2}$, *where C is a positive constant independent of mesh size or time-step, such that*

$$||Q(u+z) - Q(u)||_{\mathcal{M}} \leq \delta_1 ||z||_{\mathcal{M}} \ \forall u, v \in V_h.$$

Using this lemmas, we can prove the following convergence result.

Theorem 1. *If we choose* $0 < \lambda < 2\delta_0/\delta_1^2$ *then* **ASR** *converges in the $\mathcal{M}$ norm to the solution u^* of* (5), *i.e.*

$$||u^k - u^*||_{\mathcal{M}}^2 \leq P(\lambda)^k ||u^0 - u^*||_{\mathcal{M}}^2,$$

where $P(\lambda) = 1 - 2\lambda\delta_0 + \lambda^2\delta_1^2$.

Proof. We define the error $e^k = u^k - u^*$ and the residual $r^k = Q(u^k) - Q(u^*)$. The error of the $k+1$ step of the ASR-iteration can be expressed in terms of the error and residual at the k step:

$$e^{k+1} = u^{k+1} - u^* = u^k - \lambda r^k - u^* = e^k - \lambda r^k.$$

Using the linearity of $(\cdot,\cdot)_{\mathcal{M}}$:

$$\begin{aligned} ||e^{k+1}||_{\mathcal{M}}^2 &= (e^{k+1}, e^{k+1})_{\mathcal{M}} = (e^k - \lambda r^k, e^k - \lambda r^k)_{\mathcal{M}} \\ &= ||e^k||_{\mathcal{M}}^2 - 2\lambda(e^k, r^k)_{\mathcal{M}} + \lambda^2 ||r^k||_{\mathcal{M}}^2. \end{aligned}$$

Lemma 1 implies:

$$\begin{aligned} -(e^k, r^k)_{\mathcal{M}} &= -(u^k - u^*, Q(u^k) - Q(u^*))_{\mathcal{M}} \\ &= -(u^k - u^*, Q(u^k - u^* + u^*) - Q(u^*))_{\mathcal{M}} \\ &\leq -\delta_0 ||e^k||_{\mathcal{M}}^2. \end{aligned}$$

From Lemma 2 we have

$$||r_k||^2_{\mathcal{M}} = ||Q(u^k) - Q(u^*)||^2_{\mathcal{M}} = ||Q(u^k - u^* + u^*) - Q(u^*)||^2_{\mathcal{M}} \le \delta_1^2 ||e^k||^2_{\mathcal{M}},$$

hence

$$||e^{k+1}||^2_{\mathcal{M}} \le (1 - 2\lambda\delta_0 + \lambda^2\delta_1^2)||e_k||^2_{\mathcal{M}}.$$

We define $P(\lambda) = 1 - 2\lambda\delta_0 + \lambda^2\delta_1^2$. If we choose $0 < \lambda < \frac{2\delta_0}{\delta_1^2}$ then $P(\lambda) < 1$ and the convergence holds. We remark that $P(\lambda)$ has its minimum in $\lambda_{min} = \frac{\delta_0}{\delta_1^2}$ and $P(\lambda_{min}) = 1 - \frac{\delta_0^2}{\delta_1^2} < 1$.

Remark 1. If we drop the coarse space V_0 and we define $Q(u) = \sum\limits_{i=1}^{N} Q_i(u)$, the ASR-algorithm is convergent. In this case, it is possible to prove that $\delta_0 = \frac{1}{C_0^2}$ and $\delta_1 = C_0\rho(\epsilon)\omega(1 + \tilde{L})$.

Remark 2. The algorithm depends on the choice of the parameter λ. Numerical tests have shown that the step-length selection described in [10] performs well.

4 Numerical Results

We consider the variational nonlinear parabolic problem: given $u(t_0, x) = u_0(x)$ and $T > t_0$, for all $t \le T$ find $u(t) \in H_0^1(\Omega)$ such that

$$\left(\frac{\partial u(t)}{\partial t}, v\right) + a(u(t), v) + (f(u(t)), v) = (g, v) \quad \forall v \in H_0^1(\Omega),$$

where

$$a(u, v) = \int\limits_{\Omega} \sum_{i,j} a_{ij} \frac{\partial u}{\partial x_i}\frac{\partial v}{\partial x_j}\ dx,$$

with $a_{ij} \in C^1(\Omega)$ such that $a_{ij}(x) = a_{ji}(x), \quad \forall x \in \Omega,\ \forall i, j$ and f monotone.

Table 1. *Scalability of 1-level and 2-level ASR method for fixed overlap size $\delta = h$, subdomain size $H/h = 4$ and increasing number of subdomains (and nodes).*

$\lambda = 0.4$

N	1-level		2-level	
	iter	err	iter	err
2×2	40	6.75e-3	44	6.75e-3
4×4	70	1.67e-3	37	1.67e-3
6×6	123	7.44e-4	37	7.44e-4
8×8	197	4.18e-4	38	4.18e-4
10×10	293	2.67e-4	38	2.67e-4
12×12	410	1.85e-4	39	1.85e-4
14×14	-	-	39	1.36e-4

λ : step-length selection

N	1-level		2-level	
	iter	err	iter	err
2×2	25	6.75e-3	24	6.75e-3
4×4	37	1.67e-3	25	1.67e-3
6×6	69	7.44e-4	24	7.44e-4
8×8	117	4.18e-4	24	4.18e-4
10×10	157	2.67e-4	27	2.67e-4
12×12	223	1.85e-4	22	1.85e-4
14×14	-	-	25	1.36e-4

Table 2. *Iteration counts and relative errors for fixed overlap size $\delta = h$, mesh size $h = 1/48$ and increasing number of subdomains.*

$\lambda = 0.4$

N	1-level		2-level	
	iter	err	iter	err
2×2	155	1.74e-4	70	1.74e-4
4×4	192	1.74e-4	58	1.74e-4
6×6	245	1.74e-4	47	1.74e-4
8×8	301	1.74e-4	41	1.74e-4

λ : step-length selection

N	1-level		2-level	
	iter	err	iter	err
2×2	86	1.74e-4	43	1.74e-4
4×4	114	1.74e-4	32	1.74e-4
6×6	149	1.74e-4	27	1.74e-4
8×8	-	-	23	1.74e-4

The numerical tests were performed for the bilinear form $a(u, v) = (\nabla u, \nabla v)$ and the nonlinear function $f(u) = 0.5u+u^3$. The domain is the unit square $\Omega = (0, 1)\times(0, 1)$ and g is chosen so that $u^*(t, x) = t\sin(\pi x)\sin(\pi y)$ is the exact solution. We consider $t_0 = 0,\ u_0(x) = 0$ and we compute the solution for $t = \tau = 0.01$. The iterations process is stopped when $||r_k||_{\mathcal{M}}/||r_0||_{\mathcal{M}} \leq 1e-8$ and we denote the relative error by $err = ||u - u^*||_{l^2(\Omega)}/||u^*||_{l^2(\Omega)}$.

Our additive Schwarz preconditioner is build as in the linear case. We partition the domain Ω into shape regular nonoverlapping subdomains $\{\Omega_i,\ 1 \leq i \leq N\}$ of diameter H defining a shape-regular coarse mesh $\mathcal{T}_H$. Each subregion Ω_i is extended to a larger one, Ω_i' such that the fine mesh $\mathcal{T}_h$ gives rise to N local meshes $\mathcal{T}_{h,i}$, and the partition $\{\Omega_i'\}$ satisfies the finite covering assumption [12]. Using the above decomposition, a 1-level method is defined by the local spaces $V_i = \{v \in H_0^1(\Omega_i')|\ v|_T$ is linear, $\forall T \in T_{h,i}\},\ 1 \leq i \leq N$, and the local bilinear forms $\tilde{a}_{\tau,i}(u_i, v_i) = a_\tau(R_i^T u_i, R_i^T v_i),\ \forall u_i,\ v_i \in V_i$, with zero extension interpolation operators $R_i^T : V_i \longrightarrow V,\ 1 \leq i \leq N$. We then build a 2-level algorithm by defining the coarse finite element space $V_0 = \{v \in H_0^1(\Omega)|\ v$ is continuous and $v|_T$ is linear, $\forall T \in T_H\}$ and the operator R_0^T which interpolates the coarse functions onto the fine mesh. It can be proved that the stable decomposition constant is

$$C_0^2 = C\max\{1+\frac{H}{\delta},\ 1+\frac{\tau}{H\delta}\}\ \text{(1-level)}, \qquad C_0^2 = C(1+\frac{H}{\delta}),\ \text{(2-level)}, \tag{7}$$

where δ measures the width of region $\Omega_i' \backslash \Omega_i$, i.e. the overlap size.

Table 3. *Iteration counts and relative errors for fixed mesh size $h = 1/48$, number of subdomain $N = 2 \times 2$, $\lambda = 0.4$ and increasing the overlap size δ*

overlap	1-level		2-level	
	iter	err	iter	err
h	155	1.74e-4	70	1.74e-4
2h	82	1.74e-4	46	1.74e-4
3h	59	1.74e-4	37	1.74e-4
4h	49	1.74e-4	37	1.74e-4

Table 1 reports the iteration counts and relative errors of our ASR method with fixed overlap $\delta = h$, increasing the number of nodes and subdomains so that $H/h = 4$ is kept fixed (scaled speedup). The parameter λ is fixed at 0.4 (left table) or chosen by

Table 4. *Same as Table 1 but with random right-hand side;* $\lambda = 0.4$.

N	1-level iter	2-level iter
2×2	40	42
4×4	70	38
6×6	122	38
8×8	196	38
10×10	291	41
12×12	407	42

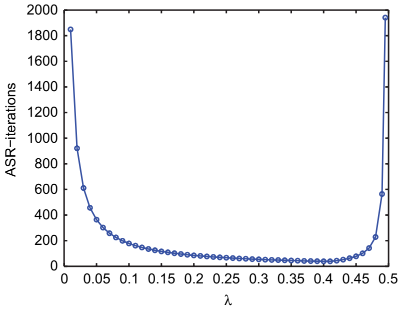


Fig. 1. *ASR iterations counts as a function of the parameter* λ

the step-length strategy of [10] (right table). According to the theory, in the 1-level case the number of iterations increases, because H decreases to zero while δ and τ are kept constant in (7). On the other hand, the iteration counts of the 2-level method remain bounded, because H/δ is kept fixed in (7). The same quantities are reported in Table 2, keeping now $h = 1/48$ fixed and increasing the number of subdomains (standard speedup). Only in the 2-level case the iteration counts improve as the subdomain size decreases. Table 3 shows that the iteration counts improve with increasing overlap size, as in the linear case and Table 4 is the same as Table 1 with $\lambda = 0.4$ but with random right-hand size. Finally, Fig. 1 confirms the theoretical prediction of Theorem 1, showing the ASR iteration counts as a function of the parameter λ for $h = 1/16, N = 2 \times 2, \delta = h$: the ASR convergence rate attains a minimum inside an interval $(0, \alpha)$, $\alpha > 0$ and degenerates at the interval endpoints.

Acknowledgement. Thanks are due to Maksymilian Dryja for very helpful comments and suggestions.

References

[1] H-B. An. On convergence of the Additive Schwarz Preconditioned Inexact Newton Method. *SIAM J. Numer. Anal.*, 43(5):1850–1871, 2005.

[2] X-C. Cai and M. Dryja. Domain decomposition methods for monotone nonlinear elliptic problems. In J. Xu D.E. Keyes, editor, *Seventh International Conference*

on Domain Decomposition Methods in Scientific and Engineering Computing, pages 21–27, 1994. Contemp. Math. 180, Amer. Math. Soc., Providence, RI.

[3] X-C. Cai, W.D. Gropp, D.E. Keyes, R.G. Melvin, and D.P. Young. Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation. *SIAM. J. Sci. Comput.*, 19(1):246–265, 1998.

[4] X-C. Cai, W.D. Gropp, D.E. Keyes, and M.D. Tidriri. Newton-Krylov-Schwarz methods in CFD. In *Proceedings of the International Workshop on Numerical Methods for the Navier-Stokes Equations*, pages 183–200, 1995. Vieweg, Braunschweig.

[5] X-C. Cai and D.E. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SIAM. J. Sci. Comput.*, 24(1):183–200, 2002.

[6] M. Dryja and W. Hackbush. On the nonlinear domain decomposition method. *BIT*, 37(2):296–311, 1997.

[7] F-N. Hwang and X-C. Cai. A parallel nonlinear additive Schwarz preconditioned inexact Newton algorithm for incompressible Navier-Stokes equations. *J. Comput. Phys.*, 204(2):666–691, 2005.

[8] M. Munteanu. *Domain Decomposition Methods for Nonlinear Reaction-Diffusion Problems*. PhD thesis, Univ. of Milan. Dept. of Math., 2007, in preparation.

[9] M. Murillo and X-C. Cai. A fully implicit parallel algorithm for simulating the non-linear electrical activity of the heart. *Numer. Linear Algebra Appl.*, 11(2-3):261–277, 2004.

[10] J. Nocedal and S.J. Wright. *Numerical Optimization*. Springer-Verlag, New-York, 1999. Springer Series in Operations Research.

[11] S. Ovtchinnikov and X-C. Cai. One-level Newton-Krylov-Schwarz algorithm for unsteady non-linear radiation diffusion problem. *Numer.Linear Algebra Appl.*, 11(10):867–881, 2004.

[12] A. Toselli and O. Widlund. *Domain Decomposition Methods-Algorithms and Theory*. Springer-Verlag, Berlin, 2005. Springer Series in Computationl Mathematics, 34.

[13] A. Ženišek. *Nonlinear Elliptic and Evolution Problems and Their Finite Element Approximtions*. Academic Press, Inc., London, 1990. Computational Mathematics and Applications.

Parallel Numerical Solution of Intracellular Calcium Dynamics

Chamakuri Nagaiah[1], Sten Rüdiger[2,3], Gerald Warnecke[1], and Martin Falcke[2]

[1] Institute for Analysis and Numerics, Otto-von-Guericke University, 39106 Magdeburg, Germany. Nagaiah.Chamakuri@Student.Uni-Magdeburg.De, Gerald.Warnecke@Mathematik.Uni-Magdeburg.De

[2] Department of Theoretical Physics, Hahn-Meitner Institute, 14109 Berlin, Germany. falcke@hmi.de

[3] Institut für Physik, Humboldt-Universität zu Berlin, Newtonstr. 15, 12489 Berlin, Germany. sten.ruediger@physik.hu-berlin.de

Summary. We present a parallel numerical approach for intracellular calcium dynamics. Calcium is an important second messenger in cell communication. The dynamics of intracellular calcium is determined by the liberation and uptake by cellular stores as well as reactions with buffers. We develop models and numerical tools to study the liberation of calcium from the endoplasmic reticulum (ER). This process is characterized by the existence of multiple length scales. The modeling of the problem leads to a nonlinear reaction-diffusion system with natural boundary conditions in 2D. We used piecewise linear finite elements for the spatial discretization and time discretization by a linearly implicit Runge-Kutta scheme. We used the CHACO package for the domain decomposition. In our description the dynamics of IP_3-controlled channels remains discrete and stochastic. It is implemented in the numerical simulation by a stochastic source term in the reaction diffusion equation. The strongly localized temporal behavior due to the on-off behavior of channels as well as their spatial localization is treated by an adaptive numerical method.

1 Introduction

Ca^{2+} acts as an intracellular messenger regulating multiple cellular functions such as gene expression, secretion, muscle contraction or synaptic plasticity. The Ca^{2+} signal employed by a variety of processes is a transient increase of the concentration in the cytosol. This is modeled by a system of reaction-diffusion equations with stochastic source terms for which we present numerical simulations.

In this article, we will outline the following important factors in the numerical solution of the problem: grid adaptivity, space and time discretization, coupling between space and time approximations, and parallelization. Briefly, it is very important to have a adaptive grid refinement in the area of clusters to obtain an efficient and fast numerical solutions. The finite element method is very suitable to handle these unstructured grids and complex geometry. We use a linearly implicit methods to avoid nonlinear algebraic systems which arise for fully implicit methods after the time discretization. The classical embedding technique for ordinary differential equation integrators is employed to estimate the error in time. An automatic step size selection procedure ensures that the step size is as large as possible to guarantee the desired precision. To speed up the calculations parallelization is essential. Here the domain decomposition enters at the level of solution of algebraic system, see [1].

The paper is organized as follows. In the second Section we present the model which comprises calcium-buffer binding, diffusion and transport through the ER membrane. We will then introduce our method and strategies for grid adaptation, finite-element discretization and time-stepping in Section 3. Section 4 presents test results using sequential calculations and based on the domain decomposition method which is basic to our parallel code. Section 5 gives a short discussion of our work.

2 Governing Equations

The model consists of equations for the following deterministic quantities: calcium concentration in the cytosol and the ER as well as concentrations of several buffers. The current 2D model describes the concentrations of the involved chemical species in a thin layer on both sides of an idealized plane ER membrane. More details regarding the 2D modeling can be found in [2]. The evolution of concentrations will be determined by diffusion, transport of calcium through the ER membrane, and the binding and unbinding of buffer molecules to calcium:

$$\frac{\partial c}{\partial t} = D_c \Delta c + (P_l + P_c(r))(E - c) - P_p \frac{c^2}{K_d^2 + c^2} - \sum_i H_i(c, b_i), \tag{1}$$

$$\frac{\partial E}{\partial t} = D_E \Delta E + \gamma \left[(P_l + P_c(r))(E - c) - P_p \frac{c^2}{K_d^2 + c^2} \right] - \sum_j K_j(c, b_{E,j}), \tag{2}$$

$$\frac{\partial b_i}{\partial t} = D_{b,i} \nabla^2 b_i + H_i(c, b_i), \quad i = s, m, d, \tag{3}$$

$$\frac{\partial b_{E,j}}{\partial t} = D_{E,j} \nabla^2 b_{E,j} + K_j(E, b_{E,j}), \quad j = s, m. \tag{4}$$

Here c is the concentration of Ca^{2+} in the cytosol, E is the concentration in the ER. The buffer concentration of bound calcium in the cytosol and the ER is given by b_i or $b_{E,j}$, respectively. We have $i = s, d, m$ and $j = s, m$, where s denotes a stationary, d a dye and m a mobile buffers. All buffers are assumed to be distributed homogeneously in the initial state. Immobile buffers are modeled by setting their diffusion coefficient to zero. Total buffer concentrations in the cytosol and the ER are denoted by B_i or G_j, respectively. The buffer binding and unbinding of calcium is modeled by the usual mass-action kinetic terms:

$$H_i = k_{b,i}^+ (B_i - b_i) c - k_{b,i}^- b_i \,, \quad K_j = k_{E,j}^+ (G_j - b_{E,j}) E - k_{E,j}^- b_{E,j}. \tag{5}$$

The second to fourth terms on the right hand sides of (1)-(2) model the transport of calcium through the membrane: leak current, current through IP_3 controlled channels, and pump current, respectively. Channels are typically clustered on the membrane [2]. If a channel is open it contributes within the model to an effective circular source area given by the channel flux term

$$P_c(\mathbf{r}) = \begin{cases} P_{ch} & \text{if } \|\mathbf{r} - \mathbf{x}_n\| < R_n \quad \text{for a cluster n,} \\ 0 & \text{otherwise.} \end{cases}$$

Here the radius R_n of the cluster n with $N_{\text{open},n}$ open channels is then given by $R_n = R_s \sqrt{N_{\text{open},n}}$. The parameter R_s is the source area of a cluster with one open channel. The position of the cluster is given by a fixed position $\mathbf{x}_n$.

