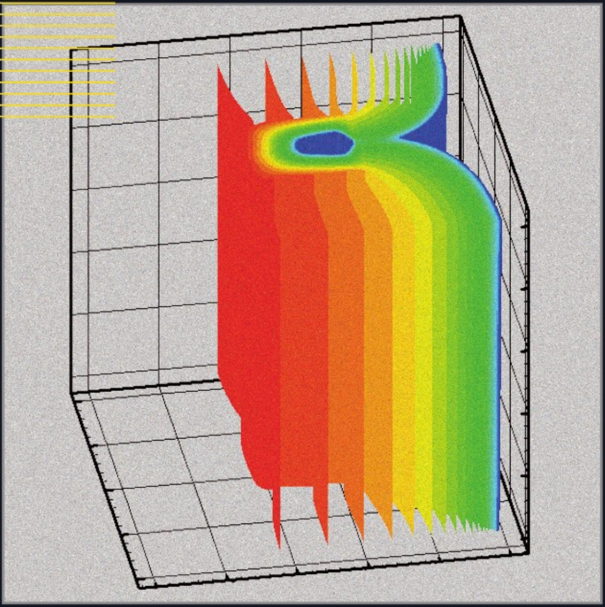
Lecture Notes in Computational Science and Engineering

55



Olof B. Widlund
David E. Keyes
Editors

Domain Decomposition Methods in Science and Engineering XVI

Springer

Lecture Notes in Computational Science and Engineering

55

Olof B. Widlund David E. Keyes (Eds.)

Domain Decomposition Methods in Science and Engineering XVI

With 222 Figures and 99 Tables

Springer

Editors

Olof B. Widlund
Courant Institute of Mathematical Sciences
New York University
251 Mercer Street
New York, NY 10012-1185, USA
E-mail: widlund@cims.nyu.edu

David E. Keyes
Department of Applied Physics & Mathematics
Columbia University
500 W. 120th Street, MC 4701
New York, NY 10027 USA
E-mail: david.keyes@columbia.edu

Library of Congress Control Number: 2006935914

Mathematics Subject Classification (2000): 65M55, 15-06, 35-06, 74-06, 76-06, 78-06

ISBN-10 3-540-34468-3 Springer Berlin Heidelberg New York
ISBN-13 978-3-540-34468-1 Springer Berlin Heidelberg New York

Springer is a part of Springer Science+Business Media
springer.com

The use of general descriptive names, registered names, trademarks, etc. in this publication does not imply, even in the absence of a specific statement, that such names are exempt from the relevant protective laws and regulations and therefore free for general use.

Typesetting: by the authors and techbooks using a Springer LaTeX macro package
Cover design: *design & production* GmbH, Heidelberg

Printed on acid-free paper SPIN: 11753995 46/techbooks 5 4 3 2 1 0

Preface

This volume is the definitive technical record of advances in the analysis algorithmic development, large-scale implementation, and application of domain decomposition methods in science and engineering presented at the Sixteenth International Conference on Domain Decomposition Methods. The conference was held in New York City, January 11-15, 2005. The largest meeting in this series to date, it registered 228 participants from 20 countries. The Courant Institute of Mathematical Sciences of New York University hosted the technical sessions. The School of Engineering and Applied Science of Columbia University hosted a pre-conference workshop on software for domain decomposition methods.

1 Background of the Conference Series

The International Conference on Domain Decomposition Methods has been held in eleven countries throughout Asia, Europe, and North America, beginning in Paris in 1987. Originally held annually, it is now spaced out at roughly 18-month intervals. A complete list of past meetings appears below.

The sixteenth instance of the International Conference on Domain Decomposition Methods was the sixth in the United States, and the first since 1997. In 1997, ASCI Red, the world's first Teraflops-scale computer, was just being placed into service at Sandia National Laboratories. The Bell Prize was won by an application that sustained 170 Gflop/s that year. An entirely new fleet of machines, algorithms, and codes has swept the research community in the intervening years. Now the Top 500 supercomputers in the world all sustain 2.0 Teraflop/s or more on the ScaLAPACK benchmark and nearly 200 Tflop/s have been sustained in simulations submitted to the Bell Prize competition.

The principal technical content of the conference has always been mathematical, but the principal motivation has been to make efficient use of distributed memory computers for complex applications arising in science and engineering. Thus, contributions from mathematicians, computer scientists, engineers, and scientists have always been welcome. Though the conference has grown up in the wake of commercial massively parallel processors, it is worth noting that many interesting applications of domain decomposition are not massively parallel at all. "Gluing together" just two subproblems to effectively exploit a different solver on each is also part of the technical fabric of the conference. Even as multiprocessing becomes commonplace, multiphysics modeling is in ascendancy, so the International Conference on Domain Decomposition Methods remains as relevant and as fundamentally interdisciplinary as ever. While research in domain decomposition methods is presented at numerous venues, the International Conference on Domain Decomposition Methods is the only regularly occurring international forum dedicated to interdisciplinary technical interactions between theoreticians and practitioners working in the creation, analysis, software implementation, and application of domain decomposition methods.

International Conferences on Domain Decomposition Methods:

- Paris, France, 1987
- Los Angeles, USA, 1988
- Houston, USA, 1989
- Moscow, USSR, 1990
- Norfolk, USA, 1991
- Como, Italy, 1992
- University Park (Pennsylvania), USA, 1993
- Beijing, China, 1995
- Ullensvang, Norway, 1996
- Boulder, USA, 1997
- Greenwich, UK, 1998
- Chiba, Japan, 1999
- Lyon, France, 2000
- Cocoyoc, Mexico, 2002
- Berlin, Germany, 2003
- New York, USA, 2005

International Scientific Committee on Domain Decomposition Methods:

- Petter Bjørstad, Bergen
- Roland Glowinski, Houston
- Ronald Hoppe, Augsburg & Houston
- Hideo Kawarada, Chiba
- David Keyes, New York
- Ralf Kornhuber, Berlin
- Yuri Kuznetsov, Houston
- Ulrich Langer, Linz
- Jacques Périaux, Paris
- Olivier Pironneau, Paris
- Alfio Quarteroni, Lausanne
- Zhong-ci Shi, Beijing
- Olof Widlund, New York
- Jinchao Xu, University Park

2 About the Sixteenth Conference

The 3.5-day conference featured 14 invited speakers, who were selected from about three times this number of nominees by the International Scientific Committee, with the goals of mixing traditional leaders and “new blood,” featuring mainstream and new directions, and reflecting the international diversity of the community. There were 160 presentations altogether. Sponsorship from several U.S. scientific agencies and organizations (listed below) made it possible to offer about 20 travel fellowships to graduate students and post-docs from the U.S. and abroad.

Sponsoring Organizations:

- Argonne National Laboratory
- Lawrence Livermore National Laboratory
- Sandia National Laboratories
- U. S. Army Research Office
- U. S. Department of Energy, National Nuclear Security Administration
- U. S. National Science Foundation
- U. S. Office of Naval Research

Cooperating Organizations:

- Columbia University, School of Engineering & Applied Sciences
- New York University, Courant Institute of Mathematical Sciences
- Society for Industrial and Applied Mathematics, Activity Group on Supercomputing

Local Organizing Committee Members:

- Randolph E. Bank, University of California, San Diego
- Timothy J. Barth, NASA Ames Research Center
- Marsha Berger, New York University
- Susanne Brenner, University of South Carolina
- Charbel Farhat, University of Colorado
- Donald Goldfarb, Columbia University
- David E. Keyes, Columbia University (Co-Chair)
- Michael L. Overton, Courant Institute, New York University
- Charles Peskin, New York University
- Barry Smith, Argonne National Laboratory
- Marc Spiegelman, Columbia University
- Ray Tuminaro, Sandia National Laboratory
- Panayot Vassilevski, Lawrence Livermore National Laboratory
- Olof Widlund, New York University (Co-Chair)
- Margaret H. Wright, New York University

3 About Domain Decomposition Methods

Domain decomposition, a form of divide-and-conquer for mathematical problems posed over a physical domain, as in partial differential equations, is the most common paradigm for large-scale simulation on massively parallel, distributed, hierarchical memory computers. In domain decomposition, a large problem is reduced to a collection of smaller problems, each of which is easier to solve computationally than the undecomposed problem, and most or all of which can be solved independently and concurrently. Typically, it is necessary to iterate over the collection of smaller problems, and much of the theoretical interest in domain decomposition algorithms lies in ensuring that the number of iterations required is very small. Indeed, the best domain decomposition methods share with their cousins, multigrid methods, the property that the total computational work is linearly proportional to the size of the input data, or that the number of iterations required is at most logarithmic in the number of degrees of freedom of individual subdomains.

Algorithms whose work requirements are linear in the size of the input data in this context are said to be "optimal." Near optimal domain decomposition algorithms are now known for many, but certainly not all, important classes of problems that arise science and engineering. Much of the contemporary interest in domain decomposition algorithms lies in extending the classes of problems for which optimal algorithms are known.

Domain decomposition algorithms can be tailored to the properties of the physical system as reflected in the mathematical operators, to the number of processors available, and even to specific architectural parameters, such as cache size and the ratio of memory bandwidth to floating point processing rate.

Domain decomposition has proved to be an ideal paradigm not only for execution on advanced architecture computers, but also for the development of reusable, portable software. The most complex operation in a typical domain decomposition method — the application of the preconditioner — carries out in each subdomain steps nearly identical to those required to apply a conventional preconditioner to the undecomposed domain. Hence software developed for the global problem can readily be adapted to the local problem, instantly presenting lots of "legacy" scientific code for to be harvested for parallel implementations. Furthermore, since the majority of data sharing between subdomains in domain decomposition codes occurs in two archetypal communication operations — ghost point updates in overlapping zones between neighboring subdomains, and global reduction operations, as in forming an inner product — domain decomposition methods map readily onto optimized, standardized message-passing environments, such as MPI.

Finally, it should be noted that domain decomposition is often a natural paradigm for the modeling community. Physical systems are often decomposed into two or more contiguous subdomains based on phenomenological considerations, such as the importance or negligibility of viscosity or reactivity, or any other feature, and the subdomains are discretized accordingly, as independent tasks. This physically-based domain decomposition may be mirrored in the software engineering of the corresponding code, and leads to threads of execution that operate on contiguous subdomain blocks. These can be either further subdivided or aggregated to fit the granularity of an available parallel computer.

4 Bibliography of Selected Books and Survey Articles

1. P. Bjørstad, M. Espedal and D. E. Keyes, eds., *Proc. Ninth Int. Symp. on Domain Decomposition Methods for Partial Differential Equations* (Ullensvang, 1997), Wiley, New York, 1999.
2. S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods (2nd edition)*, Springer, New York, 2002.
3. T. F. Chan and T. P. Mathew, *Domain Decomposition Algorithms*, Acta Numerica, 1994, pp. 61-143.
4. T. F. Chan, R. Glowinski, J. Périaux and O. B. Widlund, eds., *Proc. Second Int. Symp. on Domain Decomposition Methods for Partial Differential Equations* (Los Angeles, 1988), SIAM, Philadelphia, 1989.
5. T. F. Chan, R. Glowinski, J. Périaux, O. B. Widlund, eds., *Proc. Third Int. Symp. on Domain Decomposition Methods for Partial Differential Equations* (Houston, 1989), SIAM, Philadelphia, 1990.
6. T. Chan, T. Kako, H. Kawarada and O. Pironneau, eds., *Proc. Twelfth Int. Conf. on Domain Decomposition Methods in Science and Engineering* (Chiba, 1999), DDM.org, Bergen, 2001.
7. N. Débit, M. Garbey, R. Hoppe, D. Keyes, Yu. A. Kuznetsov and J. Périaux, eds., *Proc. Thirteenth Int. Conf. on Domain Decomposition Methods in Science and Engineering* (Lyon, 2000), CINME, Barcelona, 2002.
8. C. Farhat and F.-X. Roux, *Implicit Parallel Processing in Structural Mechanics*, Computational Mechanics Advances **2**, 1994, pp. 1–124.
9. R. Glowinski, G. H. Golub, G. A. Meurant and J. Périaux, eds., *Proc. First Int. Symp. on Domain Decomposition Methods for Partial Differential Equations* (Paris, 1987), SIAM, Philadelphia, 1988.
10. R. Glowinski, Yu. A. Kuznetsov, G. A. Meurant, J. Périaux and O. B. Widlund, eds., *Proc. Fourth Int. Symp. on Domain Decomposition Methods for Partial Differential Equations* (Moscow, 1990), SIAM, Philadelphia, 1991.
11. R. Glowinski, J. Périaux, Z.-C. Shi and O. B. Widlund, eds., *Eighth International Conference of Domain Decomposition Methods* (Beijing, 1995), Wiley, Strasbourg, 1997.
12. W. Hackbusch, *Iterative Methods for Large Sparse Linear Systems*, Springer, Heidelberg, 1993.

13. I. Herrera, D. Keyes, O. Widlund and R. Yates, eds. *Proc. Fourteenth Int. Conf. on Domain Decomposition Methods in Science and Engineering* (Cocoyoc, Mexico, 2003), National Autonomous University of Mexico (UNAM), Mexico City, 2003.
14. D. E. Keyes, T. F. Chan, G. A. 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.
15. D. E. Keyes, Y. Saad and D. G. Truhlar, eds., *Domain-based Parallelism and Problem Decomposition Methods in Science and Engineering*, SIAM, Philadelphia, 1995.
16. D. E. Keyes and J. Xu, eds. *Proc. Seventh Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (University Park, 1993), AMS, Providence, 1995.
17. R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Widlund and J. Xu, eds., *Proc. Fifteenth Int. Conf. on Domain Decomposition Methods* (Berlin, 2003), Springer, Heidelberg, 2004.
18. C.-H. Lai, P. Bjørstad, M. Cross and O. Widlund, eds., *Proc. Eleventh Int. Conf. on Domain Decomposition Methods* (Greenwich, 1999), DDM.org, Bergen, 2000.
19. P. Le Tallec, *Domain Decomposition Methods in Computational Mechanics*, Computational Mechanics Advances **2**, 1994, pp. 121–220.
20. J. Mandel, C. Farhat, and X.-C. Cai, eds, *Proc. Tenth Int. Conf. on Domain Decomposition Methods in Science and Engineering* (Boulder, 1998), AMS, Providence, 1999.
21. L. Pavarino and A. Toselli, *Recent Developments in Domain Decomposition Methods*, Volume 23 of *Lecture Notes in Computational Science & Engineering*, Springer, Heidelberg, 2002.
22. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford, 1999.
23. A. Quarteroni, J. Périaux, Yu. A. Kuznetsov and O. B. Widlund, eds., *Proc. Sixth Int. Conf. on Domain Decomposition Methods in Science and Engineering* (Como, 1992), AMS, Providence, 1994.
24. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1996.
25. B. F. Smith, P. E. Bjørstad and W. D. Gropp, *Domain Decomposition: Parallel Multilevel Algorithms for Elliptic Partial Differential Equations*, Cambridge Univ. Press, Cambridge, 1996.
26. A. Toselli and O. Widlund, *Domain Decomposition Methods: Algorithms and Theory*, Springer, New York, 2004.
27. B. I. Wolhmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, Volume 17 of *Lecture Notes in Computational Science & Engineering*, Springer, Heidelberg, 2001.
28. J. Xu, *Iterative Methods by Space Decomposition and Subspace Correction*, SIAM Review **34**, 1991, pp. 581-613.

5 Note Concerning Abstracts and Presentations

Within each section of plenary, minisymposium, and contributed papers, the edited proceedings appear in alphabetical order by first-listed author.

6 Acknowledgments

The Organizers are exceedingly grateful to Kara A. Olson for her editorial work in finalizing all of the chapters of this manuscript and for conversion to Springer multi-author latex style.

New York,
June 2006

Olof B. Widlund
David E. Keyes

Contents

Part I Plenary Presentations

Part II Minisymposia

Part III Contributed Presentations

Part I

Plenary Presentations

A Domain Decomposition Solver for a Parallel Adaptive Meshing Paradigm

Randolph E. Bank *

Department of Mathematics, University of California at San Diego, La Jolla,
California 92093-0112, USA. rbank@ucsd.edu

Summary. We describe a domain decomposition algorithm for use in the parallel adaptive meshing paradigm of Bank and Holst. Our algorithm has low communication, makes extensive use of existing sequential solvers, and exploits in several important ways data generated as part of the adaptive meshing paradigm. Numerical examples illustrate the effectiveness of the procedure.

1 Bank-Holst Algorithm

In [4, 3], we introduced a general approach to parallel adaptive meshing for systems of elliptic partial differential equations. This approach was motivated by the desire to keep communications costs low, and to allow sequential adaptive software (such as the software package PLTMG used in this work) to be employed without extensive recoding. Our discussion is framed in terms of continuous piecewise linear triangular finite element approximations used in PLTMG, although most ideas generalize to other approximation schemes.

Our original paradigm, called *Plan A* in this work, has three main components:

> **Step I: Load Balancing.** We solve a small problem on a coarse mesh, and use a posteriori error estimates to partition the mesh. Each subregion has approximately the same error, although subregions may vary considerably in terms of numbers of elements or gridpoints.
> **Step II: Adaptive Meshing.** Each processor is provided the complete coarse mesh and instructed to sequentially solve the *entire* problem, with the stipulation that its adaptive refinement should be limited largely to

*The work of this author was supported by the National Science Foundation under contract DMS-0208449. The UCSD Scicomp Beowulf cluster was built using funds provided by the National Science Foundation through SCREMS Grant 0112413, with matching funds from the University of California at San Diego.

its own partition. The target number of elements and grid points for each problem is the same. At the end of this step, the mesh is regularized such that the global mesh described in Step III is conforming.

Step III: Global Solve. The final global mesh consists of the union of the refined partitions provided by each processor. A final solution is computed using domain decomposition.

With this paradigm, the load balancing problem is reduced to the numerical solution of a small elliptic problem on a single processor, using a sequential adaptive solver such as PLTMG without requiring any modifications to the sequential solver. The bulk of the calculation in the adaptive meshing step also takes place independently on each processor and can also be performed with a sequential solver with no modifications necessary for communication. The only parts of the calculation requiring communication are (1) the initial fan-out of the mesh distribution to the processors at the beginning of the adaptive meshing step, once the decomposition is determined by the error estimator in load balancing; (2) the mesh regularization, requiring communication to produce a global conforming mesh in preparation for the final global solve in Step III; and (3) the final solution phase, that requires communicating certain information about the interface system (see Section 2).

In [2], we considered a variant of the above approach in which the load balancing occurs on a much finer mesh. The motivation was to address some possible problems arising from the use of a coarse grid in computing the load balance. In particular, we assume in Plan A that $N_c \gg p$ where N_c is the size of the coarse mesh and p is the number of processors. This is necessary to allow the load balance to do an adequate job of partitioning the domain into regions with approximately equal error. We also assume that N_c is sufficiently large and the mesh sufficiently well adapted for the a posteriori error estimates to accurately reflect the true behavior of the error. For the second step of the paradigm, we assume that $N_p \gg N_c$ where N_p is the target size for the adaptive mesh produced in Step II of the paradigm. Taking $N_p \gg N_c$ is important to marginalize the cost of redundant computations.

If any of these assumptions is weakened or violated, there might be a corresponding decline the effectiveness of the paradigm. In this case, we consider the possibility of modifying Steps I and II of the paradigm as follows. This variant is called *Plan B* in this work.

Step I: Load Balancing. On a single processor we adaptively create a *fine* mesh of size N_p, and use a posteriori error estimates to partition the mesh such that each subregion has approximately equal error, similar to Step I of the original paradigm.

Step II: Adaptive Meshing. Each processor is provided the complete adaptive mesh and instructed to sequentially solve the *entire* problem. However, in this case each processor should adaptively *coarsen* regions corresponding to other processors, and adaptively refine its own subregion. The size of the problem on each processor remains N_p, but this adaptive rezoning strategy concentrates the degrees of freedom in the processor's subregion. At the end of this step, the mesh is regularized such that the global mesh is conforming.

Step III: Global Solve. This step is the same as Plan A.

With Plan B, the initial mesh can be of any size. Indeed, our choice of N_p is mainly for convenience and to simplify notation; any combination of coarsening and refinement could be allowed in Step II. Allowing the mesh in Step I to be finer increases the cost of both the solution and the load balance in Step I, but it allows flexibility in overcoming potential deficiencies of a very coarse mesh in Plan A.

2 A Domain Decomposition Algorithm

In developing a domain decomposition solver appropriate for Step III, we follow a similar design philosophy. In particular, our DD solver has low communications costs, and recycles the sequential solvers employed in the Steps I and II. Furthermore, we use the existing partially refined global meshes distributed among the processors as the basis of local subdomain solves. This results in an overlapping DD algorithm in which the overlap is global, and provides a natural built-in coarse grid space on each processor. Thus no special coarse grid solve is necessary. Finally, a very good initial guess is provided by taking the fine grid parts of the solution on each processor.

The DD algorithm is described in detail in [6, 9]; some convergence analysis for a related algorithm in the symmetric, positive definite case can be found in [5]. To simplify the discussion, we initially consider the case of only two processors. We imagine the fine grid solutions for each of the two regions glued together using Lagrange multipliers to impose continuity along the interface. This leads to a block 5 × 5 system

$$\begin{pmatrix} A_{11} & A_{1\gamma} & 0 & 0 & 0 \\ A_{\gamma 1} & A_{\gamma\gamma} & 0 & 0 & I \\ 0 & 0 & A_{\nu\nu} & A_{\nu 2} & -I \\ 0 & 0 & A_{2\nu} & A_{22} & 0 \\ 0 & I & -I & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta U_1 \\ \delta U_\gamma \\ \delta U_\nu \\ \delta U_2 \\ \Lambda \end{pmatrix} = \begin{pmatrix} R_1 \\ R_\gamma \\ R_\nu \\ R_2 \\ U_\nu - U_\gamma \end{pmatrix}. \tag{1}$$

Here U_1 and U_2 are the solutions for the interior of regions 1 and 2, while U_γ and U_ν are the solutions on the interface. R_* are the corresponding residuals. The blocks A_{11}, A_{22} correspond to interior mesh points for regions 1 and 2, while $A_{\gamma\gamma}$, $A_{\nu\nu}$ correspond to the interface. Λ is Lagrange multiplier; the identity matrix I appears because global mesh is conforming.

In a similar fashion, we can imaging the fine grid on processor 1 glued to the coarse grid on processor 1 using a similar strategy. This results in a similar block 5 × 5 system

$$\begin{pmatrix} A_{11} & A_{1\gamma} & 0 & 0 & 0 \\ A_{\gamma 1} & A_{\gamma\gamma} & 0 & 0 & I \\ 0 & 0 & \bar{A}_{\nu\nu} & \bar{A}_{\nu 2} & -I \\ 0 & 0 & \bar{A}_{2\nu} & \bar{A}_{22} & 0 \\ 0 & I & -I & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta U_1 \\ \delta U_\gamma \\ \delta \bar{U}_\nu \\ \delta \bar{U}_2 \\ \Lambda \end{pmatrix} = \begin{pmatrix} R_1 \\ R_\gamma \\ R_\nu \\ 0 \\ U_\nu - U_\gamma \end{pmatrix} \qquad (2)$$

where the barred quantities (e.g. $\bar{A}_{22}$) refer to the coarse mesh. The right hand side of (2) is a subset of (1), except that we have set $\bar{R}_2 \equiv 0$. If local solves in Step II of the procedure were done exactly, then the initial guess would produce zero residuals for all interior points in the global system (1). We thus assume $R_1 \approx 0$, $R_2 \approx 0$ at all steps. This approximation substantially cuts communication and calculation costs.

Next, on processor 1 we reorder the linear system (2) as

$$\begin{pmatrix} 0 & -I & 0 & I & 0 \\ -I & \bar{A}_{\nu\nu} & 0 & 0 & \bar{A}_{\nu 2} \\ 0 & 0 & A_{11} & A_{1\gamma} & 0 \\ I & 0 & A_{\gamma 1} & A_{\gamma\gamma} & 0 \\ 0 & \bar{A}_{2\nu} & 0 & 0 & \bar{A}_{22} \end{pmatrix} \begin{pmatrix} \Lambda \\ \delta \bar{U}_\nu \\ \delta U_1 \\ \delta U_\gamma \\ \delta \bar{U}_2 \end{pmatrix} = \begin{pmatrix} U_\nu - U_\gamma \\ R_\nu \\ R_1 \\ R_\gamma \\ 0 \end{pmatrix}$$

and formally eliminate the upper 2 × 2 block. The resulting local Schur complement system is given by

$$\begin{pmatrix} A_{11} & A_{1\gamma} & 0 \\ A_{\gamma 1} & A_{\gamma\gamma} + \bar{A}_{\nu\nu} & \bar{A}_{\gamma 2} \\ 0 & \bar{A}_{2\nu} & \bar{A}_{22} \end{pmatrix} \begin{pmatrix} \delta U_1 \\ \delta U_\gamma \\ \delta \bar{U}_2 \end{pmatrix} = \begin{pmatrix} R_1 \\ R_\gamma + R_\nu + \bar{A}_{\nu\nu}(U_\nu - U_\gamma) \\ 0 + \bar{A}_{2\nu}(U_\nu - U_\gamma) \end{pmatrix}. \qquad (3)$$

The system matrix in (3) is just the stiffness matrix for the conforming mesh on processor 1. To solve this system, processor 1 must receive R_ν, and U_ν from processor 2 (and in turn send R_γ, and U_γ to processor 2). With this information, the right hand side can be computed and the system solved sequentially with no further communication. We use δU_1 and δU_γ to update U_1 and U_γ; we discard $\delta \bar{U}_2$. The update could be local ($U_1 \leftarrow U_1 + \delta U_1$, $U_\gamma \leftarrow U_\gamma + \delta U_\gamma$) or could require communication. In PLTMG, the update procedure is a Newton line search. Here is a summary of the calculation on processor 1.

1. locally compute R_1 and R_γ.
2. exchange boundary data (send R_γ and U_γ; receive R_ν and U_ν).
3. locally compute the right-hand-side of the Schur complement system (3).
4. locally solve the linear system (3) via the multigraph iteration.
5. update U_1 and U_γ using δU_1 and δU_γ.

We now consider the case of the global saddle point system in the general case of p processors. Now the global system has the form

$$\begin{pmatrix} A_{ss} & A_{sm} & A_{si} & I \\ A_{ms} & A_{mm} & A_{mi} & -Z^t \\ A_{is} & A_{im} & A_{ii} & 0 \\ I & -Z & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta U_s \\ \delta U_m \\ \delta U_i \\ \Lambda \end{pmatrix} = \begin{pmatrix} R_s \\ R_m \\ R_i \\ ZU_m - U_s \end{pmatrix}. \tag{4}$$

Here U_i are the interior unknowns for all subregions, and A_{ii} is a block diagonal matrix corresponding to the interiors of all subregions; as before we expect $R_i \approx 0$. For the interface system, we (arbitrarily) designate one unknown at each interface point as the *master* unknown, and all others as *slave* unknowns; there will be more than one slave unknown at cross points (where more than 2 subregions share a single interface point). As before we impose continuity at interface points using Lagrange multipliers; $Z \neq I$ in general due to cross points. If we reorder (4) and eliminate the Lagrange multipliers and slave unknowns, the resulting Schur complement system is

$$\begin{pmatrix} A_{mm} + A_{ms}Z + Z^tA_{sm} + Z^tA_{ss}Z & A_{mi} + Z^tA_{si} \\ A_{im} + A_{is}Z & A_{ii} \end{pmatrix} \begin{pmatrix} \delta U_m \\ \delta U_i \end{pmatrix} = \begin{pmatrix} R_m + Z^tR_s - (A_{ms} + Z^tA_{ss})(ZU_m - U_s) \\ R_i - A_{is}(ZU_m - U_s) \end{pmatrix}. \tag{5}$$

The system matrix is just the stiffness matrix for the global conforming finite element space. The right hand side is the conforming global residual augmented by some “jump” terms arising from the Lagrange multipliers.

The situation on processor k is analogous; we imagine gluing the fine subregion on processor k to the $p-1$ coarse subregions on processor k. The resulting saddle point problem has the form

$$\begin{pmatrix} \bar{A}_{ss} & \bar{A}_{sm} & \bar{A}_{si} & I \\ \bar{A}_{ms} & \bar{A}_{mm} & \bar{A}_{mi} & -\bar{Z}^t \\ \bar{A}_{is} & \bar{A}_{im} & \bar{A}_{ii} & 0 \\ I & -\bar{Z} & 0 & 0 \end{pmatrix} \begin{pmatrix} \delta \bar{U}_s \\ \delta \bar{U}_m \\ \delta \bar{U}_i \\ \Lambda \end{pmatrix} = \begin{pmatrix} \bar{R}_s \\ \bar{R}_m \\ \bar{R}_i \\ \bar{Z}\bar{U}_m - \bar{U}_s \end{pmatrix}. \tag{6}$$

The matrix $\bar{A}_{ii}$ and the vector $\bar{U}_i$ are fine for region k and coarse for the $p-1$ other regions. The residual $\bar{R}_i$ corresponds to R_i on region k, and is zero for the coarse subregions. Master interface variables are chosen from region k if possible; this part of the local interface system on processor k corresponds exactly to the global interface system. For other parts of the local interface system, master unknowns can be chosen arbitrarily; in PLTMG, they are actually defined using arithmetic averages, but that detail complicates the notation and explanation here. The vectors $\bar{R}_m$ and $\bar{R}_s$ are subsets of R_m and R_s, respectively.

A local Schur complement system on processor k is computed analogously to (5). This system has the form

$$\begin{pmatrix} \bar{A}_{mm} + \bar{A}_{ms}\bar{Z} + \bar{Z}^t\bar{A}_{sm} + \bar{Z}^t\bar{A}_{ss}\bar{Z} & \bar{A}_{mi} + \bar{Z}^t\bar{A}_{si} \\ \bar{A}_{im} + \bar{A}_{is}\bar{Z} & \bar{A}_{ii} \end{pmatrix} \begin{pmatrix} \delta\bar{U}_m \\ \delta\bar{U}_i \end{pmatrix} = \begin{pmatrix} \bar{R}_m + \bar{Z}^t\bar{R}_s - (\bar{A}_{ms} + \bar{Z}^t\bar{A}_{ss})(\bar{Z}\bar{U}_m - \bar{U}_s) \\ \bar{R}_i - \bar{A}_{is}(\bar{Z}\bar{U}_m - \bar{U}_s) \end{pmatrix}. \quad (7)$$

As in the 2 processor case, the system matrix is just the conforming finite element stiffness matrix for the partially refined global mesh on processor k. To compute the right hand side of (7), processor k requires interface solution values and residuals for the global interface system. Once this is known, the remainder of the solution can be carried out with no further communication. To summarize, on processor k, one step of the DD algorithm consists of the following.

1. locally compute $\bar{R}_i$ and parts of R_s and R_m from subregion k.
2. exchange boundary data, obtaining the complete fine mesh interface vectors R_m, R_s, U_m and U_s.
3. locally compute the right-hand-side of (7) (using averages).
4. locally solve the linear system (7) via the multigraph iteration.
5. update the fine grid solution for subregion k using subsets of $\delta\bar{U}_i$, $\delta\bar{U}_m$.

3 Numerical Experiments

We now present several numerical illustrations; the details of the example problems are summarized below.

Example 1: Our first example is the Poisson equation

$$\begin{aligned} -\Delta u &= 1 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \end{aligned} \qquad (8)$$

where Ω is the domain shown in Figure 1.

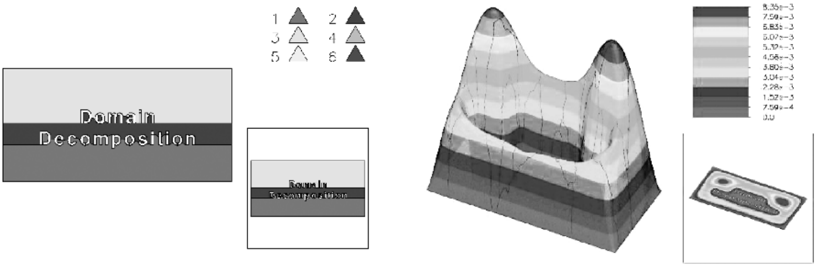


Fig. 1. The domain (left) and solution (right) for the Poisson equation (8).

Example 2: Our second example is the convection-diffusion equation

$$\begin{aligned} -\Delta u + \beta u_y &= 1 \quad \text{in } \Omega, \\ u &= 0 \quad \text{on } \partial\Omega, \\ \beta &= 10^5, \end{aligned} \tag{9}$$

where Ω is the domain shown in Figure 2.

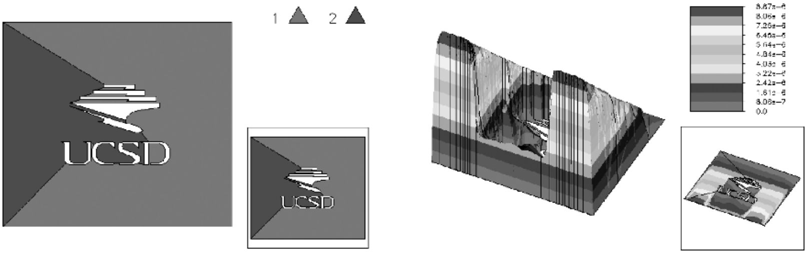


Fig. 2. The domain (left) and solution (right) for the convection-diffusion equation (9).

Example 3: Our third example is the anisotropic equation

$$\begin{aligned} -a_1 u_{xx} - a_2 u_{yy} - f = 0 \qquad & \text{in } \Omega, \\ (a_1 u_x, a_2 u_y) \cdot \mathbf{n} = c - \alpha u \qquad & \text{on } \partial\Omega, \end{aligned} \tag{10}$$

where Ω is the domain shown in Figure 3. Values of the coefficient functions are given in Table 1.

Table 1. Coefficient values for equation (10). Region numbers refer to Figure 3.

Region	a_1	a_2	f	side	c	α
1	25	25	0	left	0	0
2	7	0.8	1	top	1	3
3	5.0	10^{-4}	1	right	2	2
4	0.2	0.2	0	bottom	3	1
5	0.05	0.05	0			

Example 4: Our fourth example is the optimal control problem

$$\begin{gathered} \min \int_\Omega (u - u_0)^2 + \gamma\lambda^2 \, dx \;\text{ such that} \\ -\Delta u = \lambda \quad \text{in } \Omega \equiv (0,1) \times (0,1), \\ u = 0 \quad \text{on } \partial\Omega, \\ 1 \le \lambda \le 10, \qquad \gamma = 10^{-4}, \\ u_0 = \sin(3\pi x)\sin(3\pi y). \end{gathered} \tag{11}$$

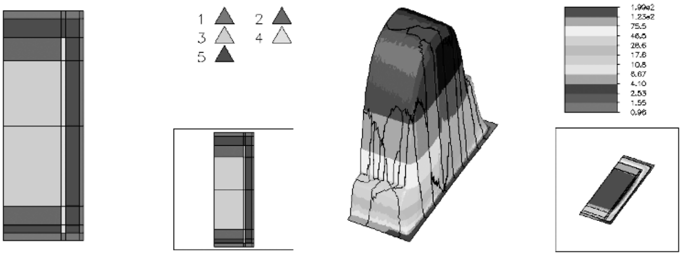


Fig. 3. The domain (left) and solution (right) for the anisotropic equation (10).

This problem is solved by an interior point method described in [7, 1]. Three finite element functions are computed; the state variable u, the Lagrange multiplier v, and the optimal control λ.

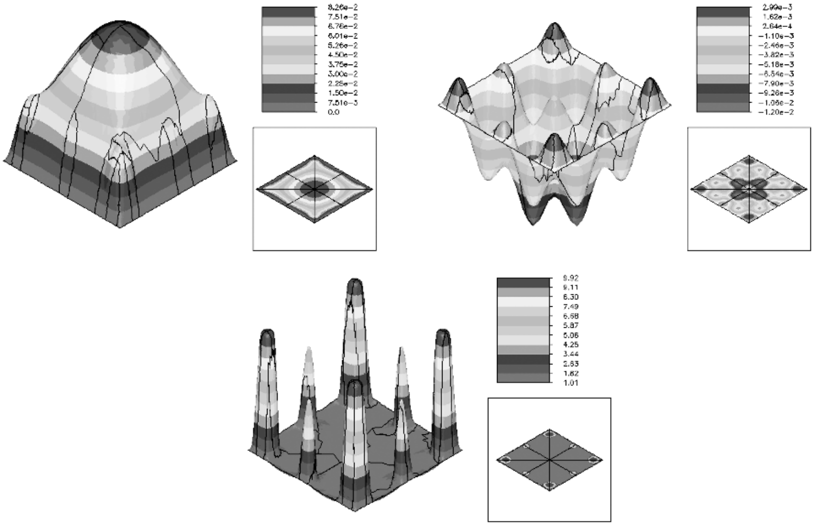


Fig. 4. The state variable (left, top), Lagrange multiplier (right, top) and optimal control (bottom) for equation (11).

Our Linux cluster consists of 20 dual 1800 Athlon-CPU nodes with 2GB of memory each, with a dual Athlon 1800 file server, also with 2GB of memory. Communication is provided via a 100Mbit CISCO 2950G Ethernet switch. The cluster runs the NPACI Rocks version of Linux, using Mpich.

In the case of the original paradigm, Plan A, in Step I for each problem we created an adaptive mesh with $N \approx 10000$ vertices. This mesh was then partitioned for $p = 8, 16, 32, 64, 128$ processors, and the coarse problem was broadcast to all processors[2]. In Step II of the paradigm, we adaptively created a mesh with $N \approx 100000$ vertices. In particular, we first adaptively refined to $N \approx 40000$, solved that problem, adaptively refined to $N \approx 100000$, and then regularized the mesh. In Step III, PLTMG first solved the local problem with $N \approx 100000$, in order to insure that interior residuals were small and validate the assumption that coarse interior residuals could be set to zero in the DD solver. This local solve was followed by several iterations of the DD solver.

In the case of the variant paradigm, Plan B, in Step I we created an adaptive mesh with $N \approx 100000$. As in Plan A, this mesh was then partitioned for $p = 8, 16, 32, 64, 128$ processors, and broadcast to all processors. In Step II, through a process of adaptive unrefinement/refinement, each processor transferred approximately 50000 vertices from outside its subregion to inside, so that the total number of vertices remained $N \approx 100000$. This mesh was them made conforming as in Step II of Plan A. In Step III, the local problem was solved, followed by several iterations of the DD solver.

For both Plan A and Plan B, the convergence criteria for the DD iteration was

$$\frac{\|\delta\mathcal{U}^k\|_G}{\|\mathcal{U}^k\|_G} \le \max\left(\frac{\|\delta\mathcal{U}^0\|_G}{\|\mathcal{U}^0\|_G}, \frac{\|\nabla e_h\|_{\mathcal{L}^2}}{\|\nabla u_h\|_{\mathcal{L}^2}}\right) \times 10^{-1}$$

$$\frac{\|\mathcal{R}^k\|_{G^{-1}}}{\|\mathcal{R}^0\|_{G^{-1}}} \le 10^{-2}$$

Here G is the diagonal of the finite element mass matrix, introduced to account for nonuniformity of the global finite element mesh. u_h and e_h are the finite element solution and a posteriori error estimate, respectively, introduced to include the approximation error in the convergence criteria. The norms in various terms are different, but we have not observed any difficulties arising as a result. For the multigraph iteration on each processor, the convergence criteria was

$$\frac{\|\overline{\mathcal{R}}^j\|_{\ell^2}}{\|\overline{\mathcal{R}}^0\|_{\ell^2}} \le 10^{-3}.$$

The stronger criteria was to insure that the approximation on coarse interior residuals by zero remained valid.

In Tables 2-5 we summarize the results of our computations. In these tables, p is the number of processors, N is the number of vertices on the final global mesh, and DD is the number of domain decomposition iterations used in Step III. Execution times, in seconds, at the end of Steps I, II, and III are also reported. Step I is done on a single processor. For Steps II and III, average times across all processors are reported; the range of times is also included in parentheses.

[2] Since our cluster had only 20 nodes, the results are simulated using Mpich for the larger values of p.

p	N	DD	Breakpoints		
			Step I	Step II	Step III
Poisson Equation: Plan A					
8	657464	2	2.8	20.1 (17.2-21.3)	61.6 (58.2-65.4)
16	1240805	2	3.0	19.6 (16.4-21.3)	62.4 (56.7-67.0)
32	2329953	2	3.2	20.4 (17.4-22.5)	67.8 (58.1-72.7)
64	4361844	2	3.3	20.0 (15.5-21.8)	68.5 (56.1-76.9)
128	8057638	3	3.5	20.0 (15.2-22.2)	77.0 (59.8-88.7)
Poisson Equation: Plan B					
8	478398	1	75.0	92.7 (90.2-95.2)	123.5 (121.2-126.2)
16	827827	1	82.4	98.8 (97.4-103.0)	130.9 (126.2-136.0)
32	1472509	1	87.2	106.1 (103.0-109.2)	139.9 (133.0-145.0)
64	2626624	1	90.1	109.1 (105.0-112.1)	143.8 (136.6-150.9)
128	4641395	1	97.9	116.7 (113.4-119.6)	151.6 (143.7-157.9)

Table 2. Numerical results for problem (8).

p	N	DD	Breakpoints		
			Step I	Step II	Step III
Convection Diffusion Equation: Plan A					
8	698859	2	3.3	18.7 (17.2-19.7)	47.4 (45.7-49.7)
16	1345203	2	3.5	18.9 (16.6-21.1)	47.5 (45.4-50.1)
32	2543913	2	3.8	18.9 (16.6-21.0)	49.3 (43.5-54.8)
64	4665741	2	4.0	19.0 (16.6-21.3)	51.1 (46.0-63.1)
128	8547289	2	4.2	19.3 (16.0-22.4)	53.1 (44.6-65.0)
Convection Diffusion Equation: Plan B					
8	493182	2	66.7	79.5 (77.2-82.4)	105.1 (100.5-111.3)
16	872605	2	76.4	89.0 (86.6-91.7)	115.8 (111.6-119.1)
32	1598504	1	81.4	94.3 (91.6-97.6)	118.8 (113.7-123.7)
64	2941956	1	84.8	97.6 (95.3-100.6)	122.6 (118.9-126.5)
128	5305634	2	83.8	96.6 (94.3-100.3)	128.0 (122.0-132.6)

Table 3. Numerical results for problem (9).

- The times for Step I are much larger for Plan B than Plan A due to the larger size of the problem. The increase in time with increasing p is due mostly to eigenvalue problems that are solved are part of the spectral bisection load balancing scheme.
- The distribution of times in Steps II and III is due mainly to differences in the local sequential algorithms, for example using one instead of two multigraph V-cycles in a local solve.
- The DD algorithm in [5] is shown to converge independently of N, which was empirically verified in [6] for the version implemented here. There is some slight, empirically logarithmic, dependence on p.

p	N	DD	Breakpoints		
			Step I	Step II	Step III
			Anisotropic Equation: Plan A		
8	646293	1	3.0	20.7 (19.3-22.0)	49.7 (46.0-54.5)
16	1169837	1	3.3	20.7 (19.0-22.6)	49.8 (44.9-54.8)
32	2038184	2	3.6	21.6 (18.7-23.2)	59.9 (48.0-68.5)
64	3500678	2	3.8	21.9 (19.4-24.6)	61.4 (51.6-71.8)
128	5729057	2	4.0	22.1 (19.5-24.9)	62.2 (53.8-75.6)
			Anisotropic Equation: Plan B		
8	484972	1	56.5	73.7 (71.4-77.0)	100.7 (97.1-106.7)
16	832360	1	62.5	79.2 (76.6-82.2)	105.8 (101.4-111.1)
32	1460881	1	68.0	86.3 (83.0-89.3)	115.1 (109.9-119.8)
64	2512231	1	74.8	92.9 (89.7-95.4)	122.7 (116.4-130.5)
128	4102960	1	81.4	99.3 (96.2-103.1)	129.8 (123.7-140.4)

Table 4. Numerical results for problem (10).

p	N	DD	Breakpoints		
			Step I	Step II	Step III
			Optimal Control Problem: Plan A		
8	677913	1	6.0	31.8 (29.8-35.6)	109.9 (101.2-114.3)
16	1297918	1	6.3	32.1 (29.0-37.7)	119.3 (106.1-131.3)
32	2421235	1	6.5	33.3 (29.0-38.3)	131.3 (110.4-141.9)
64	4514511	1	6.6	33.8 (30.4-38.3)	137.3 (105.9-157.3)
128	8324507	2	7.0	34.2 (29.9-42.2)	173.6 (135.5-212.5)
			Optimal Control Problem: Plan B		
8	481309	1	139.0	156.7 (155.0-159.0)	215.6 (205.5-229.5)
16	830641	1	143.0	160.2 (157.0-163.1)	221.0 (210.6-236.7)
32	1482322	1	150.8	168.2 (164.0-171.1)	235.9 (220.1-248.4)
64	2600084	2	154.9	172.5 (169.0-175.6)	256.7 (242.0-275.6)
128	4507853	2	143.0	160.7 (156.5-166.0)	249.4 (230.3-268.3)

Table 5. Numerical results for problem (11).

- For the convection-diffusion problem a multigraph preconditioned Bi-CG algorithm was used, while for the Poisson equation and the anisotropic equation regular preconditioned CG was used. Details of the multigraph solver are given in [8].
- For the optimal control problem, the block linear systems were of order $3N$, and each iteration required the solution of four linear systems with the $N \times N$ finite element stiffness matrix, and one system with an $N \times N$ matrix similar to the finite element mass matrix. See [1] for details.

In viewing the results as a whole, both paradigms scale reasonably well as a function of p; since Step III is a very costly part of the calculation, it is clearly worthwhile to try to make the convergence rate independent of p as well as N, or at least to reduce the dependence on p. This is a topic of current research interest.

References

1. R. E. Bank, *PLTMG: A Software Package for Solving Elliptic Partial Differential Equations. Users' Guide 7.0*, SIAM, Philadelphia, PA, 1990.
2. ———, *Some variants of the Bank-Holst parallel adaptive meshing paradigm*, Comput. Vis. Sci., (2006). Accepted.
3. R. E. Bank and M. Holst, *A new paradigm for parallel adaptive meshing algorithms*, SIAM Review, 45 (2003), pp. 291–323.
4. R. E. Bank and M. J. Holst, *A new paradigm for parallel adaptive mesh refinement*, SIAM J. Sci. Comput., 22 (2000), pp. 1411–1443.
5. R. E. Bank, P. K. Jimack, S. A. Nadeem, and S. V. Nepomnyaschikh, *A weakly overlapping domain decomposition preconditioner for the finite element solution of elliptic partial differential equations*, SIAM J. Sci. Comput., 23 (2002), pp. 1817–1841.
6. R. E. Bank and S. Lu, *A domain decomposition solver for a parallel adaptive meshing paradigm*, SIAM J. Sci. Comput., 26 (2004), pp. 105–127.
7. R. E. Bank and R. F. Marcia, *Interior Methods for a Class of Elliptic Variational Inequalities*, vol. 30 of Lecture Notes in Computational Science and Engineering, Springer, 2003, pp. 218–235.
8. R. E. Bank and R. K. Smith, *An algebraic multilevel multigraph algorithm*, SIAM J. Sci. Comput., 23 (2002), pp. 1572–1592.
9. S. Lu, *Parallel Adaptive Multigrid Algorithms*, PhD thesis, University of California at San Diego, Department of Mathematics, 2004.

Algebraic Multigrid Methods Based on Compatible Relaxation and Energy Minimization

James Brannick[1] and Ludmil Zikatanov[2]

[1] Department of Applied Mathematics, University of Colorado, Boulder, CO 80309, USA. James.Brannick@colorado.edu

[2] Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA. ludmil@psu.edu

Summary. This paper presents an adaptive algebraic multigrid setup algorithm for positive definite linear systems arising from discretizations of elliptic partial differential equations. The proposed method uses *compatible relaxation* to select the set of coarse variables. The nonzero supports for the coarse-space basis are determined by approximation of the so-called two-level "ideal" interpolation operator. Then, an energy minimizing coarse basis is formed using an approach aimed to minimize the trace of the coarse–level operator. The variational multigrid solver resulting from the presented setup procedure is shown to be effective, without the need for parameter tuning, for some problems where current algorithms exhibit degraded performance.

Key words: algebraic multigrid, compatible relaxation, trace minimization

1 Introduction

In this paper, we consider solving linear systems of equations,

$$A\mathbf{u} = \mathbf{f}, \tag{1}$$

via algebraic multigrid (AMG), where $A \in \Re^{n\times n}$ is assumed to be symmetric positive definite (SPD). Our AMG approach for solving (1) involves a stationary linear iterative smoother and a coarse-level correction. The corresponding two-grid method gives rise to an error propagation operator having the following form,

$$E_{TG} = (I - P(P^tAP)^{-1}P^tA)(I - M^{-1}A), \tag{2}$$

where $P : \Re^{n_c} \mapsto \Re^n$ is the interpolation operator and M is the approximate inverse of A that defines the smoother. It is well known that if A is symmetric, then this variational form of the correction step is optimal in the energy norm. As usual, a multilevel algorithm is obtained by recursion, that is, by solving the coarse-level residual problem, involving $A_c = P^t AP$, again by using a two-grid method. The efficiency of such an approach depends on proper interplay between the smoother and the coarse-level correction. In AMG, the smoother is typically fixed and the coarse-level correction is formed to compensate for its deficiencies. The primary task is, of course, the selection of P. It is quite common to use only the information from the current level in order to compute P and, hence, the next coarser space, because such a procedure can be implemented efficiently and at a low computational cost. A general process for constructing P is described by the following generic two-level algorithm:

- Choose a set of n_c coarse degrees of freedom;
- Choose a sparsity pattern of interpolation $P \in \mathbb{R}^{n \times n_c}$;
- Define the weights of the interpolation (i.e., the entries of P), giving rise to the next level operator as $A_c = P^t AP \in \mathbb{R}^{n_c \times n_c}$.

Standard algebraic multigrid setup algorithms are based on properties of M-matrices (e.g., the assumption that algebraically-smooth error varies slowly in the direction of strong couplings – typically defined in terms of the relative size of the entries of the matrix) in their setup to construct P. Although these traditional approaches have been shown to be extremely effective for a wide range of problems [1, 14, 13, 15], the use of heuristics based on M-matrix properties still limits their range of applicability. In fact, the components and parameters associated with these approaches are often problem dependent. Developing more robust AMG solvers is currently a topic of intense research.

General approaches for selecting the set of coarse variables are presented in [12, 4]. These approaches use compatible relaxation (CR) to gauge the quality of (as well as construct) the coarse variable set, an idea first introduced by Brandt [2]. In [3], an energy-based strength-of-connection measure is developed and shown to extend the applicability of Classical AMG when coupled with adaptive AMG interpolation [7]. Recent successes in developing a more general form of interpolation include [7, 6, 17, 19]. These methods are designed to allow efficient attenuation of error in a subspace characterized locally by a given set of error components, regardless of whether they are smooth or oscillatory in nature. In [7, 6], these components are computed automatically in the setup procedure using a multilevel power method iteration based on the error propagation operator of the method itself.

The algorithm we propose for constructing P is motivated by the recently developed two-level theory introduced in [9] and [10]. We explore the use of this theory in developing a robust setup procedure in the setting of classical AMG. In particular, as in classical AMG, we assume that the coarse-level variables are a subset of the fine-level variables. Our coarsening algorithm constructs the coarse variable set using the CR-based algorithm introduced by Brannick and Falgout in [4]. The notion of strength of connection we use in determining the nonzero sparsity pattern of the columns of P is based on a sparse approximation of the so-called two-level ideal interpolation operator. Given the sparsity pattern of the columns of P, the values of the nonzero entries of the columns of P are computed using the trace minimization algorithm proposed by Wan, Chan, and Smith [18], based on the efficient implementation developed by Xu and Zikatanov [19].

2 Preliminaries and motivation

We begin by introducing notation. Since, in the presented algorithm, the coarse-level degrees of freedom are viewed as a subset of the fine-level degrees of freedom, prolongation P has the form $P = \begin{bmatrix} W \\ I \end{bmatrix}$, where I is the $n_c \times n_c$ identity and $W \in \mathbb{R}^{n_s \times n_c}$, $n_s = n - n_c$, contains the rest of the interpolation weights. In this way the coarse space $V_c \subset \mathbb{R}^n$ is defined as Range(P).

In what follows, we use several projections on the Range(P). These projections are defined for any SPD matrix X as follows:

$$\pi_X = P(P^tXP)^{-1}P^tX,$$

where, for $X = I$, we omit the subscript and write π instead of π_I. To relate the construction of interpolation to a compatible relaxation procedure, we introduce two operators: $R = [0, I]$ and S, where R has the dimensions of P^t and S has the dimensions of P. The fact that the coarse-level degrees of freedom are a subset of the fine-level degrees of freedom is reflected in the form of R. The matrix S corresponds to the complementary degrees of freedom, i.e. fine-level degrees of freedom, and can be chosen in many different ways, as long as $RS = 0$. In the approach presented here, we assume that $S = [I, 0]^t$. With R and S in hand, we define the 2×2 block splitting of any $X \in \mathbb{R}^{n\times n}$ by

$$X = \begin{bmatrix} X_{ff} & X_{fc} \\ X_{cf} & X_{cc} \end{bmatrix}, \tag{3}$$

where $X_{ff} = S^tXS$, $X_{fc} = S^tXR^t$, $X_{cf} = RXS$, and $X_{cc} = RXR^t$. We also need the Schur complement of X with respect to this splitting, defined as $\mathcal{S}(X) = X_{cc} - X_{cf}X_{ff}^{-1}X_{fc}$.

Given the smoother's M, the F-relaxation form of compatible relaxation (CR) we use in our algorithm yields an error propagation operator having the following form:

$$E_f = (I - M_{ff}^{-1}A_{ff}). \tag{4}$$

The associated symmetrized smoother is then defined as $\widetilde{M} := M^t(M^t + M - A)^{-1}M$, where $M^t + M - A$ is assumed to be SPD, a sufficient condition for convergence. To simplify the presentation here, we also assume that M is symmetric, in which case $2M - A$ being SPD is also necessary for the convergence of the smoothing iteration.

2.1 Some convergence results

The convergence result motivating our approach is a theorem proved in [10], giving the precise convergence factor of the two-grid algorithm.

Theorem 1. *Let E_{TG} be defined as in* (2). *Then*

$$\|E_{TG}\|_A^2 = 1 - \frac{1}{K(P)}, \qquad K(P) = \sup_{\mathbf{v}} \frac{\|(I - \pi_{\widetilde{M}})\mathbf{v}\|_{\widetilde{M}}^2}{\|\mathbf{v}\|_A^2}.$$

Assuming that the set of coarse degrees of freedom have been selected (i.e. R is defined), the remaining task is defining a P to minimize $K(P)$. Finding such a P is of course not at all straightforward, because the dependence of $K(P)$ on P given in Theorem 2 is complicated. To make this more practical we consider minimizing an upper bound of K, which is easily obtained by replacing $\pi_{\widetilde{M}}$ with π, the ℓ_2 projection on Range(P). We then obtain a measure for the quality of the coarse space defined as follows:

$$\mu(P) = \sup_{\mathbf{v}} \frac{\|(I - \pi)\mathbf{v}\|_{\widetilde{M}}^2}{\|\mathbf{v}\|_A^2}.$$

Note that $\mu(P) \geq K(P)$ for all P. Also, this measure suggests that error components consisting of eigenvectors associated with small eigenvalues (i.e., error not effectively treated by relaxation) must be well approximated by P. The following result from [9] gives $P_\star$ that minimizes $\mu(P)$.

Theorem 2. *Assume that R, S, and μ are defined as above. Then*

$$\mu(P_\star) = \min_P \mu(P), \quad \textit{where} \quad P_\star = \left[-A_{fc}^t A_{ff}^{-1} \; ; I\right]^t.$$

Moreover, the asymptotic convergence factor of CR provides an upper bound for the above minimum as follows (see Theorem 5.1 in [9]).

Theorem 3. *If the number of non-zeros per row in A is bounded, then there exists a constant c, such that*

$$\mu(P_\star) \leq \frac{c}{1 - \rho_f}, \qquad \rho_f = \|E_f\|_{A_{ff}}^2.$$

A conclusion that follows immediately from this theorem is that ρ_f provides a *computable* measure of the quality of the coarse space, that is, a measure of the ability of the set of coarse variables to represent error not eliminated by relaxation.

The main ideas of our algorithm, described next, are based on observations and conclusions drawn from the above results.

3 Compatible relaxation based coarsening

In this section, we give more details on the first step of the algorithm, selecting the coarse degrees of freedom. The quality of the set of coarse-level degrees of freedom, C, depends on two conflicting criteria:

C1: *algebraically-smooth error should be approximated well by some vector interpolated from C, and*

C2: *C should have substantially fewer variables than on the fine level.*

In our adaptive AMG solver, the set of coarse variables is selected using the CR-based coarsening approach developed in [4]. This coarsening scheme is based on the two-level multigrid theory outlined in § 2: for a given splitting of fine-level variables Ω into C and F, F denoting the fine-level only variables, if CR is fast to converge, then there exists a P such that the resulting two-level method is uniformly convergent. The algorithm ties the selection of C to the smoother. The set of coarse variables is constructed using a multistage coarsening algorithm, where a single stage consists of: (1) running several iterations of CR (based on the current set F) and (2) if CR is slow to converge, adding an independent set of fine-level variables (not effectively treated by CR) to C. Steps (1) and (2) are applied repeatedly until the convergence of CR is deemed sufficient, giving rise to a sequence of coarse variable sets:

$$\emptyset = C_0 \subseteq C_1 \subseteq ... \subseteq C_m,$$

where, for the accepted coarse set $C := C_m$, convergence of CR is below a prescribed tolerance. Hence, this algorithm constructs C so that **C1** is strictly enforced and **C2** is satisfied as much as possible. The details of this algorithm are given in [4].

An advantage of this approach, over the two-pass algorithm employed in classical AMG, is the use of the asymptotic convergence factor of compatible relaxation as a measure of the quality of C and, thus, the ability to adapt C when necessary. An additional advantage of this approach is that the algorithm does not rely on the notion of strength of connections to form C, instead, only the graph of matrix A and the error generated by the CR process are used to form C. This typically results in more aggressive coarsening than in traditional coarsening approaches, especially on coarser levels where *stencils* tend to grow. Additionally, this approach has been shown to work for a wide range of problems without the need for parameter tuning [4].

We conclude this section by proving the following proposition relating the spectral radii of E_f to the condition number of A_{ff}.

Proposition 1. *Consider compatible relaxation defined by E_f and let*

$$\rho(E_f) \leq a < 1. \tag{5}$$

Then

$$\kappa(A_{ff}) \leq \kappa(M_{ff}) \frac{1+a}{1-a}.$$

Proof. Let λ be any eigenvalue of $M_{ff}^{-1}A_{ff}$. Then $1-\lambda$ is an eigenvalue of $(I - M_{ff}^{-1}A_{ff})$. From (5) we have that

$$|1-|\lambda|| \leq |1-\lambda| \leq a, \qquad \text{implying} \qquad 1-a \leq |\lambda| \leq 1+a.$$

Thus $\kappa(M_{ff}^{-1}A_{ff}) \leq (1+a)/(1-a)$. From the assumption on the CR convergence factor, it follows that M_{ff} is positive definite. The smallest eigenvalue of A_{ff} is then estimated as follows:

$$\begin{aligned}\lambda_{\min}(A_{ff}) = \inf_{x\neq 0}\frac{(A_{ff}x,x)}{(x,x)} &\geq \frac{\lambda_{\min}(M_{ff}^{-1/2}A_{ff}M_{ff}^{-1/2})}{\lambda_{\max}(M_{ff}^{-1})}\\ &= \frac{\lambda_{\min}(M_{ff}^{-1}A_{ff})}{\lambda_{\max}(M_{ff}^{-1})} \geq (1-a)\lambda_{\min}(M_{ff}).\end{aligned}$$

Estimating the maximum eigenvalue of A_{ff} in a similar fashion leads to the inequality

$$\lambda_{\max}(A_{ff}) \leq (1+a)\lambda_{\max}(M_{ff}). \tag{6}$$

The proof is then completed by using the last two inequalities in an obvious way.