An additional complexity of the model stems from the stochastic behavior of channel openings and closings, which needs to be incorporated into the computational approach. For an introduction to the hybrid algorithm to couple deterministic and stochastic simulations see the recent paper by [5].

3 Numerical Method

3.1 Spatial Discretization Using Finite Elements

The domain $\Omega \subseteq \mathbb{R}^2$ is a convex polygonal subset with piecewise smooth boundary Γ. The state variables $c(\underline{x}, t)$, $E(\underline{x}, t)$, $b_m(\underline{x}, t)$ and $b_{Em}(\underline{x}, t)$ are functions of space and time with values in $\Omega \times [0, T]$. We shall denote by $L^2(\Omega)$ the space of square-integrable functions over Ω. This space is equipped with the standard inner product $\langle u, v\rangle = \int_\Omega uv\,dx$ and $\|u\|_0 = \langle u, u\rangle^{1/2}$. Next we define a Sobolev space of square integrable functions and derivatives

$$H^1(\Omega) = \{v \in L^2(\Omega), \partial_i v \in L^2(\Omega), 1 \le i \le d\}\,.$$

3.2 Semi Discretization in Space

The partial differential equations can be written in the following general form

$$\begin{aligned} \tfrac{\partial \mathbf{u}(\mathbf{x},t)}{\partial t} - \nabla \cdot (\mathbf{A}(\mathbf{x})\nabla \mathbf{u}(\mathbf{x}, t)) + \mathbf{r}(\mathbf{u}(\mathbf{x}, t)) = \mathbf{f} \text{ in } & \ \Omega \times (0, T]\,, \\ \mathbf{u}(\mathbf{x}, t) = \mathbf{u}_0(\mathbf{x}) \text{ on } & \ \Omega \times t = 0\,, \\ \mathbf{n} \cdot \nabla \mathbf{u}(\mathbf{x}, t) = 0 \text{ on } & \ \partial\Omega \times [0, T]\,, \end{aligned} \tag{6}$$

where $\mathbf{u}(\mathbf{x}, t)$ is unknown, $\mathbf{A}(\mathbf{x}) > 0$ is the diffusion matrix and $\mathbf{r}(\mathbf{u}(\mathbf{x}, t))$ is the reaction function. Letting $V = H^1(\Omega)$, multiplying the above equation for a given time t by $v \in V$, integrating over Ω and using Green's formula, we get the *variational formulation*. Now let V_h be a finite dimensional subspace of V with basis $\{w_1, \ldots, w_N\}$. Specifically we take continuous functions that are piecewise linear on a quasi-uniform triangulation of Ω with mesh size h. Finally, we get a system of ordinary differential equations in the form

$$\mathsf{M}\dot{\mathbf{u}_h} + \mathsf{A}\mathbf{u}_h + \mathbf{s}(\mathbf{u}_h) = \mathsf{f}\,, \tag{7}$$

where M is the mass matrix, A is the stiffness matrix and $\mathbf{s}(\mathbf{u}_h)$ is the vector depending on reaction term. The matrices are defined as follows

$$\mathsf{M} = \langle w_i, w_j\rangle\,, \quad \mathsf{A} = \langle \mathbf{A}(\mathbf{x})\nabla w_i, \nabla w_j\rangle\,, \quad \mathbf{s}(\mathbf{u}_h) = \langle \mathbf{r}(\sum_{i=1}^{N} u_i(t) w_i(x)), w_j\rangle\,.$$

It is common practice to approximate the mass matrix M by a diagonal matrix, which can be invertible easily. This is called a *lumping* process, see [4].

3.3 Temporal Time-stepping of Continuous Equations

The ordinary differential equation system, acquired from the semi discretization in space is solved numerically with finite difference methods. We considered the ODE problem

$$\frac{\partial \mathbf{u}}{\partial t} = \mathbf{G}(\mathbf{u}), \qquad \mathbf{u}(t^0) = \mathbf{u}^0. \tag{8}$$

The notation for time step is $\tau^i = t^{i+1} - t^i$ and $\mathbf{u}^i$ to be the numerical solution at time t^i. The i-th time step of a W-method (linearly implicit Runge-Kutta type method) of order p with embedding of order $\hat{p} \neq p$ has the form

$$(\mathbf{I} - \tau^i\gamma\mathbf{J})\mathbf{k}_j = \mathbf{G}\left(t^i + \tau^i a_j, \mathbf{u}^i + \tau^i \sum_{l=1}^{j-1} b_{lj}\mathbf{k}_l\right) + \sum_{l=1}^{j-1} c_{lj}\mathbf{k}_l, \quad j = 1, \dots, s, \tag{9}$$

$$\mathbf{u}^{i+1} = \mathbf{u}^i + \tau^i \sum_{l=1}^{s} d_l\mathbf{k}_l\,, \quad \hat{\mathbf{u}}^{i+1} = \mathbf{u}^i + \tau^i \sum_{l=1}^{s} \hat{d}_l\mathbf{k}_l. \tag{10}$$

The method coefficients $\gamma, a_j, b_{jk}, c_{jk}, d_j$, and $\hat{d}_j$ are chosen such that the local error of $\mathbf{u}$ is of order τ_i^{p+1}, the local error of $\hat{\mathbf{u}}$ is of order $\tau_i^{\hat{p}+1}$, and these orders are independent of the matrix $\mathbf{J}$ that is used. We assume $p > \hat{p}$ which is reasonable since one would prefer to continue the integration with the higher order solution $\mathbf{u}$. In our computations we use a W-method with $s = 3$ stages and for the coefficients, see [6].

After the i-th integration step the value $\epsilon = \left\|\mathbf{u}^{i+1} - \hat{\mathbf{u}}^{i+1}\right\|$ is taken as an estimator of the local temporal error. A new time step τ_{new} is computed by

$$\bar{\tau} := \beta\tau^i \left(\frac{TOL_t}{\epsilon}\right)^{\frac{1}{\hat{p}+1}}, \quad \tau_{\text{new}} = \begin{cases} \beta_{\max}\tau^i, & \bar{\tau} > \beta_{\max}\tau^i, \\ \beta_{\min}\tau^i, & \bar{\tau} < \beta_{\min}\tau^i, \\ \bar{\tau}, & \text{otherwise.} \end{cases} \tag{11}$$

The parameter $\beta > 0$ is safety factor. The factors $\beta_{\min}$ and $\beta_{\max}$ restrict time step jumps. If $\epsilon < TOL_t$ we proceed to the next time step, otherwise the time step has to be shortened and repeated. Finally, after time discretization, we get system of algebraic equations in each stage. For solving the system in each stage we used the BiCGSTAB method with ILU preconditioning.

3.4 Grid Adaptivity

As spatial adaptivity criterion we used the Z^2 error estimator of [8], see also [7]. For the refinement we used the following strategy. Let $\lambda(T) \in \mathbb{N}_0$ be the refinement level of triangle $T \in \mathcal{T}$, $\lambda_{max} \in \mathbb{N}_0$ be a given maximum refinement level, and $\phi_1, \dots, \phi_{\lambda_{max}}$ be given real numbers satisfying $0 \leq \phi_1 \dots \leq \phi_{\lambda_{max}}$. Here we used the scaled indicator $\phi_T := \eta_{Z,T}/\sqrt{T}$. For the initial grid and grid adaption we used the program package UG, [1]. We refine the mesh until minimum 4 grid points lie in the area of each channel. For the Test Cases 1 and 2 the initial triangulation a diameter of 700 nm for the triangle is considered.

Test Case 1. In this case we considered one cluster with 20 channels and the domain size is [0,18000 nm] × [0,18000 nm]. The final mesh for this test case can be seen in the left hand Fig. 1.

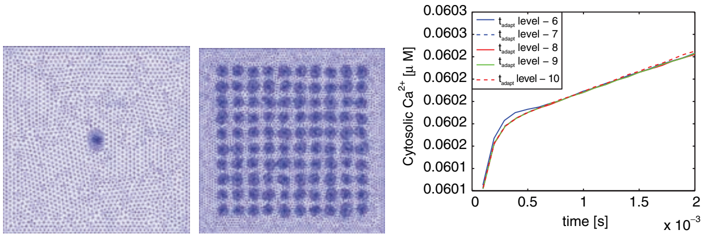


Fig. 1. Mesh level 6 for 1 cluster and 100 clusters, convergence result of cytosolic calcium at different adaptive levels.

Test Case 2. In this case we considered 100 clusters with a distance of 4 μm and each cluster consists of 20 channels. The domain size is [0,48000 nm] × [0,48000 nm]. The final mesh for this test case can be seen in the middle Fig. 1.

4 Numerical Results

In this subsection we will present the convergence results of one cluster with 1 opening channel. In all simulations we used the parameters $D_c = 200\ \mu m^2\, s^{-1},\ D_E = 200\ \mu m^2 s^{-1},\ D_m = 40\ \mu m^2 s^{-1},\ D_s = 0.01\ \mu m^2 s^{-1},\ P_{ch} = 3.0\ \mu m\, s^{-1},\ P_l = 0.025\ \mu m\, s^{-1},\ P_p = 200\ \mu m\, \mu M\, s^{-1}, R_s = 18\ nm,\ K_d = 0.04\ \mu M$ and initial solutions for $c_0 = 0.06\ \mu M,\ E_c = 700\ \mu M$. First let us consider that in the numerical simulation one channel is open for a while. We tested the result with temporally adapted different grid levels. For different levels the average value of cytosol calcium concentration is shown in the right hand Fig. 1. The average value of the solution is calculated by using the integral average $\bar{f} = \frac{1}{|\Omega|} \int_\Omega f(x)\, dx$. In the next case, see Fig. 2, we have incorporated grid adaptivity during the intermediate time steps at mesh level 7. Here channel opening is considered in the stochastic regime. Initially mesh level 7 contains 2737 nodes and 5284 elements, at time $t = 6.504\ s$ has 3216 nodes and 6242 elements, at time $t = 8.92\ s$ it reaches to 18493 nodes and 36796 elements. In Fig. 3 the cytosolic calcium at different times with 100 clusters is shown.

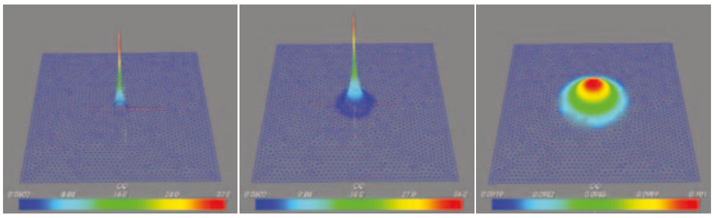


Fig. 2. The numerical result of cytosolic calcium at time $t = 6.504\ s,\ 6.68\ s,\ 8.92\ s$ in 1 cluster.

Here the channel opening is simulated stochastically.

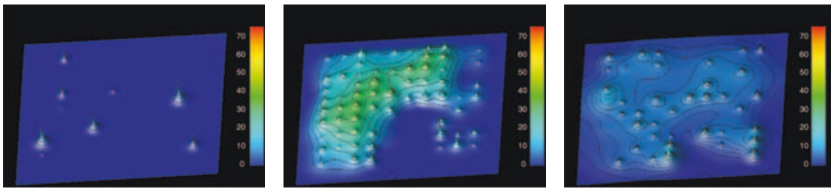


Fig. 3. The numerical result of cytosolic calcium at times $= 5.50\ s,\ 6.13\ s,\ 9.60\ s$ in 100 cluster.

4.1 Numerical Results Using Domain Decomposition Methods

In our numerical code to run the simulation time 100 s on a single processor, the CPU time takes around 50 days. To reduce the computational time and to be able to increase the number of mesh elements to millions the use of parallel computer architectures is mandatory. For the domain decomposition we used the graph partitioning package CHACO of [3]. The load balancing scheme Recursive Inertial Bisection (RIB) serves well for this problem. Load balancing has been achieved as follows: the meshes of level-0 to level-5 have been kept on one processor and the level-6 mesh has been distributed to all processors. The mesh decomposition to different processors is shown in Fig. 4.1. Computations for this problem have been carried out on HP-UX B.11.11 machines with 2GB RAM for each processor this is connected to a 64 node cluster with 3GFOLPS processor speed at our Institute.

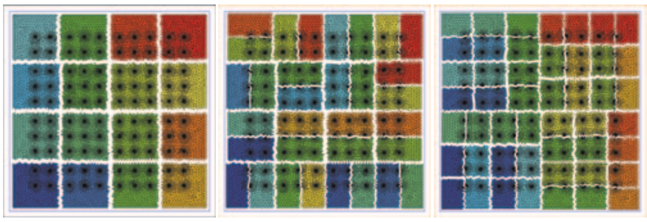


Fig. 4. Domain decomposition using 16, 32 and 48 processors

Performance data of the simulations are presented in Table 1. The time step size is kept constant in all simulations for the sake of comparison. The first column shows the number of processors used and the last column shows the efficiency of the processors. This efficiency can be calculated using $\frac{1}{P}\frac{T(1)}{T(P)}$, where $T(1)$ and $T(P)$ are total CPU time for 1 processor and P processors. Efficiency is increased if we

Table 1. Comparison of CPU times using different processors

no. of procs	unknowns	time steps	cpu time	efficiency
1	133,296	10	26m 46s	-
16	133,296	10	2m 16s	0.7381
32	133,296	10	1m 2s	0.8095
48	133,296	10	38s	0.8805
56	133,296	10	32s	0.8962

increase the number of processors, because of the data structure of the programming package. The increase of the efficiency for 56 processors is 89.62%.

5 Conclusions

In this article we have presented sequential and parallel numerical results for intracellular calcium dynamics in 2 dimensions. In the sequential case, we presented the results of hybrid deterministic and stochastic models. In a test, we obtained good agreement between all mesh levels when channels open for a prescribed time. We observed that spatial adaptivity in time is important if channels open and close stochastically. It is challenging to extend the computations to higher numbers of clusters and 3 dimensions. Furthermore, we presented parallel numerical results using domain decomposition for a setup, where the channels open in a prescribed deterministic way. Here we obtained a reasonably accurate numerical solution upon increasing the number of processors. Extension of our parallel program to stochastic channel dynamics is in progress.

Acknowledgement. The authors gratefully acknowledge the Deutsche Forschungsgemeinschaft (DFG), Germany for financial support under the project grants (Wa 633/16-1, Fa 350/6-1).

References

[1] P. Bastian, K. Birken, S. Lang, K. Johannsen, N. Neuß, H. Rentz-Reichert, and C. Wieners. UG: A flexible software toolbox for solving partial differential equations. *Comput. Vis. Sci.*, 1:27–40, 1997.

[2] M. Falcke. On the role of stochastic channel behavior in intracellular Ca^{2+} dynamics. *Biophys. J.*, 84(1):42–56, 2003.

[3] B. Hendrickson and R. Leland. The CHACO user's guide 1.0. Technical Report SAND93–2339, Sandia National Laboratories, 1993.

[4] A. Quarteroni and A. Valli. *Numerical Approximation of Partial Differential Equations.* Springer-Verlag, Berlin, 1994.

[5] S. Rüdiger, J. W. Shuai, W. Huisinga, Ch. Nagaiah, G. Warnecke, I. Parker, and M. Falcke. Hybrid stochastic and deterministic simulations of calcium blips, 2006. In preparation.

[6] B. A. Schmitt and R. Weiner. Matrix-free W-methods using a multiple Arnoldi iteration. *Appl. Numer. Math.*, 18:307–320, 1995.

[7] R. Verfürth. *A Review of A Posteriori Error Estimation and Adaptive Mesh-Refinement Techniques.* Teubner and Wiley, Stuttgart, 1996.

[8] O. C. Zienkiewicz and J. Z. Zhu. A simple error estimator and adaptive procedure for practical engineering analysis. *Internat. J. Numer. Methods Engrg.*, 24:337–357, 1987.

AMDiS - Adaptive Multidimensional Simulations: Parallel Concepts

Angel Ribalta, Christina Stoecker, Simon Vey, and Axel Voigt

Crystal Growth Group, Research Center Caesar, Ludwig-Erhard-Allee 2, 53175 Bonn, Germany. {ribalta,stoecker,vey,voigt}@caesar.de

Summary. We extend the parallel adaptive meshing algorithm by [2] to further reduce communication requirements in solving PDEs by parallel algorithms. Implementation details and first results for time dependent problems are given.

1 Introduction

In this work we describe the parallelization concepts of our finite element software AMDiS [7], which is a C++ toolbox for solving systems of partial differential equations. It uses adaptive mesh refinement based on local error estimates, to keep the number of unknowns as small as possible. Bank and Holst [2] introduced a new approach for parallelizing such adaptive simulation software, which can be summarized in the following steps:

1. Solve the problem on a relative coarse mesh. Estimate element errors and create partitions with approximately equal error.
2. Each processor solves the problem on the whole domain Ω, but does adaptive refinements only within its assigned local domain (including some overlap).
3. Construct the global solution out of all local solutions by a partition of unity method.

This idea differs from the classical domain decomposition approach in two main points. First, the load balancing is not done before every iteration, but only once at the beginning of computation. Second, the processors do not restrict computations to their local partitions, but refine only within these partitions. Both points lead to reduced communication needs between the processors, and mainly the second point makes it relatively easy to port a sequential software into a parallel one, because necessary code changes are reduced to a minimum.

In section 2 we explain how the domain decomposition is realized in AMDiS, followed by a description of necessary changes in the parallel adaptation loop in section 3. The construction of the global solution is subject of section 4. In section 5 we show some numerical results also for time dependent problems. And finally section 6 contains some conclusions and an outlook to further work.

2 Domain Decomposition

To prepare parallel computations, first the decision must be made, how the work is distributed amongst the given number of processors. Like already mentioned in section 1, the main idea is to do a domain decomposition on a relative coarse mesh, such that the sum of estimated errors in the different domains Ω_i is approximately the same. The coarse mesh can be constructed by a given number of adaptive refinement steps starting from a (very coarse) macro triangulation. These initial refinement steps can be done by one processor which sends the result to all other processes, or they can be done by all processors in parallel. The second approach has the benefit, that no communication has to be done after this phase, because all processors come to the same result in nearly the same time. After domain decomposition each processor i does its calculation on the whole domain, but refines the mesh only within its partition Ω_i^+, which is the domain Ω_i including some overlap with neighboring partitions.

The domain decomposition is computed with help of the parallel graph partitioning and sparse matrix ordering library ParMETIS [5]. ParMETIS first creates the dual graph of the mesh and then partitions this graph considering the error estimates as element weights. When constructing the dual graph, the degree of connectivity among the vertices in the graph can be given, by specifying the number of nodes that two elements at least have to share, so that the corresponding vertices are connected by an edge in the dual graph. For the partitioning of AMDiS meshes this number is set to the dimension of the mesh, because this is the number of common vertices of two neighboring simplices.

Setting this number to one, a dual graph with a higher connectivity is constructed, which can be used for an efficient overlap computation. The overlap is computed on the coarse mesh which is used for the domain decomposition. An overlap of Ω_i of size k includes all elements of the coarse mesh, that have a distance of at most k to any element within Ω_i. Two elements have a distance of 1, if they have at least one common node. A breadth-first search on the dual graph with higher connectivity can be used to determine all elements with distance $d \leq k$ to domain Ω_i.

In Figure 1 the overlap of size 1 for domain Ω_1 on the coarse two dimensional mesh is shown (left hand side). In the middle one can see the global fine mesh after parallel computations, and on the right hand side the corresponding fine mesh of rank 1 is shown. The dashed line at the overlap boundary indicates, that Ω_i^+ is an open domain. The finite element basis functions that are located at this boundary do not belong to partition i, which is important for the partition of unity method.

3 Parallel Adaptation Loop

The adaptation loop keeps nearly the same as in sequential computation. The only thing to do here, is to ensure, that error estimations and refinements are done only on the local partition (including overlap). One goal for the parallel adaptation is to achieve a mesh, which is as similar as possible to that of the sequential computation. To reach this goal, the right refinement strategy has to be chosen. If the strategy depends on global values like the maximal error estimated on Ω, synchronization

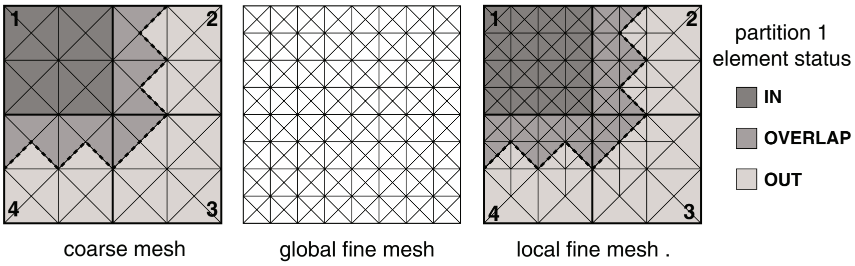


Fig. 1. Overlap of partition 1 on the coarse mesh (left), global fine mesh (middle), and fine mesh of rank 1 (right)

after each iteration is needed to determine and communicate this value. This synchronization could slow down the parallel computation drastically. So we use the equidistribution strategy described in [6] where the refinement depends only on the total error tolerance which is known in advance.

4 Building the Global Solution

After the parallel adaptation loop, every process has computed the solution on the whole domain Ω. But refinements for process *rank* was only done in Ω^{+}_{rank}. Out of this region the solution was computed on a very coarse mesh, and so it may not be very accurate. In this section we describe the construction of a final global solution out of the local rank solutions by a parallel partition of unity method. Here each process *rank* computes a weighted sum of all local solutions on its domain Ω_{rank} using the other solutions u_i on Ω^{i}_{rank} for all $i \neq rank$. In section 4.1 the concept of *mesh structure codes* is described, which provides an efficient way to exchange information about the binary mesh structure at different processes. Using these information, it is easy to synchronize meshes and exchange local solutions in the overlap regions, what is explained in section 4.2. Finally section 4.3 shows, how the parallel partition of unity is built locally by the single processes.

4.1 Mesh Structure Code

To be able to construct a global solution using the partition of unity method, first the local solutions in overlap regions must be exchanged between the different processes. An easy way to do this, is first to synchronize the meshes within these regions, and than exchange the values of corresponding nodes. The concept of mesh structure codes allows to synchronize the meshes in a very efficient way using the MPI communication protocol.

Refinement in AMDiS is done by bisectioning of simplicial elements. The two new elements are stored as children of the bisected element. So a binary tree arises, where each element has either two children or no children. An inner element is represented by a 1 in the mesh structure code, a leaf element is represented by a 0. Furthermore a unique order of tree elements must be given, which is independent

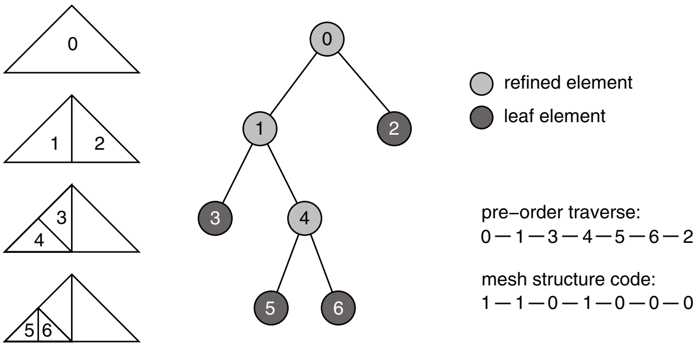


Fig. 2. Mesh structure code of an adaptively refined triangle

of the sequence of adaptive refinements and coarsenings. This order is defined by a pre-order traversal of the tree elements. In Figure 2 the construction of the mesh structure code for an adaptive refined triangle is shown. The resulting binary code in this example can be represented by the decimal value 104. Depending on the size of the binary tree and of the internal integer representation, not one but a vector of integers is necessary to represent the whole tree. To exchange the mesh structures between different processors only one or a few integer values for each macro element has to be sent over MPI. The goal of mesh synchronization is to create the composition of all meshes in overlap regions. To do this, mesh structure codes can be merged very efficiently at binary level. Then the local mesh can be adjusted to this composite mesh structure code.

4.2 Mesh Synchronization and Value Exchange

The process with rank *rank* will construct the global solution within its domain Ω_{rank}. Here fore it needs the local solution u_i on Ω^i_{rank} of every other process $i \neq rank$ which has an overlap with Ω_{rank}. After adapting the local mesh according to the composite mesh structure code the different meshes have common nodes at least in the overlap region. But due to different adaptation sequences in the meshes, these nodes can have different indices, which makes the value exchange difficult. A unique node order can be constructed by sorting the nodes lexicographically in ascending order by their coordinates. So for every other rank i a sending order is created on Ω^{rank}_i and a receiving order on Ω^i_{rank}. To avoid serialization in the MPI communication, first a process sends in a non blocking way to all other processes, and after that receives in a blocking way from the other processes.

4.3 Parallel Partition of Unity

Now process *rank* has all relevant informations to construct the global solution on Ω_{rank} by a partition of unity (see [1]), which is defined by a weighted sum over all local solutions u_i:

$$u_{PU}(x) := \sum_{i=1}^{numRanks} \gamma_i(x) u_i(x) \quad \forall x \in \Omega \tag{1}$$

where $\sum_{i=1}^{numRanks} \gamma_i(x) = 1$ for all $x \in \Omega$. We set $\gamma_i(x) := \frac{W_i(x)}{\sum_{j=1}^{N} W_j(x)}$ with $W_i(x) := \sum_{\phi \in \Phi_i^c} \phi(x)$ where Φ_i^c is the set of linear coarse grid basis functions within Ω_i^+. The partition of unity now is evaluated at all coordinates where fine grid basis functions of Ω_i are located. In [3] an upper bound for the error in H^1 semi norm resulting from the partition of unity is given. Assume $u \in H^2(\Omega)$, then $\|u - u_{PU}\|_{H^1} \leq C(h + H^2)$, where h is the maximal edge size of mesh i in Ω_i and H is the maximal edge size of mesh i in $\Omega \setminus \Omega_i$. In particular, if $h \leq \sqrt{H}$:

$$\|u - u_{PU}\|_{H^1} \leq C(h)\,. \tag{2}$$

5 Numerical Results

In this chapter we present two examples that demonstrate the functionality of the parallelization approach. In section 5.1 the Poisson equation is solved on the unit square. Section 5.2 shows an application in the field of image processing and provides an extension of the approach to time dependent problems.

5.1 Poisson Equation

We solve the Poisson equation

$$-\nabla u(x) = f(x) \quad \forall x \in \Omega \tag{3}$$

$$u(x) = g(x) \quad \forall x \in \partial\Omega \tag{4}$$

with $\Omega = [0,1] \times [0,1]$, $f(x) = -(400x^2 - 40)e^{-10x^2}$, and $g(x) = e^{-10x^2}$. The analytical solution is $u(x) = e^{-10x^2}$, shown in Figure 3 a). In Figure 3 b) the time lines of a parallel computation with eight processors are shown. In this case a speedup of 5.8 to the serial computation was reached. The optimal speedup is not reached due to the overhead of parallel initialization at the beginning and the partition of unity at the end, as well as by a certain load imbalance during the parallel adaptation loop. To get an impression of the error that arises due to the partition of unity, we compared the solution after parallel computation with the true analytical solution and computed the pointwise error, the error in L^2 norm, and the error in the H^1 semi norm. Afterwards we synchronized the meshes on the whole domain Ω and solved the problem on this global mesh again on one single processor. In Table 1 the measured errors for the single cases are listed. The final solve step reduced the pointwise and L^2 error by about one order of magnitude. However the error in H^1 semi norm is reduced only by about ten percent.

Table 1. Errors for a computation with 4 processors before and after a final solve step

	pointwise error	L^2 error	H^1 error
before final solve	$2.726 \cdot 10^{-2}$	$6,207 \cdot 10^{-5}$	$3.508 \cdot 10^{-3}$
after final solve	$1.120 \cdot 10^{-3}$	$6.418 \cdot 10^{-6}$	$3.327 \cdot 10^{-3}$

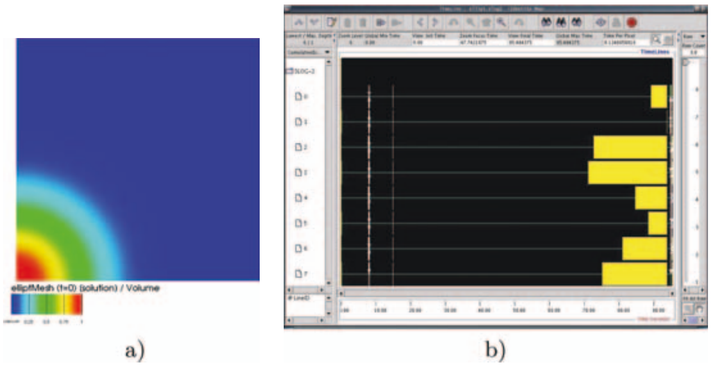


Fig. 3. Analytical solution of the Poisson equation and time lines of a parallel computation with 8 processors

5.2 Perona-Malik Denoising

We now extend the approach to time dependent problems. For this we use the Perona-Malik equation (see [4]) to reduce the noise level in a monochrome image. The gray values of the image are interpreted as height field. A dogma in image processing is that images are of high interest where the gradient of this height field is large. So the Perona-Malik equation smooths regions with a small gradient and sharpens regions with a large gradient. The equation reads

$$u_t = div(g(|\nabla u|)\nabla u) \tag{5}$$

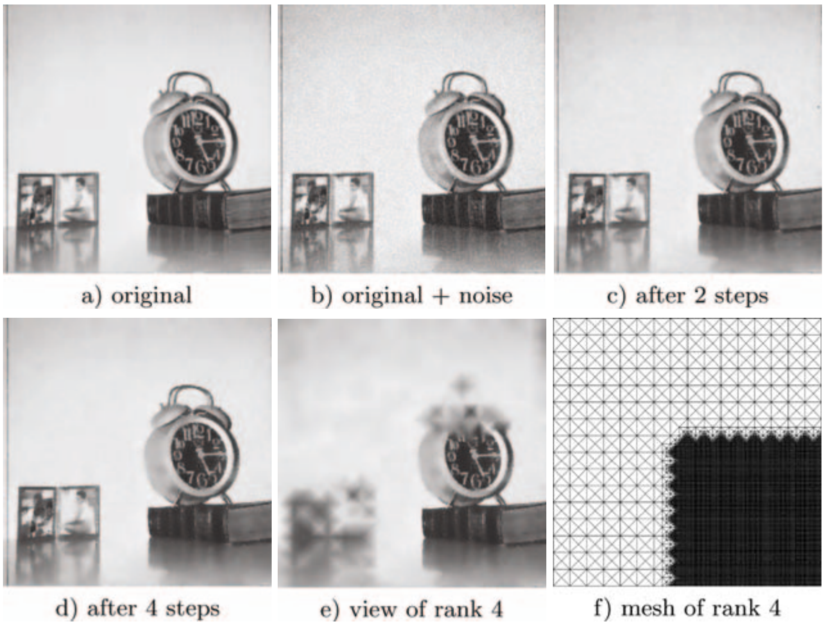
with

$$g(s) := e^{-(\frac{s}{2\lambda})^2} \quad . \tag{6}$$

The parameter λ determines the smoothing/sharpening properties. The time step size determines the degree of denoising. In this example we do not use adaptive mesh refinement, but use a fixed hierarchical mesh, which on the finest level represents the resolution of the image. The domain decomposition and overlap computation is done on a certain lower level of the mesh. After each time step a partition of unity of the local rank solutions was computed. As initial solution we added random noise of the interval $[-30, 30]$ to a picture with gray values between 0 (black) and 255 (white). The picture domain is $\Omega = [0, 1] \times [0, 1]$. The parameter λ was set to 3000 and the time step size is 10^{-4}. In Figure 4 a) the original in b) the noised picture is shown. Figure 4 c) shows the denoising result after two time steps, 4 d) after four time steps. To illustrate the view of one processor, in Figure 4 e) the local solution of rank 4 after time step 4 is shown. In Figure 4 f) the corresponding local mesh is presented.

6 Conclusions and Outlook

The concepts presented in this paper, allowed a parallelization of our finite element software AMDiS with a comparatively small amount of redesign and reimplementation. With the ParMETIS library domain decomposition was done efficiently and



a) original

b) original + noise

c) after 2 steps

d) after 4 steps

e) view of rank 4

f) mesh of rank 4

Fig. 4. Parallel denoising of a monochrome picture

the concept of mesh structure codes enabled an easy way of mesh synchronization. An aspect of future work is the treatment of time dependent problems with adaptive refinements.

References

[1] I. Babuska and J. M. Melenk. The partition of unity method. *Internat. J. Numer. Methods Engrg.*, 40:727–758, 1997.

[2] R.E. Bank and M. Holst. A new paradigm for parallel adaptive meshing algorithms. *SIAM Rev.*, 45(2):291–323, 2003.

[3] M. Holst. Applications of domain decomposition and partition of unity methods in physics and geometry. *Proceedings of the Fourteenth International Conference on Domain Decomposition Methods*, pages 63–78, 2002.

[4] P. Perona and J. Malik. Detecting and localizing edges composed of steps, peaks and roofs. Technical Report UCB/CSD-90-590, EECS Department, University of California, Berkeley, 1990.

[5] K. Schloegel, G. Karypis, and V. Kumar. Parallel static and dynamic multi-constraint graph partitioning. *Concurrency and Computation: Practice and Experience*, 14:219–240, 2002.

[6] A. Schmidt and K.G. Siebert. *Design of Adaptive Finite Element Software*, volume 42 of *LNCSE*. Springer, 2005.

[7] S. Vey and A. Voigt. ADMiS: adaptive multidimensional simulations. *Comput. Vis. Sci.*, 10(1):57–67, 2007.

Generalization of Lions' Nonoverlapping Domain Decomposition Method for Contact Problems

Taoufik Sassi[1], Mohamed Ipopa[1], and François Xavier Roux[2]

[1] Laboratoire de Mathématiques Nicolas Oresme, LMNO, Université de Caen, France. {sassi,ipopa}@math.unicaen.fr

[2] ONERA, 29 av. de la Division Leclerc, BP72, 92322 Châtillon, France. roux@onera.fr

Summary. We propose a Robin domain decomposition algorithm to approximate a frictionless Signorini contact problem between two elastic bodies. The present method is a generalization to variational inequality of Lions' nonoverlapping domain decomposition method. The Robin algorithm is a parallel one, in which we have to solve a contact problem on each domain.

1 Introduction

Contact problems take an important place in computational structural mechanics (see [8, 10, 13] and the references therein). Many numerical procedures have been proposed in the literature. They are based on standard numerical solvers for the solution of global problem in combination with a special implementation of the nonlinear contact conditions (see [5, 4]).

The numerical treatment of such nonclassical contact problems leads to very large (due to the large ratio of degrees of freedom concerned by contact conditions) and ill-conditioned systems. Domain decomposition methods are good alternative to overcome this difficulties (see [2, 3, 15, 11, 14]).

The aim of this paper is to present and study an efficient iterative schemes based on domain decomposition techniques for a nonlinear problem modeling the frictionless contact of linear elastic bodies. The present method is a generalization to variational inequality of the method described in [17, 9]. It can be interpreted as a nonlinear Robin-Robin type preconditioner.