Hence, fast-to-converge CR and M_{ff} being well conditioned imply that A_{ff} is well conditioned. For many discrete PDE problems, M_{ff} is very well conditioned. This, together with the result from the next section, shows that fast convergence of CR indicates the existence of a sparse and local approximation to the inverse of A_{ff} and, hence, a good approximation to the two-level ideal interpolation operator. We note that, when M is ill conditioned, simple rescaling can often be used to reduce the problem to the well-conditioned case. For example, replacing A by $D^{-1/2}AD^{-1/2}$ and M by$D^{-1/2}MD^{-1/2}$, where D is the diagonal of A, may produce a well conditioned M_{ff} so that the above conclusions apply.

4 Inverse of sparse matrices and supports of coarse grid basis vectors

We describe now the parts of our algorithm that relate to the choice of the sparsity pattern of P. Set $\Omega = \{1,\ldots,n\}$ and assume that the coarse grid degrees of freedom are $C = \{n_s+1,\ldots,n\}$, where $n_s = n - n_c$. This leads to a 2×2 splitting of A, as given by (3). We aim to construct a covering of Ω with n_c sets $\{\Omega_i\}_{i=1}^{n_c}$, such that $\cup_{i=1}^{n_c}\Omega_i = \Omega$ contain information on the non-zero structure of the entries of P. We desribe our approach using some elementary tools from graph theory.

With matrix A_{ff}, we associate a graph, G, whose set of vertices is $\Omega\setminus C$, and set of edges is

$$\mathcal{E} = \{(i,j) \in \Omega \setminus C \quad \text{if and only if} \quad [A_{ff}]_{ij} \neq 0\}.$$

By graph distance between vertices i and j, denoted by $|i-j|_G$, we mean the length (i.e., the number of edges) of a shortest path connecting i and j in G. We assume without loss of generality that G is connected, so that the graph distance between any i and j is well defined. An important observation (see, for example, [11]) related to the sparsity of A is that $(A_{ff}^k e_i, e_j) = 0$ holds for all k, i, and j such that $1 \leq k < |i-j|_G$. This in turn shows that, for any polynomial $p(x)$ of degree less than $|i-j|_G$, we have that

$$[A_{ff}^{-1}]_{ij} = (A_{ff}^{-1} e_i, e_j) = ((A_{ff}^{-1} - p(A_{ff}))e_i, e_j).$$

Taking the infimum over all such polynomials and using a standard approximation theory result for approximating $1/x$ with polynomials on the interval $[\lambda_{\min}(A_{ff}), \lambda_{\max}(A_{ff})]$, we arrive at the following inequality:

$$[A_{ff}^{-1}]_{ij} \leq c\, q^{|i-j|_G - 1}, \tag{7}$$

where $q < 1$ depends on condition number, κ, of A_{ff} and can be taken to be $\dfrac{\kappa^{1/2}-1}{\kappa^{1/2}+1}$, and c is a constant. The estimate on the decay of $[A_{ff}^{-1}]_{ij}$ given in (7) was contributed by Vassilevski [16]. It is related to similar results for banded matrices due to Demko [8]. This reference was also brought to our attention by Vassilevski [16].

A simple and important observation from (7) is that a polynomial (or close to polynomial) approximation to the inverse A_{ff}^{-1} indicates exactly where the large entries of A_{ff}^{-1} are. Such an approximation can be constructed efficiently, since if A_{ff} is well-conditioned, the degree of the polynomial can taken to be rather small and, hence, the approximation will be sparse.

We use this observation in our algorithm to construct sets Ω_i in the following way: We first fix the cardinality of each Ω_i to be n_i (i.e. the number of nonzeros per column of P). Then, starting with initial guess $W_0 = 0 \in \mathbb{R}^{n_s \times n_c}$, we iterate towards the solution of $A_{ff}W = A_{fc}$ by ℓ steps of damped Jacobi iterations ($\ell \leq 5$):

$$W_k = W_{k-1} + \omega D_{ff}^{-1}(A_{fc} - A_{ff}W_{k-1}), \quad k = 1, \ldots, \ell. \tag{8}$$

Since this iteration behaves like a polynomial approximation to A_{ff}^{-1}, by (7), it follows that the largest entries in A_{ff}^{-1} will in fact show as large entries in W_ℓ. Thus to define Ω_i we pick the largest n_i entries in each column of W_ℓ.

There are also other methods that we are currently implementing for obtaining a polynomial approximation of A_{ff}^{-1}, such as a Conjugate Gradient approximation and also changing n_i adaptively. This is ongoing research. We point out that for the numerical results reported in 6, the approximations are based on the Jacobi iteration given in (8) with n_i fixed at the beginning.

5 On the best approximation to $P_\star$ in the trace norm

Since a covering of Ω was constructed in § 4, we proceed with the part of the algorithm for finding the interpolation weights. From the form of the iteration given in (8) for the sets $\{\Omega_i\}_{i=1}^{n_c}$, we have the following

$$\text{Each } \Omega_i \text{ contains exactly one index from } C. \tag{9}$$

To explore the relations between P obtained via trace minimization and the minimizer of $\mu(\cdot)$ introduced in § 2 consider the following affine subspaces of $\mathbb{R}^{n\times n_c}$:

$$\begin{aligned} \mathcal{X} &= \{Q \ : Q = \begin{bmatrix} W \\ I \end{bmatrix}, \ W \in \mathbb{R}^{n_s\times n_c}\}, \\ \mathcal{X}_H &= \{Q \ : Q \in \mathcal{X}, \quad Q_{ji} = 0, \quad \text{for all} \quad j \notin \Omega_i; \ Q\mathbf{1}_c = \mathbf{e}\}. \end{aligned} \tag{10}$$

Here, $\mathbf{e}$ is an arbitrary nonzero element of $\mathbb{R}^n$ (as seen from (9) $\mathbf{e}$ is subject to the restriction that it is equal to 1 at the coarse grid degrees of freedom).

The interpolation that we use in our algorithm is then defined as the unique solution of the following constrained minimization problem:

$$P = \arg\min J(Q) := \arg\min \operatorname{trace}(Q^t AQ), \quad Q \in \mathcal{X}_H. \tag{11}$$

Various relevant properties of this minimizer can be found in the literature. Existence and uniqueness are shown in [18, 19]. A proof that P is piecewise "harmonic" if $\mathbf{e}$ is harmonic can be found in [19]. It is also well known that the i-th column of the solution to (11) is given by

$$[P]_i = I_i A_i^{-1} I_i^t M_a \mathbf{e}, \quad M_a^{-1} = \sum_{i=1}^{n_c} I_i A_i^{-1} I_i^t, \tag{12}$$

where $I_i \in \mathbb{R}^{n\times n_i}$ and $(I_i)_{kl} = \delta_{kl}$ if both k and l are in Ω_i and zero otherwise, and $A_i = I_i^t A I_i$. Associate with each Ω_i a vector space, V_i, defined as:

$$V_i = \operatorname{span}\{e_j, \ j \in \Omega_i\}, \quad \dim V_i = n_i.$$

where e_j is the j-th standard canonical Euclidean basis vectors. Then, in (12), the matrix M_a^{-1} is the standard additive Schwarz preconditioner for A based on the splitting $\displaystyle\sum_{i=1}^{n_c} V_i = \mathbb{R}^n$.

We also have that, for any pair $Q_1 \in \mathcal{X}$ and $Q_2 \in \mathcal{X}$,

$$(Q_1 - Q_2)^t A P_\star = 0. \tag{13}$$

From this relation, in the extreme case, when each Ω_i contains $\{1, \dots, n_s\}$ and $\mathbf{e} = P_\star \mathbf{1}_c$, we can easily obtain that $P_\star \in \mathcal{X}_H$, $P_\star$ minimizes $J(\cdot)$ and $J(P_\star) =$

trace($\mathcal{S}(A)$). Remember that $\mathcal{S}(A)$ is the Schur complement associated with the 2×2 splitting of A.

Since $J(Q)$ is in fact also a norm (equivalent to the usual Frobenius norm for Q), for convenience, we denote it by $|||Q|||_A^2 := J(Q)$. We have the following result:

Theorem 4. *Let P be the unique solution of* (11). *Then*

$$|||P_\star - P|||_A = \min_{Q \in \mathcal{X}_H} |||P_\star - Q|||_A \tag{14}$$

Proof. Let $Q \in \mathcal{X}_H$ be arbitrary. We use formula (13) and write

$$J(Q) = J(P_\star + (Q - P_\star)) = \text{trace}(\mathcal{S}(A)) + |||P_\star - Q|||_A^2 \,. \tag{15}$$

If we take the the minimum on the left side in (15) with respect to all $Q \in \mathcal{X}_H$, then we must also achieve a minimum on the right side. Hence

$$|||P_\star - P|||_A = \min_{Q \in \mathcal{X}_H} |||P_\star - Q|||_A \,,$$

which concludes the proof of the theorem.

In fact, this theorem, provides a way to estimate $|||P_\star - P|||_A$, and also to choose $\mathbf{e}$ (an error component to be represented exactly on coarser level). Since, as is well known (and can be directly computed), $J(P) = (M_a\mathbf{e}, \mathbf{e})$, from (15), we have that

$$|||P_\star - P|||_A^2 = (M_a\mathbf{e}, \mathbf{e}) - \text{trace}(\mathcal{S}(A)). \tag{16}$$

We can now take the minimum with respect to $\mathbf{e}$ on both sides of (16) and arrive at

$$|||P_\star - P|||_A^2 = \text{trace}[\mathcal{S}(M_a) - \mathcal{S}(A)], \tag{17}$$

where $\mathcal{S}(M_a)$ is the Schur complement of M_a and this equality holds for $\mathbf{e} = \begin{bmatrix} -M_{a,ff}^{-1} M_{a,fc} \mathbf{1}_c \\ \mathbf{1}_c \end{bmatrix}$. If we want to estimate the actual error of the best approximation, we need to estimate both quantities on the right side of (17). In fact, the first term, trace[$\mathcal{S}(M_a)$], can be obtained explicitly since (9) implies that $\mathcal{S}(M_a)$ is diagonal. This can be easily seen by using the expression for M_a^{-1}, given in (12), in terms of A_i and I_i, and also the obvious relation $M_a^{-1} = \begin{bmatrix} * & * \\ * & [\mathcal{S}(M_a)]^{-1} \end{bmatrix}$. To get an accurate and computable estimate on the other quantity appearing on the right side of (16), namely, trace($\mathcal{S}(A)$), we use the result from § 4 to get the following approximation

$$\text{trace}(\mathcal{S}(A)) \approx \text{trace}(A_{cc} - G_{cc}),$$

where, as in § 4, $G_{cc} = A_{cf} p(A_{ff}) A_{fc}$, and $p(x)$ is a polynomial approximating x^{-1} on $[\lambda_{\min}(A_{ff}), \lambda_{\max}(A_{ff})]$. Such estimates and also the relations between optimizing the right hand side of (17), CR, and the optimal $\mathbf{e}$ (optimal for the norm $|||\cdot|||_A$), are also subject to an ongoing research. Currently in the numerical experiments we use an error component, $\mathbf{e}$, obtained during the CR iteration.

6 Numerical Results

We consider several problems of varying difficulty to demonstrate the effectiveness of our approach. Our test problems correspond to the bilinear finite element discretization of

$$-\nabla \cdot D(x,y)\nabla u(x,y) = f \quad \texttt{in} \quad \Omega = [0,1]\times[0,1] \tag{18}$$

$$u(x,y) = 0 \quad \texttt{on} \quad \partial\Omega \tag{19}$$

on a uniform rectangular grid. Our first test problem is Laplace's equation ($D \equiv 1$), a problem for which AMG works well. We consider the more difficult second problem defined by taking $D = \begin{bmatrix} 1 & 0 \\ 0 & 10^{-1} \end{bmatrix}$. In [5], numerical experiments demonstrate the degraded performance classical AMG exhibits for this problem without appropriate tuning of the strength parameter (θ). This is an example of the fragility of current AMG methods. For our last test, we let $D = 10^{-8}$ in 20 percent of the elements (randomly selected) and $D = 1$ in the remaining elements. This type of rough coefficient problem becomes increasingly difficult with problem size. Classical AMG performance has been shown to degrade with increasing problem size for this problem as well [7].

To test asymptotic convergence factors, we use $\mathbf{f} = 0$ and run 40 iterations of $V(1,1)$ cycles with Gauss-Seidel relaxation. The trace minimization form of interpolation is computed using five iterations of an additive Schwarz preconditioned Conjugate Gradient solver.

The results in Table 1 demonstrate that our algorithm exhibits multigrid-like optimality for test problems one and two. Test two points to one advantage of our approach, namely, that our solver maintains optimality without parameter tuning being necessary. Although the convergence factor of our solver grows with increasing problem size for test problem three, this is a rather difficult problem for any iterative solver, and our results are promising when compared to existing multilevel algorithms. To obtain a more complete picture of the overall effectiveness of our multigrid iteration, we examine also *operator complexity*, defined as the number of nonzero entries stored in the operators on all levels divided by the number of non-zero entries in the finest-level matrix. The operator complexity can be viewed as indicating how expensive the entire V-cycle is compared to performing only the finest-level relaxations of the V-cycle. We note that the operator complexities are acceptable for all of the test problems and remain bounded with repsect to problem size.

N	Problem 1	Problem 2	Problem 3
128^2	.085 / 5 / 1.29	.110 / 5 / 1.31	.098 / 5 / 1.79
256^2	.113 / 6 / 1.31	.124 / 6 / 1.35	.139 / 7 / 1.83
512^2	.118 / 7 / 1.33	.125 / 7 / 1.38	.197 / 9 / 1.87

Table 1. Asymptotic convergence factors / number of levels / operator complexities for test Problems 1-3.

7 Conclusions

Our current approach is only a first step towards developing a more general AMG algorithm. Using CR in constructing C and a trace minimization form of interpolation, we are able to efficiently solve problems arising from scalar PDEs. For systems of PDEs, there are other approaches that fit quite well in the framework described here. The CR algorithm can be extended in a straightforward way to include block smoothers as well as to incorporate more general algorithms for trace minimization (such as the one described in [17]). Another attractive alternative is presented by using adaptive coarse space definition, namely by running simultaneous V-cycle iterations on the linear system that we want to solve and the corresponding homogeneous system (the latter with random initial guess) and using the error of the homogeneous iteration to define the constraint in the trace minimization formulation. Although expensive (part of the setup process has to be performed on every iteration), this procedure should be very robust and work in cases when there are many algebraically smooth error components that need to be approximated.

Acknowledgments

The authors would like to thank Rob Falgout and Panayot Vassilevski of the Center of Applied and Scientific Computing at the Lawrence Livermore National Laboratory (LLNL), Marian Brezina, Scott MacLachlan, Tom Manteuffel, Steve McCormick and John Ruge of the CU Boulder Applied Math Department, and Jinchao Xu of the Penn State Math Department for their insightful comments on this work. This work was sponsored by the Department of Energy under grant numbers DE-FC02-01ER25479 and DE-FC02-04ER25595, Lawrence Livermore National Laboratory under contracts number B533502 and number B551021, Sandia National Laboratories under contract number 15268, and the National Science Foundation under grant number DMS-0410318.

References

1. A. Brandt, *Algebraic multigrid theory: The symmetric case*, Appl. Math. Comput., 19 (1986), pp. 23–56.

2. ———, *Generally highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal., 10 (2000), pp. 1–20.
3. J. Brannick, M. Brezina, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *An energy-based AMG coarsening strategy*, Numer. Linear Algebra Appl., 12 (2006), pp. 133–148.
4. J. Brannick and R. Falgout, *Compatible relaxation and coarsening in algebraic multigrid.* In preparation.
5. M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Sci. Comput., 22 (2000), pp. 1570–1592.
6. M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA)*, SIAM J. Sci. Comput., 25 (2004), pp. 1896–1920.
7. ———, *Adaptive algebraic multigrid methods*, SIAM J. Sci. Comput., 27 (2006), pp. 1261–1286.
8. S. Demko, W. F. Moss, and P. W. Smith, *Decay rates of inverse band matrices*, Math. Comp., 43 (1984), pp. 491–499.
9. R. D. Falgout and P. S. Vassilevski, *On generalizing the algebraic multigrid framework*, SIAM J. Numer. Anal., 42 (2004), pp. 1669–1693.
10. R. D. Falgout, P. S. Vassilevski, and L. T. Zikatanov, *On two-grid convergence estimates*, Numer. Linear Algebra Appl., 12 (2005), pp. 471–494.
11. A. Gibbons, *Algorithmic Graph Theory*, Cambridge University Press, 1985.
12. O. E. Livne, *Coarsening by compatible relaxtion*, Numer. Linear Algebra Appl., 11 (2004), pp. 205–227.
13. J. W. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1987, pp. 73–130.
14. U. Trottenberg, C. W. Oosterlee, and A. Schüller, *Multigrid*, Academic Press, London, 2001.
15. P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid based on smoothed aggregation for second and fourth order problems*, Computing, 56 (1996), pp. 179–196.
16. P. Vassilevski, *Exponential decay in sparse matrix inverses.* rivate communication, July 2004.
17. P. S. Vassilevski and L. T. Zikatanov, *Multiple vector preserving interpolation mappings in algebraic multigrid*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1040–1055.
18. W. L. Wan, T. F. Chan, and B. Smith, *An energy-minimizing interpolation for robust multigrid methods*, SIAM J. Sci. Comput., 21 (2000), pp. 1632–1649.
19. J. Xu and L. Zikatanov, *On an energy minimizing basis for algebraic multigrid methods*, Comput. Vis. Sci., 7 (2004), pp. 121–127.

Lower Bounds in Domain Decomposition

Susanne C. Brenner

Center for Computation and Technology, Johnston Hall, Louisiana State University, Baton Rouge, LA 70803, USA. brenner@math.lsu.edu

1 Introduction

An important indicator of the efficiency of a domain decomposition preconditioner is the condition number of the preconditioned system. Upper bounds for the condition numbers of the preconditioned systems have been the focus of most analyses in domain decomposition [21, 20, 23]. However, in order to have a fair comparison of two preconditioners, the sharpness of the respective upper bounds must first be established, which means that we need to derive lower bounds for the condition numbers of the preconditioned systems.

In this paper we survey lower bound results for domain decomposition preconditioners [7, 3, 8, 5, 22] that can be obtained within the framework of additive Schwarz preconditioners. We will describe the results in terms of the following model problem.

Find $u_h \in V_h$ such that

$$\int_{\Omega} \nabla u_h \cdot \nabla v \, dx = \int_{\Omega} f v \, dx \qquad \forall \, v \in V_h, \tag{1}$$

where $\Omega = [0,1]^2$, $f \in L_2(\Omega)$, and V_h is the P_1 Lagrange finite element space associated with a uniform triangulation $\mathcal{T}_h$ of Ω. We assume that the length of the horizontal (or vertical) edges of $\mathcal{T}_h$ is a dyadic number $h = 2^{-k}$.

We recall the basic facts concerning additive Schwarz preconditioners in Section 2 and present the lower bound results for one-level and two-level additive Schwarz preconditioners, Bramble-Pasciak-Schatz preconditioner and the FETI-DP preconditioner in Sections 3–6. Section 7 contains some concluding remarks.

2 Additive Schwarz Preconditioners

Let V be a finite dimensional vector space and $A : V \longrightarrow V'$ be an SPD operator, i.e., $\langle Av_1, v_2 \rangle = \langle Av_2, v_1 \rangle \; \forall \, v_1, v_2 \in V$ and $\langle Av, v \rangle > 0$ for any

$v \in V \setminus \{0\}$, where $\langle \cdot, \cdot \rangle$ denotes the canonical bilinear form between a vector space and its dual.

The ingredients for an additive Schwarz preconditioner B for A are (i) auxiliary finite dimensional vector spaces V_j for $1 \leq j \leq J$, (ii) SPD operators $A_j : V_j \longrightarrow V_j'$ and (iii) connection operators $I_j : V_j \longrightarrow V$. The preconditioner $B : V' \longrightarrow V$ is then given by

$$B = \sum_{j=1}^{J} I_j A_j^{-1} I_j^t,$$

where $I_j^t : V' \longrightarrow V_j'$ is the transpose of I_j, i.e. $\langle I_j^t \phi, v \rangle = \langle \phi, I_j v \rangle \ \forall \, \phi \in V'$ and $v \in V_j$.

Under the condition $V = \sum_{j=1}^{J} I_j V_j$, the operator B is SPD and the maximum and minimum eigenvalues of $BA : V \longrightarrow V$ are characterized by the following formulas [26, 1, 25, 14, 21, 8, 23]:

$$\lambda_{\max}(BA) = \max_{v \in V \setminus \{0\}} \frac{\langle Av, v \rangle}{\displaystyle\min_{\substack{v = \sum_{j=1}^{J} I_j v_j \\ v_j \in V_j}} \sum_{j=1}^{J} \langle A_j v_j, v_j \rangle}, \tag{2}$$

$$\lambda_{\min}(BA) = \min_{v \in V \setminus \{0\}} \frac{\langle Av, v \rangle}{\displaystyle\min_{\substack{v = \sum_{j=1}^{J} I_j v_j \\ v_j \in V_j}} \sum_{j=1}^{J} \langle A_j v_j, v_j \rangle}. \tag{3}$$

3 One-Level Additive Schwarz Preconditioner

Let $A_h : V_h \rightarrow V_h'$ be defined by

$$\langle A_h v_1, v_2 \rangle = \int_{\Omega} \nabla v_1 \cdot \nabla v_2 \, dx \quad \forall \, v_1, v_2 \in V_h.$$

We can precondition the operator A_h using subdomain solves from an overlapping decomposition, which is created by (i) dividing Ω into $J = H^{-2}$ nonoverlapping squares (H is a dyadic number $\gg h$) and (ii) enlarging the nonoverlapping subdomains by an amount of δ ($\leq H$) so that each of the overlapping subdomains $\Omega_1, \ldots, \Omega_J$ is the union of triangles from $\mathcal{T}_h$ (cf. Figure 1). We take the auxiliary space $V_j \subset H_0^1(\Omega_j)$ to be the finite element space associated with the triangulation of Ω_j by triangles from $\mathcal{T}_h$, and define the SPD operator $A_j : V_j \longrightarrow V_j'$ by

$$\langle A_j v_1, v_2 \rangle = \int_{\Omega_j} \nabla v_1 \cdot \nabla v_2 \, dx \quad \forall\, v_1, v_2 \in V_j.$$

The space V_j is connected to V_h by the trivial extension map I_j and the one-level additive Schwarz preconditioner [19] B_{OL} for A_h is defined by

$$B_{OL} = \sum_{j=1}^{J} I_j A_j^{-1} I_j^t. \tag{4}$$

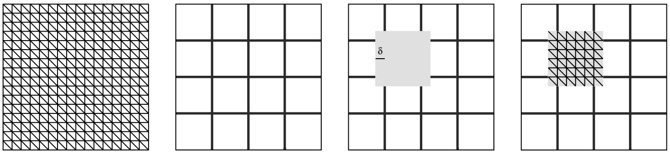


Fig. 1. An overlapping domain decomposition

It is well-known that the preconditioner B_{OL} does not scale. Here we give a lower bound for the condition number $\kappa(B_{OL}A_h)$ that explains this phenomenon. We use the notation $A \lesssim B$ ($B \gtrsim A$) to represent the inequality $A \leq (\text{constant})B$, where the positive constant is independent of h, J, δ and H. The statement $A \approx B$ is equivalent to $A \lesssim B$ and $A \gtrsim B$.

Theorem 1. *Under the condition $\delta \approx H$, it holds that*

$$\kappa(B_{OL}A_h) = \lambda_{\max}(B_{OL}A_h)/\lambda_{\min}(B_{OL}A_h) \gtrsim J. \tag{5}$$

Proof. Since the connection maps I_j preserve the energy norm (in other words, $\langle A_h I_j v, I_j v \rangle = \langle A_j v, v \rangle \; \forall\, v \in V_j$), it follows immediately from (2) that

$$\lambda_{\max}(B_{OL}A_h) \geq 1. \tag{6}$$

Let $v_* \in H_0^1(\Omega)$ be the piecewise linear function with respect to the triangulation of Ω of mesh size $1/4$ such that v_* equals 1 on the four central squares (cf. the first figure in Figure 2). Since v_* is independent of h, we have

$$\langle A_h v_*, v_* \rangle = |v_*|^2_{H^1(\Omega)} \approx 1 \tag{7}$$

as $h \downarrow 0$. We will show that, for this function $v_* \in V_h$, the estimate

$$\sum_{j=1}^{J} \langle A_j v_j, v_j \rangle \gtrsim J \langle A_h v_*, v_* \rangle \tag{8}$$

holds whenever

$$v_* = \sum_{j=1}^{J} I_j v_j \qquad \text{and} \quad v_j \in V_j \quad \text{for} \quad 1 \le j \le J. \tag{9}$$

It follows immediately from (3), (8) and (9) that

$$\lambda_{\min}(B_{OL}A_h) \lesssim 1/J, \tag{10}$$

which together with (6) implies (5).

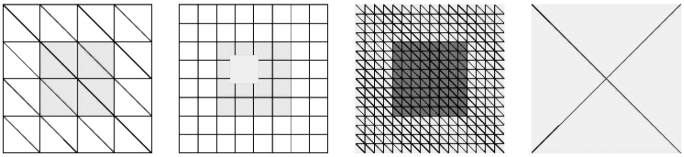


Fig. 2. Subdomains for Theorem 1

In order to derive (8), we first focus on a single subdomain Ω_j that overlaps with the square where v_* is identically 1 (cf. the second figure in Figure 2), and without loss of generality, assume that $\delta = H/4$. Condition (9) then implies $v_j = 1$ in the central area of Ω_j (cf. the third figure of Figure 2).

We can construct a weak interpolation operator Π from $H^1(\Omega_j)$ into the space of functions that are piecewise linear with respect to the triangulation of Ω_j by its two diagonals (cf. the fourth figure of Figure 2). For $v \in H^1(\Omega_j)$, we define the value of Πv at the four corners of Ω_j to be the mean of v on $\partial\Omega_j$ and the value of Πv at the center of Ω_j to be the mean of v on the central area of Ω_j. It follows that Πv_j equals 1 at the center of Ω_J and vanishes identically on $\partial\Omega_j$. A simple calculation shows that $|\Pi v_j|^2_{H^1(\Omega_j)} \approx 1$. On the other hand, the weak interpolation operator satisfies the estimate $|\Pi v_j|_{H^1(\Omega_j)} \lesssim |v_j|_{H^1(\Omega_j)}$. We conclude that

$$\langle A_j v_j, v_j \rangle = |v_j|^2_{H^1(\Omega_j)} \gtrsim 1. \tag{11}$$

Since there are $J/4$ such subdomains, (8) follows from (7) and (11).

Remark 1. The estimate (5) implies that, for a given tolerance, the number of iterations for the preconditioned conjugate gradient method grows at the rate of $O(\sqrt{J}) = O(1/H)$, a phenomenon that has been observed numerically [21]. See also the discussion on page 17 of [23].

4 Two-Level Additive Schwarz Preconditioner

To obtain scalability for the additive Schwarz overlapping domain decomposition preconditioner, Dryja and Widlund [10] developed a two-level preconditioner by introducing a coarse space.

Let $\mathcal{T}_H$ be a coarse triangulation of Ω obtained by adding diagonals to the underlying nonoverlapping squares whose sides are of length H (cf. the second figure in Figure 1) and $V_H \subset H_0^1(\Omega)$ be the corresponding P_1 finite element space. The coarse space V_H is connected to V_h by the natural injection I_H, and $A_H : V_H \longrightarrow V_H'$ is defined by

$$\langle A_H v_1, v_2 \rangle = \int_\Omega \nabla v_1 \cdot \nabla v_2 \, dx \qquad \forall\, v_1, v_2 \in V_H.$$

The two-level preconditioner $B_{TL} : V_h' \longrightarrow V_h$ is then given by

$$B_{TL} = I_H A_H^{-1} I_H^t + B_{OL} = I_H A_H^{-1} I_H^t + \sum_{j=1}^J I_j A_j^{-1} I_j^t. \tag{12}$$

It follows from the well-known estimate [11]

$$\kappa(B_{TL} A_h) \lesssim 1 + \frac{H}{\delta} \tag{13}$$

that B_{TL} is an optimal preconditioner when $\delta \approx H$ (the case of generous overlap). However, in the case of small overlap where $\delta \ll H$, the number $1 + (H/\delta)$ becomes significant and it is natural to ask whether the estimate (13) can be improved. That the estimate (13) is sharp is established by the following lower bound result [3].

Theorem 2. *In the case of minimal overlap where $\delta = h$, it holds that*

$$\kappa(B_{TL} A_h) \gtrsim \frac{H}{h}. \tag{14}$$

We will sketch the derivation of (14) in the remaining part of this section and refer to [3] for the details.

First observe that, by comparing (4) and (12), the estimate

$$\lambda_{\max}(B_{TL} A_h) \geq \lambda_{\max}(B_{OL} A_h) \geq 1 \tag{15}$$

follows immediately from (2) and (6).

In the other direction, it suffices to construct a finite element function $v_* \in V_h$ such that, for any decomposition $v_* = I_H v_H + \sum_{j=j}^J I_j v_j$ where $v_H \in V_H$ and $v_j \in V_j$,

$$\frac{H}{h} \langle A_h v_*, v_* \rangle \lesssim \langle A_H v_H, v_H \rangle + \sum_{j=1}^J \langle A_j v_j, v_j \rangle. \tag{16}$$

The estimate $\lambda_{\min}(B_{TL} A_h) \lesssim h/H$ then follows from (3) and (16), and together with (15) it implies (14).

Since the subdomains are almost nonoverlapping when $\delta = h$, we can construct v_* using techniques from nonoverlapping domain decomposition. Let $\hat{\Omega}_j$ $(1 \le j \le J)$ be the underlying nonoverlapping decomposition of Ω (cf. the second figure in Figure 1) from which we construct the overlapping decomposition, and $\Gamma = \bigcup_{j=1}^{J} \partial\hat{\Omega}_j \setminus \partial\Omega$ be the interface of $\hat{\Omega}_1, \ldots, \hat{\Omega}_J$. The space $V_h(\Gamma)$ of discrete harmonic functions is defined by

$$V_h(\Gamma) = \{v \in V_h : \int_\Omega \nabla v \cdot \nabla w \, dx = 0 \quad \forall\, w \in V_h, w\big|_\Gamma = 0\}.$$

We will choose v_* from $V_h(\Gamma)$. Note that a discrete harmonic function is uniquely determined by its restriction on Γ.

Let E be an edge of length H shared by two nonoverlapping subdomains $\hat{\Omega}_1$ and $\hat{\Omega}_2$. Let g be a function defined on E such that (i) g is piecewise linear with respect to the uniform subdivision of E of mesh size $H/8$, (ii) g is identically zero within a distance of $H/4$ from either one of the endpoints of E, (iii) g is $L_2(E)$-orthogonal to all polynomials on E of degree ≤ 1. (It is easy to see that such a function g exists by a dimension argument.) We then define $v_* \in V_h(\Gamma)$ to be g on E and 0 on $\Gamma \setminus E$.

It follows from property (ii) of g and standard properties of discrete harmonic functions [2, 6, 23] that

$$\begin{aligned} \langle A_h v_*, v_* \rangle = |v_*|^2_{H^1(\Omega)} &\approx \sum_{j=1}^{2} |v_*|^2_{H^{1/2}(\partial\hat{\Omega}_j)} \\ &\approx |g|^2_{H^{1/2}(E)} \approx \frac{1}{H}\|g\|^2_{L_2(E)} = \frac{1}{H}\|v_*\|^2_{L_2(E)}. \end{aligned} \tag{17}$$

Suppose $v_* = I_H v_H + \sum_{j=1}^{J} I_j v_j$ where $v_H \in V_H$ and $v_j \in V_j$ for $1 \le j \le J$. Let E_c be the set of points in E whose distance from the endpoints of E exceed $H/4$. Since $v_H\big|_E$ is a polynomial of degree ≤ 1, property (iii) of g implies that

$$\|v_*\|^2_{L_2(E_c)} \le \|v_* - v_H\|^2_{L_2(E_c)} = \|\sum_{j=1}^{J} v_j\|^2_{L_2(E_c)} = \|v_1 + v_2\|^2_{L_2(E_c)}, \tag{18}$$

where we have also used the fact that $v_j = 0$ on E_c for $j \ne 1, 2$ because $\delta = h$.

Finally, since v_1 (resp. v_2) vanishes on $\partial\Omega_1$ (resp. $\partial\Omega_2$) which is within one layer of elements from E, a simple calculation shows that

$$\|v_j\|^2_{L_2(E_c)} \lesssim h|v_j|^2_{H^1(\Omega_j)} = h\langle A_j v_j, v_j \rangle \quad \text{for} \quad j = 1, 2. \tag{19}$$

The estimate (16) follows from (17)–(19).

Remark 2. Theorem 2 also holds for nonconforming finite elements [7] and mortar elements [22]. It can also be extended to fourth order problems [8, 7] in which case the right-hand side of (14) becomes $(H/h)^3$.

5 Bramble-Pasciak-Schatz Preconditioner

Let Γ be the interface of a nonoverlapping decomposition of Ω and $V_h(\Gamma)$ be the space of discrete harmonic functions as described in Section 4. By a parallel subdomain solve, we can reduce (1) to the following problem.

Find $\bar{u}_h \in V_h(\Gamma)$ such that

$$\langle S_h \bar{u}_h, v\rangle = \int_\Omega f v\,dx \qquad \forall\, v \in V_h(\Gamma),$$

and the Schur complement operator $S_h : V_h(\Gamma) \longrightarrow V_h(\Gamma)'$, defined by

$$\langle S_h v_1, v_2\rangle = \int_\Omega \nabla v_1 \cdot \nabla v_2\,dx \qquad \forall\, v_1, v_2 \in V_h(\Gamma),$$

is the operator that needs a preconditioner.

The auxiliary spaces for the Bramble-Pasciak-Schatz preconditioner [2] are the coarse space $V_{\scriptscriptstyle H}$ introduced in Section 4, and the edge spaces $V_\ell = \{v \in V_h(\Gamma) : v = 0 \text{ on } \Gamma \setminus E_\ell\}$ associated with the edges E_ℓ of the interface Γ. The space $V_{\scriptscriptstyle H}$ is equipped with the SPD operator $A_{\scriptscriptstyle H}$ introduced in Section 4, and is connected to $V_h(\Gamma)$ by the map $I_{\scriptscriptstyle H}$ that maps $v \in V_{\scriptscriptstyle H}$ to the discrete harmonic function that agrees with v on Γ. The edge space V_ℓ is connected to $V_h(\Gamma)$ by the natural injection I_j, and is equipped with the Schur complement operator $S_\ell : V_\ell \longrightarrow V_\ell'$ defined by

$$\langle S_\ell v_1, v_2\rangle = \int_\Omega \nabla v_1 \cdot \nabla v_2\,dx \qquad \forall\, v_1, v_2 \in V_\ell.$$

The preconditioner $B_{\scriptscriptstyle BPS} : V_h(\Gamma)' \longrightarrow V_h(\Gamma)$ is then given by

$$B_{\scriptscriptstyle BPS} = I_{\scriptscriptstyle H} A_{\scriptscriptstyle H}^{-1} I_{\scriptscriptstyle H} + \sum_{\ell=1}^{L} I_\ell S_\ell^{-1} I_\ell^t.$$

The sharpness of the well-known estimate [2]

$$\kappa(B_{\scriptscriptstyle BPS} S_h) \lesssim \Big(1 + \ln \frac{H}{h}\Big)^2 \tag{20}$$

follows from the following lower bound result [8].

Theorem 3. *It holds that*

$$\kappa(B_{\scriptscriptstyle BPS} S_h) \gtrsim \Big(1 + \ln \frac{H}{h}\Big)^2. \tag{21}$$

Since the natural injection I_ℓ preserves the energy norm, it follows immediately from (2) that

$$\lambda_{\max}(B_{\scriptscriptstyle BPS} S_h) \geq 1. \tag{22}$$

To complete the proof of (21), it suffices to construct $v_* \in V_h(\Gamma)$ such that, for the unique decomposition $v_* = I_H v_H + \sum_{\ell=1}^{L} v_\ell$ where $v_H \in V_H$ and $v_\ell \in V_\ell$,

$$\langle A_H v_H, v_H\rangle + \sum_{\ell=1}^{L} \langle S_\ell v_\ell, v_\ell\rangle \gtrsim \Big(1 + \ln\frac{H}{h}\Big)^2 \langle S_h v_*, v_*\rangle, \tag{23}$$

which together with (3) implies that $\lambda_{\min}(B_{BPS} S_h) \lesssim \Big(1 + \ln\frac{H}{h}\Big)^{-2}$ and thus, in view of (22), completes the proof of (21). Below we will sketch the construction of v_* and refer to [8] for the details.

Since the derivation of (20) depends crucially on the discrete Sobolev inequality [2, 6, 23] that relates the L_∞ norm and the H^1 norm of finite element functions on two-dimensional domains, v_* is intimately related to piecewise linear functions on an interval with special property with respect to the Sobolev norm of order $\frac{1}{2}$. Let $I = (0, 1)$. A key observation is that

$$|v|^2_{H^{1/2}_{00}(I)} \approx \sum_{n=1}^{\infty} n|v_n|^2 \qquad \forall\, v \in H^{1/2}_{00}(I), \tag{24}$$

where $\sum_{n=1}^{\infty} v_n \sin(n\pi x)$ is the Fourier sine-series expansion of v.

Let $\mathcal{T}_\rho$ ($\rho = 2^{-k}$) be a uniform dyadic subdivision of I and $\mathcal{L}_\rho \subset H^1_0(I)$ be the space of piecewise linear functions on I (with respect to $\mathcal{T}_\rho$) that vanish at 0 and 1. The special piecewise linear functions that we need come from the functions S_N ($N = 2^k = \rho^{-1}$) defined by

$$S_N(x) = \sum_{n=1}^{N} \Big(\frac{1}{4n-3}\Big) \sin\big((4n-3)\pi x\big). \tag{25}$$

From (24) and (25) we find

$$|S_N|^2_{H^{1/2}_{00}(I)} \approx \ln N \approx |\ln \rho|, \tag{26}$$

and a direct calculation shows that

$$|S_N|^2_{H^1(I)} \approx N = \rho^{-1}. \tag{27}$$

Now we define $\sigma_\rho \in \mathcal{L}_\rho$ to be the nodal interpolant of S_N. It follows from (26), (27) and an interpolation error estimate that

$$|\sigma_\rho|^2_{H^{1/2}_{00}(I)} \approx |\ln \rho|. \tag{28}$$

Remark 3. Since $\|\sigma_\rho\|_{L_\infty(I)} = \sigma_\rho(1/2) = S_N(1/2) \approx \ln N = |\ln \rho|$, the estimate (28) implies the sharpness of the discrete Sobolev inequality.

Let σ_ρ^I be the piecewise linear interpolant of S_N with respect to the coarse subdivision $\{0, 1/2, 1\}$ of I. Then a calculation using (24) yields

$$|\sigma_\rho - \sigma_\rho^I|^2_{H^{1/2}_{00}(0,1/2)} = |\sigma_\rho - \sigma_\rho^I|^2_{H^{1/2}_{00}(1/2,1)} \approx |\ln \rho|^3. \tag{29}$$

Finally we take $\rho = h/2H$ and $g(x) = \sigma_\rho\big((x+H)/2H\big)$. Then g is a continuous piecewise linear function on $[-H, H]$ with respect to the uniform partition of mesh size h. Note that S_N is symmetric with respect to the midpoint $1/2$ and hence g is symmetric with respect to 0. We can now define $v_* \in V_h(\Gamma)$ as follows: (i) v_* vanishes on Γ except on the two line segments P_1P_2 and P_3P_4 (each of length $2H$) that form the interface of the four nonoverlapping subdomains $\Omega_1, \ldots, \Omega_4$ (cf. the first figure in Figure 3), and (ii) $v_* = g$ on P_1P_2 and P_3P_4.

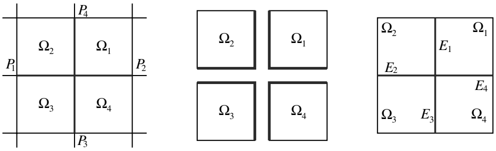


Fig. 3. The four subdomains associated with v_*

It is clear that $v_* = 0$ outside the four subdomains and, by the symmetry of g, $v_* = g$ on one half of $\partial\Omega_j$ (represented by the thick lines in the second figure in Figure 3) and vanishes at the other half, for $1 \le j \le 4$. Therefore, we have, from (28) and standard properties of discrete harmonic functions,

$$\begin{aligned}\langle S_h v_*, v_*\rangle &= \sum_{j=1}^{4} |v_*|^2_{H^1(\Omega_j)} \approx \sum_{j=1}^{4} |v_*|^2_{H^{1/2}(\partial\Omega_j)} \\ &\approx |g|^2_{H^{1/2}_{00}(-H,H)} = |\sigma_\rho|^2_{H^{1/2}_{00}(0,1)} \approx |\ln \rho| \approx \ln \frac{H}{h}. \end{aligned} \tag{30}$$

The function v_* admits a unique decomposition $v_* = I_H v_H + \sum_{\ell=1}^{4} v_\ell$, where $v_H \in V_H$, $v_\ell \in V(E_\ell)$ and E_ℓ $(1 \le j \le 4)$ are the interfaces of $\Omega_1, \ldots, \Omega_4$ (cf. the third figure in Figure 3). On each E_ℓ, $v_\ell = v - I_H v_H$ agrees with $g - g^I$, where g^I is the linear polynomial that agrees with g at the two endpoints of E_ℓ. Therefore it follows from (29) that

$$\langle S_\ell v_\ell, v_\ell\rangle \approx |\ln \rho|^3 \approx \Big(\ln \frac{H}{h}\Big)^3 \qquad \text{for} \quad 1 \le \ell \le 4, \tag{31}$$

and the estimate (23) follows from (30) and (31).

6 FETI-DP Preconditioner

Let $\Omega_1, \ldots, \Omega_J$ be a nonoverlapping decomposition of Ω aligned with $\mathcal{T}_h$ (cf. the first two figures in Figure 4) and $\tilde{V}_h = \{v \in L_2(\Omega) : v$ is a standard P_1 finite element function on each subdomain, v is not required to be continuous on the interface Γ except at the cross points and $v = 0$ on $\partial\Omega\}$. In the Dual-Primal Finite Element Tearing and Interconnecting (FETI-DP) approach [13], the problem (1) is rewritten as

$$\begin{aligned} \sum_{j=1}^{J} \int_{\Omega_j} \nabla u_h \cdot \nabla v \, dx + \langle \phi, v \rangle &= \int_{\Omega} f v \, dx \qquad && \forall\, v \in \tilde{V}_h, \\ \langle \mu, u_h \rangle \qquad\qquad\qquad\qquad &= 0 && \forall\, \mu \in M_h, \end{aligned} \tag{32}$$

where $M_h \subset \tilde{V}_h'$ is the space of Lagrange multipliers that enforce the continuity of v along the interface Γ. More precisely, for each node p on Γ that is not a cross point, we have a multiplier $\mu_p \in \tilde{V}_h'$ defined by $\langle \mu_p, v \rangle = (v|_{\Omega_j})(p) - (v|_{\Omega_k})(p)$, where Ω_j and Ω_k are the two subdomains whose interface contains p, and the space M_h is spanned by all such μ_p's.

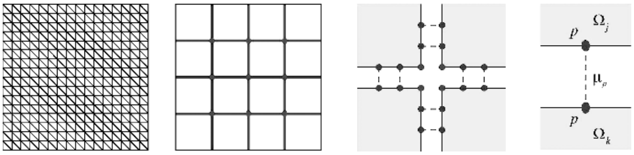


Fig. 4. FETI

By solving local SPD problems (associated with the subdomains) and a global SPD problem (associated with the cross points), the unknown u_h can be eliminated from (32), and the resulting system for ϕ involves the operator $\hat{\mathbb{S}}_h : M_h \longrightarrow M_h'$ defined by $\hat{\mathbb{S}}_h = R^t \tilde{S}_h^{-1} R$, where $R : M_h \longrightarrow [\tilde{V}_h(\Gamma)]'$ is the restriction map, $\tilde{V}_h(\Gamma)$ is the subspace of $\tilde{V}_h$ consisting of discrete harmonic functions, and $\tilde{S}_h : \tilde{V}_h(\Gamma) \longrightarrow \tilde{V}_h(\Gamma)'$ is the corresponding Schur complement operator.

Let V_j $(1 \le j \le J)$ be the space of discrete harmonic functions on Ω_j that vanish at the corners of Ω_j and $S_j : V_j \longrightarrow V_j'$ be the Schur complement operator (which is SPD). The dual spaces V_j' are the auxiliary spaces of the additive Schwarz preconditioner for $\hat{\mathbb{S}}_h$ developed by Mandel and Tezaur in [18]. Each V_j' is connected to M_h by the operator I_j defined by $\langle I_j \psi, \tilde{v} \rangle = \dfrac{1}{2} \langle \psi, v \rangle$ $\forall\, v \in V_j$, where $I_j \psi$ is a linear combination of μ_p for $p \in \Gamma_j$ and $\tilde{v} \in \tilde{V}_h$ is the trivial extension of v. The preconditioner in [18] is given by

$$B_{DP} = \sum_{j=1}^{J} I_j S_j I_j^t,$$

and the condition number estimate

$$\kappa(B_{DP}\hat{\mathbb{S}}_h) \lesssim \Big(1 + \ln\frac{H}{h}\Big)^2 \tag{33}$$

was established in [18]. The sharpness of (33) is a consequence of the following lower bound result [4].

Theorem 4. *It holds that*

$$\kappa(B_{DP}\hat{\mathbb{S}}_h) \gtrsim \Big(1 + \ln\frac{H}{h}\Big)^2.$$

Since the operator $B_{DP}\hat{\mathbb{S}}_h$ is essentially dual to the operator $B_{BPS}S_h$, Theorem 4 is derived using the special piecewise linear functions from Section 5 and duality arguments. Details can be found in [4].

7 Concluding Remarks

We present two dimensional results in this paper for simplicity. But the generalization of the results of Sections 3 and 4 to three dimensions is straightforward, and the results in Section 5 have been generalized [5] to three dimensions (wire-basket algorithm [9]) and Neumann-Neumann algorithms [12]. Since the balancing domain decomposition by constraint (BDDC) method has the same condition number as the FETI-DP method [17, 15], the sharpness of the condition number estimate for BDDC [16] also follows from Theorem 4.

We would also like to mention that the special discrete harmonic function v_* constructed in Section 5 has been used in the derivation of an upper bound for the three-level BDDC method [24].

Acknowledgement. The work in this paper was partially supported by the National Science Foundation under Grant No. DMS-03-11790.

References

1. P. E. Bjørstad and J. Mandel, *On the spectra of sums of orthogonal projections with applications to parallel computing*, BIT, 31 (1991), pp. 76–88.
2. J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp., 47 (1986), pp. 103–134.
3. S. C. Brenner, *Lower bounds for two-level additive Schwarz preconditioners with small overlap*, SIAM J. Sci. Comput., 21 (2000), pp. 1657–1669.

4. ———, *Analysis of two-dimensional FETI-DP preconditioners by the standard additive Schwarz framework*, Electron. Trans. Numer. Anal., 16 (2003), pp. 165–185.
5. S. C. Brenner and Q. He, *Lower bounds for three-dimensional nonoverlapping domain decomposition algorithms*, Numerische Mathematik, (2003).
6. S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, New York, second ed., 2002.
7. S. C. Brenner and L.-Y. Sung, *Lower Bounds for Two-Level Additive Schwarz Preconditioners for Nonconforming Finite Elements*, vol. 202 of Lecture Notes in Pure and Applied Mathematics, Marcel Dekker AG, New York, 1999, pp. 585–604.
8. ———, *Lower bounds for nonoverlapping domain decomposition preconditioners in two dimensions*, Math. Comp., 69 (2000), pp. 1319–1339.
9. 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 (1994), pp. 1662–1694.
10. M. Dryja and O. B. Widlund, *An additive variant of the Schwarz alternating method in the case of many subregions*, Tech. Rep. 339, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 1987.
11. ———, *Domain decomposition algorithms with small overlap*, SIAM J. Sci.Comput., 15 (1994), pp. 604–620.
12. ———, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Comm. Pure Appl. Math., 48 (1995), pp. 121–155.
13. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.
14. M. Griebel and P. Oswald, *On the abstract theory of additive and multiplicative Schwarz algorithms*, Numerische Mathematik, 70 (1995), pp. 163–180.
15. J. Li and O. B. Widlund, *FETI-DP, BDDC, and block Cholesky methods*, Tech. Rep. 857, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 2004.
16. J. Mandel and C. R. Dohrmann, *Convergence of a balancing domain decomposition by constraints and energy minimization*, Numer. Linear Algebra Appl., 10 (2003), pp. 639–659.
17. J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Appl. Numer. Math., 54 (2005), pp. 167–193.
18. J. Mandel and R. Tezaur, *On the convergence of a dual-primal substructuring method*, Numer. Math., 88 (2001), pp. 543–558.
19. A. M. Matsokin and S. V. Nepomnyaschikh, *A Schwarz alternating method in a subspace*, Soviet Mathematics, 29 (1985), pp. 78–84.
20. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.
21. B. F. Smith, P. E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
22. D. Stefanica, *Lower bounds for additive Schwarz methods with mortars*, C.R. Math. Acad. Sci. Paris, 339 (2004), pp. 739–743.

23. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

24. X. Tu, *Three-level BDDC in two dimensions*, Tech. Rep. 856, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 2004.

25. J. Xu, *Iterative methods by space decomposition and subspace correction*, SIAM Review, 34 (1992), pp. 581–613.

26. X. Zhang, *Studies in Domain Decomposition: Multilevel Methods and the Biharmonic Dirichlet Problem*, PhD thesis, Courant Institute, New York University, September 1991.

Heterogeneous Domain Decomposition Methods for Fluid-Structure Interaction Problems

Simone Deparis[1], Marco Discacciati[2], Gilles Fourestey[2], and Alfio Quarteroni[2,3]

[1] Mechanical Engineering Department, Massachusetts Institute of Technology, 77 Mass Ave, Cambridge MA 02139, USA. simone.deparis@epfl.ch
[2] IACS - Chair of Modeling and Scientific Computing, EPFL, CH-1015 Lausanne, Switzerland. marco.discacciati@oeaw.ac.at, gilles.fourestey@epfl.ch
[3] MOX, Politecnico di Milano, P.zza Leonardo da Vinci 32, 20133 Milano, Italy. alfio.quarteroni@epfl.ch

Summary. In this note, we propose Steklov-Poincaré iterative algorithms (mutuated from the analogy with heterogeneous domain decomposition) to solve fluid-structure interaction problems. Although our framework is very general, the driving application is concerned with the interaction of blood flow and vessel walls in large arteries.

1 Introduction

Mathematical modeling of real-life problems may lead to different kind of boundary value problems in different subregions of the original computational domain. The reason may be twofold.

Often, in order to reduce the computational cost of the simulation, a very detailed model can be used only in a region of specific interest while resorting to a simplified version of the same model sufficiently far away from where the most relevant physical phenomena occur. This is, e.g., the strategy adopted when one considers the coupling of advection-diffusion equations with advection equations, after neglecting the diffusive effects in a certain subregion (see, e.g., [11]), or when the full Navier-Stokes equations are coupled with Oseen, Stokes or even velocity potential models, the latter being adopted where the nonlinear convective effects are negligible (see, e.g., [7, 8]).

In a second circumstance, one may be obliged to consider truly different models to account for the presence of distinct physical problems within the same global domain. This case is usually indicated as multi-physics or multi-field problem.

Typical examples are given by filtration processes such as in biomechanics or in environmental applications where a fluid (e.g. blood or water) can filtrate through a porous medium (e.g. the arterial wall or the soil), so that the Navier-Stokes equations must be coupled with Darcy's (or more complicated models, e.g., Forchheimer or Brinkmann equations) to describe the underlying physics (see, e.g., [5, 10, 16, 23]).

All these problems may be cast into the same common framework of heterogeneous domain decomposition method, which extends the classical domain decomposition theory whenever two (or more) kinds of boundary value problems, say $L_i u_i = f_i$, hold in subregions Ω_i of the computational domain Ω.

A major role is played by the compatibility conditions that the unknowns u_i must satisfy across the interface which separates the subdomains. In fact, the setting of proper coupling conditions is a crucial issue to model as closely as possible the real physical phenomena. For example, when coupling the Navier-Stokes and the Oseen equations, the compatibility conditions require the continuity of the velocities and of the normal stresses across the interface. However, it is worth mentioning that they might be much less intuitive and easy to handle than in the case just mentioned (see, e.g., [5, 25]).

In this paper, we will apply the heterogeneous domain decomposition paradigm to a fluid-structure interaction problem arising in hemodynamics for modeling blood flows in large arteries. To preserve stability one should solve exactly the fluid-structure coupling, e.g. by Newton methods [9, 13] or fixed-point algorithms [2]. A Newton method with exact Jacobian has been investigated both mathematically and numerically in [9]. Segregated solvers yielding a single fluid-structure interaction in each time step do not preserve stability and may produce blow-up when the density of the structure stays below a critical threshold. On the other hand, to relax the computational complexity of fixed-point or Newton methods several inexact solution strategies can be adopted.

The Jacobian matrix can be simplified by dropping the cross block expressing the sensitivity of the fluid state to solid motion, or by replacing it by a simpler term that models *added-mass* effect (see [1, 12]). Alternative inexact solvers exploit the analogy of the fluid structure coupled problem with heterogeneous domain decomposition problems.

This approach was first presented in [24, 19] for a Stokes-linearized shell coupling and later studied also in [18], where the whole problem was first reformulated as an interface equation. In this paper we further pursue this approach. Iterative substructuring methods, typical of the domain decomposition approach, are used to solve the interface problem, exploiting the classical Dirichlet-Neumann, the Neumann-Neumann, or more sophisticated scaling (preconditioning) techniques.

After describing a precise setting of the problem (Sect. 2), we shall define the associated interface equation (Sect. 3) and illustrate possible iterative methods to solve it (Sect. 4). Finally, some numerical results will be presented (Sect. 5).

2 Problem setting

To describe the evolution of the fluid and the structure domains in time, we adopt the ALE (Arbitrary Lagrangian Eulerian) formulation for the fluid (see [6, 14]) and a purely Lagrangian framework for the structure. We denote by $\Omega(t)$ the moving domain composed of the deformable structure $\Omega^{\mathrm{s}}(t)$ and the fluid subdomain $\Omega^{\mathrm{f}}(t)$. If we denote by $\boldsymbol{d}^{\mathrm{s}}(x_0,t)$ the displacement of the solid at a time t, we can define the following mapping: $\forall t$, $\Omega_0^{\mathrm{s}} \to \Omega^{\mathrm{s}}(t)$,

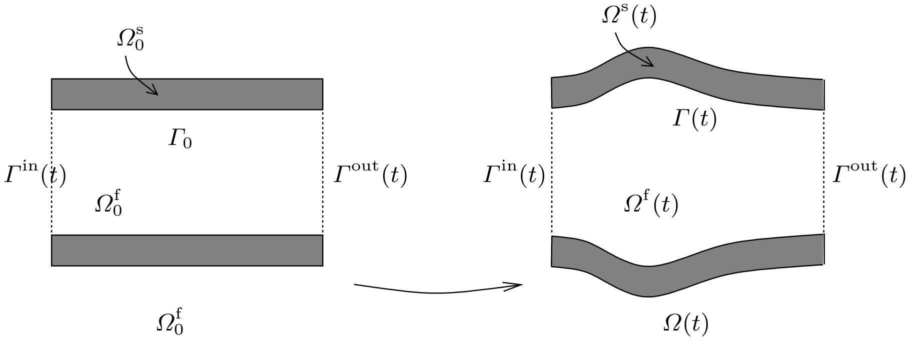


Fig. 1. ALE mapping

$$x_0 \to \boldsymbol{x}_t^{\mathrm{s}}(x_0) = x_0 + \boldsymbol{d}^{\mathrm{s}}(x_0,t), \quad x_0 \in \Omega_0^{\mathrm{s}}. \tag{1}$$

Likewise, for the fluid domain: $\forall t$, $\Omega_0^{\mathrm{f}} \to \Omega^{\mathrm{f}}(t)$,

$$x_0 \to \boldsymbol{x}_t^{\mathrm{f}}(x_0) = x_0 + \boldsymbol{d}^{\mathrm{f}}(x_0,t), \quad x_0 \in \Omega_0^{\mathrm{f}}. \tag{2}$$

The fluid domain displacement $\boldsymbol{d}^{\mathrm{f}}$ can be defined as a suitable extension of the solid interface displacement $\boldsymbol{d}^{\mathrm{s}}_{|\Gamma_0}$: $\boldsymbol{d}^{\mathrm{f}} = Ext(\boldsymbol{d}^{\mathrm{s}}_{|\Gamma_0})$ (see, e.g., [20]).

We assume the fluid to be Newtonian, viscous and incompressible, so that its behavior is described by the following fluid state problem: given the boundary data $\boldsymbol{u}_{\mathrm{in}}$, $\boldsymbol{g}_{\mathrm{f}}$, and the forcing term $\boldsymbol{f}_{\mathrm{f}}$, and denoting $\boldsymbol{w}^{\mathrm{f}} = \partial_t \boldsymbol{d}^{\mathrm{f}}$ the rate of change of the fluid domain, the velocity field $\boldsymbol{u}$ and the pressure p satisfy the momentum and continuity equations:

$$\begin{aligned} \rho_{\mathrm{f}} \left(\left.\frac{\partial \boldsymbol{u}}{\partial t}\right|_{x_0} + (\boldsymbol{u} - \boldsymbol{w}^{\mathrm{f}}) \cdot \boldsymbol{\nabla} \boldsymbol{u} \right) - \operatorname{div}[\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u},p)] &= \boldsymbol{f}_{\mathrm{f}} \ \text{ in } \Omega^{\mathrm{f}}(t), \\ \operatorname{div} \boldsymbol{u} &= 0 \quad \text{in } \Omega^{\mathrm{f}}(t), \\ \boldsymbol{u} = \boldsymbol{u}_{\mathrm{in}} \ \text{ on } \Gamma^{\mathrm{in}}(t), \quad \boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u},p) \cdot \boldsymbol{n}_{\mathrm{f}} &= \boldsymbol{g}_{\mathrm{f}} \ \text{ on } \Gamma^{\mathrm{out}}(t). \end{aligned} \tag{3}$$

We denote by ρ_{f} the fluid density, μ the fluid viscosity, $\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u},p) = -pId + 2\mu\boldsymbol{\epsilon}(\boldsymbol{u})$ the Cauchy stress tensor, Id is the identity matrix, $\boldsymbol{\epsilon}(\boldsymbol{u}) = (\boldsymbol{\nabla}\boldsymbol{u} +$

$(\nabla \boldsymbol{u})^T)/2$ the strain rate tensor. Note that (3) does not define univocally a solution $(\boldsymbol{u}, p)$ as no boundary data are prescribed on the interface $\Gamma(t)$.

Similarly, for given vector functions $\boldsymbol{g}_{\mathrm{s}}$, $\boldsymbol{f}_{\mathrm{s}}$, we consider the following structure problem whose solution is d^{s}:

$$\begin{aligned} \rho_{\mathrm{s}} \frac{\partial^2 \boldsymbol{d}^{\mathrm{s}}}{\partial t^2} - \mathrm{div}_{|x_0}(\boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{d}^{\mathrm{s}})) = \boldsymbol{f}_{\mathrm{s}} \ \text{ in } \Omega_0^{\mathrm{s}}, \\ \boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{d}^{\mathrm{s}}) \cdot \boldsymbol{n}_{\mathrm{s}} = \boldsymbol{g}_{\mathrm{s}} \ \text{ on } \partial\Omega_0^{\mathrm{s}} \setminus \Gamma_0, \end{aligned} \tag{4}$$

where $\boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{d}^{\mathrm{s}})$ is the first Piola–Kirchoff stress tensor. We remark that boundary values on Γ_0 for (4) are missing.

When coupling the two problems together, the "missing" boundary conditions are indeed supplemented by suitable matching conditions on the reference interface Γ_0. If $\lambda = \lambda(t)$ denotes the displacement of the interface, at any time t the coupling conditions on the reference interface Γ_0 are

$$\begin{aligned} \boldsymbol{x}_t^{\mathrm{s}} = x_0 + \lambda = \boldsymbol{x}_t^{\mathrm{f}}, \qquad \boldsymbol{u} \circ \boldsymbol{x}_t^{\mathrm{f}} = \frac{\partial \lambda}{\partial t}, \\ (\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u}, p) \cdot \boldsymbol{n}_{\mathrm{f}}) \circ \boldsymbol{x}_t^{\mathrm{f}} = -\boldsymbol{\sigma}_{\mathrm{s}}(\boldsymbol{d}^{\mathrm{s}}) \cdot \boldsymbol{n}_{\mathrm{s}}, \end{aligned} \tag{5}$$

imposing the matching of the interface displacements of the fluid and solid subdomains, the continuity of the velocities and of the normal stresses.