2 Weak Formulation of the Continuous Problem

Let us consider two elastic bodies, occupying two bounded domains Ω^α, $\alpha = 1, 2$, of the space $\mathbb{R}^2$. The boundary $\Gamma^\alpha = \partial\Omega^\alpha$ is assumed piecewise continuous, and composed of three complementary parts Γ_u^α, Γ_ℓ^α and Γ_c^α. The body $\overline{\Omega}^\alpha$ is fixed on the set Γ_u^α of positive measure. It is subject to surface traction forces $\Phi^\alpha \in (L^2(\Gamma_\ell^\alpha))^2$ and the body forces are denoted by $f^\alpha \in (L^2(\Omega^\alpha))^2$. In the initial configuration, both bodies have a common contact portion $\Gamma_c = \Gamma_c^1 = \Gamma_c^2$. We seek the displacement field $u = (u^1, u^2)$ (where the notation u^α stands for $u|_{\Omega^\alpha}$) and the stress tensor field $\sigma = (\sigma(u^1), \sigma(u^2))$ satisfying the following equations and conditions (1)-(3) for $\alpha = 1, 2$:

$$\begin{cases} div\, \sigma(u^\alpha) + f^\alpha &= 0 \text{ in } \Omega^\alpha, \\ \sigma(u^\alpha)n^\alpha - \Phi^\alpha &= 0 \text{ on } \Gamma_\ell^\alpha, \\ u^\alpha &= 0 \text{ on } \Gamma_u^\alpha. \end{cases} \tag{1}$$

The symbol *div* denotes the divergence operator of a tensor function and is defined as

$$div\, \sigma = \Big(\frac{\partial \sigma_{ij}}{\partial x_j}\Big)_i.$$

The summation convention of repeated indices is adopted. The elastic constitutive law, is given by Hooke's law for homogeneous and isotropic solid:

$$\sigma(u^\alpha) = A^\alpha(x)\varepsilon(u^\alpha), \tag{2}$$

where $A^\alpha(x) = (a^\alpha_{ijkh}(x))_{1\le i,j,k,h\le 2} \in (L^\infty(\Omega^\alpha))^{16}$ is a fourth-order tensor satisfying the usual symmetry and ellipticity conditions in elasticity. The linearized strain tensor $\varepsilon(u^\alpha)$ is given by

$$\varepsilon(u^\alpha) = \frac{1}{2}\Big(\nabla u^\alpha + (\nabla u^\alpha)^T\Big).$$

We will use the usual notations for the normal and tangential components of displacement and stress vector on the contact zone Γ_c:

$$u_N^\alpha = u_i^\alpha n_i^\alpha, \quad [u_N] = u^1 n^1 + u^2 n^2\,,$$
$$\sigma_N^\alpha = \sigma_{ij}(u^\alpha) n_i^\alpha n_j^\alpha, \quad \sigma_T^\alpha = \sigma_{ij}(u^\alpha) n_j^\alpha - \sigma_N^\alpha n_i^\alpha\,,$$

where n^α is the unitary normal exterior to Ω^α.
The unilateral contact law on the interface Γ_c is given by:

$$[u_N] \le 0, \quad \sigma_N \le 0, \quad \sigma_N \cdot [u_N] = 0. \tag{3}$$

The contact is supposed frictionless so on Γ_c we get:

$$\sigma_T = 0.$$

In order to give the variational formulation corresponding to the problem (1)-(3), let us introduce the following spaces

$$V^\alpha = \{v^\alpha \in (H^1(\Omega^\alpha))^2,\ v = 0 \quad \text{on} \quad \Gamma_u^\alpha\}, \quad \text{and } V = V^1 \times V^2$$

equipped with the product norm $\|\cdot\|_V = \Big(\sum_{\alpha=1}^2 \|\cdot\|^2_{(H^1(\Omega^\alpha))^2}\Big)^{\frac{1}{2}}$,

$$\mathcal{H}^{\frac{1}{2}}(\Gamma_c) = \Big\{\varphi \in (L^2(\Gamma_c))^2;\ \exists v \in V^\alpha;\ \gamma v_{|\Gamma_c} = \varphi\Big\},$$
$$H^{\frac{1}{2}}(\Gamma_c) = \Big\{\varphi \in L^2(\Gamma_c);\ \exists v \in H^1(\Omega^\alpha);\ \gamma v_{|\Gamma_c} = \varphi\Big\},$$

where γ is the usual trace operator. Now, we denote by K the following non-empty closed convex subset of V:

$$K \;=\; \left\{v=(v^1,v^2)\in V\;,\;[v_N]\leq 0 \text{ on } \Gamma_c\right\}.$$

The variational formulation of problem (1)-(3) is

$$\begin{cases} \text{Find } u\in K \text{ such that} \\ a(u,v-u)\geq L(v-u), \quad \forall v\in K, \end{cases} \tag{4}$$

where

$$a(u,v) \;=\; a^1(u,v)+a^2(u,v),$$

$$a^\alpha(u,v)=\int_{\Omega^\alpha} A^\alpha(x)\varepsilon(u^\alpha)\cdot\varepsilon(v^\alpha)dx, \tag{5}$$

and

$$L(v) \;=\; \sum_{\alpha=1}^{2}\int_{\Omega^\alpha} f^\alpha\cdot v^\alpha\,dx+\int_{\Gamma_\ell^\alpha}\Phi^\alpha\cdot v^\alpha d\sigma.$$

There exists a unique solution u to problem (4) (see [7, 13]).

3 Multibody Formulation and Algorithm

In the following, we will use some lift operators which allow us to build specific function from their values on Γ_c. For $\alpha=1,2$, let

$$\begin{aligned} R^\alpha : \mathcal{H}^{\frac12}(\Gamma_c) &\longrightarrow V^\alpha \\ \varphi &\longrightarrow R^\alpha\varphi=v^\alpha, \end{aligned} \tag{6}$$

where v^α is the solution of

$$\begin{cases} a^\alpha(v^\alpha,w)=0 & \forall w\in V^\alpha \text{ with } w=0 \text{ on } \Gamma_c \\ v^\alpha=\varphi & \text{on } \Gamma_c. \end{cases}$$

The two-body contact problem (4) is approximated by an iterative procedure involving a contact problem for each body Ω^α with a rigid foundation described by a given initial gap g^α.

Given $g_0^\alpha\in\Gamma_c$, $\alpha=1,2$, for $m\geq 1$, we build the sequence of functions $(u_m^1)_{m\geq 0}$ and $(u_m^2)_{m\geq 0}$, by solving in parallel the following problems:

Step 1:

1. Solve the contact problem

$$\begin{aligned} -div(\sigma(u_m^1)) &= f^1 \quad \text{in } \Omega^1, \\ \sigma(u_m^1)n^1 &= \Phi^1 \quad \text{on } \Gamma_\ell^1, \\ u_m^1 &= 0 \quad \text{on } \Gamma_u^1, \\ \sigma_{T,m}^1 &= 0 \quad \text{on } \Gamma_c, \\ u_m^1 n^1 &\leq g_{m-1}^1 \quad \text{on } \Gamma_c, \end{aligned} \tag{7}$$

$$\begin{aligned} \sigma^1_{N,m} &\leq 0 \quad \text{on } \Gamma_c, \\ \sigma^1_{N,m}(u^1_m n^1 - g^1_{m-1}) &= 0 \quad \text{on } \Gamma_c, \end{aligned}$$

with initial gap $g^1_{m-1} = \alpha_{S_1}(\sigma^2_{N,m-1} - \sigma^1_{N,m-1}) - u^2_{m-1}n^2$.
2. Solve the contact problem

$$\begin{aligned} -div(\sigma(u^2_m)) &= f^2 \quad \text{in } \Omega^2, \\ \sigma(u^2_m)n^2 &= \Phi^2 \quad \text{on } \Gamma^2_\ell, \\ u^2_m &= 0 \quad \text{on } \Gamma^2_u, \\ \sigma^2_{T,m} &= 0 \quad \text{on } \Gamma_c, \\ u^2_m n^2 &\leq g^2_{m-1} \quad \text{on } \Gamma_c, \\ \sigma^2_{N,m} &\leq 0 \quad \text{on } \Gamma_c, \\ \sigma^2_{N,m}(u^2_m n^2 - g^2_{m-1}) &= 0 \quad \text{on } \Gamma_c, \end{aligned} \tag{8}$$

with initial gap $g^2_{m-1} = \alpha_{S_2}(\sigma^1_{N,m-1} - \sigma^2_{N,m-1}) - u^1_{m-1}n^1$.
Step 2:
Relaxation

$$g^1_m = (1-\delta_m)g^1_{m-1} + \delta_m(\alpha_{S_2}(\sigma^1_{N,m} - \sigma^2_{N,m}) - u^2_m n^2) \;\text{ on } \Gamma_c, \tag{9}$$

$$g^2_m = (1-\delta_m)g^2_{m-1} + \delta_m(\alpha_{S_1}(\sigma^2_{N,m} - \sigma^1_{N,m}) - u^1_m n^1) \;\text{ on } \Gamma_c. \tag{10}$$

A key point in this algorithm is the choice of α_{S_i}, $i = 1, 2$:
- α_{S_i} is non-negative constant. It is the simplest choice but leads to an h-independent algorithm which is very sensible to boundary conditions.
- α_{S_i} is the Steklov-Poincaré operator defined on the interface Γ^α_c of Ω^α as introduced in [1]. This operator is not practical if the domains Ω^α are too large, but it has interesting features. Mainly, it can be defined for any geometry and for any elliptic operator, including three-dimensional anisotropic heterogeneous elasticity, and it is coercive positive selfadjoint operator. In practice, this choice consists to resolve two auxiliary problems before step 2. These auxiliary problems are written by:
$m \geq 0$, $\alpha = 1, 2$

$$\begin{aligned} -div(\sigma(w^\alpha_m)) &= 0 \quad \text{in } \Omega^\alpha, \\ \sigma(w^\alpha_m)n^\alpha &= 0 \quad \text{on } \Gamma^\alpha_\ell, \\ w^\alpha_m &= 0 \quad \text{on } \Gamma^\alpha_u, \\ \sigma(w^\alpha_m)n^\alpha &= \pm(\sigma(u^1_m)n^1 - \sigma(u^2_m)n^2) \text{ on } \Gamma_c. \end{aligned} \tag{11}$$

So the variational formulation of our algorithm takes the following form:
Given g^α_0, $\alpha = 1, 2, \in \mathcal{H}^{\frac{1}{2}}(\Gamma_c)$, for $m \geq 1$, we build the sequence of functions $(u^1_m)_{m\geq 0} \in V^1$ and $(u^2_m)_{m\geq 0} \in V^2$ by solving the following problems:

1^{st} step:

$$\begin{aligned} &\text{Find } \; u^\alpha_m \in V^\alpha_-(g^\alpha_{m-1}), \\ &a^\alpha(u^\alpha_m, v^\alpha - u^\alpha_m) \geq (f^\alpha, w^\alpha - u^\alpha_m)\, \forall v^\alpha \in V^\alpha_-(g^\alpha_{m-1}) \end{aligned} \tag{12}$$

where

$$V_{-}^{\alpha}(\varphi) = \{v \in V^{\alpha} / \; vn^{\alpha} \leq -\varphi \quad \text{on} \quad \Gamma_c^{\alpha}\}\,.$$

$\underline{2^{nd}\text{ step}}$:

$$\begin{cases} \text{Find } w_m^1 \in V^1, \\ a^1(w_m^1, v) = -a^2(u_m^2, R^2\gamma(v)) + (f^2, R^2\gamma(v)) - a^1(u_m^1, v) + (f^1, v)\, \forall v \in V^1. \\ \\ \text{Find } w_m^2 \in V^2, \\ a^2(w_m^2, v) = a^1(u_m^1, R^1\gamma(v)) - (f^1, R^1\gamma(v)) + a^2(u_m^2, v) - (f^2, v)\, \forall v \in V^2. \end{cases} \tag{13}$$

$\underline{3^{th}\text{ step}}$:

$$\begin{cases} g_m^1 = (1-\delta_m)g_{m-1}^1 + \delta_m(w_m^2 n^2 - u_m^2 n^2) \;\; \text{on } \Gamma_c, \\ \\ g_m^2 = (1-\delta_m)g_{m-1}^2 + \delta_m(w_m^1 n^1 - u_m^1 n^1) \;\; \text{on } \Gamma_c. \end{cases} \tag{14}$$

We refer to [12] for the convergence results of the continuous algorithm (12)-(14) and its finite elements approximation.

Remark 1. Another choice of α_{S_i}, for $i = 1, 2$, is to create an artificial small "dream" domain having Γ_c as one of its faces on which we define the Steklov-Poincaré operator (see [16, 18]).

4 Numerical Experiments

In this section we describe some numerical results obtained with algorithm (12)-(14) for various relaxation parameter δ and various degrees of freedom $n = n_1 + n_2$ (d.o.f in $\Omega^1 \cup \Omega^2$) and m (d.o.f. on Γ_c). The computation is based on the iterative method of successive approximations. Each iterative step requires to solve two quadratics programming problems constrained by simple bounds. Our implementation uses recently developed algorithm of quadratic programming with proportioning and gradient projections [6].

The computation efficiency shall be assessed by

$$IT_{outer}/IT_{inner},$$

where IT_{outer} (resp. IT_{inner}) denotes the number of outer iterations (resp. the total number of conjugate gradient steps i.e the number of matrix-vector multiplications by Hessians).

The numerical implementations are performed in Scilab 2.7 on Pentium 4, 2.0 GHz with 256 MB RAM. We set $tol = 10^{-8}$ and we break down iterations, if their number is greater than eight hundred. For all experiments to be described below, the stopping criterion of Algorithm (12)-(14) is

$$\frac{\|g_m^1 - g_{m-1}^1\|}{\|g_m^1\|} + \frac{\|g_m^2 - g_{m-1}^2\|}{\|g_m^2\|} \leq tol,$$

where $||\cdot||$ denotes the Euclidean norm. The precisions in the inner iterations are adaptively adjusted by the precision achieved in the outer loop.

Let us consider the plane elastic bodies

$$\Omega^1 = (0,3) \times (1,2) \quad \text{and} \quad \Omega^2 = (0,3) \times (0,1)$$

made of an isotropic, homogeneous material characterized by Young's modulus $E_\alpha = 2.1\;10^{11}$ and Poisson's ratio $\nu_\alpha = 0.277$. The decomposition of Γ^1 and Γ^2 read as:

$$\Gamma_u^1 = \{0\} \times (1,2),\; \Gamma_c^1 = (0,3) \times \{1\},\; \Gamma_l^1 = \Gamma^1 \setminus \overline{\Gamma_u^1 \cup \Gamma_c^1},$$

$$\Gamma_u^2 = \{0\} \times (0,1),\; \Gamma_c^2 = (0,3) \times \{1\},\; \Gamma_l^2 = \Gamma^2 \setminus \overline{\Gamma_u^2 \cup \Gamma_c^2}.$$

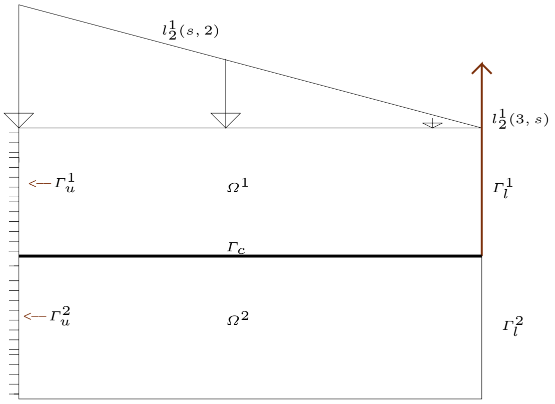


Fig. 1. Setting of the problem

The volume forces vanish for both bodies. The non-vanishing surface traction $\ell^1 = (l_1^1, l_2^1)$ (respectively, $\ell^2 = (l_1^2, l_2^2)$) act on Γ_l^1 (respectively, on Γ_l^2):

$$l_1^1(s,2) = 0, \quad l_2^1(s,2) = -3\,10^6 - 1\,10^6\,s, \quad s \in (0,3),$$

$$l_1^1(3,s) = 0, \quad l_2^1(3,s) = 2\,10^6, \quad s \in (1,2),$$

$$l_1^2(s,0) = 0, \quad l_2^2(s,0) = 0, \quad s \in (0,3),$$
$$l_1^2(3,s) = 0, \quad l_2^2(3,s) = 0, \quad s \in (0,1).$$

The Table 1 gives convergence of the algorithm (12)-(14) for different values of the relaxation parameter δ and various degrees of freedom (n and m). The results obtained show that the number of outer iterations (for an optimal value of $\delta = 0.95$) does not depend on the degrees of freedom n and m.

Table 1. Convergence of the algorithm

n/m	$\delta = 0.1$	$\delta = 0.5$	$\delta = 0.7$	$\delta = 0.95$	$\delta = 1$
12/3	287/903	76/243	62/208	47/160	–
36/6	285/899	79/272	66/237	49/179	–
288/16	270/878	74/282	79/295	45/188	–
816/24	296/957	92/332	93/340	47/204	–

References

[1] V.I. Agoshkov. Poincaré-Steklov's operators and domain decomposition methods in finite-dimensional spaces. In *First International Symposium on Domain Decomposition Methods for Partial Differential Equations (Paris, 1987)*, pages 73–112, Philadelphia, 1988. SIAM.

[2] G. Bayada, J. Sabil, and T. Sassi. Algorithme de Neumann-Dirichlet pour des problèmes de contact unilatéral: résultat de convergence. *C. R. Math. Acad. Sci. Paris*, 335(4):381–386, 2002. In French. (Neumann-Dirichlet algorithm for unilateral contact problems: convergence results).

[3] G. Bayada, J. Sabil, and T. Sassi. A Neumann-Neumann domain decomposition algorithm for the Signorini problem. *Appl. Math. Lett.*, 17(10):1153–1159, 2004.

[4] A.B. Chandhary and K.J. Bathe. A solution method for static and dynamic analysis of three-dimensional contact problems with friction. *Computers & Structures*, 24:855–873, 1986.

[5] P.W. Christensen, A. Klarbring, J.S. Pang, and N. Stromberg. Formulation and comparison of algorithms for frictional contact problems. *Internat. J. Numer. Methods Engrg.*, 42(1):145–173, 1998.

[6] Z. Dostál and J. Schöberl. Minimizing quadratic functions over non-negative cone with the rate of convergence and finite termination. *Comput. Optim. Appl.*, 30(1):23–44, 2000.

[7] G. Duvaut and Lions J.L. *Les Inéquations en Mécanique et en Physique.* Dunod, Paris, 1972.

[8] R. Glowinski, J.L. Lions, and Trémolière R. *Numerical Analysis of Variational Inequalities.* North-Holland Publishing Co., Amsterdam, 1981.

[9] W. Guo and L.S. Hou. Generalizations and accelerations of Lions' nonoverlapping domain decomposition method for linear elliptic PDE. *SIAM J. Numer. Anal.*, 41(6):2056–2080, 2003.

[10] J. Haslinger, I. Hlaváček, and J. Nečas. Numerical methods for unilateral problems in solid mechanics. In *Handbook of Numerical Analysis, Vol. IV*, pages 313–485. North-Holland, Amsterdam, 1996.

[11] J. Haslinger, M. Ipopa, R. Kučera, and T. Sassi. Implementation of Neumann-Neumann algorithm for contact problems, 2007. Submitted.

[12] M. Ipopa and T. Sassi. A Lions' domain decomposition algorithm for contact problems: Convergence results, 2007. Submitted.

[13] N. Kikuchi and J.T. Oden. *Contact Problems in Elasticity: a Study of Variational Inequalities and Finite Element Methods.* SIAM, Philadelphia, PA, 1988.

[14] J. Koko. An optimization-based domain decomposition method for a two-body contact problem. *Numer. Funct. Anal. Optim.*, 24(5-6):587–605, 2003.