3 The interface equations associated to problem (3)-(5)

We consider the coupled problem at a given time $t = t^{n+1} = (n+1)\delta t$, δt being the discrete time-step.

According to the interface conditions (5), we can envisage two possible natural choices for the interface variable: either we consider the displacement λ of the fluid-structure interface, or the normal stress exerted on it. In the following, we shall focus our attention on the case of the interface variable as the displacement; the "dual" approach using the normal stress was presented in [4] for a simple linear problem.

Thus, we define the fluid and structure interface operators as follows.

S_{f} is the *Dirichlet-to-Neumann map* in $\Omega^{\mathrm{f}}(t)$:

$$S_{\mathrm{f}} : H^{1/2}(\Gamma_0) \to H^{-1/2}(\Gamma_0), \qquad \lambda \to \sigma_{\mathrm{f}}(\lambda),$$

that operates between the trace space of displacements on the interface Γ_0 and the dual space of the normal stresses exerted on Γ_0 by the fluid. Computing $S_{\mathrm{f}}(\lambda)$ involves the extension of the interface displacement to the whole fluid domain (in order to compute the ALE velocity), the solution of a Navier-Stokes problem in $\Omega^{\mathrm{f}}(t)$ with the Dirichlet boundary condition on the interface $\boldsymbol{u}_{|\Gamma(t)} \circ \boldsymbol{x}_t^{\mathrm{f}} = (\lambda - \mathrm{d}^{\mathrm{s},n}_{|\Gamma_0})/\delta t$, and then to recover the normal stress $\sigma_{\mathrm{f}} = (\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u}, p) \cdot \boldsymbol{n}_{\mathrm{f}})_{|\Gamma(t)} \circ \boldsymbol{x}_t^{\mathrm{f}}$ as a residual of the Navier-Stokes equations on the interface.

Moreover, we consider the *Dirichlet-to-Neumann map* $S_{\rm s}$ in $\Omega_0^{\rm s}$:

$$S_{\rm s} : H^{1/2}(\Gamma_0) \to H^{-1/2}(\Gamma_0), \qquad \lambda \to \sigma_{\rm s}(\lambda),$$

that operates between the space of displacements on the interface Γ_0 and the space of the normal stresses exerted by the structure on Γ_0. Computing $S_{\rm s}(\lambda)$ corresponds to solving a structure problem in $\Omega_0^{\rm s}$ with Dirichlet boundary condition $\mathrm{d}^{\rm s}_{|\Gamma_0} = \lambda$ on Γ_0, and then to recover the normal stress $\sigma_{\rm s} = \boldsymbol{\sigma}_{\rm s}(\mathrm{d}^{\rm s})\cdot\boldsymbol{n}_{\rm s}$ on the interface, again as a residual.

The definitions of $S_{\rm f}$ and $S_{\rm s}$ involve also the boundary and forcing terms, because of the nonlinearity of the problem at hand.

Then, the coupled fluid-structure problem can be expressed in terms of the solution λ of the following nonlinear Steklov-Poincaré interface problem:

$$\text{find } \lambda \in H^{1/2}(\Gamma_0): \quad S_{\rm f}(\lambda) + S_{\rm s}(\lambda) = 0. \tag{6}$$

Remark 1. In the case of a linear coupled Stokes-shell model, Mouro [19] has given a precise characterization of these interface operators and shown that they are selfadjoint and positive.

The inverse operator $S_{\rm s}^{-1}$ is a Neumann-to-Dirichlet map that for any given normal stress σ on Γ_0 associates the interface displacement $\lambda(t^{n+1}) = \mathrm{d}^{{\rm s},n+1}$ by solving a structure problem with the Neumann boundary condition $\boldsymbol{\sigma}_{\rm s}(\mathrm{d}^s)\cdot \boldsymbol{n}_{\rm s} = \sigma$ on Γ_0 and then computing the restriction on Γ_0 of the displacement of the structure domain.

For nonlinear structural models (i.e. $\boldsymbol{\sigma}_{\rm s}(\mathrm{d}^s)$ is a nonlinear constitutive law in (4), see, e.g., [17]), we will need the *tangent* operator $S_{\rm s}'$

$$S_{\rm s}'(\bar\lambda)\delta\lambda = \lim_{h\to 0} \frac{S_{\rm s}(\bar\lambda + h\delta\lambda) - S_{\rm s}(\bar\lambda)}{h}, \qquad \forall \bar\lambda, \delta\lambda \in H^{1/2}(\Gamma_0).$$

Its inverse $(S_{\rm s}')^{-1}$ is a Neumann-to-Dirichlet map that for any given variation of the normal stress $\delta\sigma$ on Γ_0 associates the corresponding variation of the displacement $\delta\lambda$ of the interface by solving a linearized structure problem with boundary condition $\boldsymbol{\sigma}_{\rm s}(\boldsymbol{d}^{\rm s})\cdot\boldsymbol{n}_{\rm s} = \delta\sigma$ on Γ_0. Similarly, we define $S_{\rm f}'$ by

$$S_{\rm f}'(\bar\lambda)\delta\lambda = \lim_{h\to 0} \frac{S_{\rm f}(\bar\lambda + h\delta\lambda) - S_{\rm f}(\bar\lambda)}{h}, \qquad \forall \bar\lambda, \delta\lambda \in H^{1/2}(\Gamma_0).$$

This is a Dirichlet-to-Neumann map that for any variation of the interface displacement $\delta\lambda$ computes the corresponding variation of the normal stress $\delta\sigma$ on Γ_0 through the solution of linearized Navier-Stokes equations. To compute $S_{\rm f}'(\lambda)\delta\lambda$ see, e.g, [9].

The computation of the inverse operator $S_{\rm f}'(\lambda)^{-1}$ can be simplified by neglecting the shape derivatives. We then obtain the Oseen equations in the fixed configuration defined by λ that we computed while evaluating $S_{\rm f}(\lambda)$. $S_{\rm f}'(\lambda)^{-1}$ is a Neumann-to-Dirichlet map that for any given variation of the normal stress $\delta\sigma$ on Γ_0 computes the corresponding displacement $\delta\lambda$ of the interface through the solution of linearized Navier-Stokes equations with the boundary condition $(\boldsymbol{\sigma}_{\mathrm{f}}(\boldsymbol{u},p)\cdot\boldsymbol{n}_{\mathrm{f}})\circ\boldsymbol{x}^{\mathrm{f}}=\sigma$ on Γ_0.

Other possible formulations for the interface equation can be given:

$$\text{find } \lambda \text{ such that } S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda))=\lambda \text{ on } \Gamma_0, \tag{7}$$

or equivalently

$$\text{find } \lambda \text{ such that } S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda))-\lambda=0 \text{ on } \Gamma_0. \tag{8}$$

These are common formulations in fluid-structure interaction problems, but it is worth pointing out that here the unknown λ is the displacement of the sole interface, whereas classically the displacement of the whole solid domain is considered (see, e.g., [20, 9]).

4 Iterative methods for problems (6)-(8)

We consider the preconditioned Richardson method to solve the Steklov-Poincaré interface problem (6): given λ^0, for $k\geq 0$, solve

$$P_k\left(\lambda^{k+1}-\lambda^k\right)=\omega^k\left(-S_{\mathrm{f}}(\lambda^k)-S_{\mathrm{s}}(\lambda^k)\right). \tag{9}$$

The scaling operator P_k maps the space $H^{1/2}(\Gamma_0)$ of the interface variable onto the space $H^{-1/2}(\Gamma_0)$ of normal stresses, and may depend on the iterate λ^k or, more generally, on the iteration step k. The acceleration parameter ω^k can be computed via the Aitken technique (see [4]) or by line search (see [22]).

At each step k, (9) requires the solution, separately, of the fluid and the structure problems and then to apply a scaling operator. Precisely,

1. apply S_{f} to λ^k, i.e., compute the extension of λ^k to the entire fluid domain to obtain the ALE velocity, and solve the fluid problem in $\Omega^{\mathrm{f}}(t)$ with boundary condition $\boldsymbol{u}_{|\Gamma(t)}\circ\boldsymbol{x}_t^{\mathrm{f}}=(\lambda-\mathrm{d}^{\mathrm{s},n}_{|\Gamma_0})/\delta t$ on Γ_0; then, recover the normal stress σ_{f}^k on the interface;
2. apply S_{s} to λ^k, i.e., solve the structure problem with boundary condition $\mathrm{d}^{\mathrm{s},k}_{|\Gamma(t)}=\lambda^k$ on $\Gamma(t)$ and compute the normal stress σ_{s}^k;
3. apply P_k^{-1} to the total stress $\sigma^k=\sigma_{\mathrm{f}}^k+\sigma_{\mathrm{s}}^k$ on the interface.

Note that steps 1 and 2 can be performed in parallel. The crucial issue is how to choose the scaling operator (more precisely, a preconditioner in the finite dimensional case) in order for the iterative method to converge as quickly as possible.

We define a generic linear operator (more precisely, its inverse):

$$P_k^{-1}=\alpha_{\mathrm{f}}^k\,S_{\mathrm{f}}'(\lambda^k)^{-1}+\alpha_{\mathrm{s}}^k\,S_{\mathrm{s}}'(\lambda^k)^{-1}, \tag{10}$$

for two given scalars α_{f}^k and α_{s}^k, and we retrieve the following operators:

$$\textit{Dirichlet-Neumann (DN):}\ P_k = P_{DN} = S_{\mathrm{s}}'(\lambda^k),\ \text{for } \alpha_{\mathrm{f}}^k = 0,\ \alpha_{\mathrm{s}}^k = 1, \tag{11}$$

$$\textit{Neumann-Dirichlet (ND):}\ P_k = P_{ND} = S_{\mathrm{f}}'(\lambda^k),\ \text{for } \alpha_{\mathrm{f}}^k = 1,\ \alpha_{\mathrm{s}}^k = 0, \tag{12}$$

$$\textit{Neumann-Neumann (NN):}\ P_k = P_{NN} \text{ with } \alpha_{\mathrm{f}}^k + \alpha_{\mathrm{s}}^k = 1,\ \alpha_{\mathrm{f}}^k, \alpha_{\mathrm{s}}^k \neq 0. \tag{13}$$

If the structure is linear, the computational effort of a Richardson step in the DN case may be reduced to the solution of only one fluid Dirichlet problem and one structure Neumann problem.

The parameters α_{f}^k, α_{s}^k and ω^k can be chosen dynamically using a generalized Aitken technique (see [3, 4]).

Should we consider the scaling operator

$$P_k = S_{\mathrm{f}}'(\lambda^k) + S_{\mathrm{s}}'(\lambda^k), \tag{14}$$

then, we would retrieve the genuine Newton algorithm applied to the Steklov-Poincaré problem (6). Note that in order to perform the scaling step 3 in the Richardson algorithm, one must use a (preconditioned) iterative method (e.g., GMRES) and may approximate the tangent problems to accelerate the computations. Thus, using the scaling operator (14) we obtain a *domain decomposition-Newton (DD-Newton)* method; more precisely, given a solid state displacement λ^k, for $k \geq 0$, the algorithm reads

1. solve the fluid and the structure subproblems separately, as for the Richardson method, to get σ^k;
2. solve the following linear system via GMRES to compute μ^k:
$$\left[S_{\mathrm{f}}'(\lambda^k) + S_{\mathrm{s}}'(\lambda^k)\right] \mu^k = -(S_{\mathrm{f}}(\lambda^k) + S_{\mathrm{s}}(\lambda^k)) \tag{15}$$
3. update the displacement: $\lambda^{k+1} = \lambda^k + \omega^k \mu^k$.

The GMRES solver should in turn be preconditioned in order to accelerate its convergence rate. To this aim, one can use one of the previously defined scaling operators. In our numerical tests, we have considered the DN operator $S_{\mathrm{s}}'(\lambda)$, so that the preconditioned matrix of the GMRES method becomes:

$$[S_{\mathrm{s}}'(\lambda^k)]^{-1} \cdot [S_{\mathrm{f}}'(\lambda_k) + S_{\mathrm{s}}'(\lambda_k)]. \tag{16}$$

Let us briefly recall the Newton method for problem (8) in order to compare it with the previous domain decomposition approach. For a more complete discussion we refer to [4].

Let $J(\lambda)$ denote the Jacobian of $S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda))$ in λ. Given λ^0, for $k \geq 0$:

$$\begin{array}{ll} \text{solve} & (J(\lambda^k) - Id)\mu^k = -(S_{\mathrm{s}}^{-1}(-S_{\mathrm{f}}(\lambda^k)) - \lambda^k), \\ \text{update} & \lambda^{k+1} = \lambda^k + \omega^k \mu^k. \end{array} \tag{17}$$

The parameter ω^k can be computed, e.g., by a line search technique (see [22]). Note that the Jacobian in λ^k has the following expression:

$$J(\lambda^k) = -\left[S_{\rm s}'\left(S_{\rm s}^{-1}(-S_{\rm f}(\lambda^k))\right)\right]^{-1}\cdot S_{\rm f}'(\lambda^k) = -\left[S_{\rm s}'\left(\bar{\lambda}^k\right)\right]^{-1}\cdot S_{\rm f}'(\lambda^k). \quad (18)$$

The solution of the linear system (17) can be obtained by using an iterative matrix-free method such as GMRES.

In general, the Newton method applied to (8) and to the Steklov-Poincaré formulation (6) are not equivalent. However, in the case of a linear structure, they actually are (to see this, left multiply both hand sides of (15) by $S_{\rm s}^{-1}$, exploit $S_{\rm s}'(\lambda^k) = S_{\rm s}$ and compare (16) with (17)).

We remark that while the computation of $\left[S_{\rm s}'\left(\bar{\lambda}^k\right)\right]^{-1}\cdot\delta\sigma$ (for any given $\delta\sigma$) does only require the derivative with respect to the state variable at the interface, the computation of $S_{\rm f}'(\lambda^k)\cdot\delta\lambda$ is nontrivial since it also requires shape derivatives, as a variation in λ determines a variation of the fluid domain.

We finally remark that in the classical Newton method, the fluid and structure problems must be solved separately and sequentially, while the domain decomposition formulation allows us to set up parallel algorithms to solve the Steklov-Poincaré equation (6).

5 Numerical results

In this section, we present some numerical results which compare the domain decomposition methods to the classical fixed point and Newton algorithms, and illustrate their behavior with respect to the grid size h and the time step δt.

For the domain decomposition algorithms, we consider the DN preconditioner (11), and the NN preconditioner (13) in which $S_{\rm f}'$ is linearized by neglecting the shape derivatives.

Finally, we consider the DD-Newton method (14). The fluid tangent problem is considered as in [9] in its exact form. To solve (15), we apply the GMRES method possibly preconditioned by the operator DN (11).

Both problems (3) and (4) are discretized, and we adopt $\mathbb{P}_1$-bubble/$\mathbb{P}_1$ finite elements for the fluid and $\mathbb{P}_1$ elements for the structure. The simulations are performed on a dual 2.8 Ghz Pentium 4 Xeon with 3 GB of RAM.

We simulate a pressure wave in a straight cylinder of length 5 *cm* and radius 5 *mm* at rest. The structure of thickness 0.5 *mm* is linear and clamped at both the inlet and the outlet. The fluid viscosity is set to $\mu = 0.03$ *poise*, the densities to $\rho_{\rm f} = 1\ g/cm^3$ and $\rho_{\rm s} = 1.2\ g/cm^3$. We impose zero body forces and homogeneous Dirichlet boundary conditions on $\partial\Omega_0^{\rm s}\setminus\Gamma_0$. The fluid and the structure are initially at rest and a pressure (a normal stress, actually) of $1.3332\cdot 10^4\ dynes/cm^2$ is imposed on the inlet for $3\cdot 10^{-3}\ s$. We consider two computational meshes: a coarse one with 1050 nodes (4680 elements) for the fluid and 1260 nodes (4800 elements) for the solid, and a finer mesh with 2860 nodes (14100 elements) for the fluid and 2340 nodes (9000 elements) for the solid.

A comparison between the fixed point iterations for problem (7) and Richardson iterations (9) (with DN and NN preconditioners) on problem (6) is shown in table 1 for two time steps and for the coarse and the fine mesh. In this table, "FS eval" stands for the average number of evaluations per time step of either (7) or (9), while "FS' eval" represents the average number of evaluations of the corresponding linearized system per time step (that is (10) for DN, ND or NN preconditioners, (16) for the DD-Newton method (15), and (18) for the classical Newton method (17)). We can see that, using the preconditioned Richardson method (9), fewer FS evaluations than with the classical fixed point algorithm are needed. However, the computational time of the domain decomposition formulation is slightly higher than that of the fixed point formulation. The reason is that the domain decomposition formulation requires solving, at each iteration, the fluid and the structure subproblems, as well as the associated tangent problems, while the latter are indeed skipped by the fixed point procedure. Furthermore, since the operator for the structure is linear, the two approaches are very similar and since our research code is sequential, the parallel structure of the Steklov-Poincaré formulation (6) is not capitalized.

Moreover, we notice that using the NN preconditioner the number of iterations required for the convergence with respect to both parameters h and δt, does not vary appreciably.

The same table shows also the results obtained using the Newton and DD-Newton methods. The Jacobian matrices (14) and (18) have been computed exactly (see [9]) and inverted by GMRES. The number of iterations of Newton and DD-Newton is equivalent, but the inversion of the Jacobian in DD-Newton ("FS' eval") needs more GMRES iterations, a number which depends on h and δt. However, preconditioning GMRES by DN reduces the iteration numbers to the same as in Newton, and the CPU times are then quite similar. As before, the reasons reside in the linearity of the structure model and in the fact that our code is sequential.

Further improvements may be obtained resorting to more sophisticated preconditioners for the Jacobian system, derived either from the classical domain decomposition theory or from lower dimensional models (in a multiscale approach, see [21]).

We now simulate a pressure wave in the carotid bifurcation using the same fluid and structure characteristics as before. We solve the coupling using our DD-Newton algorithm with DN preconditioner for the GMRES inner iterations. The mesh that we have used was computed using an original realistic geometry first proposed in [15].

The fluid and the structure are initially at rest and a pressure of $1.3332 \cdot 10^4$ $dynes/cm^2$ is set at the inlet for a time of $3 \cdot 10^{-3}$ s. The average inflow diameter is 0.67 cm, the time step used is $\delta t = 1e - 04$ and the total number of iterations is 200. Figure 2 displays the solution computed at two different time steps. Table 2 shows the comparison between the classical Newton algorithm and our DD-Newton algorithm preconditioned by DN. Like in

Table 1. Comparison of the number of sub-iterations and computational time for the fixed point, and domain decomposition based algorithms for the coarse mesh (left) and fine mesh (right)

$\delta t = 0.001$			
Method	FS eval	FS’ eval	CPU time
Fixed point	19.8	0	1h16’
DN	19.8	19.8	1h17’
NN	17.9	17.9	1h42’
Newton	3	12	0h56’
DD-Newton	3	24	1h30’
DD-Newton DN	3	12	0h58’

$\delta t = 0.0005$			
Method	FS eval	FS’ eval	CPU time
Fixed point	32.1	0	3h27’
DN	29.2	29.2	3h50’
NN	22	22	4h20’
Newton	3	17	1h55’
DD-Newton	3	29	3h30’
DD-Newton DN	3	17	2h10’

$\delta t = 0.0001$			
Method	FS eval	FS’ eval	CPU time
Newton	3	19	11h41’
DD-Newton	3	35	16h21’
DD-Newton DN	3	19	12h39’

$\delta t = 0.001$			
Method	FS eval	FS’ eval	CPU time
Fixed point	19.9	0	4h28’
DN	19.5	19.5	4h40’
NN	17.7	17.7	6h12’
Newton	3	12	3h39’
DD-Newton	3	30	4h56’
DD-Newton DN	3	12	3h45’

$\delta t = 0.0005$			
Method	FS eval	FS’ eval	CPU time
Fixed point	33	0	12h40’
DN	29.6	29.6	12h50’
NN	22.1	22.1	15h44’
Newton	3	14	8h31’
DD-Newton	3	35	10h50’
DD-Newton DN	3	14	8h40’

$\delta t = 0.0001$			
Method	FS eval	FS’ eval	CPU time
Newton	3	19	26h40’
DD-Newton	3	37	40h26’
DD-Newton DN	3	19	27h01’

the previous test, “FS eval” and “FS’ eval” represent respectively the average number of fluid/structure evaluations and the average number of linearized fluid/structure evaluations. As expected, both methods behave in the same way with respect to the number of operator evaluations. The total computation times are also in very good agreement for the two largest time step.

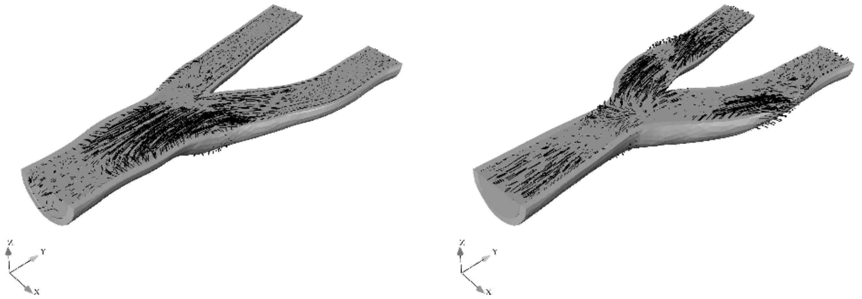


Fig. 2. Structure deformation and fluid velocity at $t = 0.005$ s (left) and $t = 0.008$ s (right)

Table 2. Convergence comparison of the computational time for the exact Newton and DD-Newton methods (case of carotid bifurcation)

	$\delta t = 0.001$			$\delta t = 0.0005$			$\delta t = 0.0001$		
Method	FS eval	FS' eval	CPU time	FS eval	FS' eval	CPU time	FS eval	FS' eval	CPU time
Newton	3	7.5	8h51'	3	10	19h41'	3	19	125h20'
DD-Newton DN	3	7.5	8h12'	3	10	19h33'	3	19	131h08'

Acknowledgement. This research has been supported by the Swiss National Science Foundation (project 20-101-800) and by the INDAM project "Integrazione di sistemi complessi in biomedicina: modelli, simulazioni, rappresentazioni".

References

1. P. Causin, J.-F. Gerbeau, and F. Nobile, *Added-mass effect in the design of partitioned algorithms for fluid-structure problems*, Comput. Methods Appl. Mech. Engrg., 194 (2005), pp. 4506–4527.
2. M. Cervera, R. Codina, and M. Galindo, *On the computational efficiency and implementation of block-iterative algorithms for nonlinear coupled problems*, Engrg. Comput., 13 (1996), pp. 4–30.
3. S. Deparis, *Numerical Analysis of Axisymmetric Flows and Methods for Fluid-Structure Interaction Arising in Blood Flow Simulation*, PhD thesis, École Polytechnique Fédérale de Lausanne, 2004.
4. S. Deparis, M. Discacciati, and A. Quarteroni, *A domain decomposition framework for fluid-structure interaction problems*, in Proceedings of the Third International Conference on Computational Fluid Dynamics, C. Groth and D. W. Zingg, eds., Springer, May 2006.
5. M. Discacciati, *Domain Decomposition Methods for the Coupling of Surface and Groundwater Flows*, PhD thesis, École Polytechnique Fédérale de Lausanne, 2004.
6. J. Donea, *Arbitrary Lagrangian Eulerian finite element methods*, in Computational Methods for Transient Analysis, vol. 1 of Computational Methods in Mechanics, Amsterdam, North-Holland, 1983, pp. 473–516.
7. L. Fatone, P. Gervasio, and A. Quarteroni, *Multimodels for incompressible flows*, J. Math. Fluid Mech., 2 (2000), pp. 126–150.
8. ———, *Multimodels for incompressible flows: iterative solutions for the Navier-Stokes/Oseen coupling*, Math. Model. Numer. Anal., 35 (2001), pp. 549–574.
9. M. A. Fernández and M. Moubachir, *A Newton method using exact Jacobians for solving fluid-structure coupling*, Comput. & Structures, 83 (2005), pp. 127–142.
10. J. C. Galvis and M. Sarkis, *Inf-sup for coupling Stokes-Darcy*, in Proceedings of the XXV Iberian Latin American Congress in Computational Methods in Engineering, A. L. et al., ed., Universidade Federal de Pernambuco, 2004.
11. F. Gastaldi, A. Quarteroni, and G. S. Landriani, *On the coupling of two-dimensional hyperbolic and elliptic equations: analytical and numerical approach*, in Third International Symposium on Domain Decomposition Methods

for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989, T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM, pp. 22–63.

12. J.-F. Gerbeau and M. Vidrascu, *A quasi-Newton algorithm based on a reduced model for fluid-structure interaction problems in blood flows*, Math. Model. Numer. Anal., 37 (2003), pp. 631–647.
13. M. Heil, *An efficient solver for the fully coupled solution of large-displacement fluid-structure interaction problems*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 1–23.
14. T. J. Hughes, W. K. Liu, and T. K. Zimmermann, *Lagrangian-Eulerian finite element formulation for incompressible flows*, Comput. Methods Appl. Mech. Engrg., 29 (1981), pp. 329–349.
15. G. Karner, K. Perktold, M. Hofer, and D. Liepsch, *Flow characteristics in an anatomically realistic compliant carotid artery bifurcation model*, Comput. Methods Biomech. Biomed. Engrg., 2 (1999), pp. 171–185.
16. W. J. Layton, F. Schieweck, and I. Yotov, *Coupling fluid flow with porous media flow*, SIAM J. Num. Anal., 40 (2003), pp. 2195–2218.
17. J. E. Marsden and T. J. Hughes, *Mathematical Foundations of Elasticity*, Dover Publications, Inc., New York, 1994. Reprint.
18. D. P. Mok and W. A. Wall, *Partitioned analysis schemes for the transient interaction of incompressible flows and nonlinear flexible structures*, in Proceedings of the International Conference Trends in Computational Structural Mechanics, K. Schweizerhof and W. A. Wall, eds., K.U. Bletzinger, CIMNE, Barcelona, 2001.
19. J. Mouro, *Interactions Fluide Structure en Grands Déplacements. Résolution Numérique et Application aux Composants Hydrauliques Automobiles*, PhD thesis, École Polytechnique, Paris, September 1996.
20. F. Nobile, *Numerical Approximation of Fluid-Structure Interaction Problems with Application to Haemodynamics*, PhD thesis, École Polytechnique Fédérale de Lausanne, 2001.
21. A. Quarteroni and L. Formaggia, *Mathematical modelling and numerical simulation of the cardiovascular system*, in Modelling of Living Systems, P. G. Ciarlet and J. L. Lions, eds., vol. 12 of Handbook of Numerical Analysis, Elsevier, Amsterdam, 2004, pp. 3–127.
22. A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Texts in Applied Mathematics, Springer, New York, 2000.
23. A. Quarteroni, A. Veneziani, and P. Zunino, *A domain decomposition method for advection-diffusion processes with application to blood solutes*, SIAM J. Sci. Comput., 23 (2002), pp. 1959–1980.
24. P. L. Tallec and J. Mouro, *Fluid structure interaction with large structural displacements*, Comput. Methods Appl. Mech. Engrg., 190 (2001), pp. 3039–3067.
25. P. Zunino, *Iterative substructuring methods for advection-diffusion problems in heterogeneous media*, in Challenges in Scientific Computing–CISC 2002, vol. 35 of Lecture Notes in Computational Science and Engineering, Springer, 2003, pp. 184–210.

Preconditioning of Saddle Point Systems by Substructuring and a Penalty Approach

Clark R. Dohrmann[*]

Structural Dynamics Research Department, Sandia National Laboratories,
Albuquerque, NM 87185-0847, USA. crdohrm@sandia.gov

Summary. The focus of this paper is a penalty-based strategy for preconditioning elliptic saddle point systems. As the starting point, we consider the regularization approach of Axelsson in which a related linear system, differing only in the (2,2) block of the coefficient matrix, is introduced. By choosing this block to be negative definite, the dual unknowns of the related system can be eliminated resulting in a positive definite primal Schur complement. Rather than solving the Schur complement system exactly, an approximate solution is obtained using a substructuring preconditioner. The approximate primal solution together with the recovered dual solution then define the preconditioned residual for the original system.

The effectiveness of the overall strategy hinges on the preconditioner for the primal Schur complement. A condition ensuring real and positive eigenvalues of the preconditioned saddle point system is satisfied automatically in certain instances if a Balancing Domain Decomposition by Constraints (BDDC) preconditioner is used. Following an overview of BDDC, we show how its constraints can be chosen to ensure insensitivity to parameter choices in the (2,2) block for problems with a divergence constraint. Example saddle point problems are presented and comparisons made with other approaches.

1 Introduction

Consider the linear system

$$\begin{bmatrix} A & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} b \\ 0 \end{bmatrix} \tag{1}$$

arising from a finite element discretization of a saddle point problem. The matrix A is assumed to be symmetric and positive definite on the kernel of B. The matrix B is assumed to have full rank and C is assumed to be symmetric

[*] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the United States Department of Energy under Contract DE-AC04-94AL85000.

and positive semidefinite. The primal and dual vectors are denoted by $u \in \mathbb{R}^n$ and $p \in \mathbb{R}^m$, respectively.

Several different preconditioners for (1) have been investigated. Many are based on preconditioning the dual Schur complement $C + BA^{-1}B^T$ by another matrix that is spectrally equivalent to the dual mass matrix. Examples include block diagonal preconditioners [18], block triangular preconditioners [10], and inexact Uzawa approaches [7]. Reformulation of the saddle point problem in (1) as a symmetric positive definite system was considered in [3] that permits an iterative solution using the conjugate gradient algorithm. Overlapping Schwarz preconditioners involving solutions of both local and coarse saddle point problems were investigated in [11]. More recently, substructuring preconditioners based on balancing Neumann-Neumann methods [16, 9, 8] and FETI-DP [12] were studied.

The approach presented here builds on the basic idea of preconditioning indefinite problems using a regularization approach [1]. Preconditioning based on regularization is motivated by the observation that the solution of a penalized problem is often close to that of the original constrained problem. Results are presented that extend [1] to cases where the penalized primal Schur complement $S_A = A + B^T\tilde{C}^{-1}B$ is preconditioned rather than factored directly. Here, $\tilde{C}$ is a symmetric positive definite penalty counterpart of C in (1).

The preconditioner for (1) is most readily applied to discretizations employing discontinuous interpolation of the dual variable. In such cases the dual variable can be eliminated at the element level and S_A has the same sparsity structure as A. Not surprisingly, the effectiveness of the approach hinges on the preconditioner for S_A.

Significant portions of this paper are based on two recent technical reports [6, 5]. Material taken directly from [6] includes a statement, without proof, of its main result in Section 2. New material related to [6] includes additional theory for the special case of $C = 0$ in Section 2, and an extension of numerical results of the cited reference in Section 5. An overview of the BDDC preconditioner is provided in Section 3. In Section 4 we show how to choose the constraints in BDDC to accommodate problems with a divergence constraint. Numerical examples in Section 5 confirm the theory and demonstrate the excellent performance of the preconditioner. Comparisons are also made with block diagonal and block triangular preconditioners for saddle point systems.

2 Penalty Preconditioner

The penalized primal Schur complement S_A is defined as

$$S_A = A + B^T\tilde{C}^{-1}B$$

where $\tilde{C}$ is symmetric and positive definite. Since A is assumed to be positive definite on the kernel of B, it follows that S_A is positive definite. We consider a preconditioner $\mathcal{M}$ of the form

$$\mathcal{M} = \begin{bmatrix} I & B^T\tilde{C}^{-1} \\ 0 & -I \end{bmatrix} \begin{bmatrix} \hat{S}_A & 0 \\ 0 & -\tilde{C} \end{bmatrix} \begin{bmatrix} I & 0 \\ \tilde{C}^{-1}B & -I \end{bmatrix}$$

where $\hat{S}_A$ is a preconditioner for S_A. The action of the preconditioner on a vector r (with primal and dual subvectors r_u and r_p) is

$$\begin{bmatrix} z_u \\ z_p \end{bmatrix} = \begin{bmatrix} I & 0 \\ \tilde{C}^{-1}B & -I \end{bmatrix} \begin{bmatrix} \hat{S}_A^{-1} & 0 \\ 0 & -\tilde{C}^{-1} \end{bmatrix} \begin{bmatrix} I & B^T\tilde{C}^{-1} \\ 0 & -I \end{bmatrix} \begin{bmatrix} r_u \\ r_p \end{bmatrix}$$

leading to the two step application of $\mathcal{M}^{-1}r$ as

1. Solve $\hat{S}_A z_u = r_u + B^T\tilde{C}^{-1}r_p$ for z_u,
2. Solve $\tilde{C}z_p = Bz_u - r_p$ for z_p.

Each application of the preconditioner requires two solves with $\tilde{C}$ and one solve with $\hat{S}_A$.

Consider the eigenvalues ν of the generalized eigenproblem

$$\mathcal{A}z = \nu\mathcal{M}z \tag{2}$$

where $\mathcal{A}$ is the coefficient matrix in (1). The following theorem is taken from [6].

Theorem 1. *If* $\alpha_1 > 1,\ 0 \le \beta_1 < \beta_2 < 1,\ \gamma_1 > 0,$ *and*

$$\alpha_1 x^T\hat{S}_A x \le x^T S_A x \le \alpha_2 x^T \hat{S}_A x \quad \forall x \in \mathbb{R}^n, \tag{3}$$

$$\beta_1 y^T\tilde{C}y \ \le y^T C y \le \ \beta_2 y^T\tilde{C}y \quad \forall y \in \mathbb{R}^m, \tag{4}$$

$$\gamma_1 y^T B\hat{S}_A^{-1}B^T y \ \le y^T\tilde{C}y \le \ \gamma_2 y^T B\hat{S}_A^{-1}B^T y \quad \forall y \in \mathbb{R}^m, \tag{5}$$

and

$$0 < y^T\tilde{C}y \quad \forall y \ne 0 \in \mathbb{R}^m,$$

then the eigenvalues of (2) are real and satisfy

$$\delta_1 \le \nu \le \delta_2$$

where

$$\delta_1 = \min\{\sigma_2(\alpha_1/\alpha_2), \beta_1 + \sigma_1(1-\beta_2)(\alpha_2\gamma_2)^{-1}\}$$

$$\delta_2 = \max\{2\alpha_2 - \sigma_2, \beta_2 + (1-\beta_1)(2-\sigma_1/\alpha_2)\gamma_1^{-1}\}$$

and σ_1, σ_2 *are arbitrary positive constants that satisfy* $\sigma_1 + \sigma_2 = 1$.

When the eigenvalues of (2) are real and positive, conjugate gradients can be used for the iterative solution of (1). Details are available in [6].

Notice in (3) that α_1 and α_2 depend on the preconditioner for S_A. In order to obtain bounds for γ_1 and γ_2 in (5), it proves useful to express A as

$$A = B^T A_1 B + B_\perp^T A_2 B_\perp + B^T A_3 B_\perp + B_\perp^T A_3^T B$$

where the columns of $B_\perp$ form an orthonormal basis for the null space of B and

$$A_1 = (BB^T)^{-1}BAB^T(BB^T)^{-1}, \quad A_2 = B_\perp AB_\perp^T, \quad A_3 = (BB^T)^{-1}BAB_\perp^T .$$

Using a similar expression for S_A^{-1} and the identity $S_A S_A^{-1} = I$ we obtain

$$BS_A^{-1}B^T = (\tilde{C}^{-1} + G)^{-1} \quad \text{where} \quad G = A_1 - A_3A_2^{-1}A_3^T = R^TR .$$

Notice that A_2 is nonsingular since A was assumed positive definite on the kernel of B. In addition, G is at least positive semidefinite since it is independent of $\tilde{C}$ and $BS_A^{-1}B$ is positive definite. Application of the Sherman-Morrison-Woodbury formula leads to

$$BS_A^{-1}B^T = \tilde{C} - \tilde{C}R^T(I + R\tilde{C}R^T)^{-1}R\tilde{C} . \tag{6}$$

We now consider the special case $C = 0$ and the parameterization $\tilde{C} = \zeta\bar{C}$. The positive scalar ζ is chosen so that

$$\zeta\|\bar{C}R^T(I + \zeta R\bar{C}R^T)^{-1}R\bar{C}\| < \epsilon\lambda_{\min}(\bar{C}) \tag{7}$$

where $\epsilon > 0$ and $\lambda_{\min}(\bar{C})$ is the smallest eigenvalue of $\bar{C}$. It then follows from (3), (6), and (7) that

$$(1/\alpha_2)y^TB\hat{S}_A^{-1}B^Ty \le y^T\tilde{C}y \le (1/\alpha_1)(1-\epsilon)^{-1}y^TBS_A^{-1}B^Ty \quad \forall y \in \mathbb{R}^m \tag{8}$$

Comparison of (5) and (8) reveals that

$$\gamma_1 \ge 1/\alpha_2 \quad \text{and} \quad \gamma_2 \le (1/\alpha_1)(1-\epsilon)^{-1} .$$

Notice from (4) for $C = 0$ that $\beta_1 = 0$ and β_2 can be chosen arbitrarily close to 0. The expressions for the eigenvalue bounds with σ_1 and σ_2 both chosen as 1/2 then simplify to

$$\delta_1 = (1-\epsilon)(\alpha_1/\alpha_2)/2, \quad \delta_2 = 2\alpha_2 - 1/2 .$$

For very small values of ϵ we see that the eigenvalue bounds depend only on the parameters α_1 and α_2 which are related to the preconditioner. This result is purely algebraic and does not involve any inf-sup constants. For α_1 and α_2 both near 1 we see that all eigenvalues are bounded between $(1-\epsilon)/2$ and 3/2. Numerical results in Section 5 suggest that these bounds could be made even tighter. In Section 4 we show how to choose the constraints of a BDDC preconditioner so that α_1 and α_2 are insensitive to mesh parameters and to values of ϵ near zero.

3 BDDC Preconditioner

A brief overview of the BDDC preconditioner is provided here for completeness. Additional details can be found in [4, 14, 15]. The domain of a finite element mesh is assumed to be decomposed into nonoverlapping substructures $\Omega_1, \ldots, \Omega_N$ so that each element is contained in exactly one substructure. The assembly of the substructure contributions to the linear system can be expressed as

$$\begin{bmatrix} A & B^T \\ B & -D \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \sum_{i=1}^{N} \begin{bmatrix} R_i^T & P_i^T \end{bmatrix} \begin{bmatrix} A_i & B_i^T \\ B_i & -D_i \end{bmatrix} \begin{bmatrix} R_i \\ P_i \end{bmatrix} \begin{bmatrix} u \\ p \end{bmatrix} = \begin{bmatrix} f \\ 0 \end{bmatrix} \tag{9}$$

where each row of R_i and P_i contains exactly one nonzero entry of unity. Throughout this section several subscripted R matrices with exactly one nonzero entry of unity in each row are used for bookkeeping purposes. For discontinuous pressure elements and compressible materials the matrices D and D_i are positive definite and block diagonal. Solving the second block of equations in (9) for p in terms of u and substituting the result back into the first block of equations leads to

$$Ku = f, \qquad p = D^{-1}Bu \tag{10}$$

where the displacement Schur complement K is given by

$$K = A + B^T D^{-1} B = \sum_{i=1}^{N} R_i^T K_i R_i$$

and

$$K_i = A_i + B_i^T D_i^{-1} B_i \, .$$

The coarse interpolation matrix Φ_i for Ω_i is obtained by solving the linear system

$$\begin{bmatrix} K_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} \Phi_i \\ \Lambda_i \end{bmatrix} = \begin{bmatrix} 0 \\ I \end{bmatrix} \tag{11}$$

where C_i is the constraint matrix for Ω_i and I is a suitably dimensioned identity matrix. A straightforward method to calculate Φ_i from (11) using solvers for sparse symmetric definite systems of equations is given in [4].

Each row of the constraint matrix C_i is associated with a specific coarse degree of freedom (dof). Moreover, each coarse dof is associated with a particular set of nodes in Ω_i that appear in at least one other substructure. Let S_i denote the set of all such nodes. The set S_i is first partitioned into disjoint node sets $\mathcal{M}_{i1}, \ldots, \mathcal{M}_{iM_i}$ via the following equivalence relation. Two nodes are related if the substructures containing the two nodes are identical. In other words, each node of S_i is contained in exactly one node set, and all nodes in a given node set are contained in exactly the same set of substructures. Additional node sets called corners are used in [4] to facilitate the numerical implementation. Each corner is obtained by removing a node from one of the node sets described above. For notational convenience, we refer to $\{\mathcal{M}_{ij}\}_{j=1}^{M_i}$ as the set of all disjoint node sets for Ω_i including corners. Rows of the constraint matrix C_i associated with node set $\mathcal{M}_{ij}$ are given by $R_{ijr}C_i$. Similarly, columns of C_i associated with node set $\mathcal{M}_{ij}$ are given by $C_i R_{ijc}^T$. In this study all node sets are used in the substructure constraint equations.

Let u_{ci} denote a vector of coarse dofs for Ω_i. The dimension of u_{ci} equals the number of rows in the constraint matrix C_i. The vector u_{ci} is related to the global vector of coarse dofs u_c by

$$u_{ci} = R_{ci} u_c \,.$$

The coarse stiffness matrix of Ω_i is defined as

$$K_{ci} = \Phi_i^T K_i \Phi_i$$

and the assembled coarse stiffness matrix K_c is given by

$$K_c = \sum_{i=1}^{N} R_{ci}^T K_{ci} R_{ci} \,.$$

Consistent with (9), the vector of substructure displacement dofs u_i are related to u by

$$u_i = R_i u \,.$$

Let u_{Ii} denote a vector containing all displacement dofs in Ω_i that are not shared with any other substructures. The vector u_{Ii} is related to u_i by

$$u_{Ii} = R_{Ii} u_i \,.$$

In order to distribute residuals to the substructures, it is necessary to define weights for each substructure dof. In this study, the diagonal substructure weight matrix W_i is defined as

$$W_i = R_{Ii}^T R_{Ii} + \sum_{j=1}^{M_i} \alpha_{ij} R_{ijc}^T R_{ijc}$$

where

$$\alpha_{ij} = \mathrm{trace}(R_{ijc} K_{ci} R_{ijc}^T) / \mathrm{trace}(R_{ijc} R_{ci} K_c R_{ci}^T R_{ijc}^T)$$

and trace denotes the sum of diagonal entries. Notice that the weights of all dofs in a node set are identical. The substructure weight matrices form a partition of unity in the sense that

$$\sum_{i=1}^{N} R_i^T W_i R_i = I \,.$$

Given a residual vector r associated with the iterative solution of (10a), the preconditioned residual is obtained using the following algorithm.

1. Calculate the coarse grid correction v_1,

$$v_1 = \sum_{i=1}^{N} R_i^T W_i \Phi_i R_{ci} K_c^{-1} r_c \quad \text{where} \quad r_c = \sum_{i=1}^{N} R_{ci}^T \Phi_i^T W_i R_i r \, .$$

2. Calculate the substructure correction v_2,

$$v_2 = \sum_{i=1}^{N} R_i^T W_i z_i \quad \text{where} \quad \begin{bmatrix} K_i & C_i^T \\ C_i & 0 \end{bmatrix} \begin{bmatrix} z_i \\ \lambda_i \end{bmatrix} = \begin{bmatrix} W_i R_i r \\ 0 \end{bmatrix} .$$

3. Calculate the static condensation correction v_3,

$$v_3 = \sum_{i=1}^{N} R_i^T R_{Ii}^T (R_{Ii} K_i R_{Ii}^T)^{-1} R_{Ii} R_i r_1 \quad \text{where} \quad r_1 = r - K(v_1 + v_2) \, .$$

4. Calculate the preconditioned residual $M^{-1} r = v_1 + v_2 + v_3$.

Residuals associated with displacement dofs in substructure interiors are removed prior to the first conjugate gradient iteration via a static condensation correction. These residuals then remain zero for all subsequent iterations.

4 BDDC Constraint Equations

In this section we show how to choose the constraint equations of BDDC so that it can be used effectively as a preconditioner for the primal Schur complement S_A. Recall that at the end of Section 2, the goal was to have a preconditioner that is insensitive to values of ϵ near zero. For problems with a divergence constraint like incompressible elasticity, this means that the performance of the preconditioner should not degrade as the norm of D in (9) approaches zero. Additional details and work related to this section can be found in [5] and [13].

The choice of constraints is guided by the goal to keep the volume change of each substructure relatively small in the presence of a divergence constraint. In particular, the volume change corresponding to a preconditioned residual should not be too large. Otherwise, the energy associated with the preconditioned residual will be excessively large and cause slow convergence of a Krylov iterative method.

Using the divergence theorem, the volume change of Ω_i resulting from u_i to first order is given by

$$\Delta V_i = \int_{\Omega_i} \nabla \cdot \mathbf{u} \, d\Omega = a_i^T u_i \tag{12}$$

where $\mathbf{u}$ is the finite element approximation of the displacement field. The vector a_i can be calculated in the same manner as the vector for a body force by summing element contributions to the divergence. All entries in a_i associated with nodes not on the boundary of Ω_i are zero. We note that a constraint of zero volume change for each substructure has been used in augmented versions of FETI algorithms for incompressible problems [19].

The nodes in node set $\mathcal{M}_{ij}$ of substructure i are also contained in one or more node sets of other substructures. As such, define

$$\mathcal{N}_{ij} = \{(k,l) : \mathcal{M}_{kl} = \mathcal{M}_{ij}\} . \tag{13}$$

For notational convenience, assume that the rows of R_{ijc} are ordered such that $R_{ijc}u_i = R_{klc}u_k$ for all $(k,l) \in \mathcal{N}_{ij}$. Let E_{ij} denote the column concatenation of all vectors $R_{klc}a_k$ such that $(k,l) \in \mathcal{N}_{ij}$. Consider the singular value decomposition

$$\tilde{E}_{ij} = U_{ij} S_{ij} V_{ij}^T \tag{14}$$

where $\tilde{E}_{ij}$ is the matrix obtained by normalizing each column of E_{ij}. Assuming the singular values s_{ijm} on the diagonal of S_{ij} are in descending numerical order, let m_{ij} denote the largest value of m such that $s_{ikm}/s_{ij1} > tol$ where in this study $tol = 10^{-8}$. The singular values along with tol are used to determine a numerical rank of E_{ij}. Let F_{ij} denote the matrix obtained by normalizing each column of $(R_{ijr}C_iR_{ijc}^T)^T$ and define

$$\tilde{F}_{ij} = F_{ij} - \tilde{U}_{ij}\tilde{U}_{ij}^T F_{ij} = \bar{U}_{ij}\bar{S}_{ij}\bar{V}_{ij}^T \tag{15}$$

where $\tilde{U}$ contains the first m_{ij} columns of U_{ij}. The columns of $\tilde{U}$ are orthogonal and numerically span the range of E_{ij}. The singular values $\bar{s}_{ijm}$ on the diagonal of $\bar{S}_{ij}$ are assumed to be in descending numerical order and $\bar{m}_{ij}$ denotes the largest value of m such that $\bar{s}_{ijm} > tol$. Define

$$G_{ij} = \left[\, \tilde{U}_{ij} \ \hat{U}_{ij} \,\right] \tag{16}$$

where $\hat{U}_{ij}$ contains the first $\bar{m}_{ij}$ columns of $\bar{U}_{ij}$. The columns of $\hat{U}$ are orthogonal and numerically span the range of the projection of F_{ij} onto the orthogonal complement of $\tilde{U}_{ij}$. Thus, the columns of G_{ij} are orthogonal. Notice that G_{ij} contains a linearly independent set of vectors for the zero divergence constraints and the original BDDC constraints for node set $\mathcal{M}_{ij}$.

Finally, the original constraint matrix C_i is replaced by the row concatenation of the matrices $G_{ij}^T R_{ijc}$ for $j = 1, \ldots, M_i$. Use of the new substructure constraint matrices ensures that preconditioned residuals will not have excessively large values of volumetric energy. The final requirement needed to ensure good scalability with respect to the number of substructures is that the coarse stiffness matrix K_c be flexible enough to approximate well the low energy modes of K. This requirement is closely tied to an inf-sup condition, but is not analyzed here. Numerical results, however, indicate good scalability in this respect.

For 2D problems a node set consists either of a single isolated node called a corner or a group of nodes shared by exactly two substructures called a face. Furthermore, m_{ij}, the number of columns in $\tilde{U}_{ij}$, is at most two for a corner and one for a face. Similarly, for 3D problems m_{ij} is at most three for a corner and one for a face. The value of m_{ij} for the remaining 3D node sets, called edges here, depends on the mesh decomposition as well as the positions of nodes in the mesh. In any case, performance of the preconditioner should not degrade in the presence of nearly incompressible materials provided that all the columns of $\tilde{U}_{ij}$ are included in G_{ij}. Including columns of $\hat{U}_{ij}$ in G_{ij} as well will reduce condition numbers of the preconditioned equations, but is not necessary to avoid degraded performance for nearly incompressible materials.

Use of the modified constraints does not cause any difficulties when both nearly incompressible materials (e.g. rubber) and materials with smaller values of Poisson ratio (e.g. steel) are present. One can exclude the incompressibility constraint for substructures not containing nearly incompressible materials simply by setting all entries of a_i in (12) to zero. Doing so may lead to a slightly smaller coarse problem, but it is not necessary.

5 Numerical Examples

In this section, (1) is solved to a relative residual tolerance of 10^{-6} using both right preconditioned GMRES [17] and preconditioned conjugate gradients (PCG) for an incompressible elasticity problem. For linear elasticity the shear modulus G and Lamé parameter λ for an isotropic material are related to the elastic modulus E and Poisson ratio ν by

$$G = \frac{E}{2(1+\nu)}, \qquad \lambda = \frac{\nu E}{(1+\nu)(1-2\nu)} .$$

For incompressible problems λ is infinite with the result that $C = 0$ in (1). All the elasticity examples in this section use $G = 1$ and $\nu = 1/2$. We consider two different preconditioners for S_A in order to better understand the saddle point preconditioner. The first is based on a direct solver where $1.00001\hat{S}_A = S_A$ while the second is the BDDC preconditioner described in the previous two sections. Note that the leading constant 1.00001 is used to satisfy the assumption $\alpha_1 > 1$. The penalty matrix $\tilde{C}$ for the elasticity problems is chosen as the negative (2,2) block of the coefficient matrix in (1) for an identical problem with the same shear modulus but a value of ν less than 1/2.

Regarding assumption (3), we note that the BDDC preconditioner used for S_A has the attractive property that $\alpha_1 \geq 1$ and α_2 is mesh independent under certain additional assumptions [15]. For the conjugate gradient algorithm we scale the preconditioned residual associated with the primal Schur complement by 1.00001 to ensure that $\mathcal{H}$ is positive definite.

For purposes of comparison, we also present results for block diagonal and block triangular preconditioners for (1). Given the primal and dual residuals r_u and r_p, the preconditioned residuals z_u and z_p for the block diagonal preconditioner are given by

$$z_u = M_A^{-1} r_u \quad \text{and} \quad z_p = M_p^{-1} r_p$$

where M_p is the dual mass matrix and either $M_A = A$ (direct solver) or M_A is the BDDC preconditioner for A. Note that the shear modulus G was chosen as 1 to obtain proper scaling of z_p. Similarly, the preconditioned residuals for the block triangular preconditioner are given by

$$z_p = -M_p^{-1} r_p \quad \text{and} \quad z_u = M_A^{-1}(r_u - B^T z_p)\,.$$

We note that the majority of computations for the block preconditioners occur in forming and applying the BDDC preconditioner for A. Thus, the setup time and time for each iteration are very similar for the preconditioner of this study and the two block preconditioners.

The first example is for a 2D plane strain problem on a unit square with all displacement degrees of freedom (dofs) on the boundary constrained to zero. The entries of the right hand side vector b were chosen as uniformly distributed random numbers in the range from 0 to 1. For this simple geometry the finite element mesh consists of stable $Q_2 - P_1$ elements. This element uses biquadratic interpolation of displacement and discontinuous linear interpolation of pressure. In 2D the element has 9 nodes for displacement and 3 element pressure dofs. A description of the $Q_2 - P_1$ discontinuous pressure element can be found in [2].

Results are shown in Table 1 for the saddle point preconditioner (SPP) applied to a problem discretized by a 32 x 32 arrangement of square elements. Condition number estimates of the preconditioned equations are shown in parenthesis for the PCG results. The BDDC preconditioner is based on a regular decomposition of the mesh into 16 square substructures. The results shown in columns 2-5 are insensitive to changes in ν near the incompressible limit of 1/2. Notice that the use of a direct solver to precondition S_A results in very small numbers of iterations for values of ν near 1/2. The final two columns in Table 1 show results for BDDC constraint equations that are not modified to enforce zero divergence of each substructure. The condition number estimates grow in this case as ν approaches 1/2.

Table 2 shows results for a growing number of substructures with $H/h = 4$ where H and h are the substructure and element lengths, respectively. Very small growth in numbers of iterations with problem size is evident in the table for all the preconditioners. Notice that the iterations required by PCG either equal or are only slightly larger than those for GMRES. The primary advantage of PCG from a solver perspective is that storage of all search directions is not required as it is for GMRES. The SPP preconditioner is clearly superior to the two block preconditioners when a direct solver is used ($1.00001\hat{S}_A = S_A$ and $M_A = A$). The performance of the SPP preconditioner compares very favorably with both of the block preconditioners when the BDDC preconditioner is used.

Table 1. Iterations needed to solve incompressible 2D plane strain problem using the saddle point preconditioner. Results are shown for different values of ν used to define $\tilde{C}$. Results in parenthesis are condition number estimates from PCG. The $\hat{S}_A$ = no mod BDDC designation is for BDDC constraint equations that cannot enforce zero divergence of each substructure.

	$1.00001\hat{S}_A = S_A$		$\hat{S}_A =$ BDDC		$\hat{S}_A =$ no mod BDDC	
ν	GMRES	PCG	GMRES	PCG	GMRES	PCG
0.3	8	10 (4.8)	19	23 (16)	19	22 (16)
0.4	7	10 (2.4)	15	17 (7.2)	15	17 (7.1)
0.49	4	5 (1.1)	11	11 (3.0)	13	13 (3.6)
0.499	3	3 (1.01)	10	10 (2.7)	17	18 (8.5)
0.4999	3	3 (1.01)	9	9 (2.7)	23	28 (7.0e1)
0.49999	3	3 (1.01)	9	9 (2.6)	25	44 (6.9e2)

Table 2. Iterations needed to solve incompressible plane strain problems with increasing numbers of substructures (N) and $H/h = 4$. The value of ν used to define $\tilde{C}$ in the SPP preconditioner is 0.49999. Block diagonal and triangular preconditioners are denoted by M_d and M_t, respectively.

N	$1.00001\hat{S}_A = S_A$ and $M_A = A$				$\hat{S}_A$ and M_A = BDDC			
	SPP		M_d	M_t	SPP		M_d	M_t
	GMRES	PCG	GMRES	GMRES	GMRES	PCG	GMRES	GMRES
4	3	3 (1.01)	17	9	6	6 (1.8)	26	16
16	3	3 (1.01)	17	9	8	8 (2.1)	30	20
36	3	3 (1.01)	17	9	9	9 (2.6)	35	23
64	3	3 (1.01)	17	9	9	10 (2.9)	38	26
100	3	3 (1.01)	17	9	10	10 (3.0)	40	28
144	3	3 (1.03)	17	9	10	10 (3.1)	42	29
196	3	3 (1.01)	17	9	10	11 (3.1)	45	30
256	3	3 (1.01)	17	9	10	11 (3.1)	47	30

References

1. O. Axelsson, *Preconditioning of indefinite problems by regularization*, SIAM J. Numer. Anal., 16 (1979), pp. 58–69.
2. K.-J. Bathe, *Finite element procedures*, Prentice Hall, Englewood Cliffs, NJ, 1996.
3. J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Mathematics of Computation, 50 (1988), pp. 1–17.
4. C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258.

5. ———, *A substructuring preconditioner for nearly incompressible elasticity problems*, Tech. Rep. SAND 2004-5393, Sandia National Laboratories, October 2004.
6. C. R. Dohrmann and R. B. Lehoucq, *A primal based penalty preconditioner for elliptic saddle point systems*, Tech. Rep. SAND 2004-5964, Sandia National Laboratories, 2004.
7. H. C. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., (1994), pp. 1645–1661.
8. P. Goldfeld, L. F. Pavarino, and O. B. Widlund, *Balancing Neumann-Neumann methods for mixed approximations of heterogeneous problems in linear elasticity*, Numer. Math., 95 (2003), pp. 283–324.
9. P. Goldfield, *Balancing Neumann-Neumann preconditioners for the mixed formulation of almost-incompressible linear elasticity*, PhD thesis, New York University, Department of Mathematics, 2003.
10. A. Klawonn, *Block-triangular preconditioners for saddle point problems with a penalty term*, SIAM J. Sci. Comput., 19 (1998), pp. 172–184.
11. A. Klawonn and L. F. Pavarino, *Overlapping Schwarz methods for elasticity and Stokes problems*, Comput. Methods Appl. Mech. Engrg., 165 (1998), pp. 233–245.
12. J. Li, *A Dual-Primal FETI method for incompressible Stokes equations*, Tech. Rep. 816, Courant Institute of Mathematical Sciences, Department of Computer Sciences, 2001.
13. J. Li and O. B. Widlund, *BDDC algorithms for incompressible Stokes equations*, Tech. Rep. TR-861, New York University, Department of Computer Science, 2005.
14. J. Mandel and C. R. Dohrmann, *Convergence of a balancing domain decomposition by constraints and energy minimization*, Numer. Linear Algebra Appl., 10 (2003), pp. 639–659.
15. J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Appl. Numer. Math., 54 (2005), pp. 167–193.
16. L. F. Pavarino and O. B. Widlund, *Balancing Neumann-Neumann methods for incompressible Stokes equations*, Comm. Pure Appl. Math., 55 (2002), pp. 302–335.
17. Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.
18. D. J. Silvester and A. J. Wathen, *Fast iterative solution of stabilised Stokes systems part II: using general block preconditioners*, SIAM J. Numer. Anal., 31 (1994), pp. 1352–1367.
19. B. Vereecke, H. Bavestrello, and D. Dureisseix, *An extension of the FETI domain decomposition method for incompressible and nearly incompressible problems*, Comput. Methods Appl. Mech. Engrg., 192 (2003), pp. 3409–3429.

Nonconforming Methods for Nonlinear Elasticity Problems *

Bernd Flemisch and Barbara I. Wohlmuth

University of Stuttgart, Institute for Applied Analysis and Numerical Simulation, Stuttgart, Germany. flemisch,wohlmuth@ians.uni-stuttgart.de

Summary. Domain decomposition methods are studied for several problems exhibiting nonlinearities in terms of curved interfaces and/or underlying model equations. In order to retain as much flexibility as possible, we do not require the subdomain grids to match along their common interfaces. Dual Lagrange multipliers are employed to generate efficient and robust transmission operators between the subdomains. Various numerical examples are presented to illustrate the applicability of the approach.

1 Introduction

We apply domain decomposition techniques to efficiently discretize nonlinear elasticity problems. The framework of mortar methods, [1, 2, 3, 8], is employed to deal with nonmatching grids. Especially for the applications discussed in Section 3, we recommend the use of dual discrete Lagrange multiplier spaces as in [5]. They are a basic ingredient for the formulation and the performance of our numerical solution procedures presented there.

In Section 2, we focus on a type of nonlinearity arising only from the geometry of the subdomain interfaces, namely, when the interfaces are curved and therefore require a nonlinear parametrization. The subdomain grids originating from a nonoverlapping decomposition may now overlap or even exhibit gaps along the curved interface. Transferring the methodology of the scalar setting to elasticity problems, we encounter a preasymptotic misbehavior when using dual Lagrange multipliers on the coarse side and present a remedy.

Section 3 deals with nonlinear elasticity model equations. First, two-body contact problems are studied, where we use an inexact primal-dual active set strategy as our solution method. The last part is devoted to the geometrically nonlinear elasticity setting and to the use of Neo–Hooke materials.

*This work was supported in part by the Deutsche Forschungsgemeinschaft, SFB 404, B8, C12.

2 Curvilinear boundaries

Scalar case. For simplicity, we first restrict ourselves to the case of two 2D subdomains sharing a closed interface curve and refer to [4] for a complete analysis for many subdomains. We consider the model problem

$$-\varDelta u = f \text{ in } \varOmega \subset \mathbb{R}^2, \qquad u = 0 \text{ on } \partial\varOmega. \tag{1}$$

for the situation depicted in Figure 1. The domain $\varOmega$ is partitioned into two subdomains $\varOmega^{\mathrm{m}}$ and $\varOmega^{\mathrm{s}}$ by a sufficiently smooth curve $\varGamma$ of length L, given in terms of an arc length parametrization $\gamma : \hat{I} \to \varGamma$, $\hat{I} = [0, L)$. By introducing the spaces $X = H^1_*(\varOmega^{\mathrm{m}}) \times H^1_*(\varOmega^{\mathrm{s}})$ and $M = H^{-1/2}(\varGamma)$, with $H^1_*(\varOmega^{\mathrm{i}})$ respecting the Dirichlet conditions on $\partial\varOmega$, $i = \mathrm{m}, \mathrm{s}$, the boundary value problem (1) can be transformed into the following saddle point problem: find $(u, \lambda) \in X \times M$ such that

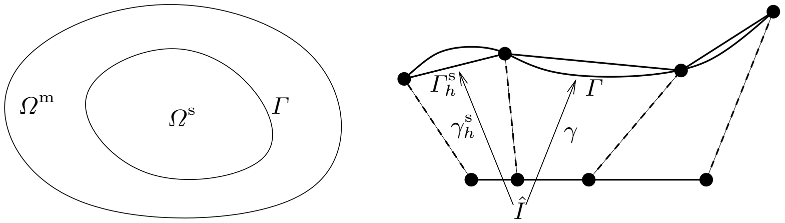


Fig. 1. Left: Decomposition into subdomains $\varOmega^{\mathrm{m}}$, $\varOmega^{\mathrm{s}}$. Right: interface $\varGamma$ and its piecewise linear interpolation $\varGamma_h^{\mathrm{s}}$.

$$\begin{aligned} a(u,v) + b(v,\lambda) &= f(v), \quad && v \in X, \\ b(u,\mu) &= 0, && \mu \in M, \end{aligned} \tag{2}$$

with the obvious meanings for $a(\cdot,\cdot)$ and $f(\cdot)$, and with the coupling bilinear form $b(\cdot,\cdot)$ given by

$$b(v,\mu) = \langle [v],\, \mu \rangle_{\varGamma}, \quad (v,\mu) \in X \times M, \tag{3}$$

where $[\cdot]$ denotes the jump across $\varGamma$. The discretization of $\varOmega$ by $\varOmega^{\mathrm{s}}$ and $\varOmega^{\mathrm{m}}$ with simplicial triangulations results in piecewise linearizations $\varGamma_h^{\mathrm{s}}$ and $\varGamma_h^{\mathrm{m}}$ of the curved interface $\varGamma$, given by piecewise linear parametrizations $\gamma_h^{\mathrm{s}} : \hat{I} \to \varGamma_h^{\mathrm{s}}$ and $\gamma_h^{\mathrm{m}} : \hat{I} \to \varGamma_h^{\mathrm{m}}$, respectively. These parametrizations enable us to uniquely identify each point on $\varGamma_h^{\mathrm{m}}$ with a point on $\varGamma_h^{\mathrm{s}}$, providing a projection operator

$$P_{\mathrm{s}} : (L^2(\varGamma_h^{\mathrm{m}}))^2 \to (L^2(\varGamma_h^{\mathrm{s}}))^2, \quad v_{\mathrm{m}} \mapsto P_{\mathrm{s}} v_{\mathrm{m}} = v_{\mathrm{m}} \circ \gamma_h^{\mathrm{m}} \circ (\gamma_h^{\mathrm{s}})^{-1}. \tag{4}$$

In order to obtain an approximate coupling bilinear form $b_h(\cdot,\cdot)$, we introduce a mesh dependent jump over the interface grid $\varGamma_h^{\mathrm{s}}$ by

$$[v]_h = v_{\mathrm{s}} - P_{\mathrm{s}} v_{\mathrm{m}}.$$

The approximation M_h of M is given by one of the common discrete Lagrange multiplier spaces on Γ_h^{s}, see e.g. [2, 3, 8, 5]. The space X is approximated by X_h using $P1$ finite elements. We define $b_h(\cdot,\cdot)$ in terms of $[\cdot]_h$ by

$$b_h(v,\mu) = (\,[v]_h,\,\mu\,)_{L^2(\Gamma_h^{\mathrm{s}})}, \quad (v,\mu) \in X_h \times M_h. \tag{5}$$

Approximating $a(\cdot,\cdot)$ and $f(\cdot)$ by $a_h(\cdot,\cdot)$ and $f_h(\cdot)$, we obtain the discrete saddle point problem of finding $(u_h,\lambda_h) \in X_h \times M_h$ as the solution of

$$\begin{aligned} a_h(u_h,v) + b_h(v,\lambda_h) &= f_h(v), \quad && v \in X_h, \\ b_h(u_h,\mu) &= 0, && \mu \in M_h. \end{aligned} \tag{6}$$

For an analysis of (6), we refer to [4]. There, in order to obtain a priori bounds for the discretization error, we proceed in two steps. In the first step, we introduce and analyze a new discrete variational problem based on blending elements, where the curved interfaces are resolved exactly, see [6]. In the second step, we interpret (6) as a perturbed blending approach, and estimate the perturbation terms obtained from the first Strang lemma. The main result is:

Theorem 1. *Let* (u,λ) *and* (u_h,λ_h) *solve* (2) *and* (6)*, respectively. Then*

$$\|u - u_h\|_{X_h} + \|\lambda - \lambda_h\|_M \le C(u) \max_{i=m,s} h_i.$$

In [4], several numerical tests in 2D are provided to verify the theoretical results. Here, we focus on a 3D example. An exact parametrization of the interface Γ is often not available. Therefore, an alternative definition of the projection operator P_{s} from (4) is required. This can be achieved for each slave element side by using the piecewise constant normal projection of the corresponding master sides, [9]. We remark that the analysis above has to be extended to this case in order to handle the lack of regularity of P_{s}. For the following example, we use this alternative projection operator to define the coupling bilinear form $b_h(\cdot,\cdot)$.

For the domain Ω, a ball of radius 0.9 is cut out of a concentric ball of radius 1.1. The subdomains Ω_1 and Ω_2 are the parts of Ω with radii greater and less than 1, respectively, their common interface Γ being the unit sphere. The exact solution depends only on the radius r and is set to be $u(r) = ar^{-2} + br$ with a, b chosen such that u describes the radial displacement when the domain is subject to a uniform internal pressure of magnitude 1. We exploit the symmetry of the problem data and reduce the computational domain to $\Omega_r = \{(x,y,z) \in \Omega : x,y,z > 0\}$, adding natural boundary conditions on the symmetry planes. Two initial triangulations with ratios 4:1 and 8:1 of the number of fine to coarse interface element sides are shown in Figure 2.

In Figure 3, we compare the error decays using different Lagrange multiplier spaces, namely, the standard Lagrange multipliers coinciding with the trace space W_h of the $P1$ finite element functions on Ω_h^{s}, with the dual

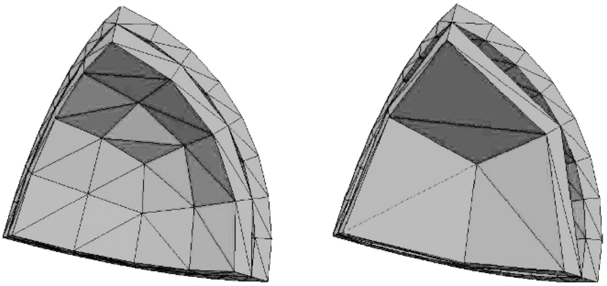


Fig. 2. Initial triangulations: ratios 4:1 and 8:1.

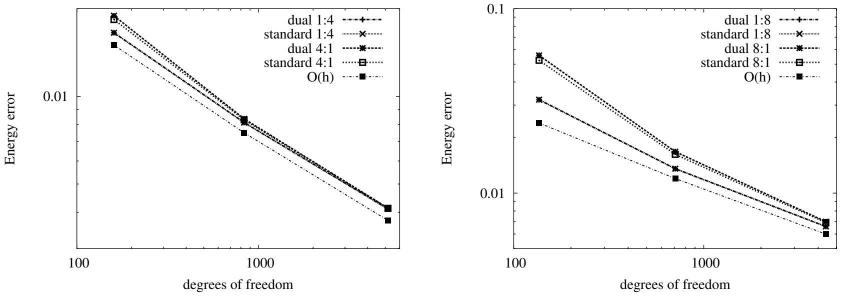


Fig. 3. 3D example: error decay using different Lagrange multiplier spaces.

ones spanned by piecewise linear discontinuous basis functions satisfying a biorthogonality relation with the nodal basis functions of W_h, see [5]. The choice of the basis functions, either standard or dual, does not greatly influence the numerical results. For very coarse meshes, the use of the coarser grid for the Lagrange multipliers provides better results than the altenative. However, this effect gets small already for very moderate numbers of unknowns.

2D elasticity. We keep the same setting as above and intend to solve (2) with spaces and (bi-)linear forms given by the weak form of the linear elasticity problem of finding a displacement vector field u such that

$$-\operatorname{div}\sigma(u) = f \text{ in } \Omega, \tag{7}$$

supplemented by boundary conditions, by the Saint-Venant Kirchhoff law

$$\sigma = \lambda(\operatorname{tr}\varepsilon)\mathrm{I} + 2\mu\,\varepsilon, \tag{8}$$

with the Lamé constants μ, λ and by the linearized strain tensor

$$\varepsilon(u) = \frac{1}{2}(\nabla u + [\nabla u]^{\mathrm{T}}) \ . \tag{9}$$

We consider the domain visualized in the left picture of Figure 4, see [5]. The ring Ω with inner radius $r_{\mathrm{i}} = 0.9$, outer radius $r_{\mathrm{o}} = 1.1$, and moduli $E = 1$,

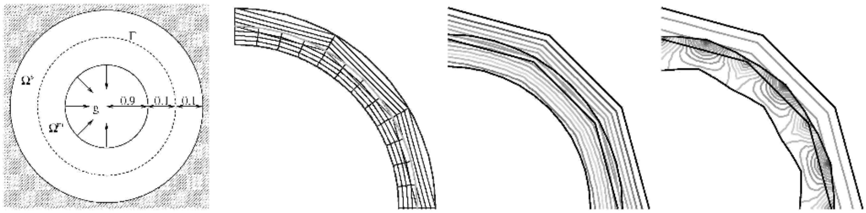


Fig. 4. Model problem, grid, stress using standard and dual multipliers.

$\nu = 0.3$, is fixed at the outer boundary, whereas at the inner boundary, a surface traction $f_\Gamma(x, y) = -(x, y)^\mathrm{T}/r_\mathrm{i}$ constant in normal direction is applied. The region is divided into two rings Ω^m and Ω^s such that their interface Γ is the unit circle. We choose the inner ring to be Ω^m, and the outer ring to be Ω^s. A part of the computational grid is shown in the second picture of Figure 4. The whole grid consists of 240 elements and is constructed in such a way that each element edge on the slave side meets four master edges. Thus, the discrete Lagrange multiplier space M_h is defined with respect to the coarse grid on Γ_h^s. Again, we compare the standard Lagrange multipliers with the dual ones. In the third and fourth picture of Figure 4, the isolines of the van Mises stresses of the numerical solutions on the deformed domains are plotted. Whereas standard Lagrange multipliers yield a visually satisfying result, the behavior of the solution using dual Lagrange multipliers suffers from strong oscillations along the master interface Γ_h^m.

The misbehavior of the dual Lagrange multipliers, which only occurs preasymptotically and only if they are chosen with respect to the coarser grid, can be explained by the fact that quantities constant in normal or tangential direction are not transferred correctly between the two grids. In [5], we introduce and analyze a modification curing this misbehavior, and at the same time preserving the advantages of the dual approach. We modify $b_h(\cdot, \cdot)$ in (5) to

$$b_h^\mathrm{mod}(v_h, \mu_h) = \int_{\Gamma_h^\mathrm{s}} \mu_h v_\mathrm{s} - \mu_h^\mathrm{mod} P_\mathrm{s} v_\mathrm{m}, \qquad v_h \in X_h,\ \mu_h \in M_h, \tag{10}$$

where we replace μ_h for the coupling to the master side by $\mu_h^\mathrm{mod} = \mu_h + \Delta\mu_h$. The modification $\Delta\mu_h$ of a discrete Lagrange multiplier $\mu_h \in M_h$ is defined edgewise on the elements of the interface grid Γ_h^s, see [5]. There, we show that the resulting discrete problem (6) with $b_h(\cdot, \cdot)$ replaced by $b_h^\mathrm{mod}(\cdot, \cdot)$ has the following properties: a diagonal matrix for the coupling on the slave side, symmetry, preservation of linear momentum, reduction to the unmodified dual approach in case of straight interfaces, and preservation of quantities constant in normal and tangential direction.

As a numerical test, we compare the error decays using the standard, dual, and modified dual approach. For the left picture of Figure 5, the ratio of slave to master edges is kept constant at 1:4. The modification already improves

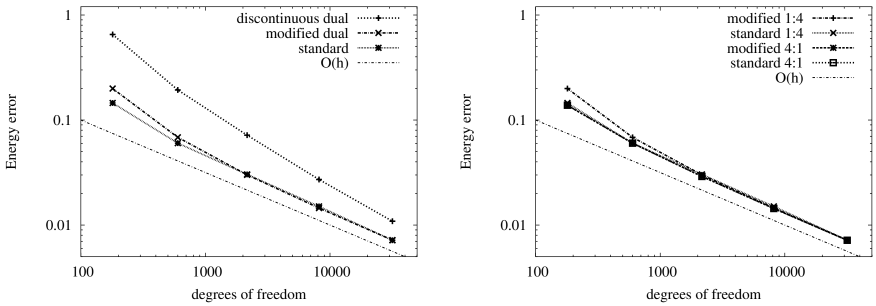


Fig. 5. Left: Decay of the energy error using standard, dual, and modified dual Lagrange multipliers. Right: Change of the Lagrange multiplier side.

the results significantly for a very moderate number of unknowns. We observe that the relative difference in the errors of the unmodified and the modified approach decreases as the number of unknowns increases. This is due to the fact that the modification only enters as a higher order term in the a priori estimates, see [5]. The right picture in Figure 5 illustrates the robustness of the standard and the modified Lagrange multipliers against a change of the master and slave side. We point out that all the benefits of the dual approach are preserved by the modification.

In many applications, symmetry of the domain and the data can be exploited to reduce the problem size. For the example above, we can reduce the computational domain to one quarter $\Omega_r = \{(x, y) \in \Omega : x, y > 0\}$. On the artificial boundaries $\Sigma_\xi = \{(x, y) \in \Omega : \xi = 0\}$, $\xi = x, y$, we have to set appropriate symmetry boundary conditions. For the elasticity setting, these are given by homogeneous Dirichlet data in the normal and homogeneous Neumann data in the tangential direction. In the framework of mortar methods, this would require us to handle the nodes $p_x = (1, 0)^{\mathrm{T}}$ and $p_y = (0, 1)^{\mathrm{T}}$ belonging to the triangulation on Ω_r^{s} as crosspoints for the normal and as usual slave nodes for the tangential components. Since this can be a tedious task to realize during the matrix assembly in existing codes, we suggest to use a simple manipulation of the saddle point system matrix $S = \left(\begin{smallmatrix} A & B^{\mathrm{T}} \\ B & 0 \end{smallmatrix}\right)$ for which the nodes p_x, p_y are handled as usual slave nodes and no Dirichlet conditions are imposed on them. We symmetrically exchange the lines and columns in B^{T} and B corresponding to the coupling of the Lagrange multipliers in the normal direction of p_x and p_y to the displacements in the normal direction on the master and slave side by Dirichlet lines and columns. This is exactly the procedure often employed to enforce Dirichlet conditions by means of Lagrange multipliers.

In Figure 6, we test four different approaches. For the calculations leading to the first two pictures, the two Dirichlet lines are inserted in the upper part of S. For the first (second) picture, the nodes p_x, p_y are handled as slave (cross) points in both directions and the Lagrange multiplier space is chosen

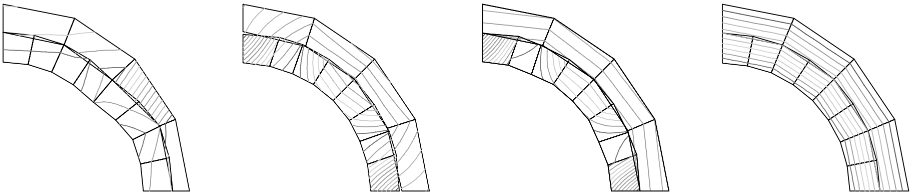


Fig. 6. Handling of symmetry boundaries.

with respect to the finer (coarser) grid. As is expected, both approaches give poor results. For the third picture, we choose the Lagrange multiplier space with respect to the coarser grid, insert only Dirichlet lines in B, and keep B^{T} unchanged. However, this is not enough. This is due to the fact that, in contrast to the full setting, the normal (w.r.t. $\varSigma_\xi$) components of the Lagrange multipliers in p_x, p_y are different from zero in the reduced setting on $\varOmega_r$, since only contributions from $\varOmega_r$ are assembled. Thus, the master nodes next to p_x, p_y are subjects to a force pushing in the wrong direction. In order to avoid that these master nodes are affected by the nonzero contribution from the Lagrange multipliers, one also has to insert the corresponding Dirichlet columns in B^{T}, resulting in the right picture of Figure 6. An equally satisfying result is obtained if the Lagrange multipliers are chosen on the finer grid.

3 Nonlinear elasticity

Contact problems. We consider a two-body nonlinear contact problem. The domain $\varOmega$ is the union of two initially disjoint bodies $\varOmega^{\mathrm{s}}$, $\varOmega^{\mathrm{m}}$, and its boundary $\varGamma = \partial\varOmega^{\mathrm{s}} \cup \partial\varOmega^{\mathrm{m}}$ is subdivided into three disjoint open sets $\varGamma_{\mathrm{D}}$, $\varGamma_{\mathrm{N}}$, $\varGamma_{\mathrm{C}}$. We intend to solve (7)-(9) with Dirichlet and Neumann boundary conditions on $\varGamma_{\mathrm{D}}$ and $\varGamma_{\mathrm{N}}$, respectively, and frictionless Signorini contact conditions on the possible contact boundary $\varGamma_{\mathrm{C}}$, given by

$$\begin{aligned} \sigma_T(u_{\mathrm{s}}) = \sigma_T(u_{\mathrm{m}}) = 0, \quad & \sigma_n(u_{\mathrm{m}})([u\,n] - g) = 0, \\ [u\,n] - g \leq 0, \quad & \sigma_n(u_{\mathrm{m}}) = \sigma_n(u_{\mathrm{s}}) \leq 0, \end{aligned} \tag{11}$$

where $\sigma_T(u_k)$ and $\sigma_n(u_k)$ are the tangential part and the normal component of the surface traction $\sigma(u_k)n$, respectively, $k =$ m,s, and $[u\,n]$ stands for the jump of the normal displacement across $\varGamma_{\mathrm{C}}$.

We arrive at the problem: find $(u, \lambda) \in X \times M^+$ such that

$$\begin{aligned} a(u,v) + b(v,\lambda) \;&=\; f(v), & v \in X, \\ b(u, \mu - \lambda) \;&\leq\; \langle g, (\mu - \lambda)\, n\rangle_{\varGamma_{\mathrm{C,s}}}, & \mu \in M^+, \end{aligned} \tag{12}$$

with $b(v,\mu) = \langle \mu\, n, [v\, n]\rangle_{\varGamma_{\mathrm{C,s}}}$ and $M^+ = \{\mu \in M : \mu_T = 0, \langle \mu\, n, v\rangle_{\varGamma_{\mathrm{C,s}}} \geq 0,\ v \in W,\ v \geq 0 \text{ on } \varGamma_{\mathrm{C,s}}\}$, where W denotes the trace space of $H^1_*(\varOmega^{\mathrm{s}})$ restricted to $\varGamma_{\mathrm{C,s}}$ and M is its dual. We use standard piecewise linear finite elements for X and discontinuous piecewise linear dual Lagrange multipliers for M. The discrete convex cone M_h^+ is defined with respect to the scalar dual basis funtions ψ_i as

$$M_h^+ = \{\mu_h \in M_h : \mu_h = \sum \alpha_i \psi_i,\ \alpha_i \in \mathbb{R}^2, \alpha_i\, n \geq 0,\ \alpha_i \times n = 0\}.$$

In [7], optimal a priori error bounds are obtained for the correspondig discrete problem formulation. Concerning the numerical solution process, we employ a primal-dual active set strategy (PDASS) in order to deal with the nonlinearity of the contact condition (11). Starting from an initial active set, the PDASS checks in each step the sign of the normal stress component for an active node to determine whether the node stays active, and for an inactive node the non-penetration condition to determine whether the node stays inactive. Proceeding like this, a new active set is calculated, and the active nodes provide Dirichlet conditions and the inactive nodes give homogeneous Neumann conditions for the linear system to be solved. The biorthogonality of the dual basis functions spanning M_h^+ is of crucial importance for the realization of the PDASS. In particular, the weak formulation of the non-penetration condition, i.e., the third equation of (11), naturally reduces to a pointwise relation which is easy to handle. Moreover, the Lagrange multiplier can be efficiently eliminated yielding a positive definite linear system for the remaining unknowns in each iteration step of the PDASS. Thus, suitable multigrid solvers can be applied. Limiting the maximum number of multigrid iterations per PDASS step yields an inexact strategy.

As a numerical example, we consider the situation depicted in Figure 7. In the left picture, a cross section of the problem definition is shown. The lower domain Ω^1 is the master, and it models a halfbowl which is fixed at its outer boundary. Against this bowl, we press the body modeled by the domain Ω^2 which is the slave. At the top of Ω^2, we apply Dirichlet data equal to $(0,\ 0,\ -0.2)^\top$. We use $r_i = 0.7$, $r_a = 1.0$, $r = 0.6$, $h = 0.5$ and $d = 0.3$, and as

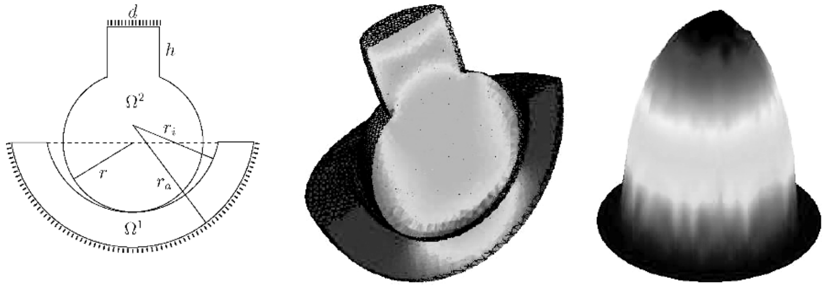


Fig. 7. Problem setting (left), cut through the distorted domains with the effective von Mises stress on level 3 (middle), and the contact stresses λ_h on level 3 (right).

material parameters, $E_1 = 400\mathrm{N/m}^2, \nu_1 = 0.3$ and $E_2 = 300\mathrm{N/m}^2, \nu_2 = 0.3$. The second and third picture in Figure 7 show a cut through the domains and the contact stress λ_h on level 3, respectively.

In Table 1, the exact PDASS is compared with the inexact version. For the inexact strategy, we apply only one multigrid step per PDASS iteration. For both strategies, we use a $\mathcal{W}$-cycle with a symmetric Gauß–Seidel smoother with 3 pre- and post-smoothing steps. The second column shows the number of degrees of freedom on level l. For the exact strategy, we denote by K_l the step in which the correct active set $\mathcal{A}$ is found for the first time, and M_l indicates the same quantity for the inexact strategy. By $\lvert\mathcal{A}_k\rvert$, we denote the number of active nodes in iteration k and multigrid step k, respectively. They are almost the same, thus, there is no need for solving the resulting linear problems in each iteration step exactly, and the cost of our nonlinear problem is very close to that of a linear problem, given the correct contact zone.

l	DOF	exact strategy					inexact strategy					
		K_l	$\lvert\mathcal{A}_k\rvert$				M_l	$\lvert\mathcal{A}_k\rvert$				
0	312	3	0	9	6		3	0	9	6		
1	1623	4	14	26	21	21	4	14	26	22	21	
2	10062	3	66	88	85		3	66	91	85		
3	71082	4	306	347	336	337	5	306	341	336	336	337

Table 1. Comparison between exact and inexact active set strategy.

Geometrically nonlinear problems and nonlinear material laws. The validity of the linearized elasticity equations (7)-(9) is restricted to small strains and small deformations. If the strains remain small but the deformations become large, one has at least to consider the geometrically nonlinear elasticity setting. This amounts to using the full Green–St. Venant tensor

$$E = \frac{1}{2}(F^{\mathrm{T}}F - \mathrm{I}) = \frac{1}{2}(C - \mathrm{I}), \tag{13}$$

instead of (9), with $F = \mathrm{I} + \nabla u$ the deformation gradient and $C = F^{\mathrm{T}}F$ the right Cauchy–Green strain tensor. We keep the constitutive law (8) as

$$S = \lambda(\mathrm{tr}\, E)\mathrm{I} + 2\mu\, E = \mathcal{C}E, \tag{14}$$

defining the second Piola–Kirchhoff stress tensor S, with $\mathcal{C}$ the Hooke-tensor. We solve

$$-\mathrm{div}\,(FS) = f, \tag{15}$$

complemented by appropriate boundary conditions. In the weak setting, this gives the linear form $a(u,\cdot)$ given by $a(u,v) = \sum_{i=1}^{4} a_i(u,v)$, where

$$a_1(u,v) = \int_\Omega \mathcal{C}\varepsilon(u) : \varepsilon(v)\,dx, \qquad a_2(u,v) = \frac{1}{2}\int_\Omega \mathcal{C}\left[(\nabla u)^\top \nabla u\right] : \nabla v\,dx,$$
$$a_3(u,v) = \int_\Omega \nabla u\,\mathcal{C}\,\nabla u : \nabla v\,dx, \quad a_4(u,v) = \frac{1}{2}\int_\Omega \nabla u\,\mathcal{C}\left[(\nabla u)^\top \nabla u\right] : \nabla v\,dx.$$

Still, the applicability of (13)–(15) is limited to small strains. In order to extend the model to large strains, we have to introduce another kind of nonlinearity by means of nonlinear material laws. In particular, to solve (15), we employ the Neo–Hooke law given by

$$S = \mu(\mathrm{I} - C^{-1}) + \frac{\lambda}{2}(J^2 - 1)C^{-1}, \tag{16}$$

with $J = \det(F)$ denoting the determinant of the deformation gradient. While in (13) the nonlinearity enters in terms of polynomials of ∇u, it is given in terms of its inverse in (16).

Despite the complexity of the nonlinear setting, the subdomain coupling via Lagrange multipliers remains the same as for linear problems. In order to calculate a numerical solution, we eliminate the discrete Lagrange multipliers and apply a Newton iteration to the constrained problem. We note that this elimination is very efficient when we use the dual basis functions for spanning the Lagrange multiplier space. Moreover, the Jacobian of the constrained system is positive definite and admits the use of multigrid solvers for the linear system in each Newton step.

For a first numerical test, we consider a square $\Omega = (0,1)^2$, decomposed into four quadrilaterals $\Omega_{ij} = ((i-1)/2, i/2) \times ((j-1)/2, j/2)$, $i,j = 1,2$. The material parameter are set to $E = 2000\,\mathrm{N/m}^2$, $\nu = 0.4$ on Ω_{11}, Ω_{22} and to $E = 300\,\mathrm{N/m}^2$, $\nu = 0.3$ on Ω_{21}, Ω_{12}. We use the linear elasticity model on Ω_{11}, Ω_{22}, and the nonlinear Neo–Hooke model on Ω_{21}, Ω_{12}. The domain is fixed at its upper and lower boundary segment, whereas on the left and right segment, a force density of magnitude $10 + y(y-1)$ pointing inside the domain is applied. The first two pictures of Figure 8 show the deformed grids with deformations magnified by a factor 100 for two ways of dealing with the crosspoint $p^{\mathrm{c}} = (1/2, 1/2)$. In the first calculation, the crosspoint is left

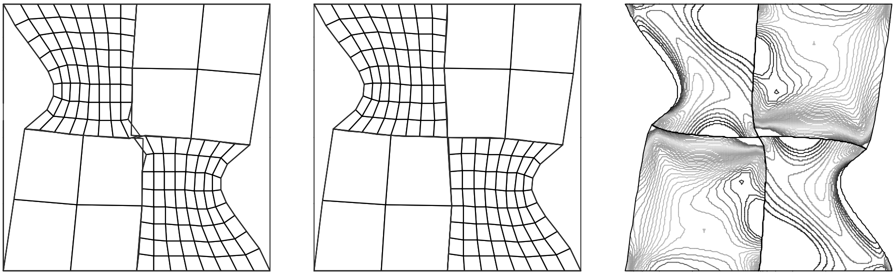


Fig. 8. Deformations without (left) and with (middle) continuity requirement.

free leading to unphysical penetrations of the subdomains. In contrast, for the second calculation, continuity is enforced; cf. [2]. We note that the undesired effect of the first calculation diminishes when the meshsize is reduced.

As 3D example, we consider an I-beam as illustrated in Figure 9. The beam is decomposed into three subdomains $\Omega_1 := (0, 50) \times (0, 10) \times (11, 13)$, $\Omega_2 := (0, 50) \times (3, 7) \times (2, 11)$ and $\Omega_3 := (0, 50) \times (0, 10) \times (0, 2)$. On all subdomains, we consider as material parameters $E = 100, \nu = 0.3$. The beam is fixed in all directions on the plane $x_3 = 0$, and in x_3-direction on the plane $x_3 = 13$. On $\Sigma_1, \Sigma_2 \subset \partial\Omega_1$ with $\Sigma_1 = (0, 50) \times \{0\} \times (11, 13), \Sigma_2 = (0, 50) \times \{10\} \times (11, 13)$, surface forces $f(x) = -2 + 4x/50$ in y-direction are applied.

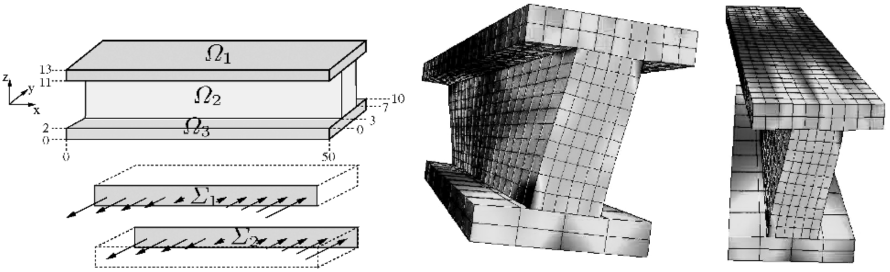


Fig. 9. Left: I-beam decomposed into three subdomains and urface forces on $\Sigma_1, \Sigma_2 \subset \partial\Omega_1$. Middle and right: deformed beam.

In the middle and the right picture of Figure 9, the deformed beam is plotted using the Neo–Hooke law on all subdomains. We note that we do not require the subdomain triangulations to match across their common interfaces; we can employ different meshsizes and uniformly structured grids as well as different models on each subdomain. The deformed grid suggests that we can employ the fully linearized one for the lower subdomain Ω_3, where only small displacements and strains occur, the geometrically nonlinear one for the upper part Ω_1 because of large displacements but small strains, and the Neo–Hooke law for the middle beam Ω_2 with both large deformations and strains.

To justify our strategy, we compare the use of different models on the individual subdomains. We indicate a configuration by ijk, $i, j, k \in \{l, g, n\}$, where l, g and n stand for linear, geometrically nonlinear and Neo–Hooke, respectively, and the position indicates the corresponding subdomain. In Figure 10, the displaments in x_1-direction along the line $(0, 50) \times \{3\} \times \{11\}$ on Ω_1 are plotted for several different settings. In the left picture, the solid, dashed, and dash-dotted lines correspond to the models nnn, lll, and ggg, respectively. Whereas the linear model is symmetric with respect to $x_1^* = 25$, the nonlinear ones exhibit a rather unsymmetric and more realistic behavior. Moreover, on each line, the markers indicate the results when the model on the lower subdomain Ω_3 is switched. There is no visible difference between using the linear or the nonlinear relationship. In the right picture, we primarly compare

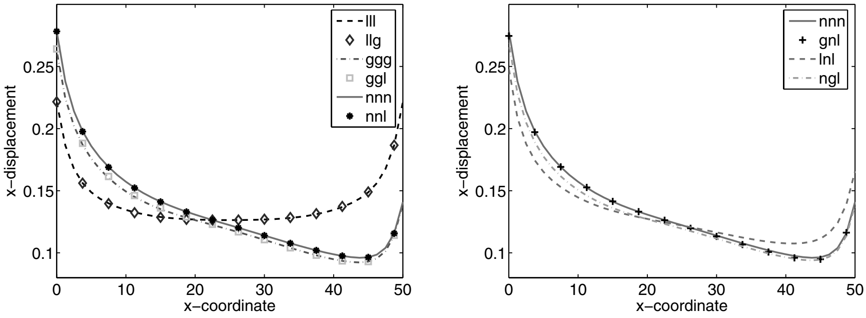


Fig. 10. Comparison of varying model equations in the subdomains.

configurations *nnn* and *gnl*, where no real difference can be observed. The results for *ngl* and *lnl* in combination with the left picture indicate that it is necessary to use the Neo–Hooke law on the middle subdomain Ω_2, while on the upper part Ω_1, the geometrically nonlinear model is required.

References

1. F. Ben Belgacem, *The mortar finite element method with Lagrange multipliers*, Numer. Math., 84 (1999), pp. 173–197.
2. C. Bernardi, Y. Maday, and A. T. Patera, *A New Non Conforming Approach to Domain Decomposition: The Mortar Element Method*, vol. 299 of Pitman Res. Notes Math. Ser., Pitman, 1994, pp. 13–51.
3. D. Braess and W. Dahmen, *Stability estimates of the mortar finite element method for 3-dimensional problems*, East-West J. Numer. Math., 6 (1998), pp. 249–264.
4. B. Flemisch, J. M. Melenk, and B. I. Wohlmuth, *Mortar methods with curved interfaces*, Appl. Numer. Math., 54 (2005), pp. 339–361.
5. B. Flemisch, M. A. Puso, and B. I. Wohlmuth, *A new dual mortar method for curved interfaces: 2D elasticity*, Internat. J. Numer. Methods Engrg., 63 (2005), pp. 813–832.
6. W. J. Gordon and C. A. Hall, *Transfinite element methods: blending-function interpolation over arbitrary curved element domains*, Numer. Math., 21 (1973), pp. 109–129.
7. S. Hüeber, M. Mair, and B. I. Wohlmuth, *A priori error estimates and an inexact primal-dual active set strategy for linear and quadratic finite elements applied to multibody contact problems*, Appl. Numer. Math., 54 (2005), pp. 555–576.
8. C. Kim, R. Lazarov, J. Pasciak, and P. Vassilevski, *Multiplier spaces for the mortar finite element method in three dimensions*, SIAM J. Numer. Anal., 39 (2001), pp. 519–538.
9. M. A. Puso, *A 3D mortar method for solid mechanics*, Internat. J. Numer. Methods Engrg., 59 (2004), pp. 315–336.

Finite Element Methods with Patches and Applications

Roland Glowinski[1], Jiwen He[1], Alexei Lozinski[2*], Marco Picasso[2], Jacques Rappaz[2], Vittoria Rezzonico[2†], and Joël Wagner[2]

[1] Department of Mathematics, University of Houston, Houston, TX 77004, USA.
[2] Institute of Analysis and Scientific Computing, Ecole Polytechnique Fédérale de Lausanne, Switzerland.
Correspondence to: V. Rezzonico, `vittoria.rezzonico@epfl.ch`

Summary. We present a new method [7] for numerically solving elliptic problems with multi-scale data using multiple levels of not necessarily nested grids. We use a relaxation method that consists of calculating successive corrections to the solution in patches of finite elements. We analyse the spectral properties of the iteration operator [6]. We show how to evaluate the best relaxation parameter and what is the influence of patches size on the convergence of the method. Several numerical results in 2D and 3D are presented.

1 Introduction

In numerical approximation of elliptic problems by the finite element method, great precision of the solutions is often required in certain regions of the domain. Efficient approaches include adaptive mesh refinement and domain decomposition methods. The objective of this paper is to present a method to solve numerically elliptic problems with multi-scale data using two levels of not necessarily nested grids.

Consider a multi-scale problem with large gradients in small sub-domains. We solve the problem with a coarse meshing of the computational domain. Therein, we consider a patch (or multiple patches) with corresponding fine mesh wherein we would like to obtain more accuracy. Thus, we calculate successively corrections to the solution in the patch. The coarse and fine discretizations are not necessarily conforming. The method is a domain decomposition method with complete overlapping. It resembles the Fast Adaptive Composite grid (FAC) method (see, e.g., [8]) or possibly a hierarchical method (see [3] for example). However it is much more flexible to use in comparison to

[*] Supported by the Swiss CTI Innovation Promotion Agency
[†] Supported by the Swiss National Science Foundation

the latter: in fact the discretizations do not need to be nested, conforming or structured. The idea of the method is strongly related to the Chimera method [4].

The outline of this paper is as follows. In Section 2, we introduce the algorithm and present an *a priori* estimate for the approximation (Prop. 1). In Section 3, we present the convergence result for the method (Prop. 3) and give sharp results for the spectral properties of the iteration operator. We give a method to estimate the optimal relaxation parameter. In Section 4, we consider computational issues and discuss the implementation. Finally, in Section 5, we assess the efficiency of the algorithm in simple two-dimensional situations and give an illustration in 3D. The reader should note that this paper contains no proofs, which can be found in [6].

2 Two-step algorithm

Let $\Omega \subset \mathbb{R}^d$, $d = 2$ or 3, be an open polygonal or polyhedral domain and consider a bilinear, symmetric, continuous and coercive form $a : H_0^1(\Omega) \times H_0^1(\Omega) \to \mathbb{R}$. The usual $H^1(\Omega)$-norm is equivalent to the a-norm defined by $||v|| = a(v,v)^{\frac{1}{2}}$, $\forall v \in H_0^1(\Omega)$. If $f \in H^{-1}(\Omega)$, due to Riesz' representation Theorem there exists a unique $u \in H_0^1(\Omega)$ such that

$$a(u,\varphi) = \langle f|\varphi\rangle, \quad \forall \varphi \in H_0^1(\Omega), \tag{1}$$

where $\langle\cdot|\cdot\rangle$ denotes the duality $H^{-1}(\Omega) - H_0^1(\Omega)$. Let us point out that (1) is the weak formulation of a problem of type $\mathcal{L}(u) = f$ in Ω, $u = 0$ on the boundary $\partial\Omega$ of Ω, where $\mathcal{L}(\cdot)$ is a second order, linear, symmetric, strongly elliptic operator.

A Galerkin approximation consists in building a finite dimensional subspace $V_{Hh} \subset H_0^1(\Omega)$, and solving the problem: Find $u_{Hh} \in V_{Hh}$ satisfying

$$a(u_{Hh},\varphi) = \langle f|\varphi\rangle, \quad \forall \varphi \in V_{Hh}. \tag{2}$$

In the following the construction of the space V_{Hh} is presented. We introduce a regular triangulation $\mathcal{T}_H$ of $\overline{\Omega}$, a union of triangles K of diameter less than or equal to H. Consider now a multi-scale situation with a solution that is very sharp, i.e., varies rapidly, in a small polygonal or polyhedral subdomain Λ of Ω, but is smooth, i.e., varies slowly, in $\Omega \setminus \Lambda$. This means that the solution can be well approximated on a coarse mesh in $\Omega \setminus \Lambda$ but needs a fine mesh in Λ. We would like to stress that $\overline{\Lambda}$ is not necessarily the union of several triangles K of $\mathcal{T}_H$. Besides Λ can be determined in practice by an a priori knowledge of the solution behaviour or an a posteriori error estimator, for example. Let $\mathcal{T}_h$ be a regular triangulation of $\overline{\Lambda}$ with triangles K such that $\mathrm{diam}(K) \leq h$.

We define $V_H = \{\psi \in H_0^1(\Omega) : \psi|_K \in \mathbb{P}_r(K), \forall K \in \mathcal{T}_H\}$, and $V_h = \{\psi \in H_0^1(\Omega) : \psi|_K \in \mathbb{P}_s(K), \forall K \in \mathcal{T}_h$ and $\psi = 0$ in $\overline{\Omega} \setminus \Lambda\}$, where $\mathbb{P}_q(K)$ is the space of polynomials of degree $\leq q$ on triangle K. We set $V_{Hh} = V_H + V_h$. Let us observe that in practice, it is not possible to determine a finite element basis of V_{Hh}. The goal of our method is to evaluate efficiently u_{Hh} without having a basis of V_{Hh}, but only a basis of V_H and a basis of V_h.

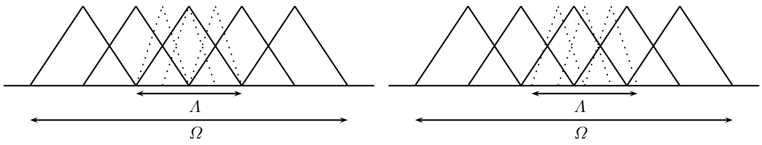
Before to show how to compute u_{Hh}, we give the following *a priori* estimate:

Proposition 1. *Let $q = \max(r, s) + 1$ and suppose that the solution u of (1) is in $H^q(\Omega)$. Then the approximate problem (2) has a unique solution u_{Hh} which satisfies the* a priori *error estimate*

$$||u - u_{Hh}|| \leq C \left(H^r ||u||_{H^q(\Omega\setminus\overline{\Lambda})} + h^s ||u||_{H^q(\Lambda)} \right), \tag{3}$$

where C is a constant independent of H, h and u.

Let us mention that a priori $V_H \cap V_h$ does not necessarily reduce to the element zero as shown in Fig. 1(a), where a 1D situation is illustrated by the hat functions in Ω and in Λ. In the case when $\mathcal{T}_H$ and $\mathcal{T}_h$ are not nested, as illustrated by Fig. 1(b), where we have translated the patch, it is not possible to easily exhibit a finite element-basis of V_{Hh} from the bases of V_H and V_h. Note also that moving from the situation depicted in Fig. 1(a) to the one in Fig. 1(b), the dimension of V_{Hh} increases by 1. All these difficulties suggest that an iterative method should be used to solve problem (2).



(a) Nested elements.

(b) Non-nested elements.

Fig. 1. Linear finite elements in 1D on Ω (plain lines) and Λ (dotted lines) .

So we suggest the following algorithm to compute u_{Hh}.

Algorithm 2

```
1. Set $u^0 \in V_H$ such that $a(u^0, \varphi) = \langle f|\varphi \rangle, \quad \forall \varphi \in V_H$,
   and choose $\omega \in (0; 2)$.
2. For $n = 1, 2, 3, \ldots$ find
   (i) $w_h \in V_h$ such that $a(w_h, \varphi) = \langle f|\varphi \rangle - a(u^{n-1}, \varphi), \quad \forall \varphi \in V_h$ ;
       $u^{n-\frac{1}{2}} = u^{n-1} + \omega w_h$ ;
   (ii)$w_H \in V_H$ such that $a(w_H, \varphi) = \langle f|\varphi \rangle - a(u^{n-\frac{1}{2}}, \varphi), \quad \forall \varphi \in V_H$ ;
       $u^n = u^{n-\frac{1}{2}} + \omega w_H$.
```

It is readily seen that this algorithm is a Schwarz type domain decomposition method [10] with complete overlapping but without any conformity between the meshes $\mathcal{T}_H$ and $\mathcal{T}_h$ (see, e.g., the work by Chan et al. [5]). It is similar to the Chimera or overset grid method [4, 11]. However, the algorithm presented in [4] is an additive method which can be changed to a multiplicative method equivalent to the above presented with $\omega = 1$.

Our multiplicative Schwarz method is also similar to a Gauss-Seidel method and can be put in the framework of the successive subspace correction algorithm by Xu and Zikatanov (see, e.g., [12]). The spaces V_H and V_h defined on the arbitrary triangulations $\mathcal{T}_H$ and $\mathcal{T}_h$ are not necessary orthognal nor do they share only the zero element as intersection. Note in particular that the sum which defines V_{Hh} is a priori not a direct sum. This property makes the above algorithm different from most known iterative schemes. For structured grid constellations, the algorithm resembles the FAC method (see, e.g., the works from McCormick et al. [9]), or possibly a hierarchical method (see, e.g., the papers from Yserentant [13], Bank et al. [2]) with a mortar method (see [1]).

We emphasize that the new aspect we introduce is to link the speed of convergence of this algorithm to the parameter $\tilde{\gamma}$, introduced here below, corresponding to the cosine of an abstract angle between the spaces V_h and V_H. Furthermore, an optimal relaxation keeps the method competitive in cases where the problem is badly conditioned (see Section 5).

3 Convergence analysis and consequences

We shall now analyse the convergence of the two-scale algorithm.[3] If $P_h : V_{Hh} \to V_h$ and $P_H : V_{Hh} \to V_H$ are orthogonal projectors from V_{Hh} upon V_h and V_H respectively with respect to the scalar product $a(\cdot,\cdot)$, and I denotes the identity operator in V_{Hh}, we set $B = (I - \omega P_H)(I - \omega P_h)$, and check that $u_{Hh} - u^n = B(u_{Hh} - u^{n-1})$.

We set $V_0 = V_H \cap V_h$ and $V_0^\perp$ the orthogonal complement of V_0 in V_{Hh} with respect to $a(\cdot,\cdot)$. We define $\tilde{V_h} = V_h \cap V_0^\perp$ and $\tilde{V}_H = V_H \cap V_0^\perp$. For $\omega \in (0;2)$ and $\tilde{\gamma} \in [0;1)$ defined by

$$\tilde{\gamma} = \begin{cases} \displaystyle\sup_{\substack{v_h \in \tilde{V}_h, v_h \neq 0 \\ v_H \in \tilde{V}_H, v_H \neq 0}} \frac{a(v_h, v_H)}{||v_h|| ||v_H||}, & \text{if } V_h \neq V_0 \text{ and } V_H \neq V_0, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

we introduce the functions

$$\rho(\tilde{\gamma}, \omega) = \begin{cases} \dfrac{\omega^2 \tilde{\gamma}^2}{2} - \omega + 1 + \dfrac{\omega\tilde{\gamma}}{2}\sqrt{\omega^2\tilde{\gamma}^2 - 4\omega + 4}, & \text{if } \omega \leq \omega_0(\tilde{\gamma}), \\ \omega - 1, & \text{otherwise}, \end{cases} \tag{5}$$

[3] An extension to a method using several patches has been analysed in [6].

where

$$\omega_0(\tilde{\gamma}) = \begin{cases} \dfrac{2 - 2\sqrt{1-\tilde{\gamma}^2}}{\tilde{\gamma}^2}, & \text{for } \tilde{\gamma} \in (0;1), \\ 1, & \text{for } \tilde{\gamma} = 0, \end{cases} \tag{6}$$

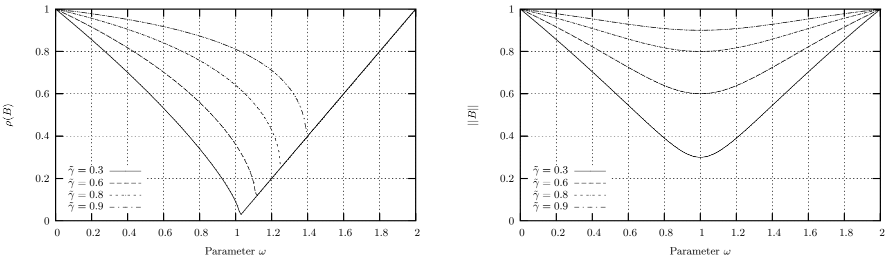
and $N(\tilde{\gamma},\omega) = \omega(2-\omega)\tilde{\gamma}/2 + \sqrt{\omega^2(2-\omega)^2\tilde{\gamma}^2/4 + (\omega-1)^2}$.

An abstract analysis of the spectral properties of the iteration operator B leads to the following result:

Proposition 3.

1. *If $\omega \in (0;2)$, then Algorithm 2 converges, i.e. $\lim_{n\to\infty} ||u^n - u_{Hh}|| = 0$.*
2. *The spectral norm of B induced by the scalar product $a(\cdot,\cdot)$ is given by $||B|| = N(\tilde{\gamma},\omega) < 1$, when $\omega \in (0;2)$.*
3. *The spectral radius of B is given by $\rho(B) = \rho(\tilde{\gamma},\omega) < 1$, when $\omega \in (0;2)$.*

Thus, we have the convergence of Algorithm 2 when $\omega \in (0;2)$, the convergence speed given by $\rho(B)$, and the factor of the reduction of the error in the norm $a(\cdot,\cdot)^{1/2}$ bounded by $||B||$. Both functions are plotted in the graphs of Fig 2. In the case $V_0 = \{0\}$, $\tilde{\gamma}$ corresponds to the constant of the strengthened Cauchy-Buniakowski-Schwarz inequality.



(a) $\rho(\omega)$ for different $\tilde{\gamma}$. (b) $||B||$ for different $\tilde{\gamma}$.

Fig. 2. Spectral radius and norm of B as a function of ω for different γ.

We remark that in [3], Bramble et al. present an abstract analysis of product iterative methods and provide an upper bound for the norm of B. Even an optimization of the constants appearing in this bound (see [6]) shows that the estimate is not always optimal. We also point out that the minimization of this known result with respect to ω does not lead to a significant value for the relaxation parameter. We show that the best convergence speed, i.e. a minimal spectral radius (5), is obtained for $\omega = \omega_0(\tilde{\gamma})$ given by (6).

Let us briefly consider a case where $\overline{\Lambda} \subset K$, for $K \in \mathcal{T}_H$ and $r = 1$. Let the scalar product be given by

$$a(\psi,\varphi)=\sum_{i,j=1}^{d}\int_{\Omega}a_{ij}\frac{\partial\psi}{\partial x_j}\frac{\partial\varphi}{\partial x_i}\,\mathrm{d}\mathbf{x},\quad \forall\psi,\varphi\in H_0^1(\Omega), \tag{7}$$

where $a_{ij}\in L^{\infty}(\Omega)$, $a_{ij}(x)=a_{ji}(x)$, $1\le i,j\le d$, and $\sum_{i,j=1}^{d}a_{ij}(x)\xi_i\xi_j\ge \alpha|\xi|^2,\forall\xi\in\mathbb{R}^d,\forall x\in\Omega$. Set $\beta=\left[\sum_{j=1}^{d}\left(\sum_{i=1}^{d}||\partial a_{ij}/\partial x_i||_{L^{\infty}(\Lambda)}\right)^2\right]^{\frac{1}{2}}$, and $\delta=\sqrt{1/\tilde{\lambda}}$, $\tilde{\lambda}$ being the Poincaré constant. In this case, we have $\tilde{\gamma}\le\dfrac{\beta\delta}{\alpha}$, i.e. an upper bound for the parameter $\tilde{\gamma}$. If furthermore the a_{ij}'s are constant over Λ, $1\le i,j\le d$, this last result implies that the algorithm converges in only one iteration.

A crucial question for running the algorithm is to know how to choose the relaxation parameter ω. By Prop. 3, if $\omega=1$, we have $\rho(B)=\tilde{\gamma}^2$. Furthermore, we can prove that $\rho(B)=\lim\limits_{n\to\infty}\sqrt[n]{||B^n u^0||}$. Hence, given an evaluation of $\tilde{\gamma}$, we obtain the optimal relaxation parameter $\omega^{\mathrm{opt}}=\omega_0(\tilde{\gamma})$ given by the formula (6). The parameter is optimal in the sense that it gives the minimum value for $\rho(B)$ directly related to the speed of convergence.

In practice, we set $\omega=1$ and $f\equiv 0$, and perform m steps of the algorithm to obtain some u^m. Following the above, we use the approximation $\rho=\sqrt[m]{||u^m||}$, and obtain with (6) and $\rho=\tilde{\gamma}^2$ that $\omega^{\mathrm{opt}}=\dfrac{2-2\sqrt{1-\rho}}{\rho}$.

Finally, we consider Algorithm 2 with two relaxation parameters ω_h and ω_H such that $u^{n-\frac{1}{2}}=u^{n-1}+\omega_h w_h$ and $u^n=u^{n-\frac{1}{2}}+\omega_H w_H$. We can prove that the spectral radius of the corresponding iteration operator is minimum when $\omega_H=\omega_h=\omega_0(\tilde{\gamma})$.

4 Implementation issues

We discuss practical aspects of constructing an efficient computer program for implementing Algorithm 2. Handling two domains with a priori nonconforming triangulations raises a couple of practical issues. At any stage the coarse and the fine parts of the solution u^n are stored separately, that is to say $u^{n-1}=u_H^{n-1}+u_h^{n-1}$ with $u_H^{n-1}\in V_H$, $u_h^{n-1}\in V_h$. We write the first step of the n-th iteration of the algorithm as follows:

Find $w_h\in V_h$ s.t. $a(w_h,\varphi)=\langle f|\varphi\rangle-a(u_H^{n-1},\varphi)-a(u_h^{n-1},\varphi),\forall\varphi\in V_h$.
Set $u_H^{n-\frac{1}{2}}=u_H^{n-1}$ and $u_h^{n-\frac{1}{2}}=u_h^{n-1}+\omega w_h$.

The same holds for the second step which appears explicitly:

Find $w_H \in V_H$ s.t. $a(w_H, \varphi) = \langle f|\varphi\rangle - a(u_H^{n-\frac{1}{2}}, \varphi) - a(u_h^{n-\frac{1}{2}}, \varphi), \forall \varphi \in V_H$.
Set $u_H^n = u_H^{n-\frac{1}{2}} + \omega w_H$ and $u_h^n = u_h^{n-\frac{1}{2}}$.

We conclude that $u_h^n = u_h^{n-1} + \omega w_h$ and $u_H^n = u_H^{n-1} + \omega w_H$.

At this point, we need to discuss the numerical integration and restrict ourselves to linear finite elements ($r = s = 1$).

Two difficulties are to be taken into account whether regions of rapid change, i.e., data needing fine meshes, of the problem comes from the right-hand side f or originates from the form a. In the first case the evaluation of $\langle f|\varphi\rangle$ needs particular attention. In the second case scalar products evaluated on the coarse grid must be considered with care. Another issue is the treatment of mixed term scalar products wherein both coarse and fine functions appear.

In the sequel, we consider these problems and illustrate our proposals with the scalar product given by (7). The evaluation of the different terms appearing in the algorithm is conforming to the following guidelines:

- If the coefficients a_{ij} defining the scalar product a are smooth in Λ, the homogeneous terms $a(\varphi_H, \psi_H)$ with $\varphi_H, \psi_H \in V_H$, and $a(\varphi_h, \psi_h)$ with $\varphi_h, \psi_h \in V_h$, of support in Ω resp. Λ are integrated using the grid $\mathcal{T}_H$ on Ω resp. $\mathcal{T}_h$ in Λ. Numerical integration in 2D is done with the standard three-point formula (in 3D we use a four-point formula). In the case of (7) this amounts to $\forall \varphi_H, \psi_H \in V_H$,

$$a(\varphi_H, \psi_H) \approx \sum_{K \in \mathcal{T}_H} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} \sum_{i,j=1}^{d} a_{ij}(\mathbf{x}_K^\alpha) \left.\frac{\partial \varphi_H}{\partial x_j}\right|_K \left.\frac{\partial \psi_H}{\partial x_i}\right|_K, \qquad (8)$$

  where $|K|$ denotes the area or volume, and $\mathbf{x}_K^\alpha$, $\alpha = 1, \ldots, d+1$, the vertices of the element K. We use the same formula for $a(\varphi_h, \psi_h)$ where $\varphi_h, \psi_h \in V_h$ with $K \in \mathcal{T}_h$ in (8).
  The mixed term $a(\varphi_h, \psi_H)$, $\varphi_h \in V_h$,$\psi_H \in V_H$, of support in Λ, is approximated by $a(\varphi_h, r_h\psi_H)$, i.e.

$$a(\varphi_h, \psi_H) \approx \sum_{K \in \mathcal{T}_h} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} \sum_{i,j=1}^{d} a_{ij}(\mathbf{x}_K^\alpha) \left.\frac{\partial \varphi_h}{\partial x_j}\right|_K \left.\frac{\partial (r_h\psi_H)}{\partial x_i}\right|_K, \qquad (9)$$

  where r_h is the standard interpolant to the space V_h. When implementing, we need to introduce a transmission grid, i.e. a fine structured grid considered over the patch Λ. This enables handling of the grids and associating fine and coarse triangles and vertices.
- If the coefficients a_{ij} are sharp in Λ, the presented approximation illustrated by (8) for the term $a(u_H^{n-\frac{1}{2}}, \varphi)$, $\varphi \in V_H$, appearing in the right-hand side of the coarse correction step needs to be rewritten in order to use a fine integration in the domain Λ. Set a_{ij}^1 and a_{ij}^2 such that $a_{ij} = a_{ij}^1 + a_{ij}^2$ and

$$a^1_{ij} = \begin{cases} a_{ij} & \text{in } \Omega \setminus \Lambda \\ 0 & \text{in } \Lambda \end{cases}, \qquad a^2_{ij} = \begin{cases} 0 & \text{in } \Omega \setminus \Lambda \\ a_{ij} & \text{in } \Lambda \end{cases}.$$

The right-hand side of relation (8) can be rewritten as $\forall \varphi_H, \psi_H \in V_H$,

$$\sum_{K \in \mathcal{T}_H} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} \sum_{i,j=1}^{d} a^1_{ij}(\mathbf{x}^\alpha_K) \left.\frac{\partial \varphi_H}{\partial x_j}\right|_K \left.\frac{\partial \psi_H}{\partial x_i}\right|_K$$
$$+ \sum_{K \in \mathcal{T}_h} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} \sum_{i,j=1}^{d} a^2_{ij}(\mathbf{x}^\alpha_K) \left.\frac{\partial (r_h\varphi_H)}{\partial x_j}\right|_K \left.\frac{\partial (r_h\psi_H)}{\partial x_i}\right|_K . \tag{10}$$

As our algorithm is a correction algorithm with corrections tending to zero, the left-hand side $a(w_H, \varphi)$, $\varphi \in V_H$, is not to be rewritten. All other terms already based on $\mathcal{T}_h$ for integration do not need to be revised.

- The term $\langle f|\varphi\rangle$, $\varphi \in V_h$ or V_H, is approximated with

$$\langle f|\varphi_H\rangle \approx \sum_{K \in \mathcal{T}_H} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} f^1(\mathbf{x}^\alpha_K)\varphi_H(\mathbf{x}^\alpha_K)$$
$$+ \sum_{K \in \mathcal{T}_h} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} f^2(\mathbf{x}^\alpha_K)(r_h\varphi_H)(\mathbf{x}^\alpha_K), \quad \forall \varphi_H \in V_H, \tag{11}$$

and

$$\langle f|\varphi_h\rangle \approx \sum_{K \in \mathcal{T}_h} \frac{|K|}{d+1} \sum_{\alpha=1}^{d+1} f^2(\mathbf{x}^\alpha_K)\varphi_h(\mathbf{x}^\alpha_K), \quad \forall \varphi_h \in V_h, \tag{12}$$

where $f = f^1 + f^2$ with $f^1 = \begin{cases} f & \text{in } \Omega \setminus \Lambda \\ 0 & \text{in } \Lambda \end{cases}$, and $f^2 = \begin{cases} 0 & \text{in } \Omega \setminus \Lambda \\ f & \text{in } \Lambda \end{cases}$.