[15] R.H. Krause and B.I. Wohlmuth. Nonconforming domain decomposition techniques for linear elasticity. *East-West J. Numer. Math.*, 8(3):177–206, 2000.

[16] P. Le Tallec and T. Sassi. Domain decomposition with nonmatching grids: augmented Lagrangian approach. *Math. Comp.*, 64(212):1367–1396, 1995.

[17] P.-L. Lions. On the Schwarz alternating method. III. A variant for nonoverlapping subdomains. In *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations (Houston, TX, 1989)*, pages 202–223. SIAM, Philadelphia, PA, 1990.

[18] F.-X. Roux, F. Magoulès, L. Series, and Y. Boubendir. Approximation of optimal interface boundary conditons for two-Lagrange multiplier FETI method. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lect. Notes Comput. Sci. Eng.*, pages 283–290. Springer, Berlin, 2005.

Multilevel Schwarz and Multigrid Preconditioners for the Bidomain System

Simone Scacchi[1] and Luca F. Pavarino[2]

[1] Dipartimento di Matematica, Università di Pavia, Via Ferrata 1, 27100 Pavia, Italy. simone.scacchi@unipv.it
[2] Dipartimento di Matematica, Università di Milano, Via Saldini 50, 20133 Milano, Italy. luca.pavarino@mat.unimi.it

Summary. Two parallel and scalable multilevel preconditioners for the Bidomain system in computational electrocardiology are introduced and studied. The Bidomain system, consisting of two degenerate parabolic reaction-diffusion equations coupled with a stiff system of several ordinary differential equations, generates very ill-conditioned discrete systems when discretized with semi-implicit methods in time and finite elements in space. The multilevel preconditioners presented in this paper attain the best performance to date, both in terms of convergence rate and solution time and outperform the simpler one-level preconditioners previously introduced. Parallel numerical results, using the PETSc library and run on Linux Clusters, show the scalability of the proposed preconditioners and their efficiency on large-scale simulations of a complete cardiac cycle.

1 Introduction

We introduce and study two parallel and scalable multilevel preconditioners for the Bidomain system in computational electrocardiology. These preconditioners improve upon the recent studies [2, 7], where one-level block Jacobi preconditioners were found to perform satisfactorily for the simplified Monodomain model but not for the more complex Bidomain system. The latter is a multiscale model of the cardiac bioelectrical activity, consisting of two degenerate parabolic reaction-diffusion equations describing the intra and extracellular potentials of the anisotropic cardiac tissue (macroscale), coupled through the nonlinear reaction term with a stiff system of several ordinary differential equations describing the ionic currents through the cellular membrane (microscale).

The numerical resolution of the Bidomain system is computationally very expensive, because of the interaction of the different scales in space and time, the degenerate nature of the PDEs involved and the very severe ill-conditioning of the discrete systems arising at each time step. Fully implicit methods in time have been considered in few studies, see e.g. [6] and require the solution of nonlinear systems at each time step. Most numerical studies employ semi-implicit (IMEX) methods in time that only require the solution of linear systems at each time step. Many different preconditioners have been proposed in order to devise efficient iterative solvers for such linear systems: diagonal preconditioners [9], Symmetric Successive Over Relaxation [8, 14], Block Jacobi (BJ) preconditioners with incomplete LU factorization (ILU) for each block [2, 7, 13], multigrid [15].

The multilevel preconditioners presented in this paper attain the best performance to date, both in terms of convergence rate and solution time and outperform the simpler one-level preconditioners previously introduced. Parallel numerical results, using the PETSc library (see [1]) and run on Linux Clusters, show the scalability of the proposed preconditioners and their efficiency on large-scale simulations of a complete cardiac cycle.

2 The Mathematical Model

The macroscopic Bidomain model represents the cardiac tissue as the superposition of two anisotropic continuous media, the intra (i) and extra (e) cellular media, coexisting at every point of the tissue and separated by a distributed continuous cellular membrane. The cardiac tissue is traditionally modeled as an arrangement of fibers rotating clockwise from epicardium to endocardium [11] and, according to [4], presents a laminar organization, which consists of a set of muscle sheets, moving radially from epicardium to endocardium. Therefore, at any point $\mathbf{x}$, it is possible to identify a triplet of orthonormal principal axes $\mathbf{a}_l(\mathbf{x})$, $\mathbf{a}_t(\mathbf{x})$, $\mathbf{a}_n(\mathbf{x})$, with $\mathbf{a}_l$ parallel to the local fiber direction, $\mathbf{a}_t$ and $\mathbf{a}_n$ tangent and orthogonal to the radial laminae respectively. The anisotropic conductivity properties of the tissue are described by the conductivity coefficients in the intra and extracellular media $\sigma_l^{i,e}$, $\sigma_t^{i,e}$, $\sigma_n^{i,e}$ measured along the corresponding direction $\mathbf{a}_l$, $\mathbf{a}_t$, $\mathbf{a}_n$ and by the conductivity tensors $\mathsf{D}_i(\mathbf{x})$ and $\mathsf{D}_e(\mathbf{x})$, given by

$$\mathsf{D}_{i,e}(\mathbf{x}) = \sigma_l^{i,e}\ \mathbf{a}_l(\mathbf{x})\mathbf{a}_l^T(\mathbf{x}) + \sigma_t^{i,e}\ \mathbf{a}_t(\mathbf{x})\mathbf{a}_t^T(\mathbf{x}) + \sigma_n^{i,e}\ \mathbf{a}_n(\mathbf{x})\mathbf{a}_n^T(\mathbf{x}).$$

The intra and extracellular electric potentials u_i, u_e in the cardiac domain Ω are described in the Bidomain model by the following parabolic reaction-diffusion system coupled with a system of ODEs for the ionic variables w:

$$\begin{cases} c_m \dfrac{\partial v}{\partial t} - \mathrm{div}(\mathsf{D}_i \nabla u_i) + I_{ion}(v,w) = 0 & \text{in } \Omega \times (0,T) \\ -c_m \dfrac{\partial v}{\partial t} - \mathrm{div}(\mathsf{D}_e \nabla u_e) - I_{ion}(v,w) = -I_{app}^e & \text{in } \Omega \times (0,T) \\ \dfrac{\partial w}{\partial t} - R(v,w) = 0, & \text{in } \Omega \times (0,T), \end{cases} \tag{1}$$

with boundary conditions $\mathbf{n}^T \mathsf{D}_{i,e} \nabla u_{i,e} = 0$ in $\partial\Omega \times (0,T)$ and initial conditions $v(\mathbf{x},0) = v_0(\mathbf{x}), w(\mathbf{x},0) = w_0(\mathbf{x})$ in Ω. Here c_m is the capacitance per unit area times the surface to volume ratio; $v = u_i - u_e$ is the transmembrane potential; I_{app}^e is the applied current; I_{ion} and R model the ionic currents and depend on the choice of the membrane model. In this work we consider the LR1 model, see [5]. Existence and regularity results for this degenerate system are proved in [3] and [12]. The system uniquely determines v, while the potentials u_i and u_e are defined only up to a same additive time-dependent constant.

3 Discretization and Numerical Methods

System (1) is discretized by the finite element method in space and a semi-implicit method in time. The space discretization is obtained meshing the cardiac domain Ω with a structured grid of hexahedral $\mathbb{Q}_1$ elements and introducing the associated finite element space V_h. A semidiscrete problem is obtained by applying a standard Galerkin procedure. We denote by M the symmetric mass matrix, by $\mathrm{A}_{i,e}$ the symmetric stiffness matrices associated to the intra and extra-cellular anisotropic conductivity tensors, respectively, and by I^h_{ion}, $\mathrm{I}^{e,h}_{app}$ the finite element interpolants of I_{ion} and I^e_{app}, respectively. The time discretization is performed by a semi-implicit method using for the diffusion term the implicit Euler method, while the nonlinear reaction term I_{ion} is treated explicitly.

As a consequence, the full evolution system is decoupled by first solving the ODEs system (given the potential $\mathbf{v}^n$ at the previous time-step)

$$\mathbf{w}^{n+1} - \Delta t\ R(\mathbf{v}^n, \mathbf{w}^{n+1}) = \mathbf{w}^n$$

and then solving for $\mathbf{u}_i^{n+1}, \mathbf{u}_e^{n+1}$ the linear system

$$\left(\frac{c_m}{\Delta t}\begin{bmatrix} \mathrm{M} & -\mathrm{M} \\ -\mathrm{M} & \mathrm{M}\end{bmatrix} + \begin{bmatrix} \mathrm{A}_i & 0 \\ 0 & \mathrm{A}_e \end{bmatrix}\right)\begin{pmatrix}\mathbf{u}_i^{n+1} \\ \mathbf{u}_e^{n+1}\end{pmatrix} =$$

$$\frac{c_m}{\Delta t}\begin{pmatrix}\mathrm{M}(\ \ \mathbf{u}_i^n - \mathbf{u}_e^n) \\ \mathrm{M}[-\mathbf{u}_i^n + \mathbf{u}_e^n]\end{pmatrix} + \begin{pmatrix}\mathrm{M}[-\mathbf{I}^h_{ion}(\mathbf{v}^n, \mathbf{w}^{n+1})] \\ \mathrm{M}[\ \ \mathbf{I}^h_{ion}(\mathbf{v}^n, \mathbf{w}^{n+1}) - \mathbf{I}^{e,h}_{app}]\end{pmatrix}, \tag{2}$$

where $\mathbf{v}^n = \mathbf{u}_i^n - \mathbf{u}_e^n$. The iteration matrix is symmetric semidefinite, having the zero eigenvalue associated to the $(\mathbf{1},\ \mathbf{1})$ eigenvector, therefore, as in the continuous model, $\mathbf{u}_i^n$ and $\mathbf{u}_e^n$ are determined only up to the same additive time-dependent constant, chosen by imposing the condition $\mathbf{1}^T M \mathbf{u}_e^n = 0$. From [2] we know that the iteration matrix is very ill conditioned and we need an efficient preconditioner.

4 Parallel Implementation and Preconditioners

The parallel strategy consists of partitioning the computational domain into subdomains of the same size and assign them to different processors. The linear system (2) is solved with the parallel PCG method of the PETSc library. We will compare three different preconditioners.

Block Jacobi Preconditioner (BJ), i.e. a block diagonal matrix with blocks built from the local restriction of matrix A to each subdomain; on each block, we use an ILU(0) solver.

V-cycle Multigrid Preconditioner (MG): the linear system at each time step is solved with a five-level V-cycle Multigrid method (MG(5)). The smoother used for all but the coarsest level is a single iteration of CG with BJ-ILU(0) preconditioner. On the coarsest level we solve the system using the PCG preconditioned by BJ-ILU(0).

Symmetrized Multiplicative Multilevel Schwarz Preconditioner (SMMS). Let be $\Omega^{(i)}$, $i = 0, ..., M$ a family of nested triangulations of Ω, coarsening from M to 0, and $\mathrm{A}^{(i)}$ the matrix obtained by discretizing (1) on $\Omega^{(i)}$: so $\mathrm{A}^{(M)} = \mathrm{A}$. $\mathrm{R}^{(i)}$ are the restriction operators from $\Omega^{(i+1)}$ to $\Omega^{(i)}$. We decompose Ω into N overlapping subdomains, hence each grid $\Omega^{(i)}$ is decomposed into N overlapping subgrids $\Omega_k^{(i)}$

for $k = 1, ..., N$, such that the overlap $\delta^{(i)}$ at level $i = 1, ..., M$ is equal to the mesh size $h^{(i)}$ of the grid $\Omega^{(i)}$. Let $R_k^{(i)}$ be the restriction operator from $\Omega^{(i)}$ to $\Omega_k^{(i)}$ and define $\mathrm{A}_k^{(i)} := \mathrm{R}_k^{(i)} \mathrm{A}^{(i)} \mathrm{R}_k^{(i)^T}$. The action of this preconditioner on a given residual $\mathbf{r}$ is given by:

$$\begin{gathered}
\mathbf{u}^{(M)} \leftarrow \sum_{k=1}^{N} \mathrm{R}_k^{(M)^T} \mathrm{A}_k^{(M)^{-1}} \mathrm{R}_k^{(M)} \mathbf{r} \\
\mathbf{r}^{(M-1)} \leftarrow \mathrm{R}^{(M-1)} (\mathbf{r} - \mathrm{A}^{(M)} \mathbf{u}^{(M)}) \\
\mathbf{u}^{(M-1)} \leftarrow \sum_{k=1}^{N} \mathrm{R}_k^{(M-1)^T} \mathrm{A}_k^{(M-1)^{-1}} \mathrm{R}_k^{(M-1)} \mathbf{r}^{(M-1)} \\
\cdots \\
\mathbf{u}^{(0)} \leftarrow \mathrm{A}^{(0)^{-1}} \mathbf{r}^{(0)}, \quad \mathbf{u}^{(1)} \leftarrow \mathbf{u}^{(1)} + \mathrm{R}^{(0)^T} \mathbf{u}^{(0)} \\
\mathbf{u}^{(1)} \leftarrow \mathbf{u}^{(1)} + \sum_{k=1}^{N} \mathrm{R}_k^{(1)^T} \mathrm{A}_k^{(1)^{-1}} \mathrm{R}_k^{(1)} (\mathbf{r}^{(1)} - \mathrm{A}^{(1)} \mathbf{u}^{(1)}) \\
\cdots \\
\mathbf{u}^{(M)} \leftarrow \mathbf{u}^{(M)} + \mathrm{R}^{(M-1)^T} \mathbf{u}^{(M-1)} \\
\mathbf{u}^{(M)} \leftarrow \mathbf{u}^{(M)} + \sum_{k=1}^{N} \mathrm{R}_k^{(M)^T} \mathrm{A}_k^{(M)^{-1}} \mathrm{R}_k^{(M)} (\mathbf{r}^{(M)} - \mathrm{A}^{(M)} \mathbf{u}^{(M)}) \\
\mathbf{u} \leftarrow \mathbf{u}^{(M)}
\end{gathered}$$

We implemented this method with 5 levels, hence in the remainder we denote it by SMMS(5). For details see [10].

5 Numerical Results

The numerical experiments were performed on two distributed memory parallel architectures, the IBM CLX/1024 Linux cluster of the Cineca Consortium (www.cineca.it), with 1024 processors Intel Xeon Pentium IV (3 GHz, 512 KB cache) grouped into 512 nodes of 2 processors (total RAM = 1 TB), and the Ulisse Linux cluster of the Department of Mathematics of the University of Milan (cluster.mat.unimi.it), with 72 processors Xeon (2.4 GHz) grouped into 36 nodes of 2 processors. Our FORTRAN code is based on the parallel library PETSc from the Argonne National Laboratory [1].

Test 1: standard speedup. We simulate the initial depolarization of a thin slab of cardiac tissue, having dimensions of $2.56 \times 2.56 \times 0.01\, cm^3$, applying a stimulus of $200\, mA/cm^3$ for $1\, ms$ on a small volume of $2 \times 2 \times 2$ elements at a vertex of the domain. The global mesh is fixed to be of $257 \times 257 \times 2$ nodes (264196 unknowns) and the number of subdomains varies from 1 to 16. The model is run for 40 time steps of $0.05\, ms$, i.e. for a time interval of $2\, ms$ on the Linux cluster of the University of Milan. In table 1, we report the average number of PCG iterations per time step, needed to reduce the l^2 norm of the residual smaller than 10^{-4}, the average condition number per time step and the average time needed to solve the linear system. Both the multilevel methods are scalable, in fact the iterations remain almost constant increasing the number of subdomains. The BJ speedup is low, because the number of iterations increases with the processors. The multilevel preconditioners behave well up to 8 processors, but with 16 the local problems are too small and the communication costs deteriorate the parallel performance.

Test 2: scaled speedup. In this test, we vary the number of subdomains from 8 to 128, keeping fixed the local mesh in each subdomain to $48 \times 48 \times 48$ nodes

Table 1. Test 1, standard speedup. IT:= average PCG iterations per time step; COND:= average condition number per time step; TIME:= average execution time per time step in seconds.

# SUB	BJ			MG(5)			SMMS(5)		
	IT.	COND.	TIME	IT.	COND.	TIME	IT.	COND.	TIME
1	95	1817	23.00	3	1.04	9.11	-	-	-
2	108	2209	22.27	3	1.04	4.63	3	1.04	4.95
4	109	2229	10.40	4	1.08	2.92	3	1.04	2.49
8	111	2367	5.31	4	1.11	1.58	3	1.04	1.28
16	114	2745	3.47	4	1.13	0.78	3	1.04	0.71

(221184 unknowns), hence varying the global number of degrees of freedom (d.o.f.) from 1.7 × 10^6 in the smallest case with 8 subdomains to 2.8 × 10^7 in the largest with 128 subdomains. As in test 1, we simulate the initial depolarization of a cardiac slab, running the model for 40 time steps on the CLX cluster of CINECA. Table 2 reports the average number of PCG iterations, the average condition number and the average solving time per time step. These results show the parallel scalability of the proposed multilevel methods, that have constant iteration counts, while the one-level BJ preconditioner has increasing iteration counts as expected. The solving time is also scalable, increasing of only 15∼20 % going from 8 to 128 processors; for SMMS(5) this increase is due only to communications, because the iterations remain constant.

Table 2. Test 2, scaled speedup. Same format as in Table 1.

# SUB	D.O.F.	BJ			MG(5)			SMMS(5)		
		IT.	COND.	TIME	IT.	COND.	TIME	IT.	COND.	TIME
8	1769472	84	1461	29.9	4	1.26	16.1	4	1.14	16.2
16	3538944	93	2119	35.7	4	1.34	16.9	4	1.14	18.1
32	7077888	106	3543	44.8	5	1.43	16.8	4	1.14	16.3
64	14155776	115	4984	42.3	5	1.61	17.8	4	1.15	18.9
128	28311552	121	5165	51.4	5	1.58	18.5	4	1.14	19.6

Table 3. Test 3, complete cardiac cycle. IT:= average PCG iterations per time step; TIME:= average execution time per time step in seconds; TOTAL TIME:= total simulation time

PREC	aver. IT.	aver. TIME	TOTAL TIME
BJ	205	46.02 sec	29 h 49 m
MG(5)	8	11.11 sec	7 h 21 m
SMMS(5)	6	9.67 sec	6 h 26 m

Test 3: complete cardiac cycle. In this last test, we simulate a complete heartbeat (400 ms) in a portion of ventricle having dimension 2 × 2 × $0.5\,cm^3$, discretized by a cartesian grid of 200 × 200 × 50 nodes (4 × 10^6 d.o.f.). We run the simulation on 36 processors of the Linux cluster of Milan. Table 3 reports the average

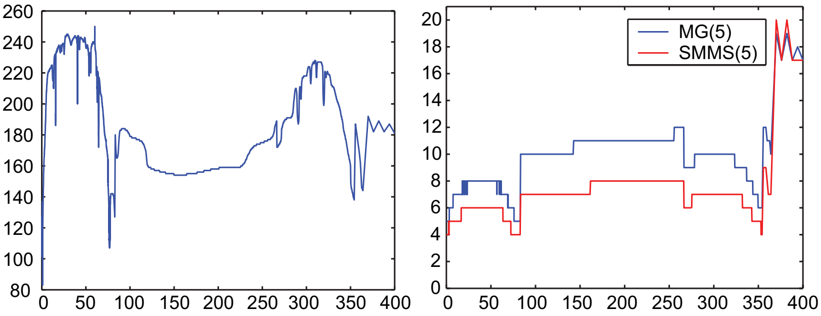


Fig. 1. Test 3. Time evolution of the PCG iterations with BJ preconditioners (left) and multilevel preconditioners MG(5), SMMS(5) (right).