5 Applications in 2D and 3D

We consider the Poisson-Dirichlet problem

$$\begin{cases} -\Delta u = f & \text{in } \ \Omega = (-1;1)^d, \ d = 2,3, \\ u = 0 & \text{on } \ \partial\Omega. \end{cases} \tag{13}$$

First, we implement the problem (13) in 2D ($d = 2$) to assess the convergence of Algorithm 2 with regard to the influence of the grids used. We take f such that the exact solution to the problem is given by $u = u_0 + \sum_{i=1}^{4} u_i$, $u_0(x,y) = \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)$ and $u_i(x,y) = \eta\chi(R_i)\exp\epsilon_f^{-2}\exp(-1/|\epsilon_f^2 - R_i^2|)$, where $R_i(x,y) = \sqrt{(x-x_i)^2 + (y-y_i)^2}$ and $\chi(R_i) = 1$ if $R_i \leq \epsilon_f$, $\chi(R_i) = 0$

if $R_i > \epsilon_f$; η, ϵ_f and (x_i, y_i), $i = 1, 2, 3, 4$ are parameters. Hence the right-hand side of (13) is given by $f = f_0 + \sum_{i=1}^{4} f_i$, where $f_0 = -\Delta u_0$ and $f_i = -\Delta u_i$, $i = 1, 2, 3, 4$. We choose $\eta = 10$, $\epsilon_f = 0.3$ and $(x_1, y_1) = (0.3, 0.3)$, $(x_2, y_2) = (0.7, 0.3)$, $(x_3, y_3) = (0.3, 0.7)$, $(x_4, y_4) = (0.7, 0.7)$.

For the triangulation of $\overline{\Omega}$, we use a coarse uniform grid with mesh size H and $r = 1$. We consider the patches Λ_i, $i = 1, 2, 3, 4$, with a fine uniform triangulation of size h and $s = 1$. Choose $\Lambda_i = (x_i - \epsilon; x_i + \epsilon) \times (y_i - \epsilon; y_i + \epsilon)$, with $\epsilon = 0.1$. We set $H = 2/N$ and $h = 2\epsilon/M$, N, M being the number of discretization points on one side of the squares Ω and Λ_i respectively.

In the following, we consider different situations including structured nested and non-nested as well as unstructured grids on the domain Ω. We always use the same structured grids for the patches. Our goal is to show that the algorithm performs well when $h \to 0$ for fixed H, and when each patch covers only a small number of coarse elements. It is particularly competitive when used with the optimal relaxation parameter in initially ill-conditioned situations (see Table 1(c), with small displacement of the nodes of the nested grid).

We introduce a stopping criterion for the algorithm, which controls the relative discrepancy $||u^n - u^{n-1}||/||u^n||$ between two iterations $n-1$ and n, $n = 1, 2, \ldots$, and measures the stagnation of the algorithm. We call n_{cvg} the number of iterations required for convergence. Conforming to our problem (13), $|| \cdot ||$ denotes here the H^1-seminorm.

All results are illustrated in the following table. In each part, we depict the considered situation by small graphics showing first the whole triangulation $\mathcal{T}_H$ with the patches, then a zoom to emphasize the region around one corner of a patch to show how $\mathcal{T}_h$ and $\mathcal{T}_H$ are related. First, we set $\omega = 1$ and run our method to obtain an estimate of $\tilde{\gamma}$ and hence of the spectral radius of the iteration operator, as discussed at the end of Section 3. Then we run the algorithm on problem (13) until convergence and report the number of iterations n_{cvg}. These values are, respectively, reported in the first rows of Tables 1(a)–1(c). Given the approximation for $\tilde{\gamma}$, we determine the optimal relaxation parameter with (6) and give the spectral radius. The last line in the tables reports the required iterations needed by the method to converge under optimal relaxation.

In a first test, we choose N and M such that the ratio H/h is of magnitude 10. In these first cases, the patches cover a small number of triangles of $\mathcal{T}_H$ leading to small coefficients $\tilde{\gamma}$ and ρ. Hence convergence is reached after a small number of iterations.

When doubling the number of fine triangles, see Table 1(b), the situation remains similar. A slight over-relaxation realises a gain of a couple of iterations. This suggests that the method is efficient in multi-scale situations, i.e. in problems with fixed H and $h \to 0$.

In the examples of Table 1(c), we increase the precision of the coarse triangulation. These cases show that the algorithm is best-suited to situations with patches covering a small number of coarse triangles. In fact, increasing the number of coarse triangles covered by the patches leads to bad condition numbers (ρ close to 1). Nevertheless optimal relaxation allows us to divide by a factor 2 the number of iterations necessary to obtain convergence. This shows that optimal relaxation is a key ingredient in our method.

These basic results show that the method is very well adapted for multiscale situations when applying small patches in the regions with large gradients.

Let us now turn to the 3D case ($d = 3$) of problem (13). We take f such that the exact solution to the problem is given by $u = u_0 + u_1$, $u_0(x, y, z) = \cos(\frac{\pi}{2}x)\cos(\frac{\pi}{2}y)\cos(\frac{\pi}{2}z)$ and $u_1(x, y, z) = \eta\chi(R)\exp\epsilon_f^{-2}\exp(-1/|\epsilon_f^2 - R^2|)$. where $R(x, y, z) = \sqrt{x^2 + y^2 + z^2}$. We choose $\eta = 10$, $\epsilon_f = 0.3$ and take $\Lambda = (-0.25, 0.25)^3$. We set $\omega = 1$. For the triangulation of $\overline{\Omega}$ resp. $\overline{\Lambda}$, we use a uniform structured grid with mesh size H resp. h. We set $H = 2/N$ and $h = 0.5/M$, N, M being the number of points per side of the cubes Ω and Λ. We use linear finite elements ($r = s = 1$). To assess the convergence of $u_{Hh} = u^{n_{\text{cvg}}}$ in H and h to the exact solution u,[4] we introduce the standard relative errors $e^n = ||u - u^n||/||u||$ and $e_{Hh} = e^{n_{\text{cvg}}} = ||u - u_{Hh}||/||u||$.

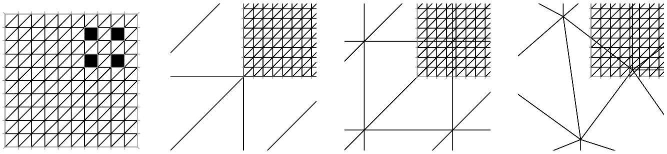
Consider the coarse triangulation ($N = 16, 32, 64$) with a patch $M = 8, 16, 32, 64$. We assess the quality of the estimate u^n at the iteration n of the algorithm by comparing it to the exact solution u. The results of e^n through n are depicted on Fig 3(a). Note that it is useful to run the algorithm through more than one iteration. Nevertheless only a couple of iterations are sufficient to obtain good results. As mentioned above, in the present cases the speed of convergence remains constant with respect to the refinement of the patch. When the error in $\Omega \setminus \Lambda$ dominates (case $N = 16$, $M = 32, 64$) a refinement of $\mathcal{T}_h$ does not improve the precision. The reduction of the error, in comparison with the sequence $M = 8$ to $M = 16$, stagnates.

Let us illustrate the efficiency of the method with respect to the memory usage. On one hand, we consider the computation of u_H on one grid with $N = 16, 32, 64$. On the other hand, we take a coarse grid ($N = 16$) with a fine grid in the patch $M = 8, 16, 32, 64$. In Fig. 3(b), we plot the error e_{Hh} with regard to the number of nodes used. Comparison of both curves leads us to conclude that the method is efficient in terms of memory usage. As above, the stagnation in the reduction of e_{Hh} stems from the error on the coarse grid becoming dominant. Similar results to those of memory usage can obtained for the CPU-time.

In Fig. 3(c), we illustrate the solution obtained after 5 iterations for the test case $N = 16$, $M = 32$.

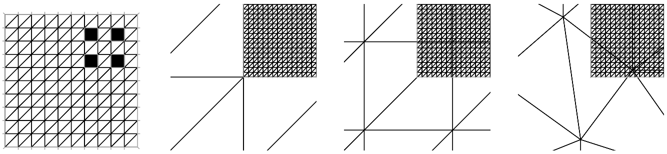
[4] An assessment of the convergence in H and h illustrating the *a priori* estimate (3) is given in [6], Fig. 6.

(a) $H/h = 10$ and $N = 10$.



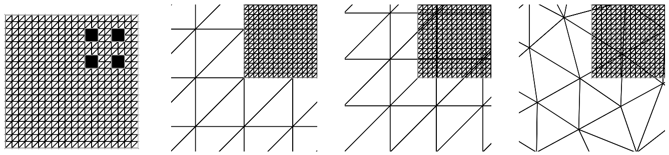
$H/h = 10$	nested $N = M = 10$	non-nested $N = 11,\ M = 10$	unstructured $N = M = 10$
$\rho(\tilde{\gamma}, 1) = \tilde{\gamma}^2$	0.28	0.30	0.34
n_{cvg}	6	8	8
$\rho(\tilde{\gamma}, \omega^{\text{opt}}) = \omega^{\text{opt}} - 1$	0.08	0.09	0.10
n_{cvg}	5	6	8

(b) $H/h = 20$ and $N = 10$.



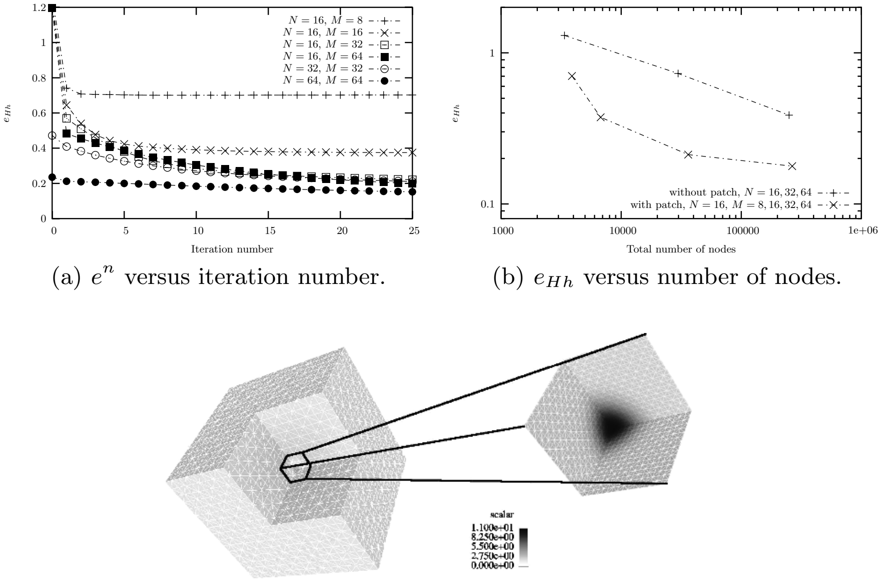
$H/h = 20$	nested $N = 10, M = 20$	non-nested $N = 11,\ M = 20$	unstructured $N = 10, M = 20$
$\rho(\tilde{\gamma}, 1) = \tilde{\gamma}^2$	0.28	0.31	0.38
n_{cvg}	6	8	9
$\rho(\tilde{\gamma}, \omega^{\text{opt}}) = \omega^{\text{opt}} - 1$	0.08	0.09	0.12
n_{cvg}	5	6	6

(c) $H/h = 20$ and $N = 20$.



$H/h = 10$	nested $N = M = 20$	non-nested $N = 21,\ M = 20$	unstructured $N = M = 20$
$\rho(\tilde{\gamma}, 1) = \tilde{\gamma}^2$	0.24	0.89	0.91
n_{cvg}	6	24	27
$\rho(\tilde{\gamma}, \omega^{\text{opt}}) = \omega^{\text{opt}} - 1$	0.07	0.50	0.54
n_{cvg}	5	13	15

Table 1. Comparison of the algorithm properties in 2D.



(a) e^n versus iteration number.

(b) e_{Hh} versus number of nodes.

(c) Estimate u^5 for $N = 16$, $M = 32$.

Fig. 3. Results in 3D and illustrations.

References

1. Y. Achdou and Y. Maday, *The mortar element method with overlapping subdomains*, SIAM J. Numer. Anal., 40 (2002), pp. 601–628.
2. R. E. Bank, T. F. Dupont, and H. Yserentant, *The hierarchical basis multigrid method*, Numer. Math., 52 (1988), pp. 427–458.
3. J. H. Bramble, J. E. Pasciak, J. Wang, and J. Xu, *Convergence estimates for multigrid algorithms without regularity assumptions*, Math. Comp., 57 (1991), pp. 23–45.
4. F. Brezzi, J.-L. Lions, and O. Pironneau, *Analysis of a Chimera method*, C.R. Acad. Sci. Paris, (2001), pp. 655–660.
5. T. F. Chan, B. F. Smith, and J. Zou, *Overlapping Schwarz methods on unstructured meshes using non-matching coarse grids*, Numer. Math., 73 (1996), pp. 149–167.
6. R. Glowinski, J. He, A. Lozinski, J. Rappaz, and J. Wagner, *Finite element approximation of multi-scale elliptic problems using patches of elements*, Numer. Math., 101 (2004), pp. 663–687.
7. R. Glowinski, J. He, J. Rappaz, and J. Wagner, *Approximation of multiscale elliptic problems using patches of finite elements*, C. R. Acad. Sci. Paris, Ser. I, 337 (2003), pp. 679–684.
8. S. McCormick and J. Thomas, *The fast adaptive composite grid (FAC) method for elliptic equations*, Math. Comp., 46 (1986), pp. 439–456.
9. S. F. McCormick and J. W. Ruge, *Unigrid for multigrid simulation*, Math. Comp., 41 (1983), pp. 43–62.

10. H. A. Schwarz, *Gesammelte Mathematische Abhandlungen*, vol. 2, AMS Chelsea Publishing, second ed., 1970, ch. Ueber einen Grenzübergang durch alternirendes Verfahren, pp. 133–143.
11. J. L. Steger and J. A. Benek, *On the use of composite grid schemes in computational aerodynamics*, Comp. Meth. Appl. Mech. Eng., 64 (1987), pp. 301–320.
12. J. Xu and L. Zikatanov, *The method of alternating projections and the method of subspace corrections in Hilbert space*, J. Amer. Math. Soc., 15 (2002), pp. 573–597.
13. H. Yserentant, *On the multi-level splitting of finite element spaces*, Numer. Math., 49 (1986), pp. 379–412.

On Preconditioned Uzawa-type Iterations for a Saddle Point Problem with Inequality Constraints

Carsten Gräser* and Ralf Kornhuber

Freie Universität Berlin, Fachbereich Mathematik und Informatik, Arnimallee 14,
D-14195 Berlin, Germany.

Summary. We consider preconditioned Uzawa iterations for a saddle point problem with inequality constraints as arising from an implicit time discretization of the Cahn-Hilliard equation with an obstacle potential. We present a new class of preconditioners based on linear Schur complements associated with successive approximations of the coincidence set. In numerical experiments, we found superlinear convergence and finite termination.

1 Introduction

Since their first appearance in the late fifties, Cahn-Hilliard equations have become the prototype class of phase-field models for separation processes, e.g., of binary alloys [7, 11, 19]. As a model problem, we consider the scalar Cahn-Hilliard equation with isotropic interfacial energy, constant mobilities and an obstacle potential [3, 4]. In particular, we concentrate on the fast solution of the algebraic spatial problems as resulting from an implicit time discretization and a finite element approximation in space [4]. Previous block Gauß-Seidel schemes [2] and the very popular ADI-type iteration by Lions and Mercier [18] suffer from rapidly deteriorating convergence rates for increasing refinement. In addition, the Lions-Mercier algorithm requires the solution of an unconstrained saddle point problem in each iteration step.

Our approach is based on a recent reformulation of the spatial problems in terms of a saddle point problem with inequality constraints [15]. Similar problems typically arise in optimal control. In contrast to interior point methods [22] or classical active set strategies we do not regularize or linearize the inequality constraints but directly apply a standard Uzawa iteration [14]. In order to speed up convergence, appropriate preconditioning is essential.

Preconditioning is well-understood in the linear case [1, 6, 12, 16] and variants for nonlinear and nonsmooth problems have been studied as well [8, 9].

*This work was supported in part by DFG under the grant KO 1806 3-1

However, little seems to be known about preconditioning of saddle point problems with inequality constraints or corresponding set-valued operators. For such kind of problems a reduced linear problem is recovered, once the exact coincidence set is known. In this case, preconditioning by the associated Schur complement would provide the exact solution in a single step. As the exact coincidence set is usually not available, our starting point for preconditioning is to use the Schur complement with respect to some approximation. General results by Glowinski et al. [14] provide convergence. To take advantage of the successive approximation of the coincidence set in the course of the iteration, it is natural to update the preconditioner in each step. In our numerical computations the resulting updated version shows superlinear convergence and finite termination. Previous block Gauß-Seidel schemes [2] are clearly outperformed. The convergence analysis and related inexact variants are considered elsewhere [15].

This paper is organized as follows. After a short review of the continuous problem and its discretization, we introduce the basic saddle point formulation. In Section 4 we present the Uzawa iterations and Section 5 is devoted to the construction of preconditioners. We conclude with some numerical experiments.

2 The Cahn-Hilliard equation with an obstacle potential

Let $\Omega \subset \mathbb{R}^2$ be a bounded domain. Then, for given $\gamma > 0$, final time $T > 0$ and initial condition $u_0 \in \mathcal{K} = \{v \in H^1(\Omega) : |v| \leq 1\}$, we consider the following initial value problem for the Cahn-Hilliard equation with an obstacle potential [3].

(P) Find $u \in H^1(0,T;(H^1(\Omega))') \cap L^\infty(0,T;H^1(\Omega))$ and $w \in L^2(0,T;H^1(\Omega))$ with $u(0) = u_0$ such that $u(t) \in \mathcal{K}$ and

$$\begin{aligned} \left\langle \frac{du}{dt}, v \right\rangle_{H^1(\Omega)} + (\nabla w, \nabla v) &= 0, && \forall v \in H^1(\Omega), \\ \gamma\left(\nabla u, \nabla v - \nabla u\right) - (u, v-u) &\geq (w, v-u)\,, && \forall v \in \mathcal{K} \end{aligned}$$

holds for a.e. $t \in (0,T)$.

Here $(\cdot,\cdot)$ stands for the L^2 scalar product and $\langle \cdot,\cdot \rangle_{H^1(\Omega)}$ is the duality pairing of $H^1(\Omega)$ and $H^1(\Omega)'$. The unknown functions u and w are called order parameter and chemical potential, respectively. The following existence and uniqueness result was shown by Blowey and Elliott [3].

Theorem 1. *Let $u_0 \in \mathcal{K}$ with $|(u_0,1)| < |\Omega|$. Then* **(P)** *has a unique solution.*

For simplicity, we assume that Ω has a polygonal boundary. Let $\mathcal{T}_h$ denote a triangulation of Ω with maximal diameter h and vertices $\mathcal{N}_h$. Then $\mathcal{S}_h$ is the corresponding space of linear finite elements spanned by the standard nodal basis φ_p, $p \in \mathcal{N}_h$. Using the lumped L^2 scalar product $\langle \cdot, \cdot \rangle$, we define the affine subspace $\mathcal{S}_{h,m} = \{v \in \mathcal{S}_h \mid \langle v, 1 \rangle = m\}$ with fixed mass m. Finally, $\mathcal{K}_h = \mathcal{K} \cap \mathcal{S}_h$ is an approximation of $\mathcal{K}$ and we set $\mathcal{K}_{h,m} = \mathcal{K} \cap \mathcal{S}_{h,m}$.

Semi-implicit Euler discretization in time and finite elements in space [2, 4, 13] lead to the following discretized problem.

($\mathbf{P}^h$) For each $k = 1, \dots, N$ find $u_h^k \in \mathcal{K}_h$ and $w_h^k \in \mathcal{S}_h$ such that

$$\begin{aligned} \langle u_h^k, v \rangle + \tau \left(\nabla w_h^k, \nabla v \right) &= \langle u_h^{k-1}, v \rangle, && \forall v \in \mathcal{S}_h, \\ \gamma \left(\nabla u_h^k, \nabla (v - u_h^k) \right) - \langle w_h^k, v - u_h^k \rangle &\geq \langle u_h^{k-1}, v - u_h^k \rangle, && \forall v \in \mathcal{K}_h. \end{aligned}$$

We select the uniform time step $\tau = T/N$. The initial condition $u_h^0 \in \mathcal{S}_h$ is the discrete L^2 projection of $u_0 \in \mathcal{K}$ given by $\langle u_h^0, v \rangle = (u_0, v)\ \forall v \in \mathcal{S}_h$. Note that the mass $m = \langle u_h^k, 1 \rangle = (u_0, 1)$, $k \geq 1$, is conserved in this way.

The following discrete analog of Theorem 1 is contained in [4], where optimal error estimates can be found as well.

Theorem 2. *There exists a solution (u_h^k, w_h^k) of* **($\mathbf{P}^h$)** *with uniquely determined u_h^k, $k = 1, \dots, N$. Moreover, w_h^k is also unique, if there is a $p \in \mathcal{N}_h$ with $|u_h^k(p)| < 1$.*

Note that non-uniqueness of w_h^k means that either the diffuse interface is not resolved by $\mathcal{T}_h$ or that u_h^k is constant.

3 A saddle point problem with inequality constraints

We consider the discrete Cahn-Hilliard system

(CH) Find $\mathbf{u} = (u, w) \in \mathcal{K}_h \times \mathcal{S}_h$ such that

$$\begin{aligned} \langle u, v \rangle + \tau (\nabla w, \nabla v) &= \langle u^{old}, v \rangle, && \forall v \in \mathcal{S}_h, \\ \gamma (\nabla u, \nabla (v - u)) - \langle w, v - u \rangle &\geq \langle u^{old}, v - u \rangle, && \forall v \in \mathcal{K}_h, \end{aligned}$$

for given $u^{old} \in \mathcal{S}_h$. Such a kind of problem arises in each time step of **($\mathbf{P}^h$)**.

Following [4, 15], we introduce the pde-constrained minimization problem

(M) Find $\mathbf{u_0} = (u, w_0) \in \mathcal{V} \subset \mathcal{K}_h \times \mathcal{S}_{h,0}$ such that

$$\mathcal{J}(\mathbf{u_0}) \leq J(\mathbf{v}) \qquad \forall \mathbf{v} \in \mathcal{V},$$

$$\mathcal{V} = \{(v_u, v_w) \in \mathcal{K}_h \times \mathcal{S}_{h,0} \mid \langle u^{old} - v_u, v \rangle - \tau (\nabla v_w, \nabla v) = 0\ \forall v \in \mathcal{S}_h\}.$$

Denoting $\mathbf{u_0} = (u, w_0), \mathbf{v} = (v_u, v_w)$, the bivariate energy functional

$$\mathcal{J}(\mathbf{u_0}) = \tfrac{1}{2} a(\mathbf{u_0}, \mathbf{u_0}) - \ell(\mathbf{u_0}), \qquad \mathbf{u_0} \in \mathcal{K}_h \times \mathcal{S}_{h,0}, \tag{1}$$

is induced by the bilinear form

$$a(\mathbf{u_0}, \mathbf{v}) = \gamma\,(\nabla u, \nabla v_u) + \gamma\,\langle u, 1\rangle\,\langle v_u, 1\rangle + \tau\,(\nabla w_0, \nabla v_w) \tag{2}$$

and the bounded linear functional

$$\ell(\mathbf{v}) = \gamma m\,\langle v_u, 1\rangle + \left\langle u^{old}, v_u\right\rangle. \tag{3}$$

The bilinear form $a(\cdot,\cdot)$ is symmetric and, by Friedrich's inequality, coercive with a constant independent of h on the Hilbert space $\mathcal{S}_h \times \mathcal{S}_{h,0}$ equipped with the inner product

$$(\mathbf{u_0}, \mathbf{v})_{\mathcal{S}_h \times \mathcal{S}_{h,0}} = \langle u, v_u\rangle + (\nabla u, \nabla v_u) + (\nabla w_0, \nabla v_w).$$

Hence, **(M)** has a unique solution (cf., e.g., [10, p. 34]).

Incorporating the pde-constraint $\mathbf{u_0} \in \mathcal{V}$ occurring in **(M)** by a Lagrange multiplier $\lambda \in \mathcal{S}_h$ we obtain the saddle point problem

(S) Find $(\mathbf{u_0}, \lambda) \in (\mathcal{K}_h \times \mathcal{S}_{h,0}) \times \mathcal{S}_h$ such that

$$\mathcal{L}(\mathbf{u_0}, \mu) \le \mathcal{L}(\mathbf{u_0}, \lambda) \le \mathcal{L}(\mathbf{v}, \lambda) \qquad \forall\,(\mathbf{v}, \mu) \in (\mathcal{K}_h \times \mathcal{S}_{h,0}) \times \mathcal{S}_h$$

with the Lagrange functional

$$\mathcal{L}(\mathbf{v}, \mu) = \mathcal{J}(\mathbf{v}) + \left\langle u^{old} - v_u, \mu\right\rangle - \tau(\nabla v_w, \nabla \mu).$$

It turns out that **(S)** is an equivalent reformulation of **(CH)** where the Lagrange parameter λ is identical with the chemical potential w. The following result is taken from [15].

Theorem 3. *Let* $\mathbf{u} = (u, w) \in \mathcal{K}_h \times \mathcal{S}_h$ *be a solution of* **(CH)***. Then* $\mathbf{u_0} = (u, w_0)$ *with* $w_0 = w - \int_\Omega w\,dx/|\Omega| \in \mathcal{S}_{h,0}$ *is the unique solution of* **(M)** *and* $(\mathbf{u_0}, w)$ *is a solution of* **(S)***. Conversely, if* $(\mathbf{u_0}, \lambda) = ((u, w_0), \lambda)$ *is a solution of* **(S)***, then* $\mathbf{u} = (u, \lambda)$ *solves* **(CH)***.*

4 Preconditioned Uzawa-type iterations

From now on, we concentrate on Uzawa-type iterations for the saddle point formulation (**S**) of the discrete Cahn-Hilliard system **(CH)**. In the light of Theorem 3, the Lagrange multiplier λ is identified with the chemical potential w. We first express the Lagrangian terms by a suitable operator Φ_S.

Lemma 1. *Let* $\langle\cdot,\cdot\rangle_S$ *be some inner product on* $\mathcal{S}_h$*. Then there is a unique Lipschitz continuous function* $\Phi_S : \mathcal{S}_h \times \mathcal{S}_{h,0} \to \mathcal{S}_h$ *with the property*

$$\left\langle u^{old} - v_u, \mu\right\rangle - \tau(\nabla v_w, \nabla \mu) = \langle \Phi_S(\mathbf{v}), \mu\rangle_S \qquad \forall \mu \in \mathcal{S}_h.$$

Furthermore $\langle \Phi_S(\cdot), \mu\rangle_S : \mathcal{S}_h \times \mathcal{S}_{h,0} \to \mathbb{R}$ *is Lipschitz continuous and convex.*

Proof. Existence and uniqueness follows directly from the representation theorem of Fréchet-Riesz. Since Φ_S is affine linear on the finite dimensional space $\mathcal{S}_h \times \mathcal{S}_{h,0}$, it is Lipschitz continuous. The same argument provides Lipschitz continuity and convexity of $\langle \Phi_S(\cdot), \mu \rangle_S$. □

Of course, Φ_S depends on the choice of the inner product $\langle \cdot, \cdot \rangle_S$ which plays the role of a preconditioner. For given $w^0 \in \mathcal{S}_h$ and $\rho > 0$ the corresponding Uzawa iteration reads as follows [14, p. 91].

Algorithm 1. (Preconditioned Uzawa iteration)

$$\begin{aligned} \mathbf{u_0^\nu} \in \mathcal{K}_h \times \mathcal{S}_{h,0} : \quad & \mathcal{L}(\mathbf{u_0^\nu}, w^\nu) \leq \mathcal{L}(\mathbf{v}, w^\nu) \quad \forall \mathbf{v} \in \mathcal{K}_h \times \mathcal{S}_{h,0} \\ & w^{\nu+1} = w^\nu + \rho \Phi_S(\mathbf{u_0^\nu}). \end{aligned} \tag{4}$$

As $a(\cdot, \cdot)$ is symmetric positive definite on $\mathcal{S}_h \times \mathcal{S}_{h,0}$ and $\mathcal{K}_h \times \mathcal{S}_{h,0}$ is a closed, convex subset, we can apply Theorem 4.1 in Chapter 2 of [14] to obtain

Theorem 4. *There are positive constants α_0, α_1 such that the iterates $\mathbf{u_0^\nu}$ provided by Algorithm 1 converge to $\mathbf{u_0}$ for $\nu \to \infty$ and all $\rho \in [\alpha_0, \alpha_1]$.*

In order to derive a more explicit formulation of Algorithm 1, it is convenient to introduce the identity I and the operators $A, C : \mathcal{S}_h \to \mathcal{S}_h$ according to

$$\langle Au, v \rangle = \gamma (\nabla u, \nabla v) + \gamma \langle u, 1 \rangle \langle v, 1 \rangle, \quad \langle Cw, v \rangle = \tau (\nabla w, \nabla v) \qquad \forall v \in \mathcal{S}_h$$

and the functions $f, g \in \mathcal{S}_h$ by

$$\langle f, v \rangle = \gamma m \langle v, 1 \rangle + \langle u^{old}, v \rangle \quad \forall v \in \mathcal{S}_h, \qquad g = -u^{old}.$$

Finally, $\partial I_{\mathcal{K}_h}$ is the subdifferential of the indicator function of $\mathcal{K}_h$. With this notation, the discrete Cahn-Hilliard system **(CH)** can be rewritten as the inclusion

$$\begin{pmatrix} A + \partial I_{\mathcal{K}_h} & -I \\ -I & -C \end{pmatrix} \begin{pmatrix} u \\ w \end{pmatrix} \ni \begin{pmatrix} f \\ g \end{pmatrix}. \tag{5}$$

Reformulating the minimization problem occurring in the first step of Algorithm 1 as a variational inclusion, we can eliminate w_0 and then insert the above operator notation to obtain the following explicit formulation

$$\begin{aligned} u^\nu &= (A + \partial I_{\mathcal{K}_h})^{-1}(f + w^\nu) \\ w^{\nu+1} &= w^\nu + \rho S^{-1}(-u^\nu - Cw^\nu - g) \end{aligned} \; . \tag{6}$$

The preconditioner $S : \mathcal{S}_h \to \mathcal{S}_h$ is the symmetric positive definite operator defined by

$$\langle Sr, v \rangle = \langle r, v \rangle_S \qquad \forall v \in \mathcal{S}_h.$$

Observe that (6) turns out to be a classical Uzawa iteration for the nonlinear, perturbed saddle point problem (5) with the preconditioner S.

5 Towards efficient preconditioning

In order to construct efficient preconditioners S, we have to find good approximations of the nonlinear Schur complement, i.e.,

$$S \approx (A + \partial I_{\mathcal{K}_h})^{-1} + C.$$

Our construction is based on the observation that the discrete Cahn-Hilliard system (5) degenerates to a reduced linear problem once the solution u on the coincidence set

$$\mathcal{N}_h^\bullet(u) = \{p \in \mathcal{N}_h \mid |u(p)| = 1\},$$

is known. To be more precise, we define the reduced linear operators

$$\begin{aligned} \left\langle \widehat{A}(u)\varphi_p, \varphi_q \right\rangle &= \begin{cases} \delta_{p,q} \langle \varphi_p, \varphi_q \rangle & \text{if } q \in \mathcal{N}_h^\bullet(u) \\ \langle A\varphi_p, \varphi_q \rangle & \text{else} \end{cases} \\ \left\langle \widehat{I}(u)\varphi_p, \varphi_q \right\rangle &= \begin{cases} 0 & \text{if } q \in \mathcal{N}_h^\bullet(u) \\ \langle \varphi_p, \varphi_q \rangle & \text{else} \end{cases} \end{aligned} \qquad p \in \mathcal{N}_h$$

and the right hand side

$$\left\langle \widehat{f}(u), \varphi_q \right\rangle = \begin{cases} u(q) \langle \varphi_q, \varphi_q \rangle & \text{if } q \in \mathcal{N}_h^\bullet(u) \\ \langle f, \varphi_q \rangle & \text{else} \end{cases}.$$

Recall that φ_p, $p \in \mathcal{N}_h$, denotes the standard nodal basis of $\mathcal{S}_h$. Then, by construction, the discrete Cahn-Hilliard system (5) has the same solution as the reduced linear system

$$\begin{pmatrix} \widehat{A}(u) & -\widehat{I}(u) \\ -I & -C \end{pmatrix} \begin{pmatrix} u \\ w \end{pmatrix} = \begin{pmatrix} \widehat{f}(u) \\ g \end{pmatrix}$$

with the Schur complement $S(u) = \widehat{A}(u)^{-1}\widehat{I}(u) + C$. Replacing the exact solution u by some approximation $\tilde{u} \approx u$, we obtain the preconditioner

$$S(\tilde{u}) = \widehat{A}(\tilde{u})^{-1}\widehat{I}(\tilde{u}) + C. \tag{7}$$

Proposition 1. *The operator $S(\tilde{u})$ is symmetric and positive semidefinite. $S(\tilde{u})$ is positive definite, if and only if $\mathcal{N}_h^\bullet(\tilde{u}) \neq \mathcal{N}_h$.*

Proof. First note that $\widehat{I}(\tilde{u}) : \mathcal{S}_h \to \mathcal{S}_h^\circ = \{v \in \mathcal{S}_h \mid v(p) = 0 \ \forall p \in \mathcal{N}_h^\bullet(\tilde{u})\}$ is orthogonal with respect to $\langle \cdot, \cdot \rangle$. The range of the restriction $A^\circ = \widehat{A}(\tilde{u})|_{\mathcal{S}_h^\circ}$ is contained in $\mathcal{S}_h^\circ$, because, for all $v \in \mathcal{S}_h^\circ$, we have by definition

$$\left\langle \widehat{A}(\tilde{u})v, \varphi_q \right\rangle = \sum_{p \in \mathcal{N}_h \setminus \mathcal{N}_h^\bullet(\tilde{u})} v(p)\delta_{p,q} \langle \varphi_p, \varphi_q \rangle = 0 \qquad \forall q \in \mathcal{N}_h^\bullet(\tilde{u}).$$

Similarly, we get $\langle A^\circ v, v'\rangle = \langle Av, v'\rangle \ \forall v, v' \in \mathcal{S}_h^\circ$ so that A° is symmetric and positive definite on $\mathcal{S}_h^\circ$. As a consequence, $\widehat{A}^{-1}(\tilde{u})\widehat{I}(\tilde{u})$ is symmetric and positive semidefinite on $\mathcal{S}_h$, because

$$\left\langle \widehat{A}^{-1}(\tilde{u})\widehat{I}(\tilde{u})v, v'\right\rangle = \left\langle (A^\circ)^{-1}\widehat{v}, v'\right\rangle = \left\langle \widehat{I}(\tilde{u})(A^\circ)^{-1}\widehat{v}, v'\right\rangle = \left\langle (A^\circ)^{-1}\widehat{v}, \widehat{v}'\right\rangle$$

denoting $\widehat{v} = \widehat{I}(\tilde{u})v$, $\widehat{v}' = \widehat{I}(\tilde{u})v'$. As C is also symmetric and positive semidefinite, the first assertion follows. It is easy to see that the kernels of $\widehat{A}^{-1}(\tilde{u})\widehat{I}(\tilde{u})$ and C have a trivial intersection, if and only if $\mathcal{N}_h^\bullet(\tilde{u}) \neq \mathcal{N}_h$. This concludes the proof. □

In the light of Theorem 4, Proposition 1 guarantees convergence of the preconditioned Uzawa iteration (6) with $S = S(\tilde{u})$ and suitable damping. The condition $\mathcal{N}_h^\bullet(u^\nu) \neq \mathcal{N}_h$ reflects the criterion $\mathcal{N}_h^\bullet(u) \neq \mathcal{N}_h$ for uniqueness of w (cf. Theorem 2). It could be removed, e.g., by imposing mass conservation $\left\langle w^{\nu+1}, 1\right\rangle = \langle w^\nu, 1\rangle$ in the singular case $\mathcal{N}_h^\bullet(\tilde{u}) = \mathcal{N}_h$.

As a straightforward approximation of u one may choose the first iterate $\tilde{u} = u^1$. It is natural to update $\tilde{u}$ in each iteration step, selecting $S = S(u^\nu)$. However, in this case convergence no longer follows from Theorem 4, because the preconditioner now depends on ν.

The following proposition is obtained by straightforward computation.

Proposition 2. *Let* $\mathcal{N}_h^\bullet(u^\nu) \neq \mathcal{N}_h$. *Then, for* $S = S(u^\nu)$ *and* $\rho = 1$ *the preconditioned Uzawa iteration* (6) *takes the form*

$$\begin{aligned} u^\nu &= (A + \partial I_{\mathcal{K}_h})^{-1}(f + w^\nu) \\ w^{\nu+1} &= S(u^\nu)^{-1}\left(-\widehat{A}(u^\nu)^{-1}\widehat{f}(u^\nu) - g\right) \end{aligned} \quad . \tag{8}$$

Note that only the actual coincidence set $\mathcal{N}_h^\bullet(u^\nu)$ and the values of u^ν on $\mathcal{N}_h^\bullet(u^\nu)$ enter the computation of $w^{\nu+1}$. Hence, (8) has the flavor of an active set strategy. As an important consequence, the Uzawa iteration (8) provides the exact solution, once the exact coincidence set $\mathcal{N}_h^\bullet(u)$ is detected. In the numerical experiments to be reported below, this required only a finite (quite moderate) number of steps. A theoretical justification will be discussed elsewhere [15].

Multigrid solvers for the subproblems. Each step of the preconditioned Uzawa iteration (8) requires a) the solution of a discretized symmetric elliptic obstacle problem with box constraints and b) the evaluation of the linear preconditioner $S(u^\nu)$.

For subproblem (8a), we apply monotone multigrid methods whose convergence speed is comparable to classical multigrid algorithms for unconstrained problems [17]. Moreover, in the non-degenerate case, the actual coincidence set $\mathcal{N}_h^\bullet(u^\nu)$ is detected after a finite number of steps. This means that we can stop the iteration on (8a) after a finite (usually quite moderate) number of steps without loosing exactness of the iteration (8). Using the Lipschitz-continuity

$$\langle A(u-u^\nu), u-u^\nu\rangle \leq \langle w-w^\nu, w-w^\nu\rangle$$

of (8a) with respect to w^ν, the *potential* accuracy of u^ν can be controlled by a posteriori estimates of the algebraic error of w^ν. Hence, the Uzawa iteration could be stopped and u^ν computed to the desired accuracy (only once!) as soon as w^ν is accurate enough.

The substep (8b) amounts to the solution of the following symmetric saddle point problem

$$\begin{pmatrix} \widehat{A}(u^\nu)\widehat{I}(u^\nu) & -\widehat{I}(u^\nu) \\ -\widehat{I}(u^\nu) & -C \end{pmatrix} \begin{pmatrix} \widehat{u} \\ w^{\nu+1} \end{pmatrix} = \begin{pmatrix} \tilde{f}(u^\nu) \\ \tilde{g}(u^\nu) \end{pmatrix} \tag{9}$$

with an auxiliary variable $\widehat{u}$ satisfying $\widehat{u} = u^\nu$ on $\mathcal{N}_h^\bullet(u^\nu)$ and the modified right-hand sides $\tilde{f}(u^\nu) = \widehat{f}(u^\nu) - \widehat{A}(u^\nu)(I-\widehat{I}(u^\nu))u^\nu$, $\tilde{g}(u^\nu) = g + (I-\widehat{I}(u^\nu))u^\nu$. For the iterative solution of (9) we apply a multigrid method with a block Gauß-Seidel smoother and canonical restriction and prolongation. Related algorithms have been investigated in [5, 20, 21, 23, 24]. In particular, multigrid convergence for a block Jacobi smoother is proved in [20].

6 Numerical experiments

We consider the Cahn-Hilliard equation **(P)** on the unit square $\Omega = (0,1)^2$ in the time interval $(0,T)$, $T = 0.5$, with $\gamma = 10^{-4}$ and its discretization by $(\mathbf{P}^h)$. The underlying triangulation $\mathcal{T}_{h_j}$ with meshsize $h_j = 2^{-j}$ results from

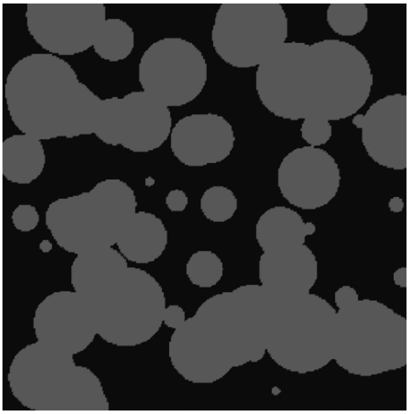
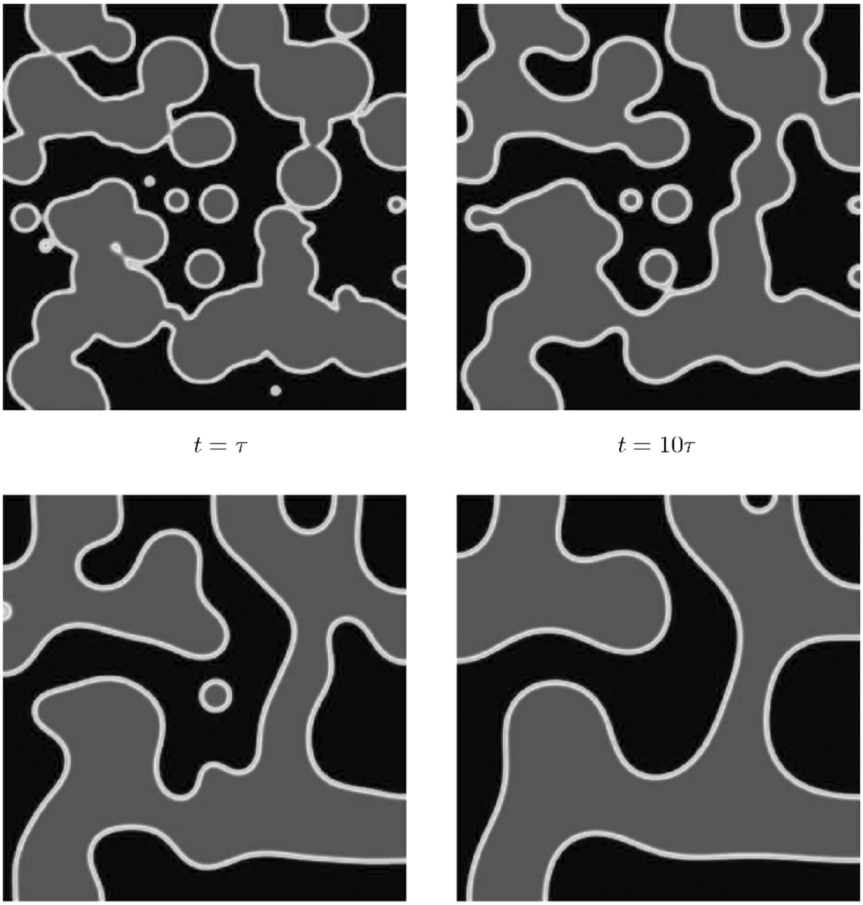


Fig. 1. Initial condition u_0

$j = 8$ uniform refinements applied to the initial triangulation $\mathcal{T}_{h_0}$ consisting of two congruent triangles. We choose the time step $\tau = \gamma$. Figure 2 illustrates the approximate solution for the initial condition u_0 as depicted in Figure 1. Observe that the initially fast dynamics slows down with decreasing curvature of the interface.



$t = \tau$

$t = 10\tau$

$t = 100\tau$

$t = 500\tau$

Fig. 2. Evolution of the phases

We now investigate the performance of the preconditioned Uzawa iteration (6). In all our experiments, we select $\rho = 1$, i.e. no damping is applied. As initial iterates w^0 we use the final approximations from the previous time step. The first time step is an exception, because no initial condition is prescribed for the chemical potential w. Here, we start with the the solution of the unconstrained reduced problem (9). Reduction takes place with respect to $\mathcal{N}_h^\bullet(u^0)$. The algebraic error is measured by the energy-type norm

$$\|\mathbf{v}\|^2 = a(\mathbf{v},\mathbf{v}) + \tau \langle v_w, v_w \rangle , \quad v = (v_u, v_w) \in \mathcal{S}_h \times \mathcal{S}_h,$$

with $a(\cdot,\cdot)$ defined in (2).

It turns out that preconditioning by $S(u^1)$ does not speed up, but slows down convergence considerably. Without preconditioning, the first spatial problem is solved to machine accuracy by about 3000 Uzawa steps. Using $S(u^1)$ as a preconditioner, 3000 steps only provide an error reduction by 10^{-1}.

From now on we only consider the preconditioner $S(u^\nu)$ which is updated in each iteration step $\nu \geq 0$. The resulting preconditioned Uzawa iteration is called uUzawa. Figure 3 illustrates the computational work for the solution of the spatial problems on the time levels $k = 1, \dots, 500$. The iteration is stopped as soon as the exact coincidence set is detected. The left picture shows the required number ν_0 of uUzawa steps. From 13 steps in the first time level, ν_0 drops down to 4 or 5 and later even to 2 or 3. This behavior clearly reflects the quality of the initial iterates w^0. The right picture shows the elapsed cpu time measured in terms of work units. One work unit is the cpu time required by one multigrid $V(3,3)$ cycle as applied to the unconstrained saddle point problem (9) on the actual refinement level j. About 15 multigrid steps are necessary to solve (9) to machine accuracy. Comparing both pictures, we find that the computational cost for each spatial problem is obtained approximately by multiplying that number with the number of Uzawa steps. The cpu time for the 4 to 7 monotone multigrid steps for detecting the actual coincidence set from each obstacle problem (8a) only plays a minor role.

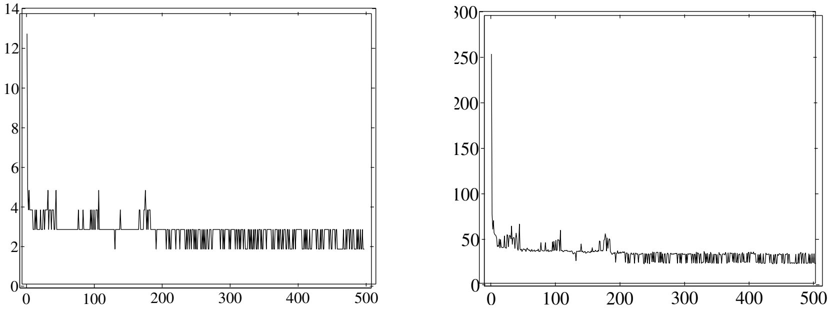


Fig. 3. Preconditioned Uzawa steps and cpu time over the time levels

To take a closer look at the convergence behavior of uUzawa, we now consider the iteration history on the first two time levels, using the refined mesh $\mathcal{T}_{h_j}$ with $j = 9$. Figure 4 shows the algebraic error $\|\mathbf{u} - \mathbf{u}^\nu\|$ over the cpu time measured in terms of work units. The "exact" solution $\mathbf{u}$ is precomputed to roundoff errors. For a comparison, we consider a recent block Gauß-Seidel iteration [2]. Reflecting the increasing accuracy of $\mathcal{N}_h^\bullet(u^\nu)$, uUzawa shows superlinear convergence throughout the whole iteration process, ending up with an error reduction by about 10^{-5} in the last iteration step. For bad initial iterates w^0, as encountered on the first time level, the efficiency of uUzawa and Gauß-Seidel is comparable in the beginning of the iteration. However, uUzawa speeds up considerably as soon as the coincidence set is approximated sufficiently well. For good initial iterates, as available on the second and all later time levels, such fast convergence takes place immediately. Even better initial iterates could be expected from nested iteration. While the convergence rates of the Gauß-Seidel scheme rapidly degenerate with decreasing mesh size, the convergence speed of uUzawa hardly depends on the refinement level. For example, the first spatial problem on the refinement levels $j = 7, 8, 9$ was solved to machine accuracy by $\nu_0 = 10, 12, 13$ iteration steps.

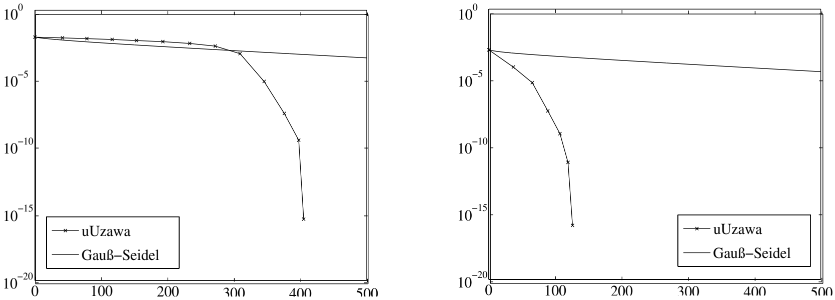


Fig. 4. Iteration history for the first 2 time levels

References

1. R. E. Bank, B. D. Welfert, and H. Yserentant, *A class of iterative methods for solving saddle point problems*, Numer. Math., 56 (1989), pp. 645–666.
2. J. W. Barrett, R. Nürnberg, and V. Styles, *Finite element approximation of a phase field model for void electromigration*, SIAM J. Numer. Anal., 42 (2004), pp. 738–772.
3. J. F. Blowey and C. M. Elliott, *The Cahn-Hilliard gradient theory for phase separation with non-smooth free energy Part I: Mathematical analysis*, European J. Appl. Math., 2 (1991), pp. 233–280.

4. ———, *The Cahn-Hilliard gradient theory for phase separation with non-smooth free energy Part II: Numerical analysis*, European J. Appl. Math., 3 (1992), pp. 147–179.
5. D. Braess and R. Sarazin, *An efficient smoother for the Stokes problem*, Appl. Numer. Math., 23 (1997), pp. 3–19.
6. J. H. Bramble, J. E. Pasciak, and A. T. Vassilev, *Analysis of the inexact Uzawa algorithm for saddle point problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1072–1092.
7. J. W. Cahn and J. E. Hilliard, *Free energy of a nonuniform system I. interfacial energy*, J. Chem. Phys., 28 (1958), pp. 258–267.
8. X. Chen, *Global and superlinear convergence of inexact Uzawa methods for saddle point problems with nondifferentiable mappings*, SIAM J. Numer. Anal., 35 (1998), pp. 1130–1148.
9. ———, *On preconditioned Uzawa methods and SOR methods for saddle-point problems*, J. Comput. Appl. Math., 100 (1998), pp. 207–224.
10. I. Ekeland and R. Temam, *Convex analysis and variational problems*, North-Holland, Amsterdam, 1976.
11. C. M. Elliott, *The Cahn-Hilliard model for the kinetics of phase separation*, in Mathematical models for phase change problems, J. F. Rodrigues, ed., Basel, 1989, Birkhäuser, pp. 35–73.
12. H. C. Elman and G. H. Golub, *Inexact and preconditioned Uzawa algorithms for saddle point problems*, SIAM J. Numer. Anal., (1994), pp. 1645–1661.
13. D. J. Eyre, *An unconditionally stable one-step scheme for gradient systems*, tech. rep., University of Utah, Salt Lake City, UT, 1998.
14. R. Glowinski, J. L. Lions, and R. Trémolières, *Numerical Analysis of Variational Inequalities*, no. 8 in Studies in Mathematics and its Applications, North-Holland Publishing Company, Amsterdam, 1981.
15. C. Gräser and R. Kornhuber, *Preconditioned Uzawa iterations for the Cahn-Hilliard equation with obstacle potential.* To appear.
16. Q. Hu and J. Zou, *Two new variants of nonlinear inexact Uzawa algorithms for saddle-point problems*, Numer. Math., 93 (2002), pp. 333–359.
17. R. Kornhuber, *Monotone multigrid methods for elliptic variational inequalities I*, Numer. Math., 69 (1994), pp. 167–184.
18. P. Lions and B. Mercier, *Splitting algorithms for the sum of two nonlinear operators*, SIAM J. Numer. Anal., 16 (1979), pp. 964–979.
19. A. Novick-Cohen, *The Cahn-Hilliard equation: Mathematical and modeling perspectives*, Adv. Math. Sci. Appl., 8 (1998), pp. 965–985.
20. J. Schöberl and W. Zulehner, *On Schwarz-type smoothers for saddle point problems*, Numer. Math., 95 (2003), pp. 377–399.
21. S. P. Vanka, *Block-implicit multigrid solution of Navier-Stokes equations in primitive variables*, J. Comput. Phys., 65 (1986), pp. 138–158.
22. S. J. Wright, *Primal-dual interior-point methods*, SIAM, Philadelphia, PA, 1997.
23. W. Zulehner, *A class of smoothers for saddle point problems*, Computing, 65 (2000), pp. 227–246.
24. ———, *Analysis of iterative methods for saddle point problems: A unified approach*, Math. Comp., 71 (2002), pp. 479–505.

Multilevel Methods for Eigenspace Computations in Structural Dynamics

Ulrich L. Hetmaniuk and Richard B. Lehoucq

Sandia National Laboratories [†], P.O. Box 5800, MS 1110, Albuquerque, NM 87185-1110, USA. ulhetma@sandia.gov, rblehou@sandia.gov.

Summary. Modal analysis of three-dimensional structures frequently involves finite element discretizations with millions of unknowns and requires computing hundreds or thousands of eigenpairs. We review in this paper methods based on domain decomposition for such eigenspace computations in structural dynamics. We distinguish approaches that solve the eigenproblem algebraically (with minimal connections to the underlying partial differential equation) from approaches that couple tightly the eigensolver with the partial differential equation.

1 Introduction

The goal of our paper is to provide a brief review of multilevel methods for eigenspace computations in structural dynamics. Our review is not meant to be exhaustive and so we apologize for relevant work not discussed. In particular, our interest is in multilevel algorithms for the numerical solution of the algebraic generalized eigenvalue problem arising from the finite element discretization of three-dimensional structures. Our interest is also restricted to methods that are scalable, both with respect to the mesh size and the number of processors of extremely large distributed-memory architectures. We start our paper by a formal discussion of the origin of the eigenvalue problem.

The dynamic analysis of a three-dimensional structure is modeled by the hyperbolic partial differential equation

$$\rho \frac{\partial^2 \mathbf{u}}{\partial t^2} - \mathcal{E}(\mathbf{u}) = \mathbf{f}(t) \quad \text{in } \Omega \tag{1}$$

where $\mathbf{u}$ is the vector of displacements, $\mathcal{E}$ is a self-adjoint elliptic differential operator, ρ is the mass density, and $\mathbf{f}$ is a vector function for loading. We assume that appropriate homogeneous boundary and initial conditions are specified on the three-dimensional simply connected domain Ω.

[†] Sandia is a multiprogram laboratory operated by Sandia Corporation, a Lockheed Martin Company, for the U.S. Department of Energy under contract DE-AC04-94AL85000.

Structural dynamic analyses are usually divided into two categories: frequency response and transient simulation. In the former category, natural frequencies of the structure and their mode shapes are determined to verify their separation from frequencies of excitation or to compute the response from a given input force at a given location. In the second category, we study the motion of the structure and its time history under prescribed loads. For these dynamic response problems, several solution methods are available and we refer the reader to [16] and the references therein for an overview. Often, modal analysis is an effective solution method because, due to the orthogonality of the modes, modal superposition gives the solution. In addition, the frequency range of excitation is usually in the low end of the natural frequencies of the structure. Consequently, high frequency modes have a much lower participation in the response than lower modes and the contribution of high frequency modes can be neglected.

The vibration frequencies and mode shapes of the structure are solutions of the problem

$$-\mathcal{E}(\mathbf{u}) = \lambda \rho \mathbf{u} \quad \text{in } \Omega \tag{2}$$

with the same homogeneous boundary conditions as (1). The eigenvalue λ is the square of the natural frequency ω. A finite element discretization of the weak form of the vibrational problem (2) leads to the generalized eigenvalue problem

$$\mathbf{K}\mathbf{u}^h = \mathbf{M}\mathbf{u}^h \lambda^h \tag{3}$$

where $\mathbf{K}$ and $\mathbf{M}$ are the stiffness and mass matrices of order n respectively that represent the elastic and inertial properties of a structure. The parameter h is the characteristic mesh size. We assume a choice of boundary conditions such that both matrices are symmetric and positive definite.

Finite element discretizations of three-dimensional structures frequently involve well over one million unknowns and modal truncation requires often hundreds or thousands of eigenpairs. Consequently, computing these eigenpairs results in a challenging linear algebra problem. The remainder of our paper reviews two approaches that can be used to compute the needed modes. We will focus on techniques to compute eigenpairs in the low end of the spectrum for two reasons. First, the frequency range of excitation and the dominant modes for the structural response are in the low end of the natural frequencies. Secondly, standard results from finite element theory [3, 48] give the following *a priori* error estimates

$$\lambda \leq \lambda^h \leq \lambda(1 + Ch^2\lambda), \tag{4}$$

assuming sufficient regularity. These estimates imply that the finite element discretization represents more accurately the modes with small natural frequency.

Our paper is organized as follows. Section 2 describes algebraic approaches to solve the eigenvalue problem (3). Section 3 discusses variational methods tightly coupled to the partial differential operator $\mathcal{E}$.

2 Algebraic approach

A popular approach is to use a block Lanczos [26] code with a shift-invert transformation $(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}$. If σ is a real number, then the standard eigenvalue problem

$$(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{u}^h = \mathbf{u}^h\nu, \quad \left(\nu = \frac{1}{\lambda^h - \sigma}\right), \tag{5}$$

results by subtracting $\sigma\mathbf{M}$ from both sides of (3) followed by *cross-multiplication.* This standard eigenvalue problem is no longer symmetric. However, a careful choice of inner product renders the operator $(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}$ symmetric (for instance, the $\mathbf{M}$-inner product).

The Lanczos algorithm builds iteratively a basis for the Krylov subspace

$$\mathcal{K}_{m+1} = \text{span}\{\mathbf{x}_0, (\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}\mathbf{x}_0, \cdots, [(\mathbf{K} - \sigma\mathbf{M})^{-1}\mathbf{M}]^m\mathbf{x}_0\} \tag{6}$$

to approximate the eigenpairs (see [20, 26, 34] for further details). At every Lanczos iteration, the action of $(\mathbf{K} - \sigma\mathbf{M})^{-1}$ on a vector or a block of vectors is required. Grimes et al. [26] solve the resulting set of linear equations by forward and backward substitution with the factors computed by a sparse direct factorization. However, performing sparse direct factorizations becomes prohibitively expensive when the dimension n is large or when the distributed-memory architecture has a large number of processors.

Other solutions are the following:

- replace the sparse direct method with a preconditioned iterative linear solver within the shift-invert Lanczos algorithm;
- replace the shift-invert Lanczos algorithm with a preconditioned eigenvalue algorithm.

These approaches are not new and we propose to review them.

For the first approach, most structural analysts choose a shift σ^*, $\sigma^* < \lambda_1^h$, so that the matrix $\mathbf{K} - \sigma^*\mathbf{M}$ is symmetric positive definite. This choice is motivated by the availability of scalable preconditioners for symmetric positive definite matrices. A scalable preconditioner for $\mathbf{K} - \sigma^*\mathbf{M}$ is desirable because the rate of convergence of the resulting preconditioned conjugate gradient iteration is independent of the mesh size and the number of processors. Recently, Farhat et al. [22] proposed a new iterative solver for symmetric indefinite matrices, *i.e.* allowing an arbitrary shift σ. Numerical experiments showed the scalability of the solver. However, to the best of our knowledge, their approach for symmetric indefinite matrices has not been coupled with a shift-invert Lanczos algorithm.

For a shift σ^* such that $\sigma^* < \lambda_1^h$, choices of scalable iterative linear solvers include FETI-DP [21], the conjugate gradient preconditioned by balanced domain-decompostion (BDDC) [19], or the conjugate gradient preconditioned by algebraic multigrid (AMG) [50, 49, 1]. No comparison is available to assess the quality of each combination. However, an efficient algorithm has been developed at Sandia National Laboratories.

Salinas [7, 8, 44] is a massively parallel implementation of finite element analysis for structural dynamics. This capability is required for high-fidelity validated models used in modal, vibrations, static, and shock analysis of weapons systems. A critical component of Salinas is scalable iterative linear algebra. The modal analysis is computed with a shift-invert Lanczos method (for a shift $\sigma^* < \lambda_1^h$) using parallel ARPACK [34, 38] and the FETI-DP iterative linear solver [23, 21]. Because the shift-invert Lanczos iteration used by ARPACK makes repeated calls to FETI-DP, the projected conjugate iteration used for computing the Lagrange multipliers retains a history of vectors computed during each FETI-DP invocation. After the first

FETI-DP call by ARPACK, the right-hand side in the projected conjugate iteration is first orthogonalized against this history of vectors. The number of projected conjugate iterations is therefore reduced as the number of Lanczos iterations needed by ARPACK increases. Besides the capability developed for Salinas, the authors are not aware of any multilevel-based modal analysis capabilities for use within a three-dimensional structural dynamics code.

Replacements for the shift-invert Lanczos algorithm include gradient schemes that attempt to minimize the Rayleigh quotient and Newton schemes that search for stationary points of the Rayleigh quotient. The gradient schemes include conjugate gradient algorithms [6, 24, 28, 31, 35, 41]. The Newton-based schemes include the Davidson-based methods [18] such as the Jacobi-Davidson algorithm [47].

All the algorithms perform a Rayleigh-Ritz analysis on a subspace $\mathcal{S}$ that is computed iteratively. At the $(m+1)$-th iteration, the current subspace $\mathcal{S}_{m+1}$ satisfies

$$\mathcal{S}_{m+1} \subset \operatorname{span}(\mathcal{S}_m, \mathbf{N}^{-1}\mathbf{R}^{(m)}) \tag{7}$$

where $\mathbf{R}^{(m)}$ is the block vector of residuals

$$\mathbf{R}^{(m)} = \mathbf{K}\mathbf{X}^{(m)} - \mathbf{M}\mathbf{X}^{(m)}\Theta^{(m)}.$$

The current iterates $\mathbf{X}^{(m)}$ are the best eigenvector approximations for $(\mathbf{K}, \mathbf{M})$ in the subspace $\mathcal{S}_m$. The matrix $\Theta^{(m)}$ is diagonal and contains the Rayleigh quotients for the iterates $\mathbf{X}^{(m)}$.

The motivation for these preconditioned eigenvalue algorithms is to avoid the requirement for a linear solve so that a single application of a preconditioner per outer iteration can be used. So $\mathbf{N}$, applied in equation (7), is in general a preconditioner for the matrix $\mathbf{K}$ (the Jacobi-Davidson algorithm is one exception, see [47] for further details). Good preconditioners are a prerequisite for any of the preconditioned algorithms to perform satisfactorily. If a scalable preconditioner $\mathbf{N}$ is available for $\mathbf{K}$, then this preconditioner is a candidate for use within a preconditioned eigenvalue algorithm. Although less studied, preconditioned iterations for the eigenvalue problem should also be independent of the mesh size. The reader is referred to [30, 32] and [42, 43] for a review of the many issues involved and convergence theory, respectively. These papers also contain numerous citations to the engineering and numerical analysis literature.

Finally, little information is available that compares the merits of shift-invert Lanczos methods versus preconditioned eigensolvers when hundreds or thousands of eigenpairs are to be computed. In particular, practical experience with preconditioned algorithms for computing eigenpairs in an interval inside the spectrum is lacking. The paper [2] compares a number of preconditioned algorithms with the shift-invert Lanczos method (for a shift $\sigma^* < \lambda_1^h$) on several large-scale eigenvalue problems arising in structural dynamics when an algebraic multigrid preconditioner is available. For these particular engineering problems, the preconditioned algorithms were competitive when the preconditioner is applied in a block fashion and the block size is selected appropriately.

Ultimately, maintaining numerical orthogonality of the basis vectors is the dominant cost of the modal analysis as the number of eigenpairs requested increases. The cost is quadratic in the number of basis vectors. The cost of maintaining numerical orthogonality is a crucial limitation that motivates the next approach.

3 Variational approach

The previous section described schemes where knowledge of the partial differential equation is only required through the application of a linear solver or a preconditioner. In contrast, the approaches in this section make extensive use of the variational form of the equation.

The leading method in the automotive industry to compute hundreds or thousands of eigenpairs is the automated multilevel substructuring method (AMLS) [4, 5]. For example, in [33], the authors show how AMLS is more efficient than the shift-invert Lanczos method [26] coupled with a sparse direct solver to compute a large number of eigenpairs for two-dimensional problems. AMLS is a variation of a component mode synthesis technique (CMS). Component mode synthesis techniques [29, 17] originated in the aerospace engineering community . These schemes decompose a structure into numerous components (or substructures), determine component modes, and then synthesize these modes to approximate the eigenpairs of (3). Their goal is to generate approximations that aptly describe the low frequency modal subspace rather than to solve iteratively the eigenproblem. The reader is referred to [46] for a review of CMS methods from a structural dynamics perspective. The variational formulation and analysis of classical CMS techniques is due to Bourquin [9, 10, 11].

To make the process concrete, suppose that the structure Ω is divided into two subdomains Ω_1 and Ω_2 with the common interface Γ. We look for solutions of

$$-\mathcal{E}(\mathbf{u}) = \lambda\rho\mathbf{u} \quad \text{in } \Omega \tag{8a}$$

$$\mathbf{u} = \mathbf{0} \qquad \text{on } \partial\Omega. \tag{8b}$$

Let $(\mathbf{u}_j^1)_{1\le j\le m_1}$ (resp. $(\mathbf{u}_j^2)_{1\le j\le m_2}$) represent eigenvectors on Ω_1 (resp. Ω_2) for the same operator $\mathcal{E}$ with homogeneous Dirichlet boundary conditions on $\partial\Omega\cap\partial\Omega_1$ (resp. on $\partial\Omega\cap\partial\Omega_2$) and specific boundary conditions on Γ that will be discussed later. Component mode synthesis techniques compute approximations to eigenpairs of (8) via a Rayleigh-Ritz analysis on an appropriate subspace coupling the information spanned by the vectors $(\mathbf{u}_j^1)_{1\le j\le m_1}$ and $(\mathbf{u}_j^2)_{1\le j\le m_2}$. These techniques differ by the boundary conditions specified on Γ and by the definition of the *coupling* subspace. In practice, the eigenpairs on Ω_1 and Ω_2 are discretized by finite elements and are computed numerically.

The family of *fixed interface* CMS methods was introduced by Hurty [29] and improved by Craig and Bampton [17]. *Fixed interface* methods impose homogeneous Dirichlet boundary condition along the interface Γ. Coupling between the local sets of vectors $(\mathbf{u}_j^1)_{1\le j\le m_1}$ and $(\mathbf{u}_j^2)_{1\le j\le m_2}$ is achieved by adding a set of vectors defined on Γ harmonically extended into Ω. The definition of these coupling vectors distinguishes the various *fixed interface* CMS methods.

Other researchers proposed *free interface* methods where a homogeneous Neumann boundary condition is imposed on Γ. Continuity on Γ for the approximation of the eigenvectors of (3) is enforced so that constraints with Lagrange multipliers appear in a subspace [36] for the final Rayleigh-Ritz analysis. The recent paper by Rixen [45] reviews several CMS techniques and introduces a dual *fixed interface* method. For a one-dimensional model problem, Bourquin [9] showed that a *fixed interface* method better approximates the eigenspace than a *free interface* method. Consequently, we focus our discussion on *fixed interface* methods.

AMLS [5] is a *fixed interface* method where the coupling modes are harmonic extension of eigenmodes for the Steklov-Poincaré and the mass complement operators. After a finite element discretization, the mass and stiffness matrices are ordered as follows, for two subdomains,

$$\mathbf{M} = \begin{bmatrix} \mathbf{M}_{\Omega_1} & \mathbf{0} & \mathbf{M}_{\Omega_1,\Gamma} \\ \mathbf{0} & \mathbf{M}_{\Omega_2} & \mathbf{M}_{\Omega_2,\Gamma} \\ \mathbf{M}_{\Omega_1,\Gamma}^T & \mathbf{M}_{\Omega_2,\Gamma}^T & \mathbf{M}_{\Gamma} \end{bmatrix} \quad \text{and} \quad \mathbf{K} = \begin{bmatrix} \mathbf{K}_{\Omega_1} & \mathbf{0} & \mathbf{K}_{\Omega_1,\Gamma} \\ \mathbf{0} & \mathbf{K}_{\Omega_2} & \mathbf{K}_{\Omega_2,\Gamma} \\ \mathbf{K}_{\Omega_1,\Gamma}^T & \mathbf{K}_{\Omega_2,\Gamma}^T & \mathbf{K}_{\Gamma} \end{bmatrix}. \qquad (9)$$

The coupling mode pencil is $(\tilde{\mathbf{K}}_\Gamma, \tilde{\mathbf{M}}_\Gamma)$, where

$$\tilde{\mathbf{K}}_\Gamma = \mathbf{K}_\Gamma - \sum_{i=1}^{2} \mathbf{K}_{\Omega_i,\Gamma}^T \mathbf{K}_{\Omega_i}^{-1} \mathbf{K}_{\Omega_i,\Gamma}$$

and $\tilde{\mathbf{M}}_\Gamma$,

$$\mathbf{M}_\Gamma - \sum_{i=1}^{2} \left(\mathbf{K}_{\Omega_i,\Gamma}^T \mathbf{K}_{\Omega_i}^{-1} \mathbf{M}_{\Omega_i,\Gamma} + \mathbf{M}_{\Omega_i,\Gamma}^T \mathbf{K}_{\Omega_i}^{-1} \mathbf{K}_{\Omega_i,\Gamma} - \mathbf{K}_{\Omega_i,\Gamma}^T \mathbf{K}_{\Omega_i}^{-1} \mathbf{M}_{\Omega_i} \mathbf{K}_{\Omega_i}^{-1} \mathbf{K}_{\Omega_i,\Gamma} \right),$$

are the Schur and mass complement matrices. The AMLS method forms these interface matrices and factors the Schur complement. For the case of two subdomains, AMLS is summarized in the following three steps

1. Compute local eigenvectors $(\mathbf{u}_j^1)_{1 \le j \le m_1}$ and $(\mathbf{u}_j^2)_{1 \le j \le m_2}$.
2. Compute coupling modes $(\mathbf{u}_j^\Gamma)_{1 \le j \le m_\Gamma}$ for the pencil $(\tilde{\mathbf{K}}_\Gamma, \tilde{\mathbf{M}}_\Gamma)$.
3. Perform a Rayleigh-Ritz analysis for the pencil $(\mathbf{K}, \mathbf{M})$ on the subspace
$$\text{span}\left\{ (\mathbf{u}_j^1)_{1 \le j \le m_1}, (\mathbf{u}_j^2)_{1 \le j \le m_2}, (E\mathbf{u}_j^\Gamma)_{1 \le j \le m_\Gamma} \right\}$$
where E denotes the harmonic extension.

For large structures, AMLS recursively divides the structure into thousands of substructures and associated interfaces. This nested decomposition results in a hierarchical tree of substructures and interfaces or, analytically, in a direct sum decomposition of $\left(H_0^1(\Omega)\right)^3$ into orthogonal subspaces. The paper [5] examines a mathematical basis for AMLS in the continuous variational setting and the resulting algebraic formulation. AMLS computes efficiently a large number of eigenpairs because the orthogonalizations of large scale vectors are eliminated. The orthogonality of the approximations is obtained by the final Rayleigh-Ritz analysis. Unfortunately, AMLS is not well suited to three-dimensional eigenvalue problems when solid elements are used. Indeed, AMLS supposes that the interface matrices are formed and, sometimes, factored. Consequently, the cost of AMLS is that of computing a sparse direct factorization for the stiffness matrix using multifrontal methods. As is well known, sparse direct methods are not scalable with respect to mesh or the number of processors.

An alternative to AMLS is to not form the Schur and mass complements. In this case, we do not subdivide the interface into a hierarchy but consider one interface. A preconditioner for the Schur complement, for instance BDDC [19], can be used within a preconditioned eigensolver for the interface eigenvalue problem. Although the interface problem is reduced in size over that of the order of (3), the application of the mass and Schur complements matrices and of the Schur complement preconditioner remains expensive. Bourquin [10] and Namar [39] consider different pencils to compute the coupling interface modes. But defining the most efficient choice of pencil remains an open question.

Finally, we comment on the eigenspace error. Bourquin [9, 10, 11] derived asymptotic results for second order elliptic differential eigenvalue problems and their finite element discretization. The error in the eigenspace computed by a CMS technique depends upon the error due to modal truncation and discretization. The bounds of Bourquin also indicate that the number of coupling modes necessary may become small when the interface Γ is small. Similarly, when the subdomains are small, the number of local modes needed is small. For further details, we refer the reader to [9, 10, 11].

To conclude this section, we review overlapping techniques to compute approximations for the eigenproblem (3). Charpentier et al. [15] defined a component mode synthesis technique using overlapping subdomains. Their approach simplifies the definition of the *coupling* space as it just combines the local sets of vectors from each subdomain. But performing the final Rayleigh-Ritz analysis on this subspace is more complex because the decomposition of $\left(H_0^1(\Omega)\right)^3$ is not a direct sum and the local sets of vectors lack orthogonality properties.

In analogy to multiplicative Schwarz preconditioners, Chan and Sharapov [14] define a multilevel technique that minimizes the Rayleigh quotient

$$\min_{\mathbf{x}\neq\mathbf{0}} \frac{\mathbf{x}^T\mathbf{K}\mathbf{x}}{\mathbf{x}^T\mathbf{M}\mathbf{x}} \tag{10}$$

with a series of subspace and coarse grid corrections. When computing the smallest eigenvalue, they show that convergence is obtained independently of the mesh size and the number of overlapping subdomains. However, experience with large-scale engineering problems is lacking.

Finally, multigrid techniques have also been used to approximate eigenpairs of (3). Neymeyr [40] reviews multigrid eigensolvers for elliptic differential operators. The Rayleigh quotient minimization algorithm [37, 25] uses corrections from each geometric grid to compute eigenpairs. Cai et al. [13] have established grid independent convergence estimates. Other researchers [27, 12] have applied multigrid as a nonlinear solver for the eigenproblem. Unfortunately, practical experience with computing many modes using multigrid techniques is lacking. Furthermore, all of the existing algorithms make use of geometry to define their set of grids. The authors are investigating the use of algebraic multigrid to define their grids and minimize the Rayleigh quotient.

4 Conclusions

We have reviewed several multilevel algorithms to compute a large number of eigenpairs for large-scale three-dimensional structures. We can distinguish two major approaches to solve this problem.

The first approach consists in using an efficient algebraic eigensolver coupled with a multilevel preconditioner or linear solver. Many of the schemes discussed are efficient. It will be interesting to see how shift-invert Lanczos can benefit from a scalable iterative solver for symmetric indefinite matrices. But, ultimately, maintaining numerical orthogonality of the basis vectors is the dominant cost of the modal analysis.

The second approach couples more tightly the eigensolver with the variational form of the partial differential equation. The corresponding schemes have the advantage of minimizing or eliminating the orthogonalization steps with large scale vectors and so are appealing. However, practical experience is needed in order to ascertain the efficiency of the resulting approach for three-dimensional problems.

Acknowledgments

We acknowledge the ongoing work of and discussions with Jeff Bennighof (UT Austin), Andrew Knyazev (CU Denver), Olof Widlund (NYU), and our colleagues in the structural dynamics and computational mathematics departments of Sandia National Laboratories.

References

1. M. Adams, *Evaluation of three unstructured multigrid methods on 3D finite element problems in solid mechanics*, Internat. J. Numer. Methods Engrg., 55 (2002), pp. 519–534.
2. P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq, and R. S. Tuminaro, *A comparison of Eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods*, Internat. J. Numer. Methods Engrg., 64 (2005), pp. 204–236.
3. I. Babuška and J. E. Osborn, *Eigenvalue problems*, vol. II of Handbook of numerical analysis, Elsevier, 1991, pp. 641–788.
4. J. K. Bennighof, M. F. Kaplan, and M. B. Muller, *Extending the frequency response capabilities of automated multi-level substructuring*, in AIAA Dynamics Specialists Conference, Atlanta, April 2000. AIAA-2000-1574.
5. J. K. Bennighof and R. B. Lehoucq, *An automated multilevel substructuring method for Eigenspace computation in linear elastodynamics*, SIAM J. Sci. Comput., 25 (2004), pp. 2084–2106.
6. L. Bergamaschi, G. Pini, and F. Sartoretto, *Approximate inverse preconditioning in the parallel solution of sparse Eigenproblems*, Numer. Linear Algebra Appl., 7 (2000), pp. 99–116.
7. M. Bhardwaj, K. Pierson, G. Reese, T. Walsh, D. Day, K. Alvin, J. Peery, C. Farhat, and M. Lesoinne, *Salinas: A scalable software for high-performance structural and solid mechanics simulations*, in Proceedings of 2002 ACM/IEEE Conference on Supercomputing, 2002, pp. 1–19. Gordon Bell Award.
8. M. Bhardwaj, G. Reese, B. Driessen, K. Alvin, and D. Day, *Salinas - an implicit finite element structural dynamics code developed for massively parallel platforms*, in Proceedings of the 41st AIAA/ASME/ASCE/AHS/ASC SDM Conference, April 2000.
9. F. Bourquin, *Analysis and comparison of several component mode synthesis methods on one-dimensional domains*, Numer. Math., 58 (1990), pp. 11–33.

10. ———, *Synthèse modale et analyse numérique des multistructures élastiques*, PhD thesis, Université Paris VI, 1991.
11. ———, *Component mode synthesis and Eigenvalues of second order operators: Discretization and algorithm*, Math. Model. Numer. Anal., 26 (1992), pp. 385–423.
12. A. Brandt, S. McCormick, and J. Ruge, *Multigrid methods for differential Eigenproblems*, SIAM J. Sci. Statist. Comput., 4 (1983), pp. 244–260.
13. Z. Cai, J. Mandel, and S. McCormick, *Multigrid methods for nearly singular linear equations and Eigenvalue problems*, SIAM J. Numerical Analysis, 34 (1997), pp. 178–200.
14. T. F. Chan and I. Sharapov, *Subspace correction multi-level methods for elliptic Eigenvalue problems*, Numer. Linear Algebra Appl., 9 (2002), pp. 1–20.
15. I. Charpentier, F. De Vuyst, and Y. Maday, *Méthode de synthèse modale avec une décomposition de domaine par recouvrement*, C. R. Acad. Sci. Paris, Série I, 322 (1996), pp. 881–888.
16. R. D. Cook, D. S. Malkus, M. E. Plesha, and R. J. Witt, *Concepts and applications of Finite Element Analysis*, John Wiley & Sons, Inc, 2002.
17. R. R. Craig, Jr. and M. C. C. Bampton, *Coupling of substructures for dynamic analysis*, AIAA Journal, 6 (1968), pp. 1313–1319.
18. E. R. Davidson, *The iterative calculation of a few of the lowest Eigenvalues and corresponding Eigenvectors of large real-symmetric matrices*, J. Comput. Phys., 17 (1975), pp. 817–825.
19. C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258.
20. T. Ericsson and A. Ruhe, *The spectral transformation Lanczos method for the numerical solution of large sparse generalized symmetric Eigenvalue problems*, Math. Comp., 35 (1980), pp. 1251–1268.
21. C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numer. Linear Algebra Appl., 7 (2000), pp. 687–714.
22. C. Farhat, J. Li, and P. Avery, *A FETI-DP method for the parallel iterative solution of indefinite and complex-valued solid and shell vibration problems*, Internat. J. Numer. Methods Engrg., 63 (2005), pp. 398–427.
23. 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 (1991), pp. 1205–1227.
24. Y. T. Feng and D. R. J. Owen, *Conjugate gradient methods for solving the smallest Eigenpair of large symmetric Eigenvalue problems*, Internat. J. Numer. Methods Engrg., 39 (1996), pp. 2209–2229.
25. T. Friese, *Eine Mehrgitter-Methode zur Lösung des Eigenwertproblems der komplexen Helmholtzgleichung*, PhD thesis, Freie Universität Berlin, 1998.
26. R. G. Grimes, J. G. Lewis, and H. D. Simon, *A shifted block Lanczos algorithm for solving sparse symmetric generalized Eigenproblems*, SIAM J. Matrix Anal. Appl., 15 (1994), pp. 228–272.
27. W. Hackbusch, *On the computation of approximate Eigenvalues and Eigenfunctions of elliptic operators by means of a multi-grid method*, SIAM J. Numerical Analysis, 16 (1979), pp. 201–215.
28. M. R. Hestenes and W. Karush, *A method of gradients for the calculation of the characteristic roots and vectors of a real symmetric matrix*, Journal of Research of the National Bureau of Standards, 47 (1951), pp. 45–61.

29. W. C. Hurty, *Vibrations of structural systems by component-mode synthesis*, Journal of the Engineering Mechanics Division, ASCE, 86 (1960), pp. 51–69.
30. A. V. Knyazev, *Preconditioned Eigensolvers–an oxymoron*, Electron. Trans. Numer. Anal., 7 (1998), pp. 104–123.
31. ———, *Toward the optimal preconditioned Eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
32. A. V. Knyazev and K. Neymeyr, *Efficient solution of symmetric Eigenvalue problems using multigrid preconditioners in the locally optimal block conjugate gradient method*, Electron. Trans. Numer. Anal., 7 (2003), pp. 38–55.
33. A. Kropp and D. Heiserer, *Efficient broadband vibro-accoustic analysis of passenger car bodies using an FE-based component mode synthesis approach*, in Fifth World Congress on Computational Mechanics (WCCM V) July 7-12, H. A. Mang, F. G. Rammerstorfer, and J. Eberhardsteiner, eds., Austria, 2002, Vienna University of Technology. ISBN 3-9501554-0-6 (http://wccm.tuwien.ac.at).
34. R. B. Lehoucq, D. C. Sorensen, and C. Yang, *ARPACK Users' Guide: Solution of Large Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*, SIAM, Phildelphia, PA, 1998.
35. D. E. Longsine and S. F. McCormick, *Simultaneous Rayleigh-quotient minimization for $Ax = \lambda Bx$*, Linear Algebra Appl., 34 (1980), pp. 195–234.
36. R. H. MacNeal, *A hybrid method of component mode synthesis*, Comput. & Structures, 1 (1971), pp. 581–601.
37. J. Mandel and S. McCormick, *A multilevel variational method for $Au = \lambda Bu$ on composite grids*, J. Comput. Phys., 80 (1989), pp. 442–452.
38. K. J. Maschhoff and D. C. Sorensen, *P_ARPACK: An efficient portable large scale Eigenvalue package for distributed memory parallel architectures*, in Applied Parallel Computing in Industrial Problems and Optimization, J. Wasniewski, J. Dongarra, K. Madsen, and D. Olesen, eds., vol. 1184 of Lecture Notes in Computer Science, Springer-Verlag, 1996.
39. R. Namar, *Méthodes de synthèse modale pour le calcul des vibrations des structures*, PhD thesis, Université Paris VI, 2000.
40. K. Neymeyr, *Solving mesh Eigenproblems with multigrid efficiency*, in Numerical methods for scientific computing. Variational problems and applications, Y. A. Kuznetsov, P. Neittaanmäki, and O. Pironneau, eds., 2003.
41. Y. Notay, *Combination of Jacobi-Davidson and conjugate gradients for the partial symmetric Eigenproblem*, Numer. Linear Algebra Appl., 9 (2002), pp. 21–44.
42. E. Ovtchinnikov, *Convergence estimates for the generalized Davidson method for symmetric Eigenvalue problems I: The preconditioning aspect*, SIAM J. Matrix Anal. Appl., 41 (2003), pp. 258–271.
43. ———, *Convergence estimates for the generalized Davidson method for symmetric Eigenvalue problems II: The subspace acceleration*, SIAM J. Matrix Anal. Appl., 41 (2003), pp. 272–286.
44. K. H. Pierson, G. M. Reese, and P. Raghavan, *Experiences with FETI-DP in a production level finite element application*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003.
45. D. J. Rixen, *A dual Craig-Bampton method for dynamic substructuring*, J. Comput. Appl. Math., 168 (2004), pp. 383–391.

46. P. Seshu, *Substructuring and component mode synthesis*, Shock and Vibration, 4 (1997), pp. 199–210.
47. G. L. G. Sleijpen and H. A. van der Vorst, *A Jacobi-Davidson iteration method for linear Eigenvalue problems*, SIAM J. Matrix Anal. Appl., 17 (1996), pp. 401–425. Reappeared in SIAM Review 42:267–293, 2000.
48. G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, N.J., 1973.
49. K. Stüben, *A review of algebraic multigrid*, J. Comput. Appl. Math., 128 (2001), pp. 281–309.
50. P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid based on smoothed aggregation for second and fourth order problems*, Computing, 56 (1996), pp. 179–196.

Recent Developments on Optimized Schwarz Methods

Frédéric Nataf[1]

Laboratoire J.L. Lions, CNRS UMR 7598, Université Pierre et Marie Curie, Boite courrier 187, 75252 Paris Cedex 05, France. nataf@ann.jussieu.fr

1 Introduction

The classical Schwarz method [31] is based on Dirichlet boundary conditions. Overlapping subdomains are necessary to ensure convergence. As a result, when overlap is small, typically one mesh size, convergence of the algorithm is slow. A first possible remedy is the introduction of Neumann boundary conditions in the coupling between the local solutions. This idea has led to the development of the Dirichlet-Neuman algorithm [10], Neumann-Neumann method [3] and FETI methods [8]. These methods are widely used and have been the subject of many studies, improvements and extensions to various scalar or systems of partial differential equations, see for instance the following books [32], [27], [37] and [35] and references therein. A second cure to the slowness of the original Schwarz method is to use more general interface conditions, Robin conditions were proposed in [19] and pseudo-differential ones in [17]. These methods are well-suited for indefinite problems [5] and as we shall see to heterogeneous problems.

We first recall the basis for the optimized Schwarz methods in section 2 and an application to the Helmholtz problem in section 2.2. Then, we consider equations with highly discontinuous coefficients in section 3. We present an optimized Schwarz method that takes properly into account of the discontinuities and make comparisons with other domain decomposition methods.

2 Generalities on Optimized Schwarz methods

2.1 Optimal Interface Conditions

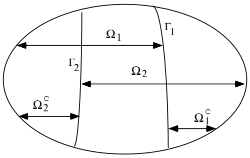
We will exhibit interface conditions which are optimal in terms of iteration counts. The corresponding interface conditions are pseudo-differential and are not practical. Nevertheless, this result is a guide for the choice of partial differential interface conditions. Moreover, this result establishes a link between the optimal interface conditions and artificial boundary conditions. This is also a help when dealing with the design of interface conditions since it gives the possibility of using the numerous papers and books published on the subject of artificial boundary conditions, see e.g. [6, 15].

We consider a general linear second order elliptic partial differential operator $\mathcal{L}$ and the problem:

Find u such that $\mathcal{L}(u) = f$ in a domain Ω and $u = 0$ on $\partial\Omega$.

The domain Ω is decomposed into two subdomains Ω_1 and Ω_2. We suppose that the problem is regular so that $u_i := u|_{\Omega_i}$, $i = 1, 2$, is continuous and has continuous normal derivatives across the interface $\Gamma_i = \partial\Omega_i \cap \bar{\Omega}_j$, $i \neq j$.

Fig. 1. A two-subdomain decomposition.



A generalized Schwarz type method is considered.

$$\begin{array}{ll}
\mathcal{L}(u_1^{n+1}) = f \quad \text{in} \quad \Omega_1 & \mathcal{L}(u_2^{n+1}) = f \quad \text{in} \quad \Omega_2 \\
u_1^{n+1} = 0 \quad \text{on} \quad \partial\Omega_1 \cap \partial\Omega & u_2^{n+1} = 0 \quad \text{on} \quad \partial\Omega_2 \cap \partial\Omega \\
\mu_1 \nabla u_1^{n+1}.\mathbf{n_1} + \mathcal{B}_1(u_1^{n+1}) & \mu_2 \nabla u_2^{n+1}.\mathbf{n_2} + \mathcal{B}_2(u_2^{n+1}) \\
= -\mu_1 \nabla u_2^{n}.\mathbf{n_2} + \mathcal{B}_1(u_2^{n}) \text{ on } \quad \Gamma_1 & = -\mu_2 \nabla u_1^{n}.\mathbf{n_1} + \mathcal{B}_2(u_1^{n}) \text{ on } \Gamma_2
\end{array} \tag{1}$$

where μ_1 and μ_2 are real-valued functions and $\mathcal{B}_1$ and $\mathcal{B}_2$ are operators acting along the interfaces Γ_1 and Γ_2. For instance, $\mu_1 = \mu_2 = 0$ and $\mathcal{B}_1 = \mathcal{B}_2 = \mathrm{Id}$ correspond to the original Schwarz method; $\mu_1 = \mu_2 = 1$ and $\mathcal{B}_i = \alpha \in \mathbf{R}$, $i = 1, 2$, has been proposed in [19] by P. L. Lions.

The question is:

> *Are there other possibilities in order to have convergence in a minimal number of steps?*

In order to answer this question, we introduce the DtN (Dirichlet to Neumann) map (a.k.a. Steklov-Poincaré) of domain $\Omega_2 \setminus \bar{\Omega}_1$: Let

$$\begin{aligned} u_0 &: \Gamma_1 \to \mathbf{R} \\ \mathrm{DtN}_2(u_0) &:= \nabla v.n_{2|\partial\Omega_1 \cap \bar{\Omega}_2}, \end{aligned} \tag{2}$$

where n_2 is the outward normal to $\Omega_2 \setminus \bar{\Omega}_1$, and v satisfies the following boundary value problem:

$$\begin{aligned} \mathcal{L}(v) = 0 \quad &\text{in} \quad \Omega_2 \setminus \bar{\Omega}_1 \\ v = 0 \quad &\text{on} \quad \partial\Omega_2 \cap \partial\Omega \\ v = u_0 \quad &\text{on} \quad \partial\Omega_1 \cap \bar{\Omega}_2. \end{aligned}$$

Similarly, we can define DtN_1 the Dirichlet to Neumann map of domain $\Omega_1 \setminus \bar{\Omega}_2$. The following optimality result is proved in [23]:

Result 1 *The use of* $\mathcal{B}_i = \mathrm{DtN}_j$ ($i = 1, 2$ *and* $i \neq j$) *as interface conditions in* (1) *is optimal: we have* (*exact*) *convergence in two iterations.*

The two-domain case for an operator with constant coefficients was first treated in [17]. The multidomain case for a variable coefficient operator with both positive results [25] and negative conjectures [26] were considered as well.

Remark 1. The main feature of this result is its generality since it does not depend on the exact form of the operator $\mathcal{L}$ and can be extended to systems or to coupled systems of equations as well with proper care of the well posedness of the algorithm.

As an application, we take $\Omega = \mathbf{R}^2$ and $\Omega_1 =]-\infty, 0\,[\times\mathbf{R}$. Using the Fourier transform along the interface (the dual variable is denoted by k), it is possible to give the explicit form of the DtN operator for a constant coefficient operator. If $\mathcal{L} = \eta - \Delta$, the DtN map is a pseudo-differential operator whose symbol is

$$B_{i,\mathrm{opt}}(k) = \sqrt{\eta + k^2},$$

i.e., $\mathcal{B}_{i,\mathrm{opt}}(u)(0, y) = \int_{\mathbf{R}} B_{i,\mathrm{opt}}(k)\hat{u}(0, k)e^{Iky}\,dk$.

The symbol is not polynomial in the Fourier variable k so that the operators and hence the optimal interface conditions are not a partial differential operators. They correspond to exact absorbing conditions. These conditions are used on the artificial boundary resulting from the truncation of a computational domain. On this boundary, boundary conditions have to be imposed. The solution on the truncated domain depends on the choice of this artificial condition. We say that it is an exact absorbing boundary condition if the solution computed on the truncated domain is the restriction of the solution of the original problem. Surprisingly enough, the notions of exact absorbing conditions for domain truncation and that of optimal interface conditions in domain decomposition methods coincide.

2.2 Optimized Interface Conditions for the Helmholtz equation

As the above example shows, the optimal interface conditions are pseudodifferential. Therefore they are difficult to implement. Moreover, in the general case of a variable coefficient operator and/or a curved boundary, the exact form of these operators is not known, although they can be approximated by partial differential operators which are easier to implement. The approximation of the DtN has been addressed by many authors since the seminal paper [6] by Engquist and Majda on this question. A first natural idea is to use these works in domain decomposition methods. As we shall see, it is better to design approximations that are optimized with respect to the domain decomposition method. We seek approximations to the Dirichlet to Neumann map by a partial differential operator

$$DtN \simeq \alpha_{opt} - \frac{\partial}{\partial\tau}(\gamma_{opt}\frac{\partial}{\partial\tau})$$

where ∂_τ is the derivative along the interface. The parameters are chosen in order to minimize the convergence rate of the algorithm. These interface conditions are called optimized of order 2 conditions (opt2). If we take $\gamma = 0$, the optimization is performed only w.r.t. α, they are called optimized of order 0 (opt0). The idea was first introduced in [34]. But the link with the optimal interface conditions was not established and made the optimization too complex.

As an example, we present here the case of the Helmholtz equation that was considered in [12]. We want to solve by a domain decomposition method:

$$\mathcal{L}(u) = (-\omega^2 - \Delta)(u) = f$$

In order to find the optimized interface conditions, we first consider a very simple geometry for which the optimization is tractable and then apply these results to an industrial case. As a first step, the domain $\Omega = \mathbb{R}^2$ is decomposed into two non overlapping subdomains $\Omega_1 = (-\infty, 0) \times \mathbb{R}$ and $\Omega_2 = (0, \infty) \times \mathbb{R}$. The algorithm is defined by (1) with $\mu_1 = \mu_2 = 1$ and $\mathcal{B}_1 = \mathcal{B}_2 = \alpha - \frac{\partial}{\partial \tau}(\gamma \frac{\partial}{\partial \tau})$. A direct computation yields the convergence rate of the iterative method in the Fourier space:

$$\rho(k;\alpha,\gamma) \equiv \begin{cases} \left| \dfrac{I\sqrt{\omega^2 - k^2} - (\alpha + \gamma k^2)}{I\sqrt{\omega^2 - k^2} + (\alpha + \gamma k^2)} \right| \text{ if } |k| < \omega \quad (I^2 = -1) \\ \\ \left| \dfrac{\sqrt{k^2 - \omega^2} - (\alpha + \gamma k^2)}{\sqrt{k^2 - \omega^2} + (\alpha + \gamma k^2)} \right| \text{ if } \omega < |k| \end{cases}$$

The convergence rate in the physical space is the maximum over k of $\rho(k;\alpha,\gamma)$. Actually, it is sufficient to consider Fourier modes that can be represented on the mesh used in the discretization of the operator. It imposes a truncation in the frequencey domain of the type $|k| < \pi/h$ where h is the mesh size. We have then to minimize the convergence rate in the physical space with respect to the parameters α and γ. We are thus led to the following min-max problem:

$$\min_{\alpha,\gamma} \max_{|k|<\pi/h} \rho(k;\alpha,\gamma).$$

Under additional simplifications, we get analytic formulas for the optimized parameters α and γ depending on ω and h, see [12].

For an arbitrary domain decomposition for instance obtained by an automatic mesh partitioner as the one shown on figure 2, we proceed in the following manner. At each node on the interface, we use the local value of the mesh size to compute the optimized parameters using the formula established in the simple case of the plane $\mathbb{R}^2$ divided into two half-planes. In table 1, we give iteration counts for various interface conditions: ABC0 means that the interface conditions are $\partial_n + I\omega$ (i.e. $\alpha = I\omega$ and $\gamma = 0$, see [2]), ABC2 corresponds to absorbing conditions of order 2 that are currently used for truncation of domains see [6] but were not designed with domain decomposition methods in mind. Notice that since the interfaces are not straight lines and the subdomains have an irregular shape, we are very far from the ideal case considered above. Nevertheless, the optimized interface conditions perform quite well.

Fig. 2. Domain decomposition of the cabin car.

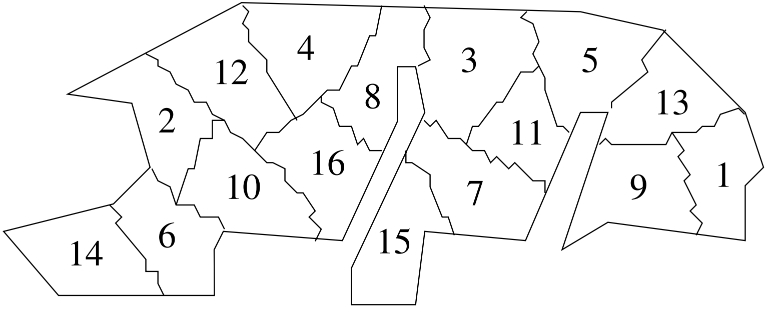


Table 1. Iteration Counts for various interface conditions and numbers of subdomains N_s.

N_s	ABC 0	ABC 2	Optimized
2	16 it	16 it	9 it
4	50 it	52 it	15 it
8	83 it	93 it	25 it
16	105 it	133 it	34 it

3 Optimized Schwarz Method for Highly Discontinuous Coefficients

We consider now a symmetric positive definite problem but with highy discontinuous coefficients. The model equation is

$$\eta u - div(\kappa \nabla u) = f \text{ in } \Omega$$

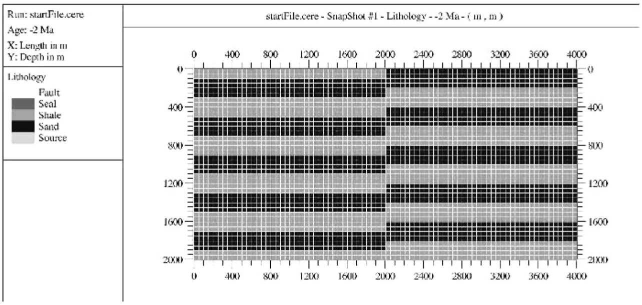
It contains some of the difficulties typical of porous media flow simulations. Indeed, the coefficients η and κ have jumps which are typically of four orders of magnitude. The tensor κ is anisotropic with large anisotropy ratios: $10^{-4} \leq \kappa_x/\kappa_y \leq 10^4$. In the situation we consider, the domain Ω is divided into two subdomains Ω_1 and Ω_2 corresponding to two different geological blocks. Each subdomain is layered so that the coefficients are discontinuous both across and along the interface, see for instance figure 3. These kinds of problems lead to very ill-conditioned linear systems so that there are plateaus in the convergence of Krylov methods even with otherwise "good" preconditioners.

In order to design optimized interface conditions, we first define more precisely the model problem we consider.

3.1 Setting of the semi-discrete problem

We consider a model problem set in an infinite tube $\Omega = \mathbb{R} \times \omega$ where ω is some bounded open set of $\mathbb{R}^p$ for some $p \geq 1$. The domain is decomposed into two non overlapping half tubes $\Omega_1 = (-\infty, 0) \times \omega$ and $\Omega_2 = (0, \infty) \times \omega$. A point in Ω will be denoted by $(x, \mathbf{y})$. Let for $i = 1, 2$

Fig. 3. Lithology.



$$\mathcal{L}_i := -\frac{\partial}{\partial x} c_i(\mathbf{y}) \frac{\partial}{\partial x} + \mathcal{C}_i(\mathbf{y}) \tag{3}$$

where c_i is a positive real valued function and $\mathcal{C}_i$ is a symmetric positive definite operator independent of the variable x. For instance, if $p = 2$ one might think of

$$\mathcal{C}_i := \eta_i(y,z) - \left(\frac{\partial}{\partial y} \kappa_{i,y}(y,z) \frac{\partial}{\partial y} + \frac{\partial}{\partial z} \kappa_{i,z}(y,z) \frac{\partial}{\partial z} \right) \tag{4}$$

with homogeneous Dirichlet boundary conditions and $\eta_i \geq 0$, $c_i, \kappa_{i,y}, \kappa_{i,z} > 0$ are given real-valued functions and $(y,z) \in \omega$.

We want to solve the following problem by a domain decomposition method

$$\begin{array}{l} \mathcal{L}_i(u_i) = f \text{ in } \Omega \\ u = 0 \text{ on } \partial\Omega \end{array}$$

with

$$C_1 \frac{\partial u_1}{\partial x} = C_2 \frac{\partial u_2}{\partial x} \quad \text{on} \quad \Gamma$$

and

$$u_2 = u_1 \quad \text{on} \quad \Gamma$$

The problem can be considered at the continuous level and then discretized (see e.g. [12], [11], [24]), or at the discrete level (see e.g. [20], [28] or [13]). We choose here a semi-discrete approach where only the tangential directions to the interface $x = 0$ are discretized whereas the normal direction x is kept continuous.

We therefore consider a discretization in the tangential directions which leads to

$$\mathcal{L}_{i,h} := -\frac{\partial}{\partial x} C_i \frac{\partial}{\partial x} + B_i \tag{5}$$

where B_i and C_i are symmetric positive matrices of order n where n is the number of discretization points of the open set $\omega \subset \mathbb{R}^p$. For instance if we take $\mathcal{C}_i$ to be defined as in (4), B_i may be obtained via a finite volume or finite element discretization of (4) on a given mesh or triangulation of $\omega \subset \mathbb{R}^2$.

We consider a domain decomposition method based on arbitrary interface conditions $\mathcal{D}_1$ and $\mathcal{D}_2$. The corresponding Optimized Schwarz method (OSM) reads:

$$\begin{array}{ll} \mathcal{L}_{1,h}(u_1^{n+1}) = f \quad \text{in} \quad \Omega_1 & \mathcal{L}_{2,h}(u_2^{n+1}) = f \quad \text{in} \quad \Omega_2 \\ \mathcal{D}_1(u_1^{n+1}) = \mathcal{D}_1(u_2^n) \quad \text{on} \quad \Gamma & \mathcal{D}_2(u_2^{n+1}) = \mathcal{D}_2(u_1^n) \quad \text{on} \quad \Gamma \end{array} \tag{6}$$

where Γ is the interface $x = 0$. It is possible to both increase the robustness of the method and its convergence speed by replacing the above fixed point iterative solver by a Krylov type method. This is made possible by expressing the algorithm in terms of interface unknowns

$$H_1 = \mathcal{D}_1(u_2)(0,.) \quad \text{and} \quad H_2 = \mathcal{D}_2(u_1)(0,.)$$

see [9].
At this point, it should be noted that the analysis of the present paper is restricted to rather idealistic geometries. However, the same formalism can be used for a domain decomposition into an arbitrary number of subdomains [12]. It has also been found there that the convergence estimates provided in this simple geometry predict very accurately the ones observed in practice even for complicated interface boundaries.

We first define interface conditions that lead to convergence in two steps of the algorithm. Let

$$\Lambda_i = C_i^{1/2} A_i^{1/2} C_i^{1/2} \tag{7}$$

where $A_i := C_i^{-1/2} B_i C_i^{-1/2}$. Taking

$$\mathcal{D}_1 = (C_1 \frac{\partial}{\partial n_1} + \Lambda_2) \quad \text{and} \quad \mathcal{D}_2 = (C_2 \frac{\partial}{\partial n_2} + \Lambda_1)$$

leads to a convergence in two steps of (1), see [9]. This result is optimal in terms of iteration counts. But, matrices Λ_i are a priori full matrices of order n costly to compute and use. Instead, we will use approximations in terms of sparse matrices denoted $\Lambda_{i,ap}$. We lose convergence in two steps. In order to have the best convergence rate, we choose optimized sparse approximations to Λ_i w.r.t the domain decomposition method.

We first consider diagonal approximations to Λ_i. At the continuous level, they correspond to Robin interface conditions. For a matrix F, let $\lambda_{m,M}(F)$ denote respectively the smallest and largest eigenvalues of F and $diag(F)$ the diagonal matrix made of the diagonal of F. We define

$$\Lambda^0_{i,ap} = \tilde{\beta}_{i,opt} \tilde{D}_i \tag{8}$$

where $\tilde{D}_i := C_i^{1/2} diag(A_i)^{1/2} C_i^{1/2}$ and

$$\tilde{\beta}_{i,opt} = \sqrt{\beta_m \, \beta_M}$$

with

$$\beta_{m,M} = \sqrt{\lambda_{m,M}(diag(A_i)^{-1/2} A_i \, diag(A_i)^{-1/2})}$$

We also consider sparse approximations that will have the same sparsity as A_i. Let $\lambda_{m,M} = \lambda_{m,M}(\tilde{D}_i^{-2} A_i)^{1/2}$, the real parameters β_1 and β_2 are defined as follows

$$\beta_1 \beta_2 = \lambda_m \, \lambda_M \tag{9}$$

$$\beta_1 + \beta_2 = \left(2\sqrt{\lambda_m \lambda_M}\,(\lambda_m + \lambda_M)\right)^{1/2} \tag{10}$$

We define

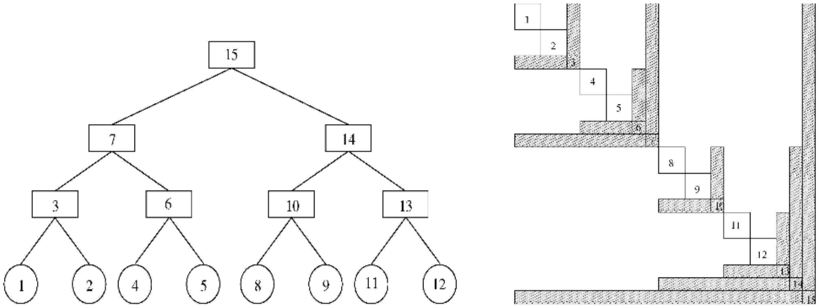
$$\Lambda^2_{ap,\beta_1,\beta_2} := C_i^{1/2} \frac{\tilde{D}_i^{-1} A_i + \beta_1 \beta_2 \tilde{D}_i}{\beta_1 + \beta_2} C_i^{1/2} \tag{11}$$

At the continuous level, they correspond to optimized of order 2 interface conditions. The motivation for definitions (8) and (11) are given in [9].

3.2 Numerical results

The substructured problems are solved by a GMRES algorithm [29]. In the tables and figures, **opt0** refers to (8) and **opt2** to formula (11). In figure 4, we compare them with interface conditions obtained using a "frozen" coefficient approach. In the latter case, the interface conditions depend only locally on the coefficients of the problem, see [36] at the continuous level, [13] at the semi-discrete level and [28] at the algebraic level. We see a plateau in the convergence curve which can be related to a few very small eigenvalues in the spectrum of the substructured problem, see figure 4. A possible cure to this problem is the use of deflation methods, [21], [16], [22] and [30]. They rely on an accurate knowledge of the eigenvectors corresponding to the "bad" eigenvalues. With the **opt2** interface conditions, no eigenvalue is close to zero and we need only extremal eigenvalues (and not the eigenvectors) of an auxiliary matrix. We also give comparisons with the Neumann-Neumann [33] [4] or FETI [18] approach, see figure 5. In the numerical tests, we have typically ten layers in each subdomain. In each layer, the diffusion tensor is anisotropic. We have jumps in the coefficients both across and along the interface. We are thus in a situation where the Neumann-Neumann or FETI methods are not necessarily optimal.

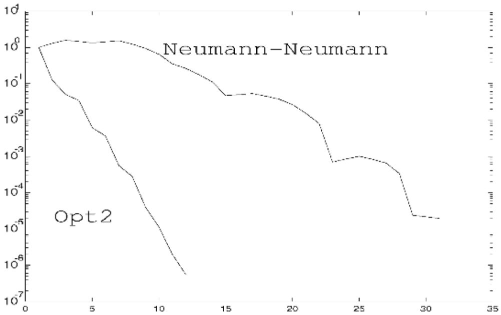
Fig. 4. Left: Convergence curve for various interface conditions. Right: Eigenvalues of the interface problem for **opt2** (cross) and "frozen" (circles) interface conditions.



4 Conclusion

We have first reviewed known results on optimized Schwarz methods for smooth coefficients operators. We have then considered problems with highly anisotropic and discontinuous coefficients, for which plateaus in the convergence of Krylov methods exist even when using "good" preconditioners. A classical remedy is to use deflated Krylov methods. We have developed in this paper a new algebraic approach in the DDM framework. We propose a way to compute optimized interface conditions for domain decomposition methods for symmetric positive definite equations. Compared

Fig. 5. residual vs. subdomain solve counts.



to deflation, only two extreme eigenvalues have to be computed. Numerical results show that the approach is efficient and robust even with highly discontinuous coefficients both across and inside subdomains. The non-symmetric case is considered in this volume at the algebraic level in a joint work with Luca Gerardo-Giorda, see also [14]. The optimization of the interface condition is then much more difficult. Let us mention that such interface conditions can be used on non-matching grids, see [1] and [7].

References

1. Y. Achdou, C. Japhet, Y. Maday, and F. Nataf, *A new cement to glue non-conforming grids with Robin interface conditions: the finite volume case*, Numer. Math., 92 (2002), pp. 593–620.
2. J. D. Benamou and B. Després, *A domain decomposition method for the Helmholtz equation and related optimal control*, J. Comp. Phys., 136 (1997), pp. 68–82.
3. J.-F. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu, *Variational formulation and algorithm for trace operator in domain decomposition calculations*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM, pp. 3–16.
4. L. C. Cowsar, J. Mandel, and M. F. Wheeler, *Balancing domain decomposition for mixed finite elements*, Math. Comp., 64 (1995), pp. 989–1015.
5. B. Després, *Décomposition de domaine et problème de Helmholtz*, C.R. Acad. Sci. Paris, 1 (1990), pp. 313–316.
6. B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp., 31 (1977), pp. 629–651.
7. I. Faille, F. Nataf, L. Saas, and F. Willien, *Finite volume methods on non-matching grids with arbitrary interface conditions and highly heterogeneous media*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Springer-Verlag, 2004, pp. 243–250. Lecture Notes in Computational Science and Engineering.

8. 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 (1992), pp. 379–396.
9. E. Flauraud and F. Nataf, *Optimized interface conditions in domain decomposition methods. Application at the semi-discrete and at the algebraic level to problems with extreme contrasts in the coefficients.*, Tech. Rep. R.I. 524, CMAP, Ecole Polytechnique, 2004.
10. D. Funaro, A. Quarteroni, and P. Zanolli, *An iterative procedure with interface relaxation for domain decomposition methods*, SIAM J. Numer. Anal., 25 (1988), pp. 1213–1236.
11. M. J. Gander and G. H. Golub, *A non-overlapping optimized Schwarz method which converges with an arbitrarily weak dependence on h*, in Fourteenth International Conference on Domain Decomposition Methods, 2002.
12. M. J. Gander, F. Magoulès, and F. Nataf, *Optimized Schwarz methods without overlap for the Helmholtz equation*, SIAM J. Sci. Comput., 24 (2002), pp. 38–60.
13. M. Genseberger, *Domain decomposition in the Jacobi-Davidson method for Eigenproblems*, PhD thesis, Utrecht University, September 2001.
14. L. G. Giorda and F. Nataf, *Optimized Schwarz methods for unsymmetric layered problems with strongly discontinuous and anisotropic coefficients*, Tech. Rep. 561, CMAP, CNRS UMR 7641, Ecole Polytechnique, France, 2004. Submitted.
15. D. Givoli, *Numerical methods for problems in infinite domains*, Elsevier, 1992.
16. I. G. Graham and M. J. Hagger, *Unstructured additive Schwarz-CG method for elliptic problems with highly discontinuous coefficients*, SIAM J. Sci. Comput., 20 (1999), pp. 2041–2066.
17. T. Hagstrom, R. P. Tewarson, and A. Jazcilevich, *Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems*, Appl. Math. Lett., 1 (1988).
18. 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 (2002).
19. 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 , held in Houston, Texas, March 20-22, 1989, T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM.
20. G. Lube, L. Mueller, and H. Mueller, *A new non-overlapping domain decomposition method for stabilized finite element methods applied to the nonstationary Navier-Stokes equations*, Numer. Lin. Alg. Appl., 7 (2000), pp. 449–472.
21. R. B. Morgan, *GMRES with deflated restarting*, SIAM J. Sci. Comput., 24 (2002), pp. 20–37.
22. R. Nabben and C. Vuik, *A comparison of deflation and coarse grid correction applied to porous media flow*, Tech. Rep. R03-10, Delft University of Technology, 2003.
23. F. Nataf, *Interface connections in domain decomposition methods*, in Modern methods in scientific computing and applications (Montréal, QC, 2001), vol. 75 of NATO Sci. Ser. II Math. Phys. Chem., Kluwer Acad. Publ., Dordrecht, 2002, pp. 323–364.

24. F. Nataf and F. Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, M^3AS, 5 (1995), pp. 67–93.
25. F. Nataf, F. Rogier, and E. de Sturler, *Optimal interface conditions for domain decomposition methods*, Tech. Rep. 301, CMAP (Ecole Polytechnique), 1994.
26. F. Nier, *Remarques sur les algorithmes de décomposition de domaines*, in Seminaire: Équations aux Dérivées Partielles, 1998–1999, École Polytech., 1999, pp. Exp. No. IX, 26.
27. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, 1999.
28. F.-X. Roux, F. Magoulès, S. Salmon, and L. Series, *Optimization of interface operator based on algebraic approach*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003.
29. Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Statist. Comput., 7 (1986), pp. 856–869.
30. Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h, *A deflated version of the conjugate gradient algorithm*, SIAM J. Sci. Comput., 21 (2000), pp. 1909–1926. Iterative methods for solving systems of algebraic equations (Copper Mountain, CO, 1998).
31. H. A. Schwarz, *Über einen Grenzübergang durch alternierendes Verfahren*, Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, 15 (1870), pp. 272–286.
32. B. F. Smith, P. E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
33. P. L. Tallec and M. Vidrascu, *Generalized Neumann-Neumann preconditioners for iterative substructuring*, in Domain Decomposition Methods in Sciences and Engineering, P. E. Bjørstad, M. Espedal, and D. Keyes, eds., John Wiley & Sons, 1997. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.
34. K. H. Tan and M. J. A. Borsboom, *On generalized Schwarz coupling applied to advection-dominated problems*, in Seventh International Conference of Domain Decomposition Methods in Scientific and Engineering Computing, D. E. Keyes and J. Xu, eds., AMS, 1994, pp. 125–130. Held at Penn State University, October 27-30, 1993.
35. A. Toselli and O. Widlund, *Domain Decomposition Methods - Algorithms and Theory*, vol. 34 of Springer Series in Computational Mathematics, Springer, 2004.
36. F. Willien, I. Faille, F. Nataf, and F. Schneider, *Domain decomposition methods for fluid flow in porous medium*, in 6th European Conference on the Mathematics of Oil Recovery, September 1998.
37. B. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, vol. 17 of Lecture Notes in Computational Science and Engineering, Springer, 2001.

Schur Complement Preconditioners for Distributed General Sparse Linear Systems*

Yousef Saad

University of Minnesota, Department of Computer Science and Engineering, 200
Union Street SE, Minneapolis, MN 55455, USA. saad@cs.umn.edu

Summary. This paper discusses the Schur complement viewpoint when developing parallel preconditioners for general sparse linear systems. Schur complement methods are pervasive in numerical linear algebra where they represent a canonical way of implementing divide-and-conquer principles. The goal of this note is to give a brief overview of recent progress made in using these techniques for solving general, irregularly structured, sparse linear systems. The emphasis is to point out the impact of Domain Decomposition ideas on the design of general purpose sparse system solution methods, as well as to show ideas that are of a purely algebraic nature.

1 Distributed sparse linear systems

The parallel solution of a linear systems of the form

$$Ax = b, \tag{1}$$

where A is an $n \times n$ large sparse matrix, typically begins by subdividing the problem into p parts with the help of a graph partitioner [24, 13, 15, 23, 8, 16]. Generally, this consists of assigning sets of equations along with the corresponding right-hand side values to 'subdomains'. It is common that if equation number i is assigned to a given subdomain then unknown number i is assigned to the same subdomain. Thus, each processor holds a set of equations (rows of the linear system) and vector components associated with these rows.

This distinction is important when taking a purely algebraic viewpoint because for highly unstructured or rectangular (least-squares) systems, this is no longer a viable or possible strategy and one needs to reconsider the standard graph partitioning approach used in Domain Decomposition. The next section is a brief discussion of graph partitioning issues.

[*] This work was supported by NSF under grants ACI-0305120 and INT-0003274 and by the Minnesota Supercomputer Institute.

2 Graph partitioning

Figure 1 shows two standard ways of partitioning a graph. On the left side is a 'vertex' partitioning which is common in the sparse matrix community. A vertex is a pair equation-unknown (equation number i and unknown number i) and the partitioner subdivides the vertex set into p partitions, i.e., p non-intersecting subsets whose union is equal to the original vertex set. On the right side of Figure 1, is a situation which is a prevalent one in finite element methods. Here it is the set of elements (rectangular in this case) that is partitioned. This can be called an element-based partitioning, or, alternatively, an 'edge-based partitioning', since in this case it also corresponds to assigning edges to subdomains.

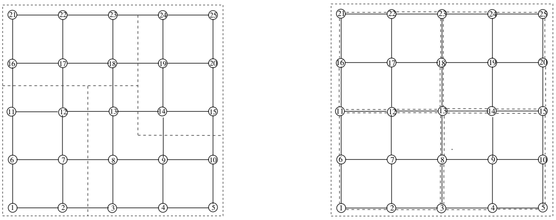


Fig. 1. Two classical ways of partitioning a graph.

The simplest criterion used to partition a graph is to try to minimize communication costs and to ensure at the same time that the work load between processors is well balanced. In this strategy, it is common to model communication costs by counting the number of edge-cuts, i.e., edges that link vertices in different subdomains. Graph partitioners such as Metis [15] and Chaco [13], attempt to partition graphs wit the quality measures just mentioned, in mind. However, a simple look at a general graph will reveal that edge-cuts will not lead to a good model for communication costs. Thus, when k edges connect a single vertex to k non-local vertices we would count k communication instances instead of one.

This observation was exploited in [8] to devise partitioners which lead to reduced communication costs. The authors of [8] used 'Hypergaphs' for this purpose. Hypergraphs are generalizations of graphs in which edges become sets (called *hyperedges* or *nets*) consisting of several vertices, instead of just two. Figure 2 shows a sparse matrix along with its traditional graph representation. Figure 3 shows the hypergraph obtained by defining hyper-edges to be the sets of column entries for each row. A hyperedge is represented by a square. Thus, hyperedge h_6, which corresponds to the 6-th row of the matrix, is the set of the 3 vertices: 1, 6, and 8, as indicated by the links from h_6 (square) to the vertices 1, 6, and 8 (bullets). Similarly $h_7 = \{1, 2, 7, 8\}$.

Note that from one viewpoint, this new representation is really that of a bipartite graph, since the nodes represented by a hyperedge (squares) are linked only to vertices of the graph (bullets). Models similar to the one just illustrated, i.e., based on setting h_i to be the set of column entries of row i, are common in hypergraph partitioning as they tend to yield better cost models for communication, see, [8]. Gains in communication will help reduce the overall run time but these gains are typically in the order of 10-30%, and they often represent a small portion of the overall execution time. One may ask whether or not the gains could be outweighed by the cost of a higher iteration count. In fact, experimental results suggest that hypergraph partitioning yields as good if not better quality partitionings from the point of convergence. More importantly, we believe that the generality and flexibility of hypergraph models has not yet been fully exploited in Domain Decomposition. Though it is difficult to rigorously build a partitioning that will yield an 'optimal' condition number for the preconditioned matrix, heuristic arguments, see, e.g., [27], may help obtain criteria that can help build good models based on weighted hypergraphs.

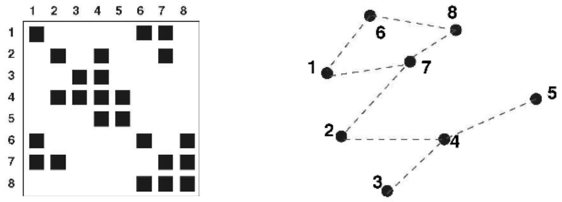


Fig. 2. A small sparse matrix and its classical graph representation.

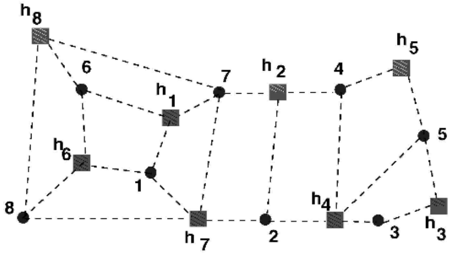


Fig. 3. One possible hypergraph representation of the matrix in Figure 2.

Another potential use of hypergraphs is for solving very irregularly structured problems which do not originate from PDEs. In these situations, the adjacency graph of the matrix may be directed (i.e., pattern of A is nonsymmetric), a situation which is not handled by standard partitioners. A common remedy is to symmetrize the graph before partitioning it, which tends to be wasteful. Domain decomposition ideas can be extended to such problems with the help of hypergraphs [12] or the closely related bipartite models [16].