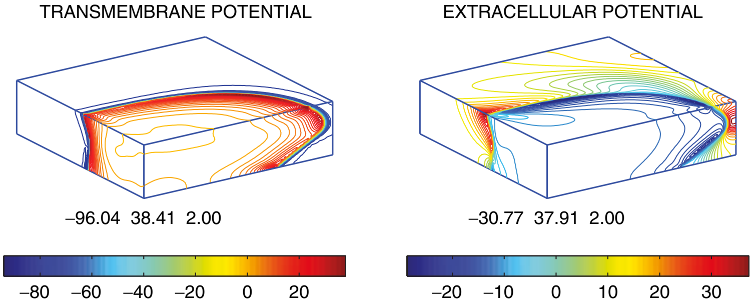


Fig. 2. Test 3. Patterns of level lines of the transmembrane and extracellular potentials during the excitation phase (t = 40 ms). Reported below each panel are the minimum, maximum and step in mV of the displayed map.

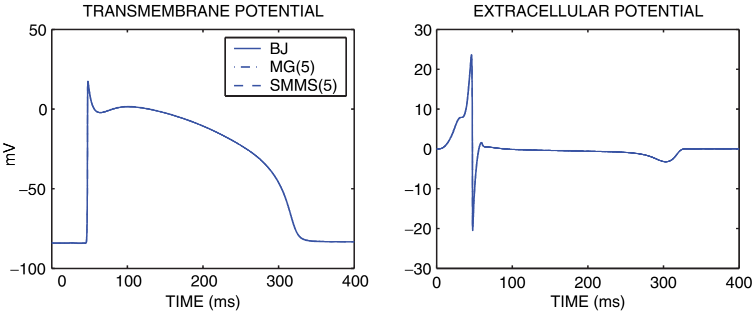


Fig. 3. Test 3. Time evolution at a fixed point of the transmembrane and extracellular potentials, computed with the three methods.

PCG iterations per time step, the average execution time per time step and the total simulation time. MG(5) and SMMS(5) are respectively 4 and 4.7 times faster than BJ. The detailed iteration counts as a function of time during the complete heartbeat are shown in Fig. 1 (left panel for BJ and right panel for MG(5) and SMMS(5)). Figure 2 shows the spatial maps of the transmembrane and extracellular potentials computed 40 ms after the stimulus was given at a vertex of the domain, i.e. during the excitation phase. Figure 3 shows the transmembrane and extracellular potentials computed in a fixed point of the domain by the three methods (the graphics are perfectly superimposed).

References

[1] S. Balay et al. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.

[2] P. Colli Franzone and L. F. Pavarino. A parallel solver for reaction-diffusion systems in computational electrocardiology. *Math. Models Methods Appl. Sci.*, 14(6):883–911, 2004.

[3] P. Colli Franzone and G. Savaré. Degenerate evolution systems modeling the cardiac electric field at micro- and macroscopic level. In A. Lorenzi and B. Ruf, editors, *Evolution Equations, Semigroups and Functional Analysis*, pages 49–78. Birkhäuser, 2002.

[4] I. J. Le Grice et al. Laminar structure of the heart: ventricular myocyte arrangement and connective tissue architecture in the dog. *Am. J. Physiol. (Heart Circ. Physiol.)*, 269(38):H571–H582, 1995.

[5] C. Luo and Y. Rudy. A model of the ventricular cardiac action potential: depolarization, repolarization, and their interaction. *Circ. Res.*, 68(6):1501–1526, 1991.

[6] M. Murillo and X.C. Cai. A fully implicit parallel algorithm for simulating the nonlinear electrical activity of the heart. *Numer. Lin. Alg. Appl.*, 11(2-3):261–277, 2004.

[7] L. F. Pavarino and P. Colli Franzone. Parallel solution of cardiac reaction-diffusion models. In R. Kornhuber et al., editor, *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Springer LNCSE*, pages 669–676. Springer-Verlag, 2004.

[8] M. Pennacchio and V. Simoncini. Efficient algebraic solution of reaction-diffusion systems for the cardiac excitation process. *J. Comput. Appl. Math.*, 145(1):49–70, 2002.

[9] K. Skouibine and W. Krassowska. Increasing the computational efficiency of a bidomain model of defibrillation using a time-dependent activating function. *Ann. Biomed. Eng.*, 28:772–780, 2000.

[10] B.F. Smith, P.E. Bjørstad, and W.D. Gropp. *Domain Decomposition.* Cambridge University Press, Cambridge, 1996.

[11] D. Streeter. Gross morphology and fiber geometry in the heart. In R. Berne, editor, *Handbook of Physiology*, volume 1 of *The Heart*, pages 61–112. Williams & Wilkins, Baltimore, 1979.

[12] M. Veneroni. Reaction-diffusion systems for the macroscopic bidomain model of the cardiac electric field. Technical report, I.M.A.T.I.-C.N.R., 2006.

[13] E. J. Vigmond, F. Aguel, and N. A. Trayanova. Computational techniques for solving the bidomain equations in three dimensions. *IEEE Trans. Biomed. Eng.*, 49(11):1260–1269, 2002.

[14] R. Weber dos Santos. *Modelling Cardiac Electrophysiology.* PhD thesis, Univ. of Rio de Janeiro, Dept. of Mathematics, 2002.

[15] R. Weber dos Santos, G. Plank, S. Bauer, and E. J. Vigmond. Parallel multigrid preconditioner for the cardiac bidomain model. *IEEE Trans. Biomed. Eng.*, 51(11):1960–1968, 2004.

Preconditioners for Low Order Thin Plate Spline Approximations

Linda Stals[1] and Stephen Roberts[1]

Department of Mathematics, Australian National University,
Canberra, ACT, 0200, Australia. stals@maths.anu.edu.au

A commonly used method for fitting smooth functions to noisy data is the thin-plate spline method. Traditional thin-plate splines use radial basis functions and consequently require the solution of a dense linear system of equations whose dimension grows linearly with the number of data points. Here we discuss a method based on low order polynomial functions with locally supported basis functions. An advantage of such an approach is that the resulting system of equations is sparse and its dimension depends linearly on the number of nodes in the finite element grid instead of the number of data points.

Another advantage is that an iterative solver, such as the conjugate gradient method, can be used. However it can be shown that the system of equations is similar to those arising from Tikhonov regularisation, and consequently the equations are ill-conditioned for certain choices of the parameters. To ensure that the method is robust an appropriate preconditioner must be used.

In this paper we present the discrete thin-plate spline method and explore a set of preconditioners. We discuss some of the properties that are unique to our particular formulation and verify that the multiplicative Schwarz method is an effective preconditioner.

1 Introduction

The thin-plate spline method is a popular data fitting technique because it is insensitive to noise in the data. For a general domain Ω the thin-plate spline f (as discussed by [10] and [3]) minimises the functional

$$\begin{aligned} J_\alpha(f) = {} & \frac{1}{n}\sum_{i=1}^{n}(f(\mathbf{x}^{(i)}) - y^{(i)})^2 \\ & + \alpha \int_\Omega \sum_{|\nu|=2} \binom{2}{\nu} (D^\nu f(\mathbf{x}))^2 d\mathbf{x}, \end{aligned} \tag{1}$$

where $\nu = (\nu_1, ..., \nu_d)$ is a d dimensional multi-index, $|\nu| = \sum_{s=1}^{d} \nu_s$, $\mathbf{x}$ is a predictor variable in $\mathbb{R}^d$, and $\mathbf{x}^{(i)}$ and $y^{(i)}$ are respectively the corresponding i-th predictor and response data value $(1 \leq i \leq n)$.

The parameter α controls the trade-off between smoothness and fit. Techniques for choosing α automatically, such as generalised cross validation (GCV), can be found in [5, 10].

Radial basis functions are often used to represent f, as they give an analytical solution of the minimiser of the functional in (1). However the resulting system of equations is dense, and furthermore its dimension is directly proportional to the number of data points.

In [7, 8] we proposed a discrete thin-plate spline method that uses piecewise functions with local support defined on a finite element mesh. In particular, the method described in Section 2 uses standard multi-linear finite element basis functions. The advantage of using functions with local support is that the dimension of the resulting system of sparse equations depends only on the number of grid points in the finite element mesh.

The system of equations resulting from the finite element discretisation can be manipulated to form a symmetric positive definite system, as shown in Section 3. However for small values of α this system is ill-conditioned and the convergence slows down markedly. Section 4 discusses some of reasons causing the difficulties with the convergence rate, and Section 5 shows that the convergence rate can be improved by using the multiplicative Schwarz preconditioner.

2 Discrete Thin Plate Splines

For simplicity most of the discussion is focused on two dimensional (2D) examples, although the theory generalises to three dimensions and the code has been developed for both two and three dimensions.

The smoothing problem from (1) can be approximated with finite elements so that the discrete smoother f is a linear combination of piecewise multi-linear basis functions (hat functions) $b_i(\mathbf{x}) \in H_0^1$,

$$f(\mathbf{x}) = \sum_{i=1}^{m} c_i b_i(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T \mathbf{c}.$$

The idea is to minimise J_α over all f of this form. The smoothing term (the second term in (1)) is not defined for piecewise multi-linear functions, but the non-conforming finite element principle can be used to introduce piecewise multi-linear functions $\mathbf{u} = (\mathbf{b}(\mathbf{x})^T\mathbf{g}_1, \mathbf{b}(\mathbf{x})^T\mathbf{g}_2)$ to represent the gradient of f. The functions f and $\mathbf{u}$ satisfy the relationship

$$\int_\Omega \nabla f(\mathbf{x}) \cdot \nabla b_j(\mathbf{x})\ d\mathbf{x} = \int_\Omega \mathbf{u}(\mathbf{x}) \cdot \nabla b_j(\mathbf{x})\ d\mathbf{x}, \tag{2}$$

for all the basis functions b_j. This relationship ensures that $\mathbf{u}$ is an approximation of the gradient of f in a weak sense.

Constraint (2) is equivalent to the relationship

$$L\mathbf{c} - G_1\mathbf{g}_1 - G_2\mathbf{g}_2 = 0, \tag{3}$$

where L is a discrete approximation to the negative Laplace operator and (G_1, G_2) is a discrete approximation to the transpose of the gradient operator.

We now consider the minimiser of the functional

$$
\begin{aligned}
J_\alpha(\mathbf{c}, \mathbf{g}_1, \mathbf{g}_2) &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{b}(\mathbf{x}^{(i)})^T\mathbf{c} - y^{(i)})^2 + \alpha \int_\Omega \sum_{s=1}^{2} \nabla(\mathbf{b}^T\mathbf{g}_s)\cdot\nabla(\mathbf{b}^T\mathbf{g}_s)\ d\mathbf{x} \\
&= \mathbf{c}^T A\mathbf{c} - 2\mathbf{d}^T\mathbf{c} + \mathbf{y}^T\mathbf{y}/n + \alpha\left(\mathbf{g}_1^T L\mathbf{g}_1 + \mathbf{g}_2^T L\mathbf{g}_2\right). \qquad (4)
\end{aligned}
$$

Our smoothing problem consists of minimising this functional over all vectors $\mathbf{c}, \mathbf{g}_1, \mathbf{g}_2$ defined on the domain Ω_h, subject to the constraint (3).

The matrices L, G_1 and G_2 are constructed independent of the data points but the matrix

$$
A = \frac{1}{n}\sum_{i=1}^{n}\mathbf{b}(\mathbf{x}^{(i)})\mathbf{b}(\mathbf{x}^{(i)})^T,
$$

and vector

$$
\mathbf{d} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{b}(\mathbf{x}^{(i)})y^{(i)},
$$

are assembled by sweeping through the data points. Matrix A is symmetric, nonnegative, and sparse. In regions where the support of the basis functions do not contain any data points the matrix is zero.

The smoothing function defined by $f(\mathbf{x}) = \mathbf{b}(\mathbf{x})^T\mathbf{c}$ has essentially the same smoothing properties as the original thin plate smoothing spline, provided the discretisation is small enough, see [6].

By using Lagrange multipliers, the minimisation problem may be rewritten as the solution of the following linear system of equations

$$
\begin{bmatrix} A & 0 & 0 & L \\ 0 & \alpha L & 0 & -G_1^T \\ 0 & 0 & \alpha L & -G_2^T \\ L & -G_1 & -G_2 & 0 \end{bmatrix}
\begin{bmatrix} \mathbf{c} \\ \mathbf{g}_1 \\ \mathbf{g}_2 \\ \mathbf{w} \end{bmatrix}
=
\begin{bmatrix} \mathbf{d} \\ 0 \\ 0 \\ 0 \end{bmatrix}
-
\begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{h}_3 \\ \mathbf{h}_4 \end{bmatrix}, \qquad (5)
$$

where $\mathbf{w}$ is a Lagrange multiplier associated with constraint (3). The vectors $\mathbf{h}_1, \cdots, \mathbf{h}_4$ store the Dirichlet boundary information.

For examples where the exact form of a the minimiser is known, the Dirichlet boundary conditions can be set accordingly, see [8]. Different boundary conditions will give different forms of the minimiser, we plan to explore this idea further as a means to incorporate prior information.

3 Solution of the Linear System

One way to solve (5) is to eliminate all the variables except $\mathbf{g}_1$ and $\mathbf{g}_2$, which gives

$$
\begin{bmatrix} \alpha L + G_1^T Z G_1 & G_1^T Z G_2 \\ G_2^T Z G_1 & \alpha L + G_2^T Z G_2 \end{bmatrix}
\begin{bmatrix} \mathbf{g}_1 \\ \mathbf{g}_2 \end{bmatrix}
=
\begin{bmatrix} G_1^T L^{-1}\mathbf{d} \\ G_2^T L^{-1}\mathbf{d} \end{bmatrix}
-
\begin{bmatrix} \mathbf{h}_2 + G_1^T L^{-1}\mathbf{h} \\ \mathbf{h}_3 + G_2^T L^{-1}\mathbf{h} \end{bmatrix}, \qquad (6)
$$

where $Z = L^{-1}AL^{-1}$, $\mathbf{h} = \mathbf{h}_1 - AL^{-1}\mathbf{h}_4$ and $\mathbf{c} = L^{-1}\left(G_1\mathbf{g}_1 + G_2\mathbf{g}_2 - \mathbf{h}_4\right)$.

Applying Z to a vector is equivalent to solving two systems of equations involving the Laplacian, so it is important to use an efficient Poisson solver. Fortunately there are techniques, such as the multigrid method, that are optimal for the solution of such problems.

System (6) is symmetric positive definite and may be solved using the preconditioned Conjugate Gradient (PCG) method.

The matrix on the left-hand side of (6) can be rewritten as

$$\left[\alpha\widehat{L} + \widehat{K}^T\widehat{K}\right], \tag{7}$$

where

$$\widehat{L} = \begin{bmatrix} L & 0 \\ 0 & L \end{bmatrix} \quad \text{and } \widehat{K}^T = \begin{bmatrix} G_1^T \\ G_2^T \end{bmatrix} L^{-1}A^{1/2}.$$

This is similar to the type of matrix arising in Tikhonov regularisation.

The system $\widehat{K}^T\widehat{K}$ is symmetric positive semidefinite, and for small values of α (6) is close to a semidefinite system. As shown in Section 5 the convergence rate of the PCG method deteriorates as α is reduced and in some cases the PCG method diverges.

3.1 Properties of Gradient Matrices

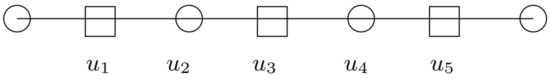


Fig. 1. Example grid in 1D used to demonstrate that the matrix G_1 may be singular.

The matrices G_s may be singular. To illustrate this consider the 1D grid shown in Figure 1. The stencil corresponding to the finite element approximation of the gradient is

$$h\begin{bmatrix}-1 & 0 & 1\end{bmatrix}$$

and thus the vector $\mathbf{u} = \begin{bmatrix}a \; 0 \; a \; 0 \; a\end{bmatrix}^T$ belongs to the null space of G_1. In other words we can assign any constant value to the points marked by squares in Figure 1 and the gradient will be zero. Similar examples can be constructed for higher dimensional grids.

Instead of the domain shown in Figure 1, consider the domain labelled Subgrid 1 in Figure 2. The matrix G_1 defined on this domain is non-singular. We require domains like those shown in Figure 1 in order to use the multigrid method to solve the Laplacian. This lead to the use of the domain decomposition method to define subgrids where the matrices G_s are non-singular, thus reducing the convergence rate (see Section 4). As the subgrids are relatively small it is possible to use a different Laplacian solver, such as a sparse direct method.

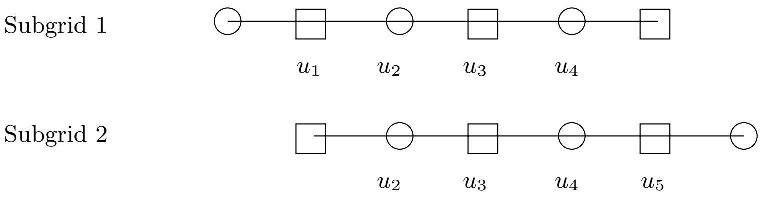


Fig. 2. Example subdivision of a grid in 1D where the gradient matrix is non-singular.

4 Preconditioners

For practical applications small α values are not usually of interest because the results will contain too much noise, however search algorithms like GCV may require some evaluations for small α before they find the optimal value of α. For larger values of α ($\alpha > 10^{-5}$) the preconditioner

$$M = \begin{bmatrix} L^{-1} & 0 \\ 0 & L^{-1} \end{bmatrix} \tag{8}$$

works very well, but for smaller α a different preconditioner is required. Finding an effective preconditioner for small values of α was a challenge.