3 The local system

Once a graph is partitioned, three types of unknowns can be distinguished: (1) Interior unknowns that are coupled only with local equations; (2) Local interface unknowns that are coupled with both non-local (external) and local equations; and

(3) External interface unknowns that belong to other subdomains and are coupled with local equations. Local points in each subdomain are often reordered so that the interface points are listed after the interior points. Thus, each local vector of unknowns x_i is split into two parts: the subvector u_i of internal vector components followed by the subvector y_i of local interface vector components. The right-hand side b_i is conformally split into the subvectors f_i and g_i. When block partitioned according to this splitting, the local system of equations can be written as

$$\underbrace{\begin{pmatrix} B_i & F_i \\ E_i & C_i \end{pmatrix}}_{A_i} \underbrace{\begin{pmatrix} u_i \\ y_i \end{pmatrix}}_{x_i} + \begin{pmatrix} 0 \\ \sum_{j \in N_i} E_{ij} y_j \end{pmatrix} = \underbrace{\begin{pmatrix} f_i \\ g_i \end{pmatrix}}_{b_i} . \tag{2}$$

Here, N_i is the set of indices for subdomains that are neighbors to the subdomain i. The term $E_{ij}y_j$ is a part of the product which reflects the contribution to the local equation from the neighboring subdomain j. The result of this multiplication affects only local interface equations, which is indicated by zero in the top part of the second term of the left-hand side of (2).

4 Schur complement techniques

Schur complement techniques consist of eliminating interior variables to define methods which focus on solving in some ways the system associated with the interface variables. For example, we can eliminate the variable u_i from (2), which gives $u_i = B_i^{-1}(f_i - F_i y_i)$ and upon substitution in the second equation,

$$S_i y_i + \sum_{j \in N_i} E_{ij} y_j = g_i - E_i B_i^{-1} f_i \equiv g_i', \tag{3}$$

where S_i is the "local" Schur complement

$$S_i = C_i - E_i B_i^{-1} F_i . \tag{4}$$

The equations (3) for all subdomains ($i = 1, \ldots, p$) constitute a linear system involving only the interface unknown vectors y_i. This reduced system has a natural block structure:

$$\underbrace{\begin{pmatrix} S_1 & E_{12} & \ldots & E_{1p} \\ E_{21} & S_2 & \ldots & E_{2p} \\ \vdots & & \ddots & \vdots \\ E_{p1} & E_{p,2} & \ldots & S_p \end{pmatrix}}_{S} \underbrace{\begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_p \end{pmatrix}}_{y} = \underbrace{\begin{pmatrix} g_1' \\ g_2' \\ \vdots \\ g_p' \end{pmatrix}}_{g'} . \tag{5}$$

The diagonal blocks in this system, the local Schur complement matrices S_i, are dense in general. The off-diagonal blocks E_{ij}, which are identical with those of the local system (2) are sparse.

If can solve the global Schur complement system (5) then the solution to the global system (1) would be trivially obtained by substituting the y_i's into the first part of (2). A key idea in domain decomposition methods is to develop preconditioners for the *global system* (1) by exploiting methods that *approximately solve the Schur complement system* (5).

Preconditioners implemented in the pARMS library [18] rely on this general approach. The system (5) is preconditioned in a number of ways, the simplest of which is to use a Block-Jacobi preconditioner exploiting the block structure of (5). The S_i's are not explicitly computed. Assuming the notation (2), and considering the LU factorization of A_i, we note that

$$\text{if} \quad A_i = \begin{pmatrix} L_{B_i} & 0 \\ E_i U_{B_i}^{-1} & L_{S_i} \end{pmatrix} \begin{pmatrix} U_{B_i} & L_{B_i}^{-1} F_i \\ 0 & U_{S_i} \end{pmatrix} \quad \text{then} \quad L_{S_i} U_{S_i} = S_i \ .$$

This yields the LU (or ILU) factorization of S_i as a by-product of the LU (resp. ILU) factorization of A_i. Setting up the preconditioner is a local process which only requires the LU (resp. ILU) factorization of A_i.

Other Schur complement preconditioners available in pARMS include methods which solve the system (5) approximately by a parallel (multicolor) version of the ILU(0) preconditioner, and a multicolor block Gauss-Seidel iteration (instead of block Jacobi). In general these work better than the simple block Jacobi technique discussed above. For details see [18].

5 Use of independent sets

Independent set orderings permute a matrix into the form

$$\begin{pmatrix} B & F \\ E & C \end{pmatrix} \tag{6}$$

where B is diagonal. The unknowns associated with the B block form an independent set (IS), which is said to be maximal if it cannot be augmented by other nodes to form a bigger independent set. Finding a maximal independent set can be done inexpensively by heuristic algorithms [9, 17, 25].

The main observation here is that the Schur complement $S = C - EB^{-1}F$ associated with the above partitioning of the matrix is again a sparse matrix *in general* since B is diagonal. Therefore, one can think of applying the reduction recursively as is illustrated in Figure 4. When the reduced system becomes small enough then it can be solved by any method. This is the idea used in ILUM [25], and in a number of related papers [7, 6, 30].

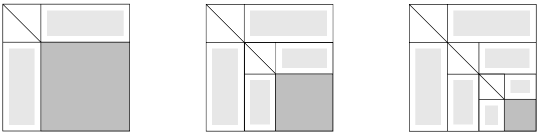


Fig. 4. Three stages of the recursive ILUM process

The notion of independent sets can easily be extended to 'group independent sets', in which the matrix B is allowed to be block-diagonal instead of just diagonal. In other words, we need to find "groups" or "aggregates" of vertices which are not coupled to each other, in the sense that no node from one group is coupled with a node of another group. Coupling within any group is allowed but not between different groups.

Define the matrix at the zeroth-th level to be $A_0 \equiv A$. The Algebraic Recursive Multilevel Solver algorithm (ARMS), see [28], is based on an approximate block factorization of the form

$$P_l A_l P_l^T = \begin{pmatrix} B_l & F_l \\ E_l & C_l \end{pmatrix} \approx \begin{pmatrix} L_l & 0 \\ E_l U_l^{-1} & I \end{pmatrix} \begin{pmatrix} I & 0 \\ 0 & A_{l+1} \end{pmatrix} \begin{pmatrix} U_l & L_l^{-1} F_l \\ 0 & I \end{pmatrix} . \tag{7}$$

Here, $L_l U_l$ is an Incomplete LU factorization of B_l, i.e., $B_l \approx L_l U_l$ and A_{l+1} approximates the Schur complement, so, $A_{l+1} \approx C_l - (E_l U_l^{-1})(L_l^{-1} F_l)$. The matrix A_{l+1} is the coefficient matrix for the linear system at the next level. It remains sparse because of the ordering selected (group independent sets) and due to the dropping of smaller terms. The L-solves associated with the above block factorization amount to a form of restriction in the PDE context, while the U-solve is similar to a prolongation. Note that the algorithm is fully recursive. At the last level (selected in advance, or by exhaustion) a simple ILU factorization is used instead of the one above.

6 Highly indefinite problems: nonsymmetric orderings

Perhaps one of the most significant advances on "general purpose iterative solvers" of the last few years is the realization that permuting a matrix in a nonsymmetric way, before applying a preconditioning, can lead to a robust iterative solution strategy [11, 10, 2]. By permuting A nonsymmetrically we mean a transformation of A of the form PAQ^T, where P and Q are two different permutations. In particular, a significant difference between this situation and the standard one where $P = Q$, is that non-diagonal entries will be moved into the main diagonal. In fact the gist of these methods is to move large entries of the matrix onto the diagonal. This was explored for many years by researchers in sparse direct methods, as a means of avoiding dynamic pivoting in Gaussian elimination [22].

In [10, 11], a (one-sided) permutation P was sought by attempting to maximize the magnitude of the product of the diagonal entries of PA. Here we briefly outline a method which also attempts to place large entries onto the diagonal, by using a more dynamic procedure based on Schur complements. The idea here is to adapt the ARMS algorithm outlined earlier by exploiting nonsymmetric permutations. We will find two permutations P (rows) and Q (columns) to transform A into

$$PAQ^T = \begin{pmatrix} B & F \\ E & C \end{pmatrix} . \tag{8}$$

No particular structure is assumed for the B block. The only requirement on P, Q is that for the resulting matrix in (8), the B block has the 'most diagonally dominant' rows (after nonsym perm) and few nonzero elements (to reduce fill-in). Once the permutations are found and the matrix is permuted as shown above, we can proceed exactly as for ARMS by invoking a multi-level procedure. So, at the l-th level we reorder A into PAQ^T, and then carry out an approximate block factorization identical with that of (7), except that the left-hand side is now PAQ^T instead of PAP^T. The rationale for this approach is that it is critical to have an accurate and well-conditioned B block, [3, 4, 5]. In the case when B is of dimension 1, one can think of this approach as a form of complete pivoting ILU.

The B block is defined by the *Matching set* $\mathcal{M}$ which is a set of n_M pairs (p_i, q_i) where $n_M \leq n$ with $1 \leq p_i, q_i \leq n$ for $i = 1, \ldots, n_M$ and $p_i \neq p_j$, for $i \neq j$ $\quad q_i \neq q_j$, for $i \neq j$ The case $n_M = n$ yields the (full) permutation pair (P, Q). A partial matching set can be easily completed into a full pair (P, Q) by a greedy approach.

The algorithm to find permutation consists of 3 phases. First, a *preselection* phase is invoked to filter out poor rows by employing a criterion based on diagonal dominance. The main goal of this preselection phase is only to reduce the cost of the next phase. Second, a *matching* phase scans candidate entries in order given by the preselection algorithm and accepts them into the $\mathcal{M}$ set, or rejects them. Heuristic arguments, mostly based on greedy procedures, are used for this. Finally, the third phase *completes the matching set* to obtain a pair of (full) permutations P, Q, using a greedy procedure.

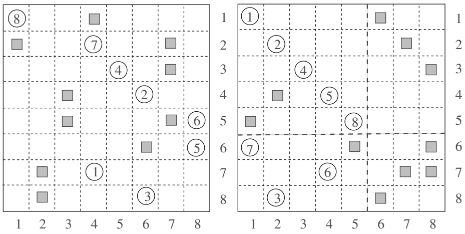


Fig. 5. Illustration of the greedy matching algorithm. Left side: a matrix after the preselection algorithm. Right side: Matrix after Matching permutation.

An illustration of the matching procedure is shown in Figure 5. The left side shows a certain matrix after the preselection procedure. The circled entries are the maximum entries in each row and they are assigned a rank based on the diagonal dominance ratio (the higher the better) and possibly the number of nonzero entries in the row (the fewer the better). The greedy matching algorithm will simply traverse these nodes in the order by which they are ranked, and then determine whether or not to assign the node to $\mathcal{M}$. Thus, entries labeled 1 (a_{74} in original matrix) and 2 ($a_{4,6}$ in original matrix) are accepted. Entry labeled 3 (a_{86}) is not because it is already in the same column as $a_{4,6}$. The algorithm continues in this manner until exhaustion of all nodes. This yields a partial permutation pair which is then completed arbitrarily. The matrix on the right shows the permuted matrix. The B block, separated by longer dash lines, is then eliminated and the process is repeated recursively on the Schur complement, in the same manner as the ARMS procedure. Details can be found in [26], along with a few more elaborate matching procedures.

As an example, Figure 6 shows an algorithm of this type in action for a highly indefinite and unstructured matrix, BP1000, obtained from the old Harwell-Boeing collection [2]. The matrix pattern is shown in the top left part of the figure. Most of the diagonal entries of the matrix are zero and as a result standard iterative methods will fail. Five levels are required by the procedure with the last block reaching a size

[2] See `http://math.nist.gov/MatrixMarket/`

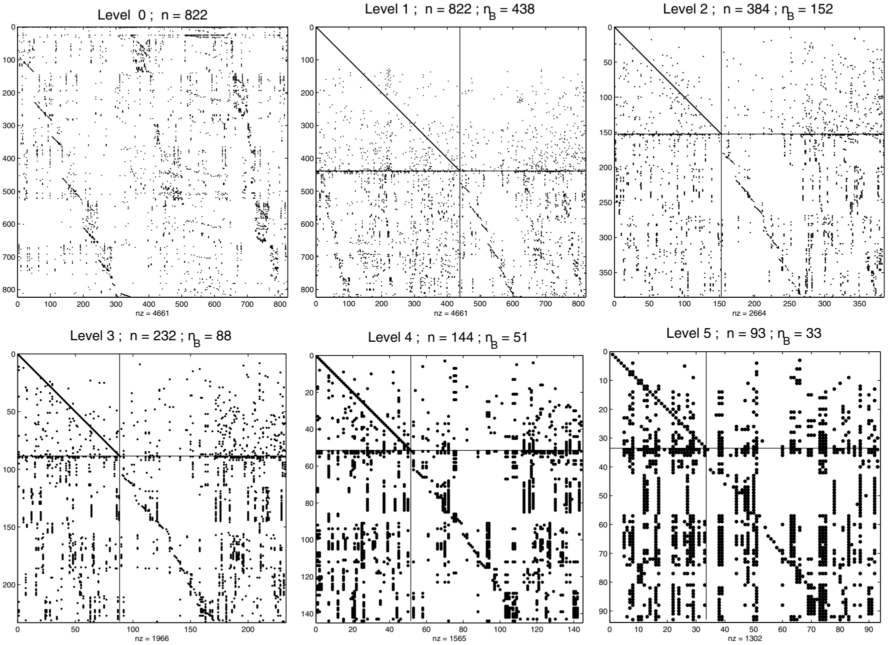


Fig. 6. The Diagonal Dominance PQ-ordering in action for a highly unstructured matrix.

of $n = 60$. With this the resulting preconditioning, GMRES converges in 17 steps. In addition this is achieved with a 'fill-factor' of 2.09, i.e., the ratio of the memory required for the preconditioner over that of the original matrix is 2.09. For additional experiments of more realistic problems see [26].

7 Wirebaskets and hierarchical graph decomposition

It was often observed in the domain decomposition literature that "cross points" play a significant role. This was exploited in [29] in a method known as the wirebasket preconditioner. Recently we have considered a method of the same type from an algebraic viewpoint [14]. This algorithm, called Parallel Hierarchical Interface Decomposition ALgorithm (PHIDAL), descends recursively into interface variables, by exploiting a hierarchy of 'interfaces'. Its main difference with the parallel version of ARMS, is that it uses a *static* ordering instead of a dynamic one. This results in fast preprocessing and, potentially, better parallelism.

To explain the algorithm, consider a graph $\mathcal{G}$ that is partitioned into p subgraphs. However, we now consider an edge-based partitioning, i.e., there are overlapping vertices. The illustration on the left side of Figure 7 shows the graph of a matrix associated with a 5-point FD discretization of a Laplacean on a 2-D domain. One can distinguish three types of nodes: interior, interface, and cross-points. Imagine now that we order the nodes according to this division: we would label all interior points first, followed by the interface points followed by the cross-points. Of course the points in the same set (in this case whether interior nodes, domain edges) are always labeled together. The result of this reordering would be the matrix shown on the right of Figure 7. We refer to the connected subsets as "connectors". The interiors of the subdomains as well as the domain edges are connectors, as are the cross-points.

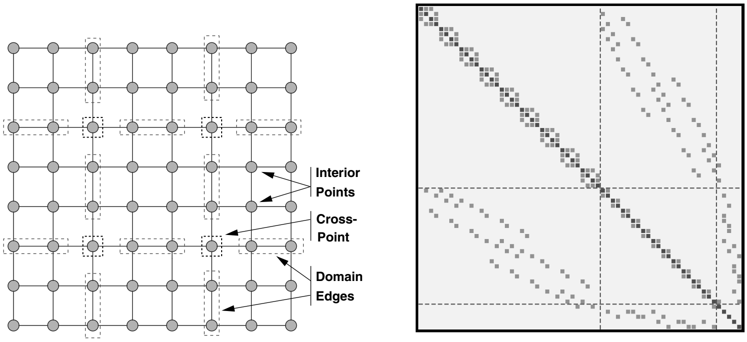


Fig. 7. A small finite difference mesh (left); Pattern of the matrix after the HID ordering.

This ordering is very appealing for parallel processing. If we do not allow any fill-in between the connectors, then the factorization will proceed in parallel at each level. For this example, there are 3 levels: one for the interior points, the second is that of the domain edges, and the 3rd is that of the cross-points. An idea similar to the one discussed here was described in [19, 20] including some analysis [21], though the setting was that of regular meshes. In [14], the above decomposition was extended to general graphs.

An extention of the above definition requires us to *partition the graph into levels of subgraphs with the requirements that the subgraphs at a given level separate those at lower levels.* We will call a connector a connected component in the adjacency graph. A level consists of a collection of connectors with the following requirements: (1) Connectors at any level should separate connectors of previous levels; (2) Connectors of the same level are not coupled (just as in ARMS).

One of the simplest (and clearly not the best) ways to obtain this decomposition is to use the number of domains to which a node belongs. We can label each node u with list $key(u)$ of domains to which it belongs and then define the Level k to be the set of nodes such that $|key(u)| = k + 1$, for $k = 1, 2, \ldots,$. The next task would be to refine the labeling of the connectors to make them independent. The simplest refinement is based on a greedy approach which would relabel a connector by a higher label if it is connected to another connector of the same level. There are many possible refinements, and the reader is referred to [14] for details.

By reordering the nodes hierarchically at the outset, it is possible to create Schur complements that can be made sparse. Once a Schur complement at a given level is constructed it is then possible to create another level. The two important ingredients of this procedure are: (1) algorithms for building a good levelization (few levels); and (2) good combination of effective dropping strategies and parallel incomplete factorization. Results shown in [14] indicate almost perfect scalability for simple model problems (Poisson's problem on a regular mesh) and good scalability for a much harder problem issued from a Magneto Hydrodynamics problem.

8 Concluding remarks

Schur complement techniques can lead to very successful parallel or sequential iterative procedures for solving general sparse linear systems. One of the most important ingredients that is exploited when taking a purely algebraic viewpoint is to reorder the equations in such a way that the next Schur complement is again sparse. This is exploited in techniques such as MRILU [7, 6] and ILUM [25], MLILU [1] and the closely related ARMS [28], and in PHIDAL [14]. Some of these techniques have their analogue in the classical DD literature, a good example being the PHIDAL preconditioner. Other types of reorderings exploit nonsymmetric permutations in order to first eliminate the easier equations. These techniques do not have obvious analogues in the classical DD literature. Because they represent an important set of tools to bridge the gap between the robustness of iterative methods and that of direct solvers, their extension to parallel computing environments, which is still lacking, is of critical importance.

9 Acknowledgments

Most of the work presented in this paper summarizes collaborative efforts with a number of co-workers. The Algebraic Recursive Multilevel Solvers (ARMS) were developed with Zhongze Li, Masha Sosonkina, and Brian Suchomel. The work on PHIDAL, which is still on-going, is in collaboration with Pascal Henon. The preliminary work on the use of hypergraphs within ARMS is joint work with Masha Sosonkina.

References

1. R. E. Bank and C. Wagner, *Multilevel ILU decomposition*, Numer. Math., 82 (1999), pp. 543–576.
2. M. Benzi, J. C. Haws, and M. Tuma, *Preconditioning highly indefinite and nonsymmetric matrices*, SIAM J. Sci. Comput., 22 (2000), pp. 1333–1353.
3. M. Bollöfer, *A robust ILU with pivoting based on monitoring the growth of the inverse factors*, Lin. Alg. Appl., 338 (2001), pp. 201–218.
4. M. Bollöfer and Y. Saad, *ILUPACK - preconditioning software package, release v1.0, may 14, 2004*. Available online at http://www.tuberlin.de/ilupack/.
5. M. Bollöfer and Y. Saad, *Multilevel preconditioners constructed from inverse-based ILUs*, SIAM J. Matrix Anal. Appl., 27 (2006), pp. 1627–1650.

6. E. F. F. Botta, A. van der Ploeg, and F. W. Wubs, *A fast linear-system solver for large unstructured problems on a shared-memory computer*, in Proceedings of the Conference on Algebraic Multilevel Methods with Applications, O. Axelsson and B. Polman, eds., 1996, pp. 105–116.
7. E. F. F. Botta and F. W. Wubs, *Matrix renumbering ILU: an effective algebraic multilevel ILU*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 1007–1026.
8. U. V. Catalyurek and C. Aykanat, *Hypergraph-partitioning-based decomposition for parallel sparse-matrix vector multiplication*, IEEE Trans. Parallel and Distributed Systems, 10 (1999), pp. 673–693.
9. T. H. Cormen, C. E. Leiserson, and R. L. Rivest, *Introduction to Algorithms*, McGraw Hill, New York, 1990.
10. I. S. Duff and J. Koster, *The design and use of algorithms for permuting large entries to the diagonal of sparse matrices*, SIAM J. Matrix Anal. Appl., 20 (1999), pp. 889–901.
11. ———, *On algorithms for permuting large entries to the diagonal of a sparse matrix*, SIAM J. Matrix Anal. Appl., 22 (2001), pp. 973–996.
12. B. Hendrickson and T. G. Kolda, *Graph partitioning models for parallel computing*, Parallel Computing, 26 (2000), pp. 1519–1534.
13. B. Hendrickson and R. Leland, *The Chaco User's Guide Version 2*, Sandia National Laboratories, Albuquerque NM, 1995.
14. P. Henon and Y. Saad, *A parallel multilevel ILU factorization based on a hierarchical graph decomposition*, Tech. Rep. UMSI-2004-74, Minnesota Supercomputer Institute, University of Minnesota, Minneapolis, MN, 2004.
15. G. Karypis and V. Kumar, *A fast and high quality multilevel scheme for partitioning irregular graphs*, SIAM J. Sci. Comput., 20 (1999), pp. 359–392.
16. T. G. Kolda, *Partitioning sparse rectangular matrices for parallel processing*, Lecture Notes in Computer Science, 1457 (1998), pp. 68–79.
17. M. R. Leuze, *Independent set orderings for parallel matrix factorizations by Gaussian elimination*, Parallel Computing, 10 (1989), pp. 177–191.
18. Z. Li, Y. Saad, and M. Sosonkina, *pARMS: a parallel version of the algebraic recursive multilevel solver*, Numer. Linear Algebra Appl., 10 (2003), pp. 485–509.
19. M. M. monga Made and H. A. van der Vorst, *A generalized domain decomposition paradigm for parallel incomplete LU factorization preconditionings*, Future Generation Computer Systems, 17 (2001), pp. 925–932.
20. ———, *Parallel incomplete factorizations with pseudo-overlapped subdomains*, Parallel Computing, 27 (2001), pp. 989–1008.
21. ———, *Spectral analysis of parallel incomplete factorizations with implicit pseudo-overlap*, Numer. Linear Algebra Appl., 9 (2002), pp. 45–64.
22. M. Olschowska and A. Neumaier, *A new pivoting strategy for Gaussian elimination*, Lin. Alg. Appl., 240 (1996), pp. 131–151.
23. F. Pellegrini, *SCOTCH 4.0 user's guide*, tech. rep., INRIA Futurs, April 2005. http://www.labri.fr/perso/pelegrin/scotch/.
24. A. Pothen, H. D. Simon, and K.-P. Liou, *Partitioning sparse matrices with Eigenvectors of graphs*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 430–452.
25. Y. Saad, *ILUM: a multi-elimination ILU preconditioner for general sparse matrices*, SIAM J. Sci. Comput., (1996), pp. 830–847.
26. ———, *Multilevel ILU with reorderings for diagonal dominance*, SIAM J. Sci. Comput., 27 (2005), pp. 1032–1057.

27. Y. Saad and M. Sosonkina, *Non-standard parallel solution strategies for distributed sparse linear systems*, in Parallel Computation: 4th international ACPC conference, Salzburg Austria, February 1999, P. Zinterhof, M. Vajtersic, and A. Uhl, eds., vol. 1557 of Lecture Notes in Computer Science, Springer-Verlag, 1999, pp. 13–27.
28. Y. Saad and B. Suchomel, *ARMS: An algebraic recursive multilevel solver for general sparse linear systems*, Numer. Linear Algebra Appl., 9 (2002), pp. 359–378.
29. B. F. Smith, *Domain Decomposition Algorithms for the Partial Differential Equations of Linear Elasticity*, PhD thesis, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, September 1990.
30. A. van der Ploeg, E. F. F. Botta, and F. W. Wubs, *Nested grids ILU decomposition (NGILU)*, J. Comp. Appl. Math., 66 (1996), pp. 515–526.

Schwarz Preconditioning for High Order Simplicial Finite Elements

Joachim Schöberl[1], Jens M. Melenk[2], Clemens G. A. Pechstein[3], and Sabine C. Zaglmayr[1]

[1] Radon Institute for Computational and Applied Mathematics (RICAM), Austria. {joachim.schoeberl,sabine.zaglmayr}@oeaw.ac.at

[2] The University of Reading, Department of Mathematics, UK. j.m.melenk@reading.ac.uk

[3] Institute for Computational Mathematics, Johannes Kepler University, Linz, Austria. clemens.pechstein@numa.uni-linz.ac.at

Summary. This paper analyzes two-level Schwarz methods for matrices arising from the p-version finite element method on triangular and tetrahedral meshes. The coarse level consists of the lowest order finite element space. On the fine level, we investigate several decompositions with large or small overlap leading to optimal or close to optimal condition numbers. The analysis is confirmed by numerical experiments for a model problem.

The first and the last author acknowledge their support by the Austrian Science Foundation FWF within project grant Start Y-192, "hpFEM : Fast Solvers and Adaptivity"

1 Introduction

High order finite element methods can lead to very high accuracy and are thus attracting increasing attention in many fields of computational science and engineering. The monographs [26, 4, 23, 15, 27] give a broad overview of theoretical and practical aspects of high order methods.

As the problem size increases (due to small mesh-size h and high polynomial order p), the cost of solving the linear systems that arise comes to dominate the solution time. Here, iterative solvers can reduce the total simulation time. We consider preconditioners based on domain decomposition methods [11, 13, 25, 28, 21]. The concept is to consider each high order element as an individual subdomain. Such methods have been studied in [17, 3, 20, 1, 2, 9, 8, 14, 24, 18, 12]. We assume that the local problems can be solved directly. On tensor product elements, one can apply optimal preconditioners for the local sub-problems as in [16, 6, 7].

In the current work, we study overlapping Schwarz preconditioners with large or small overlap. The condition numbers are bounded uniformly in the mesh size h and the polynomial order p. To our knowledge, this is a new result for tetrahedral meshes. We construct explicitly the decomposition of a global function into a coarse grid part and local contributions associated with the vertices, edges, faces, and elements of the mesh. In this paper, we sketch the analysis for the two dimensional version, and give the result for the 3D case. All proofs are given in the longer version [22].

The rest of the paper is organized as follows: In Section 2 we state the problem and formulate the main results. We sketch the 2D case in Section 3 and extend the result for 3D in Section 4. Finally, in Section 5 we give numerical results for several versions of the analyzed preconditioners.

2 Definitions and Main Result

We consider the Poisson equation on the polyhedral domain Ω with homogeneous Dirichlet boundary conditions on $\Gamma_D \subset \partial\Omega$, and Neumann boundary conditions on the remaining part Γ_N. With the sub-space $V := \{v \in H^1(\Omega) : v = 0 \text{ on } \Gamma_D\}$, the bilinear-form $A(\cdot,\cdot) : V \times V \to \mathbb{R}$ and the linear-form $f(\cdot) : V \to \mathbb{R}$ defined as

$$A(u,v) = \int_\Omega \nabla u \cdot \nabla v \, dx \qquad f(v) = \int_\Omega f v \, dx,$$

the weak formulation reads

$$\text{find } u \in V \text{ such that} \qquad A(u,v) = f(v) \qquad \forall\, v \in V. \tag{1}$$

We assume that the domain Ω is sub-divided into straight-sided triangular or tetrahedral elements. In general, constants in the estimates depend on the shape of the elements, but they do not depend on the local mesh-size. We define the set of vertices $\mathcal{V} = \{V\}$, the set of edges $\mathcal{E} = \{E\}$, the set of faces (3D only) $\mathcal{F} = \{F\}$, the set of elements $\mathcal{T} = \{T\}$. We define the sets $\mathcal{V}_f, \mathcal{E}_f, \mathcal{F}_f$ of *free* vertices, edges, and faces not completely contained in the Dirichlet boundary. The high order finite element space is

$$V_p = \{v \in V : v|_T \in P^p \; \forall T \in \mathcal{T}\},$$

where P^p is the space of polynomials up to total order p. As usual, we choose a basis consisting of lowest order affine-linear functions associated with the vertices, and of edge-based, face-based, and cell-based bubble functions. The Galerkin projection onto V_p leads to a large system of linear equations, which shall be solved with the preconditioned conjugate gradient iteration.

This paper is concerned with the analysis of additive Schwarz preconditioning. The basic method is defined by the following space splitting. In Section 5 we will consider several cheaper versions resulting from our analysis. The coarse sub-space is the global lowest order space

$$V_0 := \{v \in V : v|_T \in P^1 \; \forall T \in \mathcal{T}\}.$$

For each inner vertex we define the vertex patch $\omega_V = \bigcup_{T \in \mathcal{T} : V \in T} T$ and the vertex sub-space

$$V_V = \{v \in V_p : v = 0 \text{ in } \Omega \setminus \omega_V\}.$$

For vertices V not on the Neumann boundary, this definition coincides with $V_p \cap H_0^1(\omega_V)$. The additive Schwarz preconditioning operator is $C^{-1} : V_p^* \to V_p$ defined by

$$C^{-1} d = w_0 + \sum_{V \in \mathcal{V}} w_V$$

with $w_0 \in V_0$ such that

$$A(w_0, v) = \langle d, v \rangle \qquad \forall\, v \in V_0,$$

and $w_V \in V_V$ defined such that

$$A(w_V, v) = \langle d, v \rangle \qquad \forall\, v \in V_V.$$

This method is very simple to implement for the p-version method using a hierarchical basis. The low-order block requires the inversion of the sub-matrix according to the vertex basis functions. The high order blocks are block-Jacobi steps, where the blocks contain all vertex, edge, face, and cell unknowns associated with mesh entities containing the vertex V. The main result of this paper is to prove optimal results for the spectral bounds:

Theorem 1. *The constants λ_1 and λ_2 of the spectral bounds*

$$\lambda_1 \langle Cu, u \rangle \leq A(u, u) \leq \lambda_2 \langle Cu, u \rangle \qquad \forall\, u \in V_p$$

are independent of the mesh-size h and the polynomial order p.

The proof is based on the additive Schwarz theory, which allows us to express the C-form by means of the space decomposition:

$$\langle Cu, u \rangle = \inf_{\substack{u = u_0 + \sum_V u_V \\ u_0 \in V_0, u_V \in V_V}} \|u_0\|_A^2 + \sum \|u_V\|_A^2.$$

The constant λ_2 follows immediately from a finite number of overlapping sub-spaces. In the core part of this paper, we construct an explicit and stable decomposition of u into sub-space functions. Section 3 introduces the decomposition for the case of triangles, in Section 4 we prove the results for tetrahedra.

3 Sub-space splitting for triangles

The strategy of the proof is the following: First, we subtract a coarse grid function to eliminate the h-dependency. By stepwise elimination, the remaining function is then split into sums of vertex-based, edge-based and inner functions. For each partial sum, we give the stability estimate. This stronger result contains Theorem 1, since we can choose corresponding vertices for the edge and inner contributions (see also Section 5).

3.1 Coarse grid contribution

In the first step, we subtract a coarse grid function:

Lemma 1. *For any $u \in V_p$ there exists a decomposition*

$$u = u_0 + u_1 \tag{2}$$

such that $u_0 \in V_0$ and

$$\|u_0\|_A^2 + \|\nabla u_1\|_{L_2}^2 + \|h^{-1} u_1\|_{L_2}^2 \preceq \|u\|_A^2.$$

Proof. We choose $u_0 = \Pi_h u$, where Π_h is the Clément-operator [10]. The norm bounds are exactly the continuity and approximation properties of this operator.

>From now on, u_1 denotes the second term in the decomposition (2).

3.2 Vertex contributions

In the second step, we subtract functions u_V to eliminate vertex values. Since vertex interpolation is not bounded in H^1, we cannot use it. Thus, we construct a new averaging operator mapping into a larger space.

In the following, let V be a vertex not on the Dirichlet boundary Γ_D, and let φ_V be the piece-wise linear basis function associated with this vertex. Furthermore, for $s \in [0,1]$ we define the level sets

$$\gamma_V(s) := \{y \in \omega_V : \varphi_V(y) = s\},$$

and write $\gamma_V(x) := \gamma_V(\varphi_V(x))$ for $x \in \omega_V$. For internal vertices V, the level set $\gamma_V(0)$ coincides with the boundary $\partial\omega_V$ (cf. Figure 1). The space of functions being constant on these sets reads

$$S_V := \{w \in L_2(\omega_V) : w|_{\gamma_V(s)} = \text{const},\ s \in [0,1]\ a.e.\};$$

its finite dimensional counterpart is

$$S_{V,p} := S_V \cap V_p = \text{span}\{1, \varphi_V, ..., \varphi_V^p\}.$$

We introduce the *spider averaging operator*

$$\big(\Pi^V v\big)(x) := \frac{1}{|\gamma_V(x)|} \int_{\gamma_V(x)} v(y)\, dy, \qquad \text{for}\, v \in L_2(\omega_V).$$

To satisfy homogeneous boundary conditions, we add a correction term as follows (see Figure 2)

$$\big(\Pi_0^V v\big)(x) := \big(\Pi^V v\big)(x) - (\Pi^V v)|_{\gamma_V(0)}(1 - \varphi_V(x)).$$

Lemma 2. *The averaging operators fulfill the following algebraic properties*

(i)

$$\Pi^V V_p = S_{V,p},$$

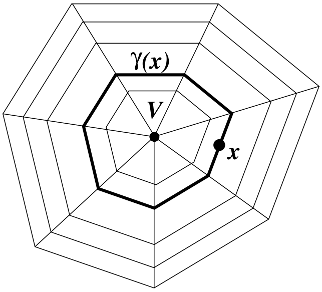


Fig. 1. The level sets $\gamma_V(x)$

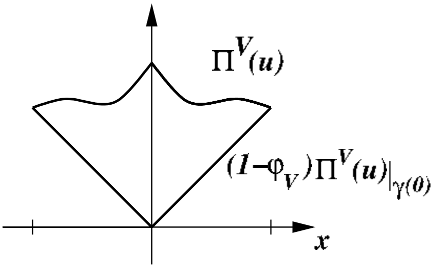


Fig. 2. Construction of Π_0^V

(ii)

$$\Pi_0^V V_p = S_{V,p} \cap V_V,$$

(iii) if u *is continuous at* V*, then*

$$(\Pi^V u)(V) = \Pi_0^V u(V) = u(V).$$

The proof follows immediately from the definitions.

We denote the distance to the vertex V, and the minimal distance to any vertex in $\mathcal{V}$ by

$$r_V(x) := |x - V| \qquad \text{and} \qquad r_{\mathcal{V}}(x) := \min_{V \in \mathcal{V}} r_V(x).$$

Lemma 3. *The averaging operators satisfy the following norm estimates*

(i)

$$\|\nabla\, \Pi^V u\|_{L_2(\omega_V)} \preceq \|\nabla u\|_{L_2(\omega_V)}$$

(ii)

$$\|r_V^{-1}\{u - \Pi^V u\}\|_{L_2(\omega_V)} \preceq \|\nabla u\|_{L_2(\omega_V)}$$

(iii)

$$\|\nabla\{\varphi_V u - \Pi_0^V u\}\|_{L_2(\omega_V)} \preceq \|\nabla u\|_{L_2(\omega_V)}$$

(iv)

$$\|r_{\mathcal{V}}^{-1}\{\varphi_V u - \Pi_0^V u\}\|_{L_2(\omega_V)} \preceq \|\nabla u\|_{L_2(\omega_V)}$$

The proof is given in [22].

The *global spider vertex operator* is

$$\Pi_{\mathcal{V}} := \sum_{V \in \mathcal{V}_f} \Pi_0^V.$$

Obviously, $u - \Pi_{\mathcal{V}} u$ vanishes in any vertex $V \in \mathcal{V}_f$. These well-defined zero vertex values are reflected by the following norm definition:

$$|\!|\!| \cdot |\!|\!|^2 := \|\nabla \cdot\|^2_{L_2(\Omega)} + \|\frac{1}{r_{\mathcal{V}}} \cdot \|^2_{L_2(\Omega)} \qquad (3)$$

Theorem 2. *Let u_1 be as in Lemma 1. Then, the decomposition*

$$u_1 = \sum_{V\in\mathcal{V}_f} \varPi_0^V u_1 + u_2 \tag{4}$$

is stable in the sense of

$$\sum_{V\in\mathcal{V}_f} \|\varPi_0^V u_1\|_A^2 + |\!|\!| u_2 |\!|\!|^2 \preceq \|u\|_A^2. \tag{5}$$

The proof is given in [22]. For the rest of this section, u_2 denotes the second term in the decomposition (4).

3.3 Edge contributions

As seen in the last subsection, the remaining function u_2 vanishes in all vertices. We now introduce an edge-based interpolation operator to carry the decomposition further, such that the remaining function, u_3, contributes only to the inner basis functions of each element.

Therefore we need a lifting operator which extends edge functions to the whole triangle preserving the polynomial order. Such operators were introduced in Babuška et al. [3], and later simplified and extended for 3D by Muñoz-Sola [19]. The lifting on the reference element T^R with vertices $(-1,0)$, $(1,0)$, $(0,1)$ and edges $E_1^R := (-1,1)\times\{0\}$, E_2^R, E_3^R reads:

$$(\mathcal{R}_1 w)(x_1,x_2) := \frac{1}{2x_2}\int_{x_1-x_2}^{x_1+x_2} w(s)ds,$$

for $w \in L_1([-1,1])$. The modification by Muñoz-Sola preserving zero boundary values on the edges E_2^R and E_3^R is

$$(\mathcal{R}w)(x_1,x_2) := (1-x_1-x_2)\,(1+x_1-x_2)\,\Big(\mathcal{R}_1\frac{w}{1-x_1^2}\Big)(x_1,x_2).$$

For an arbitrary triangle $T = F_T(T^R)$ containing the edge $E = F_T(E_1^R)$, its transformed version reads $\mathcal{R}_T w := \mathcal{R}\big[w\circ F_T\big]\circ F_T^{-1}$. The Sobolev space $H_{00}^{1/2}(E)$ on an edge $E = [V_{E,1}, V_{E,2}]$ is defined by its corresponding norm

$$\|w\|^2_{H_{00}^{1/2}(E)} := \|w\|^2_{H^{1/2}(E)} + \int\limits_E \frac{1}{r_{V_E}} w^2\, ds,$$

with $r_{V_E} := \min\{r_{V_{E,1}}, r_{V_{E,2}}\}$.

We call $\omega_E := \omega_{V_{E,1}} \cap \omega_{V_{E,2}}$ the edge patch. We define an *edge-based interpolation operator* as follows:

$$\begin{aligned} \varPi_0^E : \{v\in V_p : v = 0 \text{ in } \mathcal{V}\} &\to H_0^1(\omega_E)\cap V_p,\\ (\varPi_0^E u)|_T &:= \mathcal{R}_T \operatorname{tr}_E u. \end{aligned} \tag{6}$$

Lemma 4. *The edge-based interpolation operator $\varPi_0^E$ defined in (6) is bounded in the $|\!|\!|\cdot|\!|\!|$-norm:*

$$\|\nabla \varPi_0^E u\|_{L_2(\omega_E)} \preceq |\!|\!| u |\!|\!|_{\omega_E}$$

The proof follows from [3] and [19], and properties of the norm $|\!|\!| \cdot |\!|\!|$.

Theorem 3. *Let u_2 be as in Theorem 2. Then, the decomposition*

$$u_2 = \sum_{E \in \mathcal{E}_f} \mathit{\Pi}_0^E u_2 + u_3 \tag{7}$$

satisfies $u_3 = 0$ on $\bigcup_{E \in \mathcal{E}_f} E$ and is bounded in the sense of

$$\sum_{E \in \mathcal{E}_f} \|\nabla \mathit{\Pi}_0^E u_2\|_{L_2}^2 + \|\nabla u_3\|_{L_2}^2 \preceq |\!|\!| u_2 |\!|\!|^2. \tag{8}$$

3.4 Main result

Proof of Theorem 1 for the case of triangles: Summarizing the last subsections, we have

$$u_1 = u - \mathit{\Pi}_h u, \qquad u_2 = u_1 - \sum_{V \in \mathcal{V}_f} \mathit{\Pi}_0^V u_1, \qquad u_3 = u_2 - \sum_{E \in \mathcal{E}_f} \mathit{\Pi}_0^E u_2,$$

and the decomposition

$$u = \mathit{\Pi}_h u + \sum_{V \in \mathcal{V}_f} \mathit{\Pi}_0^V u_1 + \sum_{E \in \mathcal{E}_f} \mathit{\Pi}_0^E u_2 + \sum_{T \in \mathcal{T}} u_3|_T. \tag{9}$$

is stable in the $\| \cdot \|_A$-norm.

For any edge E or triangle T, we can find a vertex V, such that the corresponding term is in V_V. Since for each vertex only finitely many terms appear, we can use the triangle inequality and finally arrive at the missing spectral bound

$$\langle Cu, u \rangle = \inf_{\substack{u = u_0 + \sum_V u_V \\ u_0 \in V_0, u_V \in V_V}} \|u_0\|_A^2 + \sum_V \|u_V\|_A^2 \preceq \langle Au, u \rangle .$$

4 Sub-space splitting for tetrahedra

Most of the proof for the 3D case follows the strategy introduced in Section 3, so we can use the same definitions. The only principal difference is the edge interpolation operator, which has to be treated in more detail.

We define the level surfaces of the vertex hat basis functions

$$\mathit{\Gamma}_V(x) := \mathit{\Gamma}_V(\varphi_V(x)) := \{y : \varphi_V(y) = \varphi_V(x)\}.$$

As in 2D, we first subtract the coarse grid function

$$u_1 = u - \mathit{\Pi}_h u,$$

and secondly the multi-dimensional vertex interpolant to obtain

$$u_2 = u_1 - \mathit{\Pi}_{\mathcal{V}} u_1,$$

where the definitions of Π^V, Π_0^V, $\Pi_{\mathcal{V}}$ are the same as in Section 3, only the level set lines γ_V are replaced by the level surfaces Γ_V. With the same arguments, one easily shows that

$$\sum_{v \in \mathcal{V}_f} \|\Pi_0^V u_1\|_A^2 + \|\nabla u_2\|_{L_2}^2 + \|r_{\mathcal{V}}^{-1} u_2\|_{L_2}^2 \preceq \|u\|_A^2. \tag{10}$$

We define the level line corresponding to a point x in the edge-patch ω_E as

$$\gamma_E(x) := \{y : \varphi_{V_{E,1}}(y) = \varphi_{V_{E,1}}(x) \text{ and } \varphi_{V_{E,2}}(y) = \varphi_{V_{E,2}}(x)\}$$

The edge averaging operator into S_E reads

$$\big(\Pi^E v\big)(x) := \frac{1}{|\gamma_E(x)|} \int_{\gamma_E(x)} v(y)\, dy.$$

In [22], the edge interpolation operator is modified to preserve zero boundary conditions on the whole edge patch ω_E. The resulting operator is called Π_0^E. We define $\mathcal{E}_f$ as the set of are all free edges, i. e. those which do not lie completely on the Dirichlet boundary. We continue the decomposition with

$$u_3 = u_2 - \sum_{E \in \mathcal{E}_f} \Pi_0^E u_2.$$

It fulfills the stability estimate

$$\sum_{E \in \mathcal{E}_f} \|\Pi_0^E u_2\|_A^2 + \|\nabla u_3\|^2 + \|r_{\mathcal{E}}^{-1} u_3\|^2 \preceq \|\nabla u_2\|^2 + \|r_{\mathcal{V}}^{-1} u_2\|^2. \tag{11}$$

Moreover, $u_3 = 0$ on $\bigcup\limits_{E \in \mathcal{E}_f} E$. Finally, we set

$$u_4 = u_3 - \sum_{F \in \mathcal{F}_f} \Pi_0^F u_3,$$

where the face interpolation operator Π_0^F is defined similar as the edge interpolation operator in 2D.

Proof of Theorem 1 for the case of tetrahedra. The decomposition

$$u = \Pi_h u + \sum_{V \in \mathcal{V}_f} \Pi_0^V u_1 + \sum_{E \in \mathcal{E}_f} \Pi_0^E u_2 + \sum_{F \in \mathcal{F}_f} \Pi_0^F u_3 + \sum_{T \in \mathcal{T}} u_4|_T \tag{12}$$

is stable in the $\|\cdot\|_A$-norm.

5 Numerical results

In this section, we show numerical experiments on model problems to verify the theory elaborated in the last sections and to get the absolute condition numbers hidden in the generic constants. Furthermore, we study two more preconditioners.

We consider the $H^1(\Omega)$ inner product

$$A(u,v) = (\nabla u, \nabla v)_{L_2} + (u,v)_{L_2}$$

on the unit cube $\Omega = (0,1)^3$, which is subdivided into an unstructured mesh consisting of 69 tetrahedra. We vary the polynomial order p from 2 up to 10. The condition numbers of the preconditioned systems are computed by the Lanczos method.

Example 1: The preconditioner is defined by the space-decomposition with big overlap of Theorem 1:

$$V = V_0 + \sum_{V \in \mathcal{V}} V_V$$

The condition number is proven to be independent of h and p. The computed numbers are drawn in Figure 3, labeled 'overlapping V'. The inner unknowns have been eliminated by static condensation. The memory requirement of this preconditioner is considerable: For $p = 10$, the memory needed to store the local Cholesky-factors is about 4.4 times larger than the memory required for the global matrix.

In Section 2 we introduced the space splitting into the coarse space V_0 and the vertex subspaces V_V. However, our proof of Theorem 1 involves the finer splitting of a function u into a coarse function, functions in the spider spaces S_V, edge-, face-based and inner functions. Other additive Schwarz preconditioners with uniform condition numbers are induced by this finer splitting.

Example 2: Now, we decompose the space into the coarse space, the p-dimensional spider-vertex spaces $S_{V,0} = \text{span}\{\varphi_V, \ldots, \varphi_V^p\}$, and the overlapping sub-spaces V_E on the edge patches:

$$V = V_0 + \sum_{V \in \mathcal{V}} S_{V,0} + \sum_{E \in \mathcal{E}} V_E$$

The condition number is proven to be uniform in h and p. The computed values are drawn in Figure 3, labeled 'overlapping E, spider V'. Storing the local factors is now about 80 percent of the memory for the global matrix.

Example 3: The interpolation into the spider-vertex space $S_{V,0}$ has two continuity properties: It is bounded in the energy norm, and the interpolation rest satisfies an error estimate in a weighted L_2-norm, see Lemma 3 and equation (10). Now, we reduce the p-dimensional vertex spaces to the spaces spanned by the low energy vertex functions $\varphi_V^{l.e.}$ defined as solutions of

$$\min_{v \in S_{V,0},\, v(V)=1} \|v\|_A^2.$$

These low energy functions can be approximately expressed by the standard vertex functions via $\varphi_V^{l.e.} = f(\varphi_V)$, where the polynomial f solves a weighted 1D problem and can be given explicitly in terms of Jacobi polynomials, see the upcoming report [5]. The interpolation to the low energy vertex space is uniformly bounded, too. But, the approximation estimate in the weighted L_2-norm depends on p. The preconditioner is now generated by

$$V = V_0 + \sum_{V \in \mathcal{V}} \text{span}\{\varphi_V^{l.e.}\} + \sum_{E \in \mathcal{E}} V_E.$$

The computed values are drawn in Figure 3, labeled 'overlapping E, low energy V', and show a moderate growth in p. Low energy vertex basis functions obtained by orthogonalization on the reference element have also been analyzed in [8, 24].

Example 4: We also tested the preconditioner without additional vertex spaces, i.e.,

$$V = V_0 + \sum_{E \in \mathcal{E}} V_E.$$

Since vertex values must be interpolated by the lowest order functions, the condition number is no longer bounded uniformly in p. The rapidly growing condition numbers are drawn in Figure 4.

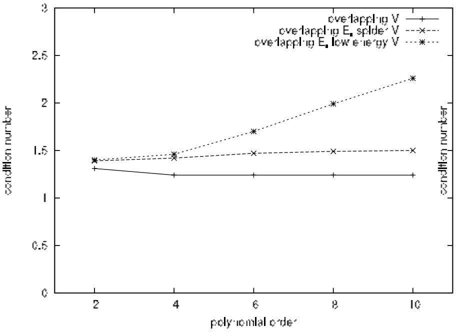


Fig. 3. Overlapping blocks

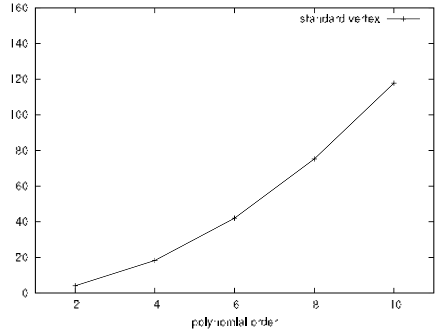


Fig. 4. Standard vertex

References

1. M. Ainsworth, *A hierarchical domain decomposition preconditioner for h-p finite element approximation on locally refined meshes*, SIAM J. Sci. Comput., 17 (1996), pp. 1395–1413.
2. ———, *A preconditioner based on domain decomposition for h-p finite element approximation on quasi-uniform meshes*, SIAM J. Numer. Anal., 33 (1996), pp. 1358–1376.
3. I. Babuška, A. Craig, J. Mandel, and J. Pitkäranta, *Efficient preconditioning for the p-version finite element method in two dimensions*, SIAM J. Numer. Anal., 28 (1991), pp. 624–661.
4. I. Babuška and M. Suri, *The p and hp versions of the finite element method: basic principles and properties*, SIAM Review, 36 (1994), pp. 578–632.
5. A. Bećirović, P. Paule, V. Pillwein, A. Riese, C. Schneider, and J. Schöberl, *Hypergeometric summation algorithms for high order finite elements*, Tech. Rep. 2006-8, SFB F013, Johannes Kepler University, Numerical and Symbolic Scientific Computing, Linz, Austria, 2006.
6. S. Beuchler, R. Schneider, and C. Schwab, *Multiresolution weighted norm equivalences and applications*, Numer. Math., 98 (2004), pp. 67–97.
7. S. Beuchler and J. Schöberl, *Optimal extensions on tensor-product meshes*, Appl. Numer. Math., 54 (2005), pp. 391–405.

8. I. Bica, *Iterative substructuring methods for the p-version finite element method for elliptic problems*, PhD thesis, Courant Institute of Mathematical Sciences, New York University, New York, September 1997.
9. M. A. Casarin, Jr., *Quasi-optimal Schwarz methods for the conforming spectral element discretization*, SIAM J. Numer. Anal., 34 (1997), pp. 2482–2502.
10. P. Clément, *Approximation by finite element functions using local regularization*, RAIRO Anal. Numer., (1975), pp. 77–84.
11. M. Dryja and O. B. Widlund, *Towards a unified theory of domain decomposition algorithms for elliptic problems*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., SIAM, Philadelphia, PA, 1990, pp. 3–21.
12. T. Eibner and J. M. Melenk, *A local error analysis of the boundary concentrated FEM*, IMA J. Numer. Anal., (2006). To appear.
13. M. Griebel and P. Oswald, *On the abstract theory of additive and multiplicative schwarz algorithms*, Numerische Mathematik, 70 (1995), pp. 163–180.
14. B. Guo and W. Cao, *An additive Schwarz method for the h-p version of the finite element method in three dimensions*, SIAM J. Numer. Anal., 35 (1998), pp. 632–654.
15. G. E. Karniadakis and S. J. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, 1999.
16. V. G. Korneev and S. Jensen, *Domain decomposition preconditioning in the hierarchical p-version of the finite element method*, Appl. Numer. Math., 29 (1999), pp. 479–518.
17. J. Mandel, *Iterative solvers by substructuring for the p-version finite element method*, Comput. Methods Appl. Mech. Eng., 80 (1990), pp. 117–128.
18. J. M. Melenk, *On condition numbers in hp-FEM with Gauss-Lobatto based shape functions*, J. Comp. Appl. Math., 139 (2002), pp. 21–48.
19. R. M. noz Sola, *Polynomial liftings on a tetrahedron and applications to the hp-version of the finite element method in three dimensions*, SIAM J. Numer. Anal., 34 (1997), pp. 282–314.
20. L. F. Pavarino, *Additive Schwarz methods for the p-version finite element method*, Numer. Math., 66 (1994), pp. 493–515.
21. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.
22. J. Schöberl, J. M. Melenk, C. G. A. Pechstein, and S. C. Zaglmayr, *Additive Schwarz preconditioning for p-version triangular and tetrahedral finite elements*, Tech. Rep. 2005-11, RICAM, Johann Radon Institute for Computational and Applied Mathematics, Austria Academy of Sciences, Linz, Austria, 2005.
23. C. Schwab, *p- and hp-Finite Element Methods: Theory and Applications in Solid and Fluid Mechanics*, Oxford Science Publications, 1998.
24. S. J. Sherwin and M. A. Casarin, *Low-energy basis preconditioning for elliptic substructured solvers based on unstructured spectral/hp element discretization*, J. Comput. Phys., 171 (2001), pp. 394–417.
25. B. F. Smith, P. E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
26. B. Szabó and I. Babuška, *Finite Element Analysis*, John Wiley & Sons, New York, 1991.

27. B. Szabó, A. Düster, and E. Rank, *The p-version of the finite element method*, in Encyclopedia of Computational Mechanics, E. Stein, R. de Borst, and T. J. R. Hughes, eds., vol. 1, John Wiley & Sons, 2004, ch. 5.
28. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

Part II

Minisymposia

MINISYMPOSIUM 1: Domain Decomposition Methods for Simulation-constrained Optimization

Organizers: Volkan Akcelik[1], George Biros[2], and Omar Ghattas[3]

[1] Carnegie Mellon University. volkan@slac.stanford.edu
[2] University of Pennslyvania. biros@seas.upenn.edu
[3] University of Texas at Austin. omar@ices.utexas.edu

By simulation we refer to numerical solution of systems governed by partial differential (or integral) equations. Tremendous strides in large-scale algorithms and hardware have provided the framework for high fidelity simulations, to the point that it is now practical to consider complex optimization problems. In such problems we wish to determine various parameters that typically consist the data of a simulation: boundary and initial conditions, material properties, distributed forces, or shape.

Due to the large size of such problems special techniques are required for their efficient solution. Domain decomposition algorithms are among the most important. This minisymposium brings scientists working in fast solvers for simulation-constrained optimization together for the development of new algorithmic approaches, and interactions with the rest of the domain decomposition community.

Robust Multilevel Restricted Schwarz Preconditioners and Applications*

Ernesto E. Prudencio[1] and Xiao-Chuan Cai[2]

[1] Advanced Computations Department, Stanford Linear Accelerator Center,
Menlo Park, CA 94025, USA. prudenci@slac.stanford.edu

[2] Department of Computer Science, University of Colorado at Boulder, 430 UCB,
Boulder, CO 80309, USA. cai@colorado.edu

Summary. We introduce a multi-level restricted Schwarz preconditioner with a special coarse-to-fine interpolation and show numerically that the new preconditioner works extremely well for some difficult large systems of linear equations arising from some optimization problems constrained by the incompressible Navier-Stokes equations. Performance of the preconditioner is reported for parameters including number of processors, mesh sizes and Reynolds numbers.

1 Introduction

There are two major families of techniques for solving Karush-Kuhn-Tucker (KKT, or optimality) Jacobian systems, namely the reduced space and the full space methods [2, 3, 12, 11]. When memory is an issue, reduced methods are preferred, although many sub-iterations might be needed to converge the outer-iterations and the parallel scalability is less ideal. As the processing speed and the memory of computers increase, full space methods become more popular because of their increased scalability. One of their main challenges, though, is how to handle the indefiniteness and ill-conditioning of those Jacobians. In addition, some of the solution components might present sharp jumps. Traditional multilevel preconditioning techniques do not work well because of the cross-mesh pollution; i.e., sharp jumps are smoothed out by inter-mesh operations.

We introduce a new multilevel restricted Schwarz preconditioner with a special coarse-to-fine interpolation and show numerically that it works extremely well for rather difficult large Jacobian systems arising from some optimization problems constrained by the incompressible Navier-Stokes equations. The preconditioner is not only scalable but also pollution-free.

Many optimization problems constrained by PDEs can be written as

*The research was supported in part by the National Science Foundation, CCR-0219190 and ACI-0305666, and in part by the Department of Energy, DE-FC02-01ER25479.

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbf{W}} \mathcal{F}(\mathbf{x}) \\ \text{s.t. } \mathbf{C}(\mathbf{x}) = \mathbf{0} \in \mathbf{Y}. \end{cases} \tag{1}$$

Here $\mathbf{W}$ and $\mathbf{Y}$ are normed spaces, $\mathbf{W}$ is the space of optimization variables, $\mathcal{F} : \mathbf{W} \to \mathbb{R}$ is the objective functional and $\mathbf{C} : \mathbf{W} \to \mathbf{Y}$ represents the PDEs. The associated Lagrangian functional $\mathcal{L} : \mathbf{W} \times \mathbf{Y}^* \to \mathbb{R}$ is defined as

$$\mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) \equiv \mathcal{F}(\mathbf{x}) + \langle \boldsymbol{\lambda}, \mathbf{C}(\mathbf{x}) \rangle_{\mathbf{Y}}, \quad \forall\ (\mathbf{x}, \boldsymbol{\lambda}) \in \mathbf{W} \times \mathbf{Y}^*,$$

where $\mathbf{Y}^*$ is the adjoint space of $\mathbf{Y}$, $\langle \cdot, \cdot \rangle_{\mathbf{Y}}$ denotes the duality pairing and variables $\boldsymbol{\lambda}$ are called Lagrange multipliers or adjoint variables. In many cases it is possible to prove that, if $\hat{\mathbf{x}}$ is a (local) solution of (1) then there exist Lagrange multipliers $\hat{\boldsymbol{\lambda}}$ such that $(\hat{\mathbf{x}}, \hat{\boldsymbol{\lambda}})$ is a critical point of $\mathcal{L}$ [10]. So, with a discretize-then-optimize approach [9] and sufficient smoothness assumptions, a solution of (1) has to necessarily solve the KKT system $\nabla \mathcal{L}(\mathbf{x}, \boldsymbol{\lambda}) = \mathbf{0}$ and each iteration of a Newton's method for solving such problem involves the Jacobian system

$$\begin{bmatrix} \boldsymbol{\nabla}_{\mathbf{xx}}\mathcal{L} & [\boldsymbol{\nabla}\mathbf{C}]^T \\ \boldsymbol{\nabla}\mathbf{C} & \mathbf{0} \end{bmatrix} \begin{pmatrix} \mathbf{p}_{\mathbf{x}} \\ \mathbf{p}_{\boldsymbol{\lambda}} \end{pmatrix} = - \begin{pmatrix} \nabla_{\mathbf{x}}\mathcal{L} \\ \mathbf{C} \end{pmatrix}. \tag{2}$$

The paper is organized as follows. Section 2 introduces a preconditioner for (2), while in Section 3 we test it on some flow control problems and report its performance for combinations of parameters including number of processors, mesh sizes and Reynolds numbers. Final conclusions are given in Section 4.

2 Multilevel pollution-removing restricted Schwarz

Schwarz methods can be used in one-level or multilevel variants and, in each case, in combination with additive and/or multiplicative algorithms [13]. They can be also used as linear [8] and nonlinear preconditioners [6].