The type of preconditioners we consider here are subspace correction preconditioners. In particular we focused on the algorithms presented in [4] and [11]. Recall that we are trying to solve (6) using the PCG method as it is a positive definite system. Solving (6) directly on a subspace is difficult because of the presence of the inverse Laplacian. Therefore we noted that (6) is a short cut to solving (5). It is straightforward to project (5) onto the subgrid and, once again, eliminate all of the variables except $\mathbf{g}_1$ and $\mathbf{g}_2$ (defined on the subgrid). Note that by using (5) and then eliminating the variables, the right-hand-side of the equation will contain some information about $\mathbf{c}$ and $\mathbf{w}$; this is how the global L^{-1} is incorporated into the system. As a consequence, values for $\mathbf{c}$ and $\mathbf{w}$ must be generated during the preconditioning step.

The first approach we looked at was the two-level preconditioner presented by [4], which is designed for matrices similar to those given by (7). Unfortunately the form of the matrices G_s would not allow the use of the injection operator as in the paper referenced above.

The next approach we tried was to use multiplicative and additive Schwarz methods with the subgrids defined in such a way as to ensure that the matrices G_s are non-singular. The PCG method was also used to solve (6) defined on each subgrid as described at the beginning of this section. The approach improved the convergence rate slightly, but not enough to offset the extra cost of generating the $\mathbf{c}$ and $\mathbf{w}$ vectors.

5 Multiplicative Schwarz

A preconditioner which gives robust results is the multiplicative Schwarz preconditioner, combined with a sparse direct method to solve (5) on each subgrids. This avoids the inverse Laplacian found in (6). The sparse direct method we used is `umfpack_di_numeric` from the UMFPACK package [2].

The (symmetric) multiplicative Schwarz approach is the same as Algorithm 3.4 in [11]. Equation 2.1 in [11] was repeatedly applied until either $\|\mathbf{u}^{k+1} - \mathbf{u}^k\|/\|\mathbf{u}^{k+1}\| < 10^{-2}$ or the number of iterations reached an upper bound (currently 20). We have chosen this approach as it is cheap and the tolerance does not have to be too accurate for the preconditioner.

We now present two examples highlighting the performance of the multiplicative Schwarz algorithm. The test runs were carried out using a code developed by Stals. The code is a parallel finite element code and we plan to move these test runs to a parallel machine, at which time we will explore the use of the additive Schwarz algorithm.

In the first example the data points $\mathbf{x}^{(i)}$ $(1 \le i \le n)$ sit on the lattice defined by dividing the square $[1/p, 1-1/p] \times [1/p, 1-1/p]$ into $p-1$ equally spaced subsquares, where $p = 100$ and hence $n = 99^2$. The values assigned to each data point were $y^{(i)} = \mathbf{x}_1^{(i)} + \mathbf{x}_2^{(i)}$. The Dirichlet boundary conditions were set so that the expected value for each entry of $\mathbf{g}_s$ is 1 and $\mathbf{w} = 0$ (for all values of α).

The multi-linear basis functions in the finite element formulation fit the solution exactly, so any error in the solution is due to algebraic error. This problem is a good test of the convergence of the PCG method. The stopping criterion used in the PCG algorithm is based on the Hestenes-Stiefel rule [1, 9]. A small tolerance of 10^{-12} was set to ensure that the error remained small for small α.

Table 1 shows the difference in the convergence between the inverse Laplacian preconditioner and the multiplicative Schwarz preconditioner for different values of m and α. The number of subgrids was kept at 4 and four levels of overlap was used. In all examples the multiplicative Schwarz preconditioner was faster than the inverse Laplacian preconditioner. There is a sudden jump in the number of iterations for the $\alpha = 10^{-7}$ and $m = 4225$ case indicating that we may need to look at increasing the overlap.

The second example also used a uniform grid with $p = 1000$ and $n = 999^2$. The values assigned to each data point were $y^{(i)} = f_y(\mathbf{x}^{(i)})$ where $f_y(\mathbf{x}) = \exp\left(-30\|0.65 - \mathbf{x}\|_2^2\right) + \exp\left(-30\|0.35 - \mathbf{x}\|_2^2\right)$. The boundary conditions are $\mathbf{h}_1 = f_y|_\Gamma$, $(\mathbf{h}_2, \mathbf{h}_3) = \nabla f_y|_\Gamma$ and $\mathbf{h}_4 = -\alpha \Delta f_y|_\Gamma$ where Γ is the boundary of Ω_h.

The solution depends on the choice of α. The tolerance in the stopping criterion was decreased to 10^{-6} and the number of subgrids remained at four with four levels of overlap. Table 2 tabulates the convergence results.

6 Conclusion

The multiplicative Schwarz algorithm with a sparse-direct solver on the subgrids is an efficient preconditioner for the systems of equations arising from the discrete thin-plate spline method.

Table 1. Convergence rate for the first test problem. The time is in seconds. The column labelled Laplace is the results for the preconditioner given by (8). The column labelled Mult S is the multiplicative Schwarz preconditioner. CG It is the total number of PCG iterations and MS It is the total number of times the Multiplicative Schwarz algorithm was called.

m	$\alpha = 10^{-6}$					$\alpha = 10^{-7}$				
	Laplace		Mult S			Laplace		Mult S		
	CG It	Time	CG It	MS It	Time	CG It	Time	CG It	MS It	Time
1089	72	16	4	21	4	266	96	5	39	7
4225	120	116	4	21	18	254	247	49	370	187
16641	147	713	6	56	146	283	1424	7	54	147
66049	141	2999	16	164	1612	309	6611	12	144	1401

Table 2. Convergence rate for the second test problem. The column labelling is the same as the first example.

m	$\alpha = 10^{-6}$					$\alpha = 10^{-7}$				
	Laplace		Mult S			Laplace		Mult S		
	CG It	Time	CG It	MS It	Time	CG It	Time	CG It	MS It	Time
1089	36	27	3	23	22	66	33	3	34	23
4225	37	66	2	17	33	93	122	3	42	45
16641	37	268	2	27	104	98	590	3	36	118
66049	38	1133	3	59	639	110	2884	3	50	581

References

[1] M. Arioli. Backward error analysis and stopping criteria for Krylov space method. In *20th Biennial Conference on Numerical Analysis*, pages 1–41, University of Dundee, June 2003.

[2] T. Davis. UMFPACK, Version 5. University of Florida, Gainsevilla, Florida, May 2006.

[3] J. Duchon. Splines minimizing rotation-invariant. In *Lecture Notes in Math*, volume 571, pages 85–100. Springer-Verlag, 1977.

[4] M. Hanke and C. R. Vogel. Two-level preconditioners for regularized inverse problems I: Theory. *Numer. Math.*, 83:385–402, 1999.

[5] M. F. Hutchinson. A stochastic estimator of the trace of the influence matrix for Laplacian smoothing splines. *Comm. Statist. Simulation Comput.*, 19(2):433–450, 1990.

[6] S. Roberts, M. Hegland, and I. Altas. Approximation of a thin plate spline smoother using continuous piecewise polynomial functions. *SIAM J. Numer. Anal.*, 41(1):208–234, 2003.

[7] L. Stals and S. Roberts. Verifying convergence rates of discrete thin-plate splines in 3D. In Rob May and A. J. Roberts, editors, *Proc. of 12th Computational Techniques and Applications Conference CTAC-2002*, volume 46, pages C515–C529, June 2005. Online `http://anziamj.austms.org.au/V46/CTAC2004/home.html`.

[8] L. Stals and S. Roberts. Smoothing and filling holes with Dirichlet boundary conditions. Submitted to the Proceedings of the International Conference on High Performance Scientific Computing, 2006.

[9] Z. Strakoš and P. Tichý. On estimation of the A-norm of the error in CG and PCG. *PAMM*, 3(1):553–554, December 2003.

[10] G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1990.

[11] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, December 1992.

Domain Decomposition Algorithms for an Indefinite Hypersingular Integral Equation in Three Dimensions

Ernst P. Stephan[1], Matthias Maischak[2], and Thanh Tran[3]

[1] Institut für Angewandte Mathematik, Leibniz Universität Hannover, 30167 Hannover, Germany. stephan@ifam.uni-hannover.de
[2] BICOM, Brunel University, Uxbridge, UB8 3PH, UK. mastmmm@brunel.ac.uk
[3] School of Mathematics and Statistics, The University of New South Wales, Sydney 2052, Australia. thanh.tran@unsw.edu.au

1 Introduction

In this paper we report on a non-overlapping and an overlapping domain decomposition method as preconditioners for the boundary element approximation of an indefinite hypersingular integral equation on a surface. The equation arises from an integral reformulation of the Neumann screen problem with the Helmholtz equation in the exterior of a screen in $\mathbb{R}^3$.

It is well-known that the linear algebraic system arising from the boundary element approximation to this integral equation is indefinite, and an iterative method like GMRES can be used to solve the system. Preconditioners by domain decomposition methods can be used to reduce the number of iterations. A non-overlapping preconditioner for the hypersingular integral equation reformulation of the 2D problem is studied in [10]. In this paper we study both non-overlapping and overlapping methods for the 3D problem. We prove that the convergence rate depends logarithmically on H/h for the non-overlapping method, and on H/δ for the overlapping method, where H and h are respectively the size of the coarse mesh and fine mesh, and δ is the overlap size. We note that domain decomposition methods with finite element approximations for the Helmholtz equation have been studied by many authors; see e.g. [2, 3, 5].

2 The Neumann Screen Problem and Boundary Integral Equation

Let Γ be a planar surface piece in $\mathbb{R}^3$ with polygonal boundary. The problem to be studied consists in finding U satisfying

$$\begin{aligned} \Delta U + k^2 U &= 0, && \text{in } \Omega_\Gamma := \mathbb{R}^3 - \overline{\Gamma}, \\ \frac{\partial U}{\partial \boldsymbol{n}} &= g, && \text{on } \Gamma, \\ \frac{\partial U}{\partial \boldsymbol{n}} - ikU &= o(1/r), && \text{as } r := |x| \to \infty, \end{aligned} \tag{1}$$

where k is a nonzero constant and g a given function. The condition at infinity is the well-known radiation condition.

The solution U can be expressed as a double-layer potential

$$U(x) = \frac{1}{4\pi} \int_\Gamma u(y) \frac{\partial}{\partial \boldsymbol{n}_y} \frac{e^{ik|x-y|}}{|x-y|} ds_y, \quad x \in \Omega_\Gamma,$$

where $u = [U]$ is the jump of U across Γ. It is shown in [9] that solving (1) is equivalent to solving

$$D_k u(x) = g(x), \quad x \in \Gamma, \tag{2}$$

where the operator D_k is defined as

$$D_k \phi(x) := -\frac{1}{4\pi} \int_\Gamma \phi(y) \frac{\partial}{\partial \boldsymbol{n}_x} \frac{\partial}{\partial \boldsymbol{n}_y} \frac{e^{ik|x-y|}}{|x-y|} ds_y, \quad x \in \Gamma. \tag{3}$$

The Sobolev spaces $\widetilde{H}^{1/2}(\Gamma)$ and $H^{1/2}(\Gamma)$ and their duals $H^{-1/2}(\Gamma)$ and $\widetilde{H}^{-1/2}(\Gamma)$ (respectively) are defined as usual; see [7]. It is shown in [9] that the operator D_0 defined as in (3) with $k = 0$ is a continuous and surjective mapping from $\widetilde{H}^{1/2}(\Gamma)$ onto $H^{-1/2}(\Gamma)$. Moreover, D_k can be written as

$$D_k = D_0 + K, \tag{4}$$

where K is a bounded operator from $\widetilde{H}^{1/2}(\Gamma)$ into $L^2(\Gamma)$. Let

$$b(v, w) := \langle D_k v, w \rangle \quad \forall v, w \in \widetilde{H}^{1/2}(\Gamma),$$

(where $\langle D_k v, w \rangle$ denotes the duality pairing which coincides with the L_2 inner product on Γ if $D_k v, w \in L_2(\Gamma)$) then the bilinear form $b(\cdot, \cdot)$ can be written as

$$b(v, w) = a(v, w) + c(v, w),$$

where $a(v, w) = \langle D_0 v, w \rangle$ and $c(v, w) = \langle K v, w \rangle$. The bilinear form $a(\cdot, \cdot)$ is a positive-definite and symmetric bilinear form on $\widetilde{H}^{1/2}(\Gamma)$ satisfying

$$a(v, v) \simeq \|v\| \widetilde{H}^{1/2}(\Gamma)^2 \quad \forall v \in \widetilde{H}^{1/2}(\Gamma), \tag{5}$$

whereas $b(\cdot, \cdot)$ is indefinite and satisfies

$$\operatorname{Re}(b(v, v)) \geq \gamma \|v\| \widetilde{H}^{1/2}(\Gamma)^2 - \eta \|v\| L^2(\Gamma)^2 \quad \forall v \in \widetilde{H}^{1/2}(\Gamma),$$

for some $\gamma > 0$ and $\eta > 0$ independent of v.

A weak form of (2) is the problem of finding

$$u \in \widetilde{H}^{1/2}(\Gamma): \quad b(u, v) = \langle g, v \rangle \quad \forall v \in \widetilde{H}^{1/2}(\Gamma). \tag{6}$$

The problem (6) will be approximated by first constructing a finite-dimensional subspace $\mathcal{S} \subset \widetilde{H}^{1/2}(\Gamma)$, and then finding

$$u_\mathcal{S} \in \mathcal{S}: \quad b(u_\mathcal{S}, v) = \langle g, v \rangle \quad \forall v \in \mathcal{S}. \tag{7}$$

3 Additive Schwarz Algorithm

3.1 General Framework

Additive Schwarz methods provide fast solutions to (7) by solving (at the same time) problems of smaller size. Let $\mathcal{S}$ be decomposed as

$$\mathcal{S} = \mathcal{S}_0 + \cdots + \mathcal{S}_J, \tag{8}$$

where $\mathcal{S}_i$, $i = 0, \ldots, J$, are subspaces of $\mathcal{S}$. Let $Q_i : \mathcal{S} \to \mathcal{S}_i$ be projections defined by

$$b_i(Q_i v, w) = b(v, w) \quad \forall v \in \mathcal{S}, \forall w \in \mathcal{S}_i, \tag{9}$$

where the bilinear forms $b_i(\cdot, \cdot)$, $i = 0, \ldots, J$, are to be defined later. Then the additive Schwarz method for (7) consists in solving the equation

$$Q u_{\mathcal{S}} = \tilde{g},$$

where $Q = Q_0 + \cdots + Q_J$ is the additive Schwarz operator and $\tilde{g}$ is given by $\tilde{g} = g_0 + \cdots + g_J$ with $g_i \in \mathcal{S}_i$ being solutions of

$$b_i(g_i, w) = \langle g, w \rangle \quad \forall w \in \mathcal{S}_i.$$

This equation is solved iteratively by the GMRES method. Starting with an initial guess u_0 and the initial residual $r_0 = \tilde{g} - Q u_0$, we compute the mth iterate u_m as $u_m = u_0 + z_m$ where z_m is chosen to minimize the residual norm $\|\tilde{g} - Q(u_{m-1} + z)\|a$, where $\|v\|a = a(v, v)$. It is proved in [4] that

$$\|r_m\|a \leq \left(1 - \frac{C_1^2}{C_2}\right)^{m/2} \|r_0\|a,$$

where $r_m = \tilde{g} - Q u_m$ and

$$C_1 = \inf_{v \in \mathcal{S}} \frac{a(v, Qv)}{a(v, v)}, \quad C_2 = \sup_{v \in \mathcal{S}} \frac{a(Qv, Qv)}{a(v, v)}. \tag{10}$$

We now define two different subspace decomposition of the form (8) which result in two different preconditioners: a non-overlapping method and an overlapping method.

3.2 Non-overlapping Algorithm

Boundary Element Space

We first define the finite-dimensional space $\mathcal{S}$ in (7) on a two-level grid.

The coarse grid. Assume that Γ is partitioned into subdomains Γ_i, $i = 1, \ldots, N$, where each subdomain Γ_i is the image of the reference square $\hat{R} = (-1, 1)^2$ under a smooth bijective mapping $\mathcal{F}_i : \hat{R} \to \Gamma_i$. Denoting by H the diameter of the subdomains, we assume that

$$\|J_{\mathcal{F}_i}\| L_\infty(\hat{R}) \preceq H \quad \text{and} \quad \|J_{\mathcal{F}_i^{-1}}\| L_\infty(\hat{R}) \preceq H^{-1},$$

where $J_{\mathcal{F}_i}$ denotes the Jacobian matrix of the transformation and the norm is a matrix norm. The partition is assumed to be conforming in the sense that the non-empty intersection of a pair of distinct subdomains is a single common vertex or edge of both subdomains, and that each vertex of the domain Γ coincides with at least one subdomain vertex.

We define on this coarse grid the space $\mathcal{V}_0$ of continuous piecewise bilinear functions, vanishing on the boundary of Γ.

The fine grid. Each subdomain Γ_i is further divided into disjoint quadrilateral or triangular elements, giving a locally uniform mesh of element of size h_i in Γ_i. We denote by h the maximum value of h_i, $i = 1, \dots, N$.

The finite-dimensional space $\mathcal{S}$ is defined as the space of continuous piecewise-bilinear functions (in the case of quadrilateral elements) or piecewise-linear functions (in the case of triangular elements) on the fine grid, vanishing on the boundary of Γ. We also define subspaces $\mathcal{V}_j = \mathcal{S} \cap \widetilde{H}^{1/2}(\Gamma_j)$ of functions in $\mathcal{S}$ supported in $\overline{\Gamma}_j$.

We denote by $\mathcal{N} = \{\boldsymbol{x}_k : k \in \mathcal{I}\}$ the set of all vertices of elements in the fine grid which are not on the boundary of Γ (where $\mathcal{I}$ is some index set), by $\mathcal{N}_w = \{\boldsymbol{x}_k \in \mathcal{N} : \boldsymbol{x}_k \text{ lies on a subdomain boundary}\}$ the wirebasket, and by $\phi_k \in \mathcal{S}$ the nodal basis function at $\boldsymbol{x}_k$, i.e., $\phi_k(\boldsymbol{x}_l) = \delta_{kl}$.

Subspace Decomposition

The non-overlapping method is defined by the subspace decomposition (8) where

$$\begin{aligned}
&\mathcal{S}_0 = \Pi_F \mathcal{V}_0, && \text{(coarse space)}\\
&\mathcal{S}_1 = \operatorname{span}\{\{\}\phi_k : \boldsymbol{x}_k \in \mathcal{N}_w\}, && \text{(wirebasket space)}\\
&\mathcal{S}_i = \mathcal{V}_{i-1} \quad \forall i = 2, \dots, N+1, && \text{(interior spaces)},
\end{aligned}$$

in which Π_F is the interpolation operator which interpolates continuous functions into functions in $\mathcal{S}$. (Note that $J = N + 1$.)

The bilinear forms $b_i(\cdot,\cdot)$ on $\mathcal{S}_i$ (see (9)) are defined as follows:

$$\begin{aligned}
b_0(v,w) &= b(\Pi_C v, \Pi_C w) \quad \forall v, w \in \mathcal{S}_0,\\
b_1(v,w) &= \sum_{j=1}^{N} \sum_{\boldsymbol{x}_k \in \partial\Gamma_j} h_j v(\boldsymbol{x}_k) w(\boldsymbol{x}_k), \quad \forall v, w \in \mathcal{S}_1,\\
b_i(v,w) &= a(v,w) \quad \forall v, w \in \mathcal{S}_i,\ i = 2, \dots, J.
\end{aligned}$$

Here Π_C is the interpolation operator that interpolates continuous functions into functions in $\mathcal{V}_0$.

Algorithm

The preconditioning technique is in practice performed by computing the action of the inverse of the preconditioner B on a residual $r \in \mathcal{S}$ when GMRES is used to solve (7) iteratively. This consists of the solution of independent problems on each of the subspaces involved in the decomposition.

1. Coarse space correction:

$$u_0 \in \mathcal{S}_0 : \quad b_0(u_0, v) = \langle r, v \rangle \quad \forall v \in \mathcal{S}_0$$

2. Wirebasket space correction:

$$u_1 \in \mathcal{S}_1 : \quad b_1(u_1, v) = \langle r, v \rangle \quad \forall v \in \mathcal{S}_1$$

3. Interior space corrections:

$$u_i \in \mathcal{S}_i : \quad b_i(u_i, v) = \langle r, v \rangle \quad \forall v \in \mathcal{S}_i,\ i = 2, \ldots, J.$$

4. Preconditioned residual:

$$B^{-1} r = \sum_{j=0}^{J} u_j.$$

Matrix Representation

Let $\boldsymbol{\Psi}$ be the set of nodal basis functions. We use the bilinear form $a(\cdot,\cdot)$ (respectively, $b(\cdot,\cdot)$) to compute the stiffness matrix $\boldsymbol{A}_a$ (respectively, $\boldsymbol{A}_b$). The coefficient vector $\boldsymbol{v}$ of a function $v \in \mathcal{S}$ is given as $v = \boldsymbol{\Psi}^T \boldsymbol{v}$, where T denotes transpose. Let $\boldsymbol{\Phi}_0$ be the vector composed of the nodal basis functions for the subspace $\mathcal{S}_0$. Then we denote by $\boldsymbol{R}_0$ the rectangular matrix that represents $\boldsymbol{\Phi}_0$ in the basis $\boldsymbol{\Psi}$, i.e., $\boldsymbol{\Phi}_0 = \boldsymbol{R}_0 \boldsymbol{\Psi}$. We also define $\boldsymbol{R}_i$, $i = 1, \ldots, J$, to be matrices of entries 0 and 1 such that $\boldsymbol{R}_i \boldsymbol{\Psi}$ forms the nodal bases for $\mathcal{S}_i$. If $v = B^{-1} r$ then $\boldsymbol{v} = \sum_{i=0}^{J} \boldsymbol{R}_i^T \boldsymbol{A}_i^{-1} \boldsymbol{R}_i \boldsymbol{M} \boldsymbol{r}$ where, noting the bilinear form used in each subspace,

$$\boldsymbol{A}_0 = \boldsymbol{R}_0 \boldsymbol{A}_b \boldsymbol{R}_0^T, \quad \boldsymbol{A}_1 = \boldsymbol{R}_1 \boldsymbol{D} \boldsymbol{R}_1^T, \quad \boldsymbol{A}_i = \boldsymbol{R}_1 \boldsymbol{A}_a \boldsymbol{R}_i^T,\ i = 2, \ldots, J.$$

The size of $\boldsymbol{A}_1$ is large; however, the matrix $\boldsymbol{D}$ computed with the bilinear form $b_1(\cdot,\cdot)$ is a diagonal matrix.

3.3 Overlapping Algorithm

Overlapping Subdomains

As in [11], we extend each subdomain Γ_j in the following way. First we define, for some $\delta > 0$ called the overlap size,

$$\tilde{\mathcal{V}_j} = \operatorname{span}\{\{\}\phi_k : \boldsymbol{x}_k \notin \overline{\Gamma}_j,\ \operatorname{dist}(\boldsymbol{x}_k, \partial\Gamma_j) \le \delta\},$$

and denote

$$\tilde{\Gamma_j} = \operatorname{supp}\{\phi_k : \phi_k \in \tilde{\mathcal{S}_j}\},$$

which is the shaded area in Figure 1. (Here the distance is defined with the max norm $\|\boldsymbol{x}\| = \max\{|x_1|, |x_2|\}$ where $\boldsymbol{x} = (x_1, x_2)$.) The extended subdomain Γ_j' is then defined as $\Gamma_j' = \overline{\Gamma}_j \cup \tilde{\Gamma}_j$. We note that Γ_j' need not be a quadrilateral domain. Also, if δ is chosen such that $\delta \in (0, H]$, then

$$\operatorname{diam}(\Gamma_i') \simeq H. \tag{11}$$

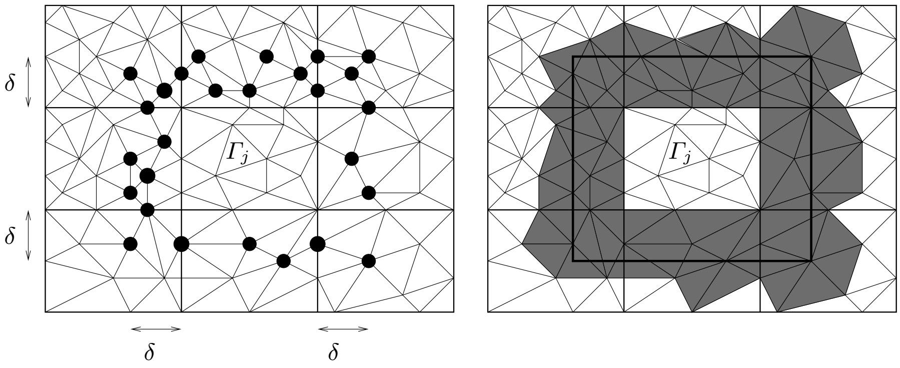


Fig. 1. • vertex at a distance δ to $\overline{\Gamma}_j$, $\tilde{\Gamma}_j$: shaded region, $\Gamma'_j = \Gamma_j \cup \tilde{\Gamma}_j$: overlapping subdomain.

Subspace Decomposition

The decomposition (8) is performed with subspaces $\mathcal{S}_j$, $j = 0, \ldots, J = N$, defined as

$$\begin{aligned} \mathcal{S}_0 &= \Pi_F \mathcal{V}_0, \\ \mathcal{S}_j &= \mathcal{V}_j \cup \tilde{\mathcal{V}}_j = \mathcal{S} \cap \widetilde{H}^{1/2}(\Gamma'_j) \quad \forall j = 1, \ldots, J. \end{aligned}$$

The bilinear forms $b_i(\cdot,\cdot)$ on $\mathcal{S}_i$ (see (9)) are defined as follows:

$$\begin{aligned} b_0(v,w) &= b(\Pi_C v, \Pi_C w) \quad \forall v, w \in \mathcal{S}_0, \\ b_i(v,w) &= a(v,w) \quad \forall v, w \in \mathcal{S}_i,\ i = 1, \ldots, J. \end{aligned}$$

Algorithm

The overlapping preconditioner is performed in the same manner as the non-overlapping version, with subspace corrections being

$$u_i \in \mathcal{S}_i : \quad b_i(u_i, v) = \langle r, v \rangle \quad \forall v \in \mathcal{S}_i,\ i = 0, \ldots, J.$$

Matrix Representation

As in the case of non-overlapping method, the updated residual vector is given by $\boldsymbol{v} = \sum_{i=0}^{J} \boldsymbol{R}_i^T \boldsymbol{A}_i^{-1} \boldsymbol{R}_i \boldsymbol{M} \boldsymbol{r}$ where

$$\boldsymbol{A}_0 = \boldsymbol{R}_0 \boldsymbol{A}_b \boldsymbol{R}_0^T, \quad \boldsymbol{A}_i = \boldsymbol{R}_1 \boldsymbol{A}_a \boldsymbol{R}_i^T,\ i = 2, \ldots, J.$$

3.4 Convergence

The preconditioned GMRES method using the non-overlapping and overlapping preconditioners converges with constants C_1 and C_2 (see (10)) slightly dependent on the mesh sizes H and h and the overlap size δ, as given in the following theorem.

Theorem 1.

- **Bound for** C_1**:** *There exists* $H_0 > 0$ *such that for all* $H \in (0, H_0]$ *and* $u \in \mathcal{S}$ *there hold*
$$\left(1 + \log^2 \frac{H}{h}\right)^{-1} a(u,u) \preceq a(u, Qu)$$
for the non-overlapping method, and
$$\left(1 + \log^2 \frac{H}{\delta}\right)^{-1} a(u,u) \preceq a(u, Qu)$$
for the overlapping method.
- **Bound for** C_2**:** *There exists* $H_1 > 0$ *such that for all* $H \in (0, H_1]$ *and* $u \in \mathcal{S}$ *there holds, for both methods,*
$$a(Qu, Qu) \preceq a(u,u)\,.$$

Proof. Sketch of the proof: First we note that

$$a(Qu, Qu) \simeq \|Qu\|{\widetilde{H}^{1/2}(\Gamma)}^2 \preceq \sum_{i=0}^{J} \|Q_i u\|{\widetilde{H}^{1/2}(\Gamma_i)}^2 \simeq \sum_{i=0}^{J} a(Q_i u, Q_i u).$$

Using this result, the boundedness of Q_0, and the definition of the projections Q_i, we can prove the bound for C_2.

The proof of the bound for C_1 is more complicated and involves the operator $P = P_0 + \cdots + P_J$ where P_i is defined as Q_i but with the bilinear form $a(\cdot,\cdot)$ in the place of $b(\cdot,\cdot)$. This operator P is in fact the additive Schwarz operator for the positive definite operator D_0 (see (4)). It is proved in [6] and [1] for the nonoverlapping method that

$$\left(1 + \log^2 \frac{H}{h}\right)^{-1} a(v,v) \preceq a(Pv, v),$$

and in [11] for the overlapping method that

$$\left(1 + \log^2 \frac{H}{\delta}\right)^{-1} a(v,v) \preceq a(Pv, v).$$

The difference in P and Q is due to the bounded operator K in (4), and further analysis to obtain similar estimates for Q involves this operator. For a detailed proof, see [8].

4 Numerical Experiments

We solve equation (2) with $k = 5$ and $g(x) \equiv 1$ on a uniform triangular mesh, by using the non-overlapping and overlapping preconditioners. In Table 1 we report on the number of iterations and CPU times (in seconds) when the equation is solved without any preconditioner, and when the non-overlapping preconditioner is used with various values of H/h. In Table 2 we report on the number of iterations and CPU times when the overlapping preconditioner is used with various values of H/δ. Choosing a suitable mesh size ratio H/h, we observe that the non-overlapping as well as the overlapping preconditioned method clearly outperform the non-preconditioned method in iteration numbers and CPU times. Here we use the GMRES without restart and stop if the relative residual is less than 10^{-10}. The local problems in computing the correction steps are solved by the GMRES or, if appropriate, by CG.

Table 1. Number of iterations and CPU times (in parentheses). WP: without preconditioner

DoF	WP	Non-overlapping			
		$H/h = 2$	$H/h = 4$	$H/h = 8$	$H/h = 16$
9	6 (0.01)	6 (0.01)			
49	17 (0.02)	17 (0.01)	17 (0.02)		
225	23 (0.02)	20 (0.03)	20 (0.02)	21 (0.04)	
961	31 (0.15)	21 (0.24)	21 (0.12)	23 (0.20)	23 (0.62)
3969	44 (3.02)	21 (4.39)	21 (1.62)	21 (1.75)	26 (4.16)
16129	63 (84.94)	21 (93.72)	21 (32.18)	21 (29.86)	24 (41.06)

Table 2. Number of iterations and CPU times (in parentheses) of overlapping method

DoF	$\delta = h$				$\delta = 2h$			
	$H/h = 2$	$H/h = 4$	$H/h = 8$	$H/h = 16$	$H/h = 2$	$H/h = 4$	$H/h = 8$	$H/h = 16$
9	6 (0.02)				6 (0.02)			
49	19 (0.01)	18 (0.02)			17 (0.03)	20 (0.02)		
225	28 (0.04)	24 (0.03)	22 (0.05)		22 (0.09)	26 (0.08)	26 (0.09)	
961	30 (0.39)	27 (0.23)	26 (0.34)	25 (0.86)	26 (0.65)	29 (0.48)	27 (0.57)	27 (1.17)
3969	30 (6.53)	28 (2.50)	27 (2.84)	28 (6.12)	31 (8.34)	31 (3.92)	28 (4.13)	28 (7.99)
16129	30 (135.47)	28 (43.76)	27 (41.14)	29 (58.46)	35 (166.31)	31 (53.10)	29 (49.80)	29 (69.04)

References

[1] M. Ainsworth and B.Q. Guo. Analysis of iterative sub-structuring techniques for boundary element approximation of the hypersingular operator in three dimensions. *Appl. Anal.*, 81:241–280, 2002.

[2] X.-C. Cai and O.B. Widlund. Domain decomposition algorithms for indefinite elliptic problems. *SIAM J. Sci. Stat. Comput.*, 13:243–258, 1992.

[3] M.A. Casarin and O.B. Widlund. Overlapping Schwarz methods for Helmholtz's equation. In C.-H. Lai, P.E. Bjørstad, M. Cross, and O.B. Widlund, editors, *Proceedings of the Eleventh International Conference on Domain Decomposition Methods*, pages 178–189. DDM.org, 1999.

[4] S.C. Eisenstat, H.C. Elman, and M.H. Schultz. Variational iterative methods for non-symmetric systems of linear equations. *SIAM J. Numer. Anal.*, 20:345–357, 1983.

[5] C. Farhat, A. Macedo, and M. Lesoinne. A two-level domain decomposition method for the iterative solution of high frequency exterior Helmholtz problems. *Numer. Math.*, 85:283–308, 2000.

[6] N. Heuer and E.P. Stephan. Iterative substructuring for hypersingular integral equations in $\mathbb{R}^3$. *SIAM J. Sci. Comput.*, 20:739–749, 1999.

[7] J.L. Lions and E. Magenes. *Non-Homogeneous Boundary Value Problems and Applications I.* Springer-Verlag, New York, 1972.

[8] M. Maischak, E.P. Stephan, and T. Tran. Additive Schwarz methods for an indefinite hypersingular integral equation in three dimensions. Technical report, 2007. In preparation.

[9] E.P. Stephan. Boundary integral equations for screen problems in $\mathbb{R}^3$. *Integral Equations Operator Theory*, 10:236–257, 1987.

[10] E.P. Stephan and T. Tran. Domain decomposition algorithms for indefinite hypersingular integral equations: the h and p versions. *SIAM J. Sci. Comput.*, 19:1139–1153, 1998.

[11] T. Tran and E.P. Stephan. An overlapping additive Schwarz preconditioner for boundary element approximations to the Laplace screen and Lamé crack problems. *J. Numer. Math.*, 12:311–330, 2004.

Editorial Policy

1. Volumes in the following three categories will be published in LNCSE:

i) Research monographs
ii) Lecture and seminar notes
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). 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 judgment 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 Lecture Notes in Computational Science and Engineering at the planning stage.

In exceptional cases some other multi-author-volumes may be considered in this category.

4. Format. Only works in English are considered. They should be submitted in camera-ready form according to Springer-Verlag's specifications.
Electronic material can be included if appropriate. Please contact the publisher.
Technical instructions and/or LaTeX macros are available via http://www.springer.com/east/home/math/math+authors?SGWID=5-40017-6-71391-0. The macros can also be sent on request.

General Remarks

Lecture Notes are printed by photo-offset from the master-copy delivered in camera-ready form by the authors. For this purpose Springer-Verlag provides technical instructions for the preparation of manuscripts. See also *Editorial Policy.*

Careful preparation of manuscripts will help keep production time short and ensure a satisfactory appearance of the finished book.

The following terms and conditions hold:

Categories i), ii), and iii):
Authors receive 50 free copies of their book. No royalty is paid. Commitment to publish is made by letter of intent rather than by signing a formal contract. Springer- Verlag secures the copyright for each volume.

For conference proceedings, editors receive a total of 50 free copies of their volume for distribution to the contributing authors.

All categories:
Authors are entitled to purchase further copies of their book and other Springer mathematics books for their personal use, at a discount of 33.3% directly from Springer-Verlag.

Addresses:

Timothy J. Barth
NASA Ames Research Center
NAS Division
Moffett Field, CA 94035, USA
e-mail: barth@nas.nasa.gov

Michael Griebel
Institut für Numerische Simulation
der Universität Bonn
Wegelerstr. 6
53115 Bonn, Germany
e-mail: griebel@ins.uni-bonn.de

David E. Keyes
Department of Applied Physics
and Applied Mathematics
Columbia University
200 S. W. Mudd Building
500 W. 120th Street
New York, NY 10027, USA
e-mail: david.keyes@columbia.edu

Risto M. Nieminen
Laboratory of Physics
Helsinki University of Technology
02150 Espoo, Finland
e-mail: rni@fyslab.hut.fi

Dirk Roose
Department of Computer Science
Katholieke Universiteit Leuven
Celestijnenlaan 200A
3001 Leuven-Heverlee, Belgium
e-mail: dirk.roose@cs.kuleuven.ac.be

Tamar Schlick
Department of Chemistry
Courant Institute of Mathematical
Sciences
New York University
and Howard Hughes Medical Institute
251 Mercer Street
New York, NY 10012, USA
e-mail: schlick@nyu.edu

Mathematics Editor at Springer:
Martin Peters
Springer-Verlag
Mathematics Editorial IV
Tiergartenstrasse 17
D-69121 Heidelberg, Germany
Tel.: *49 (6221) 487-8185
Fax: *49 (6221) 487-8355
e-mail: 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.*

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.E. Keyes, O.B. Widlund, W. Zulehner (eds.), *Domain Decomposition Methods in Science and Engineering XVII.*

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.

2. A. Quarteroni, F. Saleri, *Scientific Computing with MATLAB and Octave.*

3. H. P. Langtangen, *Python Scripting for Computational Science*.

4. H. Gardner, G. Manduchi, *Design Patterns for e-Science*.

5. M. Griebel, S. Knapek, G. Zumbusch, *Numerical Simulation in Molecular Dynamics*.

For further information on these books please have a look at our mathematics catalogue at the following URL: `www.springer.com/series/5151`