Let Ω_h be a mesh of characteristic size $h > 0$, subdivided into non-overlapping subdomains Ω_j, $j = 1, \ldots, N_S$. Let $H > 0$ denote the characteristic diameter of $\{\Omega_j\}$ and let $\{\Omega_j^{'}\}$ be an overlapping partition with overlapping $\delta > 0$. From now on we only consider simple box domains, uniform meshes and simple box decompositions, i.e., all subdomains Ω_j and $\Omega_j^{'}$ are rectangular and their boundaries do not cut through any mesh cells. Let N and N_j denote the number of degrees of freedom associated to Ω_h and $\Omega_j^{'}$, respectively. Let $\mathbf{K}$ be a $N \times N$ matrix of a linear system

$$\mathbf{K}\mathbf{p} = \mathbf{b} \tag{3}$$

that needs to be solved during the application of an algorithm for the numerical solution of a discretized differential problem. Let d indicate the number of degrees of freedom per mesh point. For simplicity let us assume that d is the same throughout the entire mesh. We define the $N_j \times N$ matrix $\mathbf{R}_j^{\delta}$ as follows: its $d \times d$ block element $(\mathbf{R}_j^{\delta})_{\alpha,\beta}$ is either (a) an identity block if the integer indices $1 \leqslant \alpha \leqslant N_j/d$ and $1 \leqslant \beta \leqslant N/d$ are related to the same mesh point and this mesh point belongs to $\Omega_j^{'}$ or (b) a zero block otherwise. The multiplication of $\mathbf{R}_j^{\delta}$ with a $N \times 1$ vector generates a smaller $N_j \times 1$ vector by discarding all components corresponding to mesh points outside $\Omega_j^{'}$. The $N_j \times N$ matrix $\mathbf{R}_j^0$ is similarly defined, with the difference that its application to a $N \times 1$ vector also zeros out all those components corresponding to mesh points on $\Omega_j^{'} \setminus \Omega_j$. Let $\mathbf{B}_j^{-1}$ be either the inverse of or a preconditioner for $\mathbf{K}_j \equiv \mathbf{R}_j^\delta \, \mathbf{K} \, \mathbf{R}_j^{\delta^T}$. The one-level classical, right restricted (r-RAS) and left restricted (ℓ-RAS) additive Schwarz preconditioners for $\mathbf{K}$ respectively are defined as [5, 7, 8]

$$\mathbf{B}_{\delta\delta}^{-1} = \sum_{j=1}^{N_s} \mathbf{R}_j^{\delta^T} \mathbf{B}_j^{-1} \mathbf{R}_j^\delta, \; \mathbf{B}_{\delta 0}^{-1} = \sum_{j=1}^{N_s} \mathbf{R}_j^{\delta^T} \mathbf{B}_j^{-1} \mathbf{R}_j^0, \; \mathbf{B}_{0\delta}^{-1} = \sum_{j=1}^{N_s} \mathbf{R}_j^{0^T} \mathbf{B}_j^{-1} \mathbf{R}_j^\delta.$$

For the description of multilevel Schwarz preconditioners, let us use index $i = 0, 1, \ldots, L-1$ to designate any of the $L \geqslant 2$ levels. Let $\mathbf{I}_i$ denote the identity operator and, for $i > 0$, let $\mathbf{R}_i^T$ denote the interpolation from level $i-1$ to level i. Multilevel Schwarz preconditioners are obtained through the combination of one-level Schwarz preconditioners $\mathbf{B}_i^{-1}$ assigned to each level. Here we focus on multilevel preconditioners that use exact coarsest solvers $\mathbf{B}_0^{-1}$ and that can be seen as multigrid V-cycle algorithms [4] having Schwarz preconditioned Richardson working as the pre and the post smoother at each level $i > 0$, with $\mathbf{B}_{i,\text{pre}}^{-1}$ preconditioning the $\mu_i \geqslant 0$ pre smoother iterations and $\mathbf{B}_{i,\text{post}}^{-1}$ preconditioning the $\nu_i \geqslant 0$ post smoother iterations. Then, as iterative methods for (3), with $\mathbf{r}^{(\ell)}$ denoting the residual at iteration $\ell = 0, 1, 2, \ldots$, they can be described in the case $L = 2$ as

$$\mathbf{r}^{(\ell+1)} = (\mathbf{I}_1 - \mathbf{K}_1 \mathbf{B}_{1,\text{post}}^{-1})^{\nu_1} (\mathbf{I}_1 - \mathbf{K}_1 \mathbf{R}_1^T \mathbf{B}_0^{-1} \mathbf{R}_1)(\mathbf{I}_1 - \mathbf{K}_1 \mathbf{B}_{1,\text{pre}}^{-1})^{\mu_1} \mathbf{r}^{(\ell)}. \qquad (4)$$

Pollution removing interpolation constitutes a key procedure in our proposed multilevel preconditioner, due to the sharp jumps that often occur for the multiplier values over those regions of Ω_h where constraints are greatly affecting the behavior of the optimized system. Although the evidence of this discontinuity property of Lagrange multipliers is just empirical in our paper, it is consistent with their interpretation [11]: the value of a Lagrange multiplier at a mesh point gives the rate of change of the optimal objective function value w.r.t. to the respective constraint at that point.

In the case of the problem corresponding to Figure 2-b, for instance, an external force causes the fluid to move clockwise and the boundary consists of rigid slip walls. The vertical walls greatly affect the overall vorticity throughout the domain, i.e., the value of the objective function, because they completely oppose the horizontal velocity component v_1. The values of λ_1 at the walls then reflect this situation. In contrast, λ_2 develops sharp jumps at the other two walls opposing v_2. In all our experiments the discontinuities are located only accross the boundary and not around it, even for very fine meshes. Common coarse-to-fine interpolation techniques will then smooth the sharp jumps present in coarse solutions, with a more gradual change, from interior mesh points towards boundary mesh points, appearing in those fine cells (elements, volumes) located inside coarse boundary ones. That is, the good correction information provided by the coarse solution is lost with a common interpolation. We refer to the smoothed jump as "pollution", in contrast to the "clean" sharp jump that is expected at the fine level as well.

We therefore propose a *modified* coarse-to-fine interpolation procedure that is based on a general and simple "removal of the pollution". Let $\mathbf{R}_i^T$ denote any unmodified interpolation procedure and $\boldsymbol{\mathcal{Z}}_i$ the operator that zeros out, from a vector at level i, the Lagrange multipliers at all those mesh points with equations that have a greater influence on the objective function. For the case of PDEs describing physical systems, the number of such points can be expected to be relatively small. Our modified interpolation is then expressed by

$$\mathbf{R}^T_{i,\mathrm{modif}} = \mathbf{R}^T_i - \boldsymbol{\mathcal{Z}}_i \mathbf{R}^T_i (\mathbf{I}_{i-1} - \boldsymbol{\mathcal{Z}}_{i-1}). \tag{5}$$

This procedure removes the smoothed contributions due to the coarse discontinuities, maintaining, at the fine level, the sharp jumps originally present at the coarse level. See Figure 1. Once $\mathbf{R}^T_i$ is available, (5) can be applied to *any* mesh in *any* dimension, with *any* number of components.

In the case of the problems in this paper, $\boldsymbol{\mathcal{Z}}_i$ zeros the Lagrange multiplier components located at the boundary. In our tests we apply the modified interpolation only for the Lagrange multiplier components of coarse solutions, while the optimization variables continue to be interpolated with $\mathbf{R}^T_i$. Also, the restriction process remains $\mathbf{R}_i$ for all variables, i.e., (4) becomes

$$\mathbf{r}^{(\ell+1)} = (\mathbf{I}_1 - \mathbf{K}_1\mathbf{B}^{-1}_{1,\mathrm{post}})^{\nu_1}(\mathbf{I}_1 - \mathbf{K}_1\mathbf{R}^T_{1,\mathrm{modif}}\mathbf{B}^{-1}_0\mathbf{R}_1)(\mathbf{I}_1 - \mathbf{K}_1\mathbf{B}^{-1}_{1,\mathrm{pre}})^{\mu_1}\mathbf{r}^{(\ell)}.$$

The Lagrange multipliers reflect the eventual "discontinuity" of the type of equations (or their physical dimensions) between equations in different regions of $\overline{\Omega}$: in the case of the problems in Section 3, between those in Ω and those on $\partial\Omega$. From this point of view, it seems "natural" to apply different interpolations to the multiplier components depending on their location.

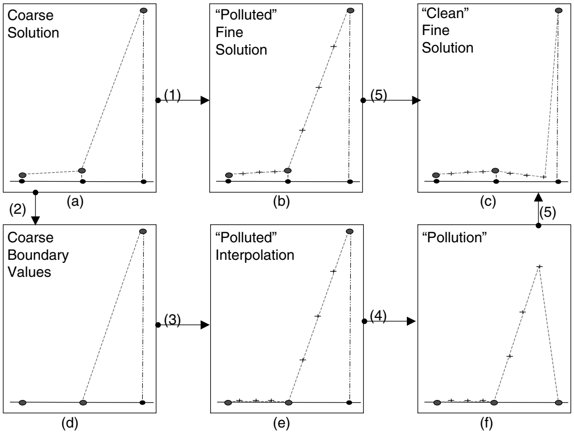


Fig. 1. Representation of the modified coarse-to-fine interpolation (5), with (a) input $\boldsymbol{\varphi}_{i-1}$ and (c) output $\boldsymbol{\varphi}_i$. The five steps are: (1) interpolation $\mathbf{R}^T_i\boldsymbol{\varphi}_{i-1}$, (2) coarse jump values $\tilde{\boldsymbol{\varphi}}_{i-1} = (\mathbf{I}_{i-1} - \boldsymbol{\mathcal{Z}}_{i-1})\boldsymbol{\varphi}_{i-1}$, (3) polluted $\tilde{\boldsymbol{\varphi}}_i = \mathbf{R}^T_i\tilde{\boldsymbol{\varphi}}_{i-1}$, (4) pollution isolation $\boldsymbol{\mathcal{Z}}_i\tilde{\boldsymbol{\varphi}}_i$, (5) pollution removal $\boldsymbol{\varphi}_i = \mathbf{R}^T_i\boldsymbol{\varphi}_{i-1} - \boldsymbol{\mathcal{Z}}_i\tilde{\boldsymbol{\varphi}}_i$.

3 Numerical experiments

Our numerical experiments in this paper focus on optimal control problems [9], where the optimization space in (1) is generally given by $\mathbf{W}=\mathbf{S}\times\mathbf{U}$, with $\mathbf{S}$ being the state space and $\mathbf{U}$ the control space. Upon discretization, one has $n=n_s+n_u$, where n_s (n_u) is the number of discrete state (control) variables. More specifically, we treat the boundary control of two-dimensional steady-state incompressible Navier-Stokes equations in the velocity-vorticity formulation: $\mathbf{v} = (v_1, v_2)$ is the velocity and ω is the vorticity. Let $\Omega \subset \mathbb{R}^2$ be an open and bounded smooth domain, Γ its boundary, $\boldsymbol{\nu}$ the unit outward normal vector along Γ and $\mathbf{f}$ a given external force defined in Ω. Let $L^2(\Omega)$ and $L^2(\Gamma)$ be the spaces of square Lebesgue integrable functions in Ω and Γ respectively. The problems consist of finding $(\mathbf{s},\mathbf{u}) = (v_1, v_2, \omega, u_1, u_2) \in L^2(\Omega)^3 \times L^2(\Gamma)^2 = \mathbf{S}\times\mathbf{U}$ such that the minimization

$$\min_{(\mathbf{s},\mathbf{u})\in\mathbf{S}\times\mathbf{U}} \mathcal{F}(\mathbf{s},\mathbf{u}) = \frac{1}{2}\int_{\Omega} \omega^2 \, d\Omega + \frac{c}{2}\int_{\Gamma} \|\mathbf{u}\|_2^2 \, d\Gamma \tag{6}$$

is achieved subject to the constraints

$$\begin{cases} -\Delta v_1 - \dfrac{\partial \omega}{\partial x_2} & = 0 \ \text{ in } \Omega, \\ -\Delta v_2 + \dfrac{\partial \omega}{\partial x_1} & = 0 \ \text{ in } \Omega, \\ -\Delta \omega + Re\ v_1 \dfrac{\partial \omega}{\partial x_1} + Re\ v_2 \dfrac{\partial \omega}{\partial x_2} - Re \ \operatorname{curl} \mathbf{f} & = 0 \ \text{ in } \Omega, \\ \mathbf{v} - \mathbf{u} & = \mathbf{0} \ \text{ on } \Gamma, \\ \omega + \dfrac{\partial v_1}{\partial x_2} - \dfrac{\partial v_2}{\partial x_1} & = 0 \ \text{ on } \Gamma, \\ \displaystyle\int_{\Gamma} \mathbf{v}\cdot\boldsymbol{\nu} \, d\Gamma & = 0, \end{cases} \tag{7}$$

where $\operatorname{curl} \mathbf{f} = -\partial f_1/\partial x_2 + \partial f_2/\partial x_1$. The parameter $c > 0$ is used to adjust the relative importance of the control norms in achieving the minimization, so indirectly constraining their sizes. The physical objective in (6)-(7) is the minimization of turbulence [9]. The last constraint is due to the mass conservation law, making $m \neq n_s$ and causing the complexity of the Jacobian computation to increase, since non-adjacent mesh points become coupled by the integral. We restrict our numerical experiments to *tangential* boundary control problems, i.e., $\mathbf{u}\cdot\boldsymbol{\nu} = 0$ on Γ, so that $m = n_s$.

Here we only report tests for $\Omega = (0,1)\times(0,1)$, $c = 10^{-2}$ and $\mathbf{f} = (f_1, f_2) = \big(-\sin^2(\pi x_1)\cos(\pi x_2)\sin^2(\pi x_2), \sin^2(\pi x_2)\cos(\pi x_1)\sin^2(\pi x_1)\big)$. For comparison, we solve simulation problems with $\mathbf{v}\cdot\boldsymbol{\nu} = 0$ and $\partial\mathbf{v}/\partial\boldsymbol{\nu} = 0$ on Γ.

We have performed tests on a cluster of Linux PCs and developed our software using the Portable, Extensible Toolkit for Scientific Computing (PETSc) from Argonne National Laboratory [1]. Table 1 shows the efficacy of the modified interpolation process, which performs much better than the unmodified one, causing the two-level preconditioner to outperform the one-level preconditioner. Table 2 shows the flexibility of the two-level preconditioner, which provides a similar average number of Krylov iterations throughout all seven situations in the table. Figure 2-*a* shows the controlled velocity field: the movement near the boundary is less intense. Figures 2-*c* and 2-*d* clearly show the stabilization on the average number of Krylov iterations provided by the two-level preconditioner with modified interpolation. The one-level preconditioner fails with 100 processors for $Re = 250$ and $Re = 300$.

Table 1. Resulting average number $\overline{\ell}$ of Krylov iterations per Newton iteration with *Re*=250, right preconditioned GMRES, a 280 × 280 mesh (631, 688 variables), 49 processors, relative overlapping $\delta/H = 1/4$ and a 70 × 70 coarse mesh, for different combinations of number L of levels, linear interpolation type, number σ of pre and post smoother iterations, and RAS preconditioner.

L	Linear Interpolation Type	σ	RAS preconditioner	
			ℓ-RAS	r-RAS
1	–	–	$\overline{\ell} = 336$	$\overline{\ell} = 973$
2	Unmodified	1	$\overline{\ell} = 1,110$	$\overline{\ell} = 1,150$
2	Unmodified	2	$\overline{\ell} = 356$	$\overline{\ell} = 222$
2	Modified	1	$\boldsymbol{\overline{\ell} = 21}$	$\boldsymbol{\overline{\ell} = 28}$

Table 2. Resulting average number $\overline{\ell}$ of Krylov iterations per Newton iteration with *Re*=300, right preconditioned GMRES and a 70×70 coarse mesh, for different situations of number N_p of processors and mesh size. To each situation corresponds a combination of the number σ of Richardson iterations, the RAS preconditioner and the relative overlapping δ/H used in the pre and post smoothers. The number of variables is 2, 517, 768 in the case of finest mesh.

N_p	$\frac{\delta}{H}$	140×140	280×280	560×560
25	$\frac{1}{4}$	$\sigma = 1$; r-RAS; $\boldsymbol{\overline{\ell} = 20}$	$\sigma = 1$; r-RAS; $\boldsymbol{\overline{\ell} = 23}$	–
49	$\frac{1}{2}$	$\sigma = 1$; r-RAS; $\boldsymbol{\overline{\ell} = 18}$	$\sigma = 1$; r-RAS; $\boldsymbol{\overline{\ell} = 21}$	–
100	$\frac{1}{2}$	$\sigma = 1$; ℓ-RAS; $\boldsymbol{\overline{\ell} = 18}$	$\sigma = 1$; ℓ-RAS; $\boldsymbol{\overline{\ell} = 25}$	$\sigma = 2$; r-RAS; $\boldsymbol{\overline{\ell} = 27}$

4 Conclusions

We have developed a multilevel preconditioner for PDE-constrained optimization that has shown a robust performance when tested on some boundary flow control problems. Our main contribution consists in the combination of a general multigrid V-cycle preconditioner with (1) RAS preconditioned Richardson smoothers and (2) a modified interpolation procedure that removes the pollution often generated by the application of common interpolation techniques to the Lagrange multipliers. Such combination is the key for the success of the two-level method in our experiments and the consequent improvement over the one-level method, handling flow control problems with higher Reynolds number, finer meshes and more processors. Surprisingly, RAS preconditioners performed much better than the classical ones.

Multilevel Schwarz is a flexible algorithm, and since it is also fully coupled (in contrast to operator-splitting, Schur complement, reduced space techniques), the original sparsity of a discretized PDE constrained optimization problem is main-

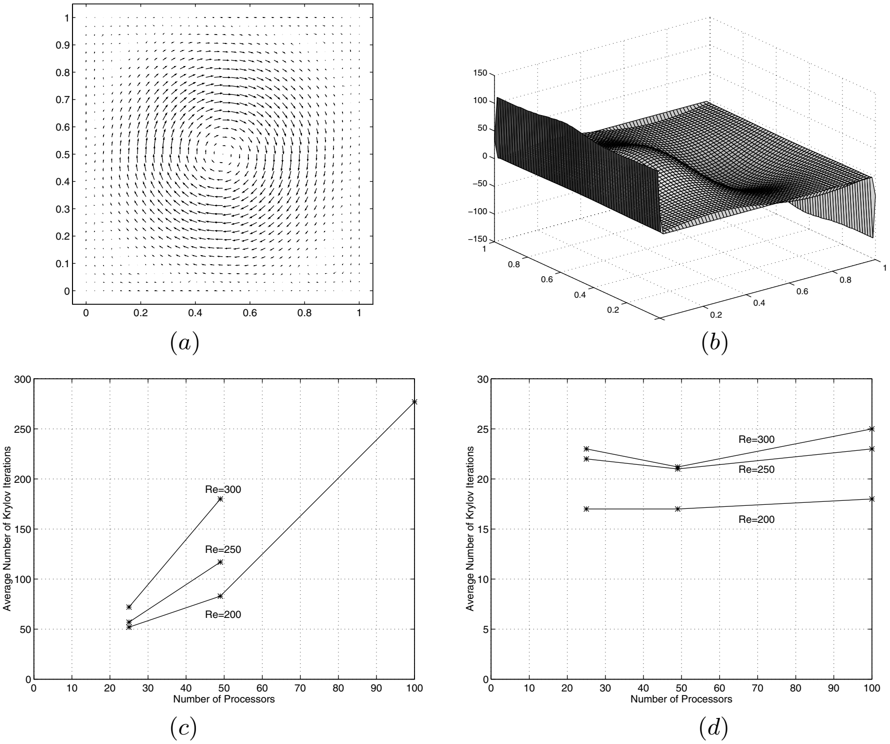


Fig. 2. Information on cavity flow problems: (a) controlled velocity field with $Re = 200$ and (b) corresponding Lagrange multiplier λ_1; results for (c) one-level and (d) two-level preconditioner with right-preconditioned GMRES, a 280×280 mesh ($631,688$ variables), and a 70×70 coarse mesh.

tained throughout its entire application and fewer sequential preconditioning steps are needed. We expect this preconditioner to have wide applications in other areas of computational science and engineering.

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*, Argonne National Laboratory, http://www.mcs.anl.gov/petsc, 2004.
2. 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 (2005), pp. 687–713.
3. ———, *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 (2005), pp. 714–739.

4. W. L. Briggs, V. E. Henson, and S. F. McCormick, *A Multigrid Tutorial*, SIAM, Philadelphia, second ed., 2000.
5. 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 (2003), pp. 1209–1231.
6. X.-C. Cai and D. E. Keyes, *Nonlinearly preconditioned inexact Newton algorithms*, SIAM J. Sci. Comput., 24 (2002), pp. 183–200.
7. X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797.
8. M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci.Comput., 15 (1994), pp. 604–620.
9. M. D. Gunzburger, *Perspectives in Flow Control and Optimization*, SIAM, Philadelphia, 2002.
10. A. D. Ioffe and V. M. Tihomirov, *Theory of Extremal Problems*, North-Holland Publishing Company, first ed., 1979. Translation from Russian edition, (c) 1974 NAUKA, Moscow.
11. E. E. Prudencio, *Parallel Fully Coupled Lagrange-Newton-Krylov-Schwarz Algorithms and Software for Optimization Problems Constrained by Partial Differential Equations*, PhD thesis, Department of Computer Science, University of Colorado at Boulder, 2005.
12. E. 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 (2006), pp. 1305–1328.
13. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

MINISYMPOSIUM 2: Optimized Schwarz Methods

Organizers: Martin Gander[1] and Fréderic Nataf[2]

[1] Swiss Federal Institute of Technology, Geneva. martin.gander@math.unige.ch
[2] Ecole Polytechnique. nataf@cmap.polytechnique.fr

Optimized Schwarz methods are based on the classical Schwarz algorithm, but they use instead of Dirichlet transmission conditions more general transmission conditions between subdomains to enhance the convergence speed, to permit methods to be used without overlap, and to obtain convergent methods for problems for which the classical Schwarz method is not convergent, such as, for example, for the Helmholtz problem from acoustics.

Over the last decade, much progress has been made in the understanding of the optimized Schwarz methods, in the development of effective transmission conditions, both at the continuous and at the discrete level, and in optimization that gives rise to methods that converge fast enough even without Krylov acceleration. This minisymposium gives an overview over the latest results for optimized Schwarz methods, at the continuous, discretized and algebraic level, for stationary partial differential equations.

Optimized Schwarz Methods in Spherical Geometry with an Overset Grid System

Jean Côté[1], Martin J. Gander[2], Lahcen Laayouni[3], and Abdessamad Qaddouri[4]

[1] Recherche en prévision numérique, Environment of Canada, Quebéc, Canada. jean.cote@ec.gc.ca

[2] Section de Mathématiques, Université de Genève, Suisse. Martin.Gander@math.unige.ch

[3] Department of Mathematics and Statistics, McGill University, Montreal, Quebéc, Canada. laayouni@math.mcgill.ca

[4] Recherche en prévision numérique, Environment of Canada, Quebéc, Canada. abdessamad.qaddouri@ec.gc.ca

Summary. In recent years, much attention has been given to domain decomposition methods for solving linear elliptic problems that are based on a partitioning of the domain of the physical problem. More recently, a new class of Schwarz methods known as optimized Schwarz methods was introduced to improve the performance of the classical Schwarz methods. In this paper, we investigate the performance of this new class of methods for solving the model equation $(\eta - \Delta)u = f$, where $\eta > 0$, in spherical geometry. This equation arises in a global weather model as a consequence of an implicit (or semi-implicit) time discretization. We show that the Schwarz methods improved by a non-local transmission condition converge in a finite number of steps. A local approximation permits the use of the new optimized methods on a new overset grid system on the sphere called the Yin-Yang grid.

1 Introduction

Meteorological operational centers are using increasingly parallel computer systems and need efficient strategies for their real-time data assimilation and forecast systems. This motivates the present study, where parallelism based on domain decomposition methods is analyzed for a new overset grid system on the sphere introduced in [6] called the Yin-Yang grid.

We investigate domain decomposition methods for solving $(\eta - \Delta)u = f$, where $\eta > 0$, in spherical geometry. The key idea underlying the optimal Schwarz method has been introduced in [4] in the context of non-linear problems. A new class of Schwarz methods based on this idea was then introduced in [1] and further analyzed in [7] and [5] for convection diffusion problems. For the case of the Poisson equation, see [2], where also the terms optimal and optimized Schwarz were introduced. Optimal Schwarz methods have non-local transmission conditions at the interfaces between subdomains, and are therefore not as easy to use as classical Schwarz methods. Optimized Schwarz methods use local approximations of the optimal non-local transmission conditions of optimal Schwarz at the interfaces, and are therefore as easy to use as classical Schwarz, but have a greatly enhanced performance.

In Section 2, we introduce the model problem on the sphere and the tools of Fourier analysis, we also recall briefly some proprieties of the associated Legendre functions, which we will need in our analysis. In Section 3, we present the Schwarz algorithm for the model problem on the sphere with a possible overlap. We show that asymptotic convergence is very poor in particular for low wave-number modes. In Section 4, we present the optimal Schwarz algorithm for the same configuration. We prove convergence in two iterations for the two subdomain decomposition with non-local convolution transmission conditions. We then introduce a local approximation which permits the use of the new method on a new overset grid system on the sphere called the Yin-Yang grid which is pole-free. In Section 5 we illustrate our findings with numerical experiments.

2 The problem setting on the sphere

Throughout this paper we consider a model problem governed by the following equation

$$\mathcal{L}(u) = (\eta - \Delta)(u) = f, \qquad \text{in} \quad S \subset \mathbb{R}^3, \tag{1}$$

where S is the unit sphere centered at the origin. Using spherical coordinates, equation (1) can be rewritten in the form

$$\mathcal{L}(u) = \left(\eta - \frac{1}{r^2}\frac{\partial}{\partial r}(r^2\frac{\partial}{\partial r}) - \frac{1}{r^2\sin^2\phi}\frac{\partial^2}{\partial\theta^2} - \frac{1}{r^2\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial}{\partial\phi})\right)(u) = f, \tag{2}$$

where ϕ stands for the colatitude, with 0 being the north pole and π being the south pole, and θ is the longitude. For our case on the surface of the unit sphere, we consider solutions independent of r, e.g., $r = 1$, which simplifies (2) to

$$\mathcal{L}(u) = \left(\eta - \frac{1}{\sin^2\phi}\frac{\partial^2}{\partial\theta^2} - \frac{1}{\sin\phi}\frac{\partial}{\partial\phi}(\sin\phi\frac{\partial}{\partial\phi})\right)(u) = f. \tag{3}$$

Our results are based on Fourier analysis. Because u is periodic in θ, it can be expanded in a Fourier series,

$$u(\phi,\theta) = \sum_{m=-\infty}^{\infty} \hat{u}(\phi,m)e^{im\theta}, \quad \hat{u}(\phi,m) = \frac{1}{2\pi}\int_0^{2\pi} e^{-im\theta}u(\phi,\theta)d\theta.$$

With the expanded u, equation (3) becomes a family of ordinary differential equations. For any positive or negative integer m, we have

$$-\frac{\partial^2\hat{u}(\phi,m)}{\partial\phi^2} - \frac{\cos\phi}{\sin\phi}\frac{\partial\hat{u}(\phi,m)}{\partial\phi} + (\eta + \frac{m^2}{\sin^2\phi})\hat{u}(\phi,m) = \hat{f}(\phi,m). \tag{4}$$

By linearity, it suffices to consider only the homogeneous problem, $\hat{f}(\phi,m) = 0$, and analyze convergence to the zero solution. Thus, for m fixed, the homogeneous problem in (4), can be written in the following form

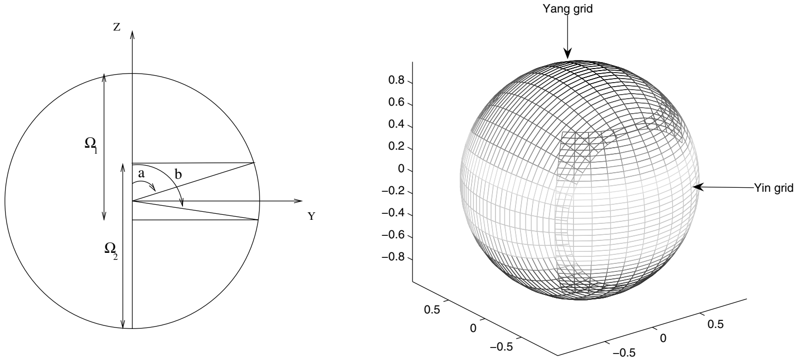


Fig. 1. Left: Two overlapping subdomains. **Right:** The Yin-Yang grid system.

$$\frac{\partial^2 \hat{u}(\phi,m)}{\partial \phi^2} + \frac{\cos\phi}{\sin\phi}\frac{\partial \hat{u}(\phi,m)}{\partial \phi} + (\nu(\nu+1) - \frac{m^2}{\sin^2\phi})\hat{u}(\phi,m) = 0, \tag{5}$$

where $\nu = -1/2 \pm 1/2\sqrt{1-4\eta}$. Note that the solution of equation (5) is independent of the sign of m, and thus, for simplicity, we assume in the sequel that m is a positive integer. Equation (5) is the associated Legendre equation and admits two linearly independent solutions with real values, namely $P_\nu^m(\cos\phi)$ and $P_\nu^m(-\cos\phi)$, see e.g., [3], where $P_\nu^m(\cos\phi)$ is called the conical function of the first kind.

Remark 1. The associated Legendre function can be expressed in terms of the hypergeometric function and one can show that the function $P_\nu^m(\cos\phi)$ has a singularity at $\phi = \pi$ and is monotonically increasing in the interval $[0,\pi]$. Furthermore, the derivative of the function $P_\nu^m(z)$ with respect to the variable z is given by

$$\frac{\partial P_\nu^m(z)}{\partial z} = \frac{1}{1-z^2}\left(-mzP_\nu^m(z) - \sqrt{1-z^2}P_\nu^{m+1}(z)\right). \tag{6}$$

3 The classical Schwarz algorithm on the sphere

We decompose the sphere into two overlapping domains as shown in Fig. 1 on the left. The Schwarz method for two subdomains and model problem (1) is then given by

$$\begin{aligned} \mathcal{L}u_1^n &= f, \text{ in } \Omega_1, \quad u_1^n(b,\theta) = u_2^{n-1}(b,\theta), \\ \mathcal{L}u_2^n &= f, \text{ in } \Omega_2, \quad u_2^n(a,\theta) = u_1^{n-1}(a,\theta), \end{aligned} \tag{7}$$

and we require the iterates to be bounded at the poles of the sphere. By linearity it suffices to consider only the case $f = 0$ and analyze convergence to the zero solution. Taking a Fourier series expansion of the Schwarz algorithm (7), and using the condition on the iterates at the poles, we can express both solutions using the transmission conditions as follows

$$\hat{u}_1^n(\phi,m) = \hat{u}_2^{n-1}(b,m)\frac{P_\nu^m(\cos\phi)}{P_\nu^m(\cos b)}, \;\; \hat{u}_2^n(\phi,m) = \hat{u}_1^{n-1}(a,m)\frac{P_\nu^m(-\cos\phi)}{P_\nu^m(-\cos a)}. \tag{8}$$

Evaluating the second equation at $\phi = b$ for iteration index $n-1$ and inserting it into the first equation, evaluating this latter at $\phi = a$, we get over a double step the relation

$$\hat{u}_1^n(a,m) = \frac{P_\nu^m(-\cos b)P_\nu^m(\cos a)}{P_\nu^m(-\cos a)P_\nu^m(\cos b)}\hat{u}_1^{n-2}(a,m). \tag{9}$$

Therefore, for each m, the convergence factor $\rho(m,\eta,a,b)$ of the classical Schwarz algorithm is given by

$$\rho_{cla} = \rho_{cla}(m,\eta,a,b) := \frac{P_\nu^m(-\cos b)P_\nu^m(\cos a)}{P_\nu^m(-\cos a)P_\nu^m(\cos b)}. \tag{10}$$

A similar result also holds for the second subdomain and we find by induction

$$\hat{u}_1^{2n}(a,m) = \rho_{cla}^n \hat{u}_1^0(a,m), \qquad \hat{u}_2^{2n}(b,m) = \rho_{cla}^n \hat{u}_2^0(b,m). \tag{11}$$

Because of Remark 1, the fractions are less than one and this process is a contraction and hence convergent. We have proved the following

Proposition 1. *For each m, the Schwarz iteration on the sphere partitioned along two colatitudes $a < b$ converges linearly with the convergence factor*

$$\rho_{cla} = \rho_{cla}(m,\eta,a,b) := \frac{P_\nu^m(-\cos b)P_\nu^m(\cos a)}{P_\nu^m(-\cos a)P_\nu^m(\cos b)} < 1.$$

The convergence factor depends on the problem parameters η, the size of the overlap $L = b-a$ and on the frequency parameter m. Fig. 2 on the left, shows the dependence of the convergence factor on the frequency m for an overlap $L = b - a = \dfrac{1}{100}$ and $\eta = 2$. This shows that for small values of m the rate of convergence is very poor, but the Schwarz algorithm can damp high frequencies very effectively.

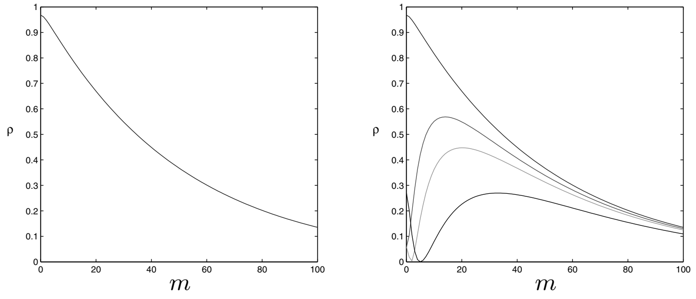


Fig. 2. Left: Behavior of the convergence factor ρ_{cla}. **Right**: Comparison between ρ_{cla} (top curve), ρ_{T0} (2^{nd} curve), ρ_{T2} (3^{th} curve) and ρ_{O0} (bottom curve). In both plots $a = \pi - L/2$ and the overlap is $L = b - a = \dfrac{1}{100}$ and $\eta = 1$.

4 The optimal Schwarz algorithm

Following the approach in [2], we now introduce a modified algorithm by imposing new transmission conditions,

$$\begin{array}{ll}\mathcal{L}(u_1^n) = f, \text{ in } \Omega_1, & (S_1 + \partial_\phi)(u_1^n)(b,\theta) = (S_1 + \partial_\phi)(u_2^{n-1})(b,\theta), \\ \mathcal{L}(u_2^n) = f, \text{ in } \Omega_2, & (S_2 + \partial_\phi)(u_2^n)(a,\theta) = (S_2 + \partial_\phi)(u_1^{n-1})(a,\theta),\end{array} \tag{12}$$

where S_j, $j = 1, 2$, are operators along the interface in the θ direction. As for the classical Schwarz method, it suffices by linearity to consider the homogeneous problem only, $f = 0$, and to analyze convergence to the zero solution. Taking a Fourier series expansion of the new algorithm (12) in the θ direction, we obtain

$$\begin{aligned}(\sigma_1(m) + \partial_\phi)(\hat{u}_1^n)(b,m) &= (\sigma_1(m) + \partial_\phi)(\hat{u}_2^{n-1})(b,m), \\ (\sigma_2(m) + \partial_\phi)(\hat{u}_2^n)(a,m) &= (\sigma_2(m) + \partial_\phi)(\hat{u}_1^{n-1})(a,m),\end{aligned} \tag{13}$$

where σ_j, $j = 1, 2$, denotes the symbol of the operators S_j, $j = 1, 2$, respectively. To simplify the notation, we introduce the function

$$q_{\nu,m}(x) = \frac{P_\nu^{m+1}(\cos x)}{P_\nu^m(\cos x)}.$$

As in the case of the classical Schwarz method, we have to choose $P_\nu^m(\cos\phi)$ as solution in the first subdomain and $P_\nu^m(-\cos\phi)$ as solution in the second subdomain. Using the transmission conditions and the definition of the derivative of the Legendre function in (6), we find the subdomain solutions in Fourier space to be

$$\begin{aligned}\hat{u}_1^n(\phi,m) &= \frac{\sigma_1(m) + m\cot b - q_{\nu,m}(\pi - b)}{\sigma_1(m) + m\cot b + q_{\nu,m}(b)} \frac{P_\nu^m(\cos\phi)}{P_\nu^m(\cos b)} \hat{u}_2^{n-1}(b,m), \\ \hat{u}_2^n(\phi,m) &= \frac{\sigma_2(m) + m\cot a + q_{\nu,m}(a)}{\sigma_2(m) + m\cot a - q_{\nu,m}(\pi - a)} \frac{P_\nu^m(-\cos\phi)}{P_\nu^m(-\cos a)} \hat{u}_1^{n-1}(a,m).\end{aligned} \tag{14}$$

Evaluating the second equation at $\phi = b$ for iteration index $n - 1$ and inserting it into the first equation, we get after evaluation at $\phi = a$,

$$\hat{u}_1^n(a,m) = \rho_{opt}(m,a,b,\eta,\sigma_1,\sigma_2)\hat{u}_1^{n-2}(a,m), \tag{15}$$

where the new convergence factor ρ_{opt} is given by

$$\rho_{opt} := \frac{\sigma_1(m) + m\cot b - q_{\nu,m}(\pi - b)}{\sigma_1(m) + m\cot b + q_{\nu,m}(b)} \frac{\sigma_2(m) + m\cot a + q_{\nu,m}(a)}{\sigma_2(m) + m\cot a - q_{\nu,m}(\pi - a)} \rho_{cla}. \tag{16}$$

As in the classical case, we can prove the following

Proposition 2. *The optimal Schwarz algorithm (12) on the sphere partitioned along two colatitudes $a < b$ converges in two iterations provided that σ_1 and σ_2 satisfy*

$$\sigma_1(m) = -m\cot b + q_{\nu,m}(\pi - b) \quad \textit{and} \quad \sigma_2(m) = -m\cot a - q_{\nu,m}(a). \tag{17}$$

This is an optimal result, since convergence in less than two iterations is impossible, due to the need to exchange information between the subdomains. In practice, one needs to inverse transform the transmission conditions involving $\sigma_1(m)$ and $\sigma_2(m)$

from Fourier space into physical space to obtain the transmission operators S_1 and S_2, and hence we need

$$S_1(u_1^n) = \mathcal{F}_m^{-1}(\sigma_1(\hat{u}_1^n)), \qquad S_2(u_2^n) = \mathcal{F}_m^{-1}(\sigma_2(\hat{u}_2^n)).$$

Due to the fact that the σ_j contain associated Legendre functions, the operators S_j are non-local. To have local operators, we need to approximate the symbols σ_j with polynomials in im. Inspired by the results for elliptic problems in two-dimensional Cartesian space, we introduce the following ansatz

$$q_{\nu,m}(\phi) \approx \frac{\sin(\phi)\sqrt{\eta+m^2}}{1+\cos(\phi)}. \tag{18}$$

Based on this ansatz we can expand the symbols $\sigma_j(m)$ in (17) in a Taylor series,

$$\begin{aligned}\sigma_1(m) &= \frac{\sin(b)\sqrt{\eta}}{-\cos(b)+1} + \frac{\sin(b)m^2}{2(-\cos(b)+1)\sqrt{\eta}} + \mathcal{O}(m^4),\\ \sigma_2(m) &= -\frac{\sin(a)\sqrt{\eta}}{\cos(a)+1} - \frac{\sin(a)m^2}{2(\cos(a)+1)\sqrt{\eta}} + \mathcal{O}(m^4).\end{aligned}$$

A zeroth order Taylor approximation $T0$ is obtained by using only the first terms in the Taylor expansion of σ_j, while a second order approximation $T2$ is obtained by using both terms from the expansion. In Fig. 2 on the right, we compare the convergence factor ρ_{cla} of the classical Schwarz method with the convergence factor ρ_{T0} of the zeroth order Taylor method and the convergence factor ρ_{T2} of the second order Taylor method. Numerically, we find the optimized Robin conditions, namely $\sigma_1 \approx -5.3189$ and $\sigma_2 \approx 5.3189$, and we compare the corresponding convergence factor ρ_{O0} to the other methods.

5 Numerical experiments

We perform two sets of numerical experiments, both with $\eta = 1$. In the first set we consider our model problem on the sphere using a longitudinal co-latitudinal grid, where we adopt a decomposition with two overlapping subdomains as shown in Fig. 1 on the left. In this case, we combine a spectral method in the θ-direction with a finite difference method in the ϕ-direction. We use a discretization with 6000 points in ϕ, including the poles, and spectral modes from -10 to 10. The decomposition is done in the middle and the overlap is chosen to be $[0.49\pi, 0.51\pi]$, see Fig. 3 on the left, where the curves with (circle) and without (square) overlap of optimal Schwarz are on top of each other. In the second experiment, we solve the model problem on the Yin-Yang grid. This is a composite grid, which covers the surface of the sphere with two identical rectangles that partially overlap on their borders. Each grid is an equatorial sector having a different polar axis but uniform discretization, see Fig. 1 on the right. The Ying-Yang grid system is free from the problem of singularity at the poles, in contrast to the ordinary spherical coordinate system. In Fig. 3 on the right we show some screenshots of the exact and numerical solutions for the Yin-Yang grid using optimized Robin conditions with $\sigma_1 = -1.4$ and $\sigma_2 = 1.4$. In Table 1 we compare the classical Schwarz method to the optimized methods in the Yin-Yang grid system.

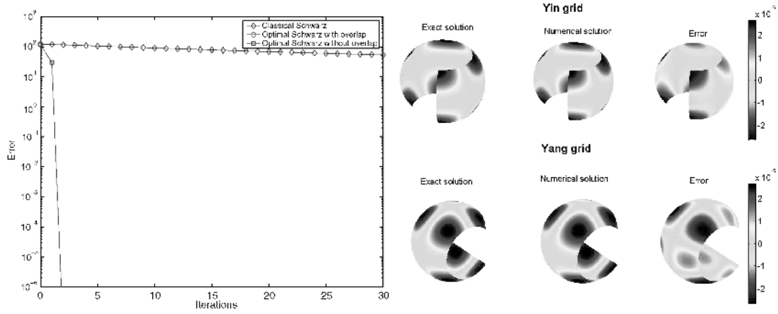


Fig. 3. Left: Convergence behavior for the methods analyzed for the two subdomain case. **Right:** Screenshots of solutions and the error for the Yin-Yang grid system. In both plots $\eta = 1$.

	Classical Schwarz		Taylor 0 method		Taylor 2 method		Optimized 0 method	
h	$L = 1/50$	$L = h$	$L = 1/50$	$L = h$	$L = 1/50$	$L = h$	$L = 1/50$	$L = h$
1/50	184	184	22	22	16	16	12	12
1/100	184	284	22	27	16	19	12	16
1/150	183	389	21	31	15	21	11	19
1/200	184	497	22	36	16	24	12	22

Table 1. Number of iterations of the classical Schwarz method compared to the optimized Schwarz methods for the Yin-Yang grid system with $\eta = 1$.

Conclusion

In this work, we show that numerical algorithms already validated for a global latitude/longitude grid can be implemented, with minor changes, for the Yin-Yang grid system. In the future we will implement optimized second order interface conditions in order to improve the convergence of the elliptic solver and we will also use Krylov methods to accelerate the algorithms.

Acknowledgement. We acknowledge the support of the Canadian Foundation for Climate and Atmospheric Sciences (CFCAS) trough a grant to the QPF network. This research was also partially supported by the Office of Science (BER), U.S. Department of Energy, Grant No. DE-FG02-01ER63199.

References

1. P. Charton, F. Nataf, and F. Rogier, *Méthode de décomposition de domaine pour l'équation d'advection-diffusion*, C. R. Acad. Sci., 313 (1991), pp. 623–626.
2. M. J. Gander, L. Halpern, and F. Nataf, *Optimized Schwarz methods*, in Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan, T. Chan, T. Kako, H. Kawarada, and O. Pironneau, eds., Bergen, 2001, Domain Decomposition Press, pp. 15–28.

3. I. S. Gradshteyn and I. M. Ryzhik, *Tables of Series, Products and Integrals*, Verlag Harri Deutsch, Thun, 1981.
4. T. Hagstrom, R. P. Tewarson, and A. Jazcilevich, *Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems*, Appl. Math. Lett., 1 (1988).
5. C. Japhet, *Optimized Krylov-Ventcell method. Application to convection-diffusion problems*, in Proceedings of the 9th international conference on domain decomposition methods, P. E. Bjørstad, M. S. Espedal, and D. E. Keyes, eds., ddm.org, 1998, pp. 382–389.
6. A. Kageyama and T. Sato, *The `Yin-Yang grid': An overset grid in spherical geometry*, Geochem. Geophys. Geosyst., 5 (2004).
7. F. Nataf and F. Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, M^3AS, 5 (1995), pp. 67–93.

An Optimized Schwarz Algorithm for the Compressible Euler Equations

Victorita Dolean[1] and Frédéric Nataf[2]

[1] UMR 6621 CNRS, Université de Nice Sophia Antipolis, 06103 Nice Cedex 2, France. dolean@math.unice.fr

[2] CMAP, UMR 7641 CNRS, École Polytechnique, 91128 Palaiseau Cedex, France. nataf@cmap.polytechnique.fr

Summary. In this work, we design new interface transmission conditions for a domain decomposition Schwarz algorithm for the Euler equations in two dimensions. These new interface conditions are designed to improve the convergence properties of the Schwarz algorithm. These conditions depend on a few parameters and they generalize the classical ones. Numerical results illustrate the effectiveness of the new interface conditions.

1 Introduction

In a previous paper [4] we formulated and studied by means of Fourier analysis the convergence of a Schwarz algorithm (interface iteration which relies on the successive solving of the local decomposed problems and the transmission of the result at the interface) involving transmission conditions that are derived naturally from a weak formulation of the underlying boundary value problem. Various studies exist to deal with Schwarz algorithms applied to the scalar problems but to our knowledge, little is known about complex systems. For systems we can mention some classical works by Quarteroni and al. [5] [6] Bjorhus [1] and Cai et al.[2]. The work most related to ours belongs to Clerc [3] and it describes the principle of building very simple interface conditions for a general hyperbolic system which we will apply and extend to Euler system. In this work, we formulate and analyze the convergence of the Schwarz algorithm with new interface conditions inspired by [3], which depend on two parameters whose values are determined by minimizing the norm of the convergence rate. The paper is organized as follows. In section 2, we first formulate the Schwarz algorithm for a general linear hyperbolic system of PDEs with general interface conditions designed to have a well-posed problem. In section 3, we estimate the convergence rate at the discrete level. We will find the optimal parameters of the interface conditions at the discrete level. In section 4, we use the new optimal interface conditions in Euler computations which illustrate the improvement over the classical interface conditions (first described in [6]).

2 A Schwarz algorithm with general interface conditions

2.1 A well-posed boundary value problem

If we consider a general non-linear system of conservation laws under the hypothesis that its solution is regular, we can also use a non-conservative (or quasi-linear) equivalent form. Assume that we first proceed to an integration in time using a backward Euler implicit scheme involving a linearization of the flux functions and that we eventually symmetrize it. (We know that when the system admits an entropy it can be symmetrized by multiplying it by the hessian matrix of this entropy). This results in the linearized system:

$$\mathcal{L}(W) \equiv \frac{\mathrm{Id}}{\Delta t} W + \sum_{i=1}^{d} A_i \frac{\partial W}{\partial x_i} = f \tag{1}$$

In the following, we will define the boundary conditions that have to be imposed when solving the problem on a domain $\Omega \subset \mathbb{R}^d$. We denote by $A_{\mathbf{n}} = \sum_{i=1}^{d} A_i n_i$, the linear combination of the jacobian matrices by the components of the outward normal vector of $\partial\Omega$, the boundary of the domain. This matrix is real, symmetric and can be diagonalized $A_{\mathbf{n}} = T\Lambda_{\mathbf{n}}T^{-1}$, $\Lambda_{\mathbf{n}} = diag(\lambda_i)$. It can also be split in negative $(A_{\mathbf{n}}^-)$ and positive $(A_{\mathbf{n}}^+)$ parts using this diagonalization. This corresponds to a decomposition with local characteristic variables. A more general splitting in negative(positive) definite parts, $A_{\mathbf{n}}^{neg}$ and $A_{\mathbf{n}}^{pos}$ of $A_{\mathbf{n}}$ can be done such that these matrices satisfy the following properties:

$$\begin{cases} A_{\mathbf{n}} & = A_{\mathbf{n}}^{neg} + A_{\mathbf{n}}^{pos} \\ rank(A_{\mathbf{n}}^{neg,pos}) & = rank(A_{\mathbf{n}}^{\pm}) \\ A_{-\mathbf{n}}^{pos} & = -A_{\mathbf{n}}^{neg} \end{cases} \tag{2}$$

In the scalar case the only possible choice is $A_{\mathbf{n}}^{neg} = A_{\mathbf{n}}^-$. Using the previous formalism, we can define the following boundary condition:

$$A_{\mathbf{n}}^{neg} W = A_{\mathbf{n}}^{neg} g, \text{ on } \partial\Omega \tag{3}$$

Within this framework, we have a result of well-posedness of the boundary value problem associated to the system (1) with the boundary conditions (3) that can be found in [3]. As the boundary value problem is well-posed, the decomposition (2) enables the design of a domain decomposition method.

2.2 Schwarz algorithm with general interface conditions

We consider a decomposition of the domain Ω into N overlapping or non-overlapping subdomains $\bar{\Omega} = \bigcup_{i=1}^{N} \bar{\Omega}_i$. We denote by $\mathbf{n}_{ij}$ the outward normal to the interface Γ_{ij} bewteen Ω_i and a neighboring subdomain Ω_j. Let $W_i^{(0)}$ denote the initial appoximation of the solution in subdomain Ω_i. A general formulation of a Schwarz algorithm for computing $(W_i^{p+1})_{1\le i\le N}$ from $(W_i^p)_{1\le i\le N}$ (where p defines the iteration of the Schwarz algorithm) reads :

$$\begin{cases} \mathcal{L}W_i^{p+1} &= f \qquad \text{in}\Omega_i \\ A_{\mathbf{n}_{ij}}^{neg} W_i^{p+1} &= A_{\mathbf{n}_{ij}}^{neg} W_j^p \text{ on } \Gamma_{ij} = \partial\Omega_i \cap \Omega_j \\ A_{\mathbf{n}_{ij}}^{neg} W_i^{p+1} &= A_{\mathbf{n}_{ij}}^{neg} g \quad \text{ on } \partial\Omega \cap \partial\Omega_i \end{cases} \tag{4}$$

where $A_{\mathbf{n}_{ij}}^{neg}$ and $A_{\mathbf{n}_{ij}}^{pos}$ satisfy (2). We have a convergence result of this algorithm in the non-overlapping case, due to ([3]). The convergence rate of the algorithm defined by (4) depends of the choice of the decomposition of $A_{\mathbf{n}_{ij}}$ into $A_{\mathbf{n}_{ij}}^{neg}$ and $A_{\mathbf{n}_{ij}}^{pos}$ satisfying (2). In order to choose the right decomposition, we need to relate this choice to the convergence rate of (4).

2.3 Convergence rate of the algorithm with general interface conditions

We consider a two-subdomain non-overlapping or overlapping decomposition of the domain $\Omega = \mathbb{R}^d$, $\Omega_1 =]-\infty, \gamma[\times\mathbb{R}^{d-1}$ and $\Omega_2 =]\beta, \infty[\times\mathbb{R}^{d-1}$ with $\beta \leq \gamma$ and study the convergence of the Schwarz algorithm in the subsonic case. A Fourier analysis applied to the linearized equations allows us to derive the convergence rate of the "ξ"-th Fourier component of the error as described in detail in [4]. After having defined in a general frame the well-posedness of the boundary value problem associated to a general equation and the convergence of the Schwarz algorithm applied to this class of problems, we will concentrate on the conservative Euler equations in two-dimensions:

$$\frac{\partial W}{\partial t} + \nabla.\mathbf{F}(W) = 0,\ W = (\rho,\ \rho\mathbf{V},\ E)^T. \tag{5}$$

In the above expressions, ρ is the density, $\mathbf{V} = (u,\ v)^T$ is the velocity vector, E is the total energy per unit of volume and p is the pressure. In equation (5), $W = W(\mathbf{x}, \mathbf{t})$ is the vector of conservative variables, $\mathbf{x}$ and t, respectively denote the space and time variables and $\mathbf{F}(W) = (F_1(W), F_2(W))^T$ is the conservative flux vector whose components are given by

$$F_1(W) = \left(\rho u, \rho u^2 + p, \rho uv, u(E+p)\right)^T,\ F_2(W) = \left(\rho v, \rho uv, \rho v^2 + p, v(E+p)\right)^T.$$

The pressure is determined by the other variables using the state equation for a perfect gas $p = (\gamma_s - 1)(E - \frac{1}{2}\rho \parallel \mathbf{V} \parallel^2)$ where γ_s is the ratio of the specific heats ($\gamma_s = 1.4$ for air).

2.4 A new type of interface conditions

We will now apply the method described previously to the computation of the convergence rate of the Schwarz algorithm applied to the two-dimensional subsonic Euler equations. In the supersonic case there is only one decomposition satisfying (2), namely $\mathcal{A}^{pos} = A_{\mathbf{n}}$ and $\mathcal{A}^{neg} = 0$ and the convergence follows in 2 steps. Therefore the only case of interest is the subsonic one.

The starting point of our analysis is given by the linearized form of the Euler equations (5) which are of the form (1) to which we applied a change of variable $\tilde{W} = T^{-1}W$ based on the eigenvector factorization of $A_1 = T\tilde{A}_1T^{-1}$. We denote by $M_n = \frac{u}{c}$, $M_t = \frac{v}{c}$ respectively the normal and the tangential Mach number. Before estimating the convergence rate we will derive the general transmission conditions at the interface by splitting the matrix A_1 into a positive and negative part. We have the following general result concerning this decomposition:

Lemma 1. *Let $\lambda_1 = M_n - 1$, $\lambda_2 = M_n + 1$, $\lambda_3 = \lambda_4 = M_n$. Suppose we deal with a subsonic flow: $0 < u < c$ so that $\lambda_1 < 0$, $\lambda_{2,3,4} > 0$. Any decomposition of $A_1 = A_{\mathbf{n}}$, $\mathbf{n} = (1, 0)$ which satisfies (2) has to be of the form:*

$$\begin{aligned}\mathcal{A}^{neg} &= \frac{1}{a_1}\mathbf{u}\cdot\mathbf{u}^t,\ \mathbf{u} = (a_1, a_2, a_3, a_4)^t\\ \mathcal{A}^{pos} &= A_{\mathbf{n}} - \mathcal{A}^{neg}.\end{aligned}$$

where $(a_1, a_2, a_3, a_4) \in \mathbb{R}^4$ satisfies $a_1 \le \lambda_1 < 0$ and $\dfrac{a_1}{\lambda_1} + \dfrac{a_2^2}{a_1\lambda_2} + \dfrac{a_3^2}{a_1\lambda_3} + \dfrac{a_4^2}{a_1\lambda_4} = 1.$

We will proceed now to estimating the convergence rate using some results from [4]. Following the technique described here we estimate the convergence rate in the Fourier space in the non-overlapping case. We use the non-dimensional wave-number $\bar{\xi} = c\Delta t\xi$, and get for the general interface conditions the following:

$$\begin{cases} \rho_{2,novr}^2(\xi) = \left|1 - \dfrac{4M_n(1-M_n)(1+M_n)R(\xi)a_1^2(a + M_nR(\xi))}{D_1D_2}\right| \\ D_1 = R(\xi)[a_1(1+M_n) - a_2(1-M_n)] + a[a_1(1+M_n) \\ \qquad +a_2(1-M_n)] - i\sqrt{2}a_3\xi(1-M_n^2) \\ D_2 = M_na_1[R(\xi)[a_1(1+M_n) - a_2(1-M_n)] + a[a_1(1+M_n) + a_2(1-M_n)]] \\ \qquad +a_3(1-M_n^2)[a_3(R+a) - iM_na_1\xi\sqrt{2}] \end{cases} \tag{6}$$

In order to simplify our optimization problem, we will take $a_3 = 0$. We can thus reduce the number of parameters to two, a_1 and a_2, since we can see from the lemma that a_4 can be expressed as a function of a_1, a_2 and a_3. At the same time, for purpose of optimization only, we introduce the parameters: $b_1 = -a_1/(1 - M_n)$ and $b_2 = a_2/(1 + M_n)$ which provide a simpler form of the convergence rate. Nevertheless, solving this problem is quite a tedious task even in the non-overlapping case, where we can obtain analytical expression of the parameters only for some values of the Mach number. At the same time, we have to analyze the convergence of the overlapping algorithm. Indeed, standard discretizations of the interface conditions correspond to overlapping decompositions with an overlap of size $\delta = h$, h being the mesh size, as seen in [4]. By applying the Fourier transform technique to the overlapping case we have the following expression of the convergence rate:

$$\begin{cases} \rho_{2,ovr}^2 = \left|Ae^{-(\lambda_2(k)-\lambda_1(k))\bar{\delta}} + (B+C)e^{-(\lambda_3(k)-\lambda_1(k))\bar{\delta}}\right| \\ A \quad = \dfrac{a + M_nR(\xi)}{a - M_nR(\xi)}\cdot\left(\dfrac{b_1(R(\xi)-a) + b_2(R(\xi)+a)}{b_1(R(\xi)+a) + b_2(R(\xi)-a)}\right)^2 \\ B \quad = -\dfrac{2M_n(b_1(1-M_n) + b_2(1+M_n))R(\xi)(R(\xi)-a)(R(\xi)+a)}{(1-M_n^2)(a - M_nR(\xi))(b_1(R(\xi)+a) + b_2(R(\xi)-a))^2} \\ C \quad = \dfrac{4((1-M_n)(b_1^2 - b_1) - b_2^2(Mn+1))(a + M_nR(\xi))}{(1-M_n^2)(b_1(R(\xi)+a) + b_2(R(\xi)-a))^2} \end{cases} \tag{7}$$

where $\bar{\delta} = \dfrac{\delta}{c\Delta t}$ denotes the non-dimensional overlap between the subdomains. Analytic optimization with respect to b_1 and b_2 seems out of reach. We will have to use numerical procedures of optimization. In order to get closer to the numerical simulations we will estimate the convergence rate for the discretized equations with general transmission conditions, both in the non-overlapping and the overlapping case and then optimize numerically this quantity in order to get the best parameters for the convergence.

3 Optimized interface conditions

In this section we study the convergence of the Schwarz algorithm with general interface conditions applied to the discrete Euler equations as described in [4] for the classical transmission conditions. This BVP is discretized using a finite volume scheme where the flux at the interface of the finite volume cells is computed using a Roe [7] type solver. Afterwards, we formulate a Schwarz algorithm whose convergence rate is estimated in the Fourier space in a discrete context. Optimizing the convergence rate with respect to the two parameters is already a very difficult task on the continuous level in the non-overlapping case, we could not carry on such a process and obtain analytical results at the discrete level in the overlapping case (which is our case of interest). Therefore, we will get the theoretical optimized parameters at the discrete level by means of a numerical algorithm, by calculating the following

$$\begin{aligned} &\rho(b_1,b_2) = \max_{k\in\mathcal{D}_h} \rho_2^2(k,\Delta x, M_n, M_t, b_1, b_2) \\ &\min_{(b_1,b_2)\in\mathcal{I}_h} \rho(b_1,b_2) \end{aligned} \tag{8}$$

Here $\mathcal{D}_h$ is a uniform partition of the interval $[0,\pi/\Delta x]$ and $\mathcal{I}_h \subset \mathcal{I}$ a discretization by means of a uniform grid of a subset of the domain of the admissible values of the parameters. This kind of calculations are done once and for all for a given pair (M_n, M_t) before the beginning of the Schwarz iterations. An example of such a result is given in figure 1 for Mach number $M_n = 0.2$. The computed parameters from the relation (8) will be further referred to with a superscript *th*. The theoretical estimates are compared afterwards with the numerical ones obtained by running the

Table 1. Overlapping Schwarz algorithm.

M_n	b_1^{th}	b_2^{th}	b_1^{num}	b_2^{num}
0.1	1.6	-0.8	1.6	-0.9
0.2	1.3	-0.5	1.4	-0.6
0.3	1.25	-0.3	1.25	-0.45
0.4	1.08	-0.15	1.08	-0.28
0.5	1.03	-0.08	1.02	-0.23
0.6	1.0	0.0	1.0	0.0
0.7	1.02	0.06	1.01	0.04
0.8	1.03	0.08	1.02	0.06
0.9	1.06	0.08	1.04	0.06

Schwarz algorithm with different pairs of parameters which lie in a an interval for which the algorithm is convergent. We are thus able to estimate the optimal values for b_1 and b_2 from these numerical computations. These values will be referred to by a superscript *num*.

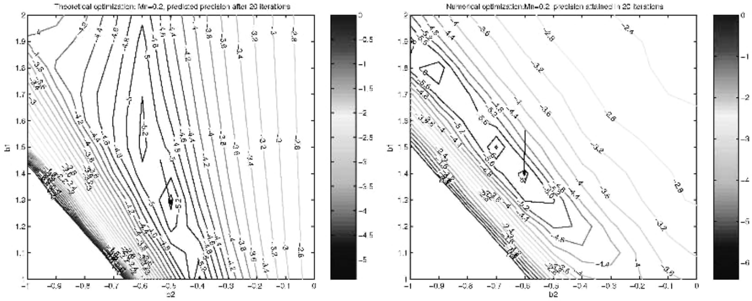


Fig. 1. Isovalues of the predicted (theoretical via formula (8)) and numerical(FV code) reduction factor of the error after 20 iterations.

4 Implementation and numerical results

We present here a set of results of numerical experiments that are concerned with the evaluation of the influence of the interface conditions on the convergence of the non-overlapping Schwarz algorithm of the form. The computational domain is given by the rectangle $[0,1]\times[0,1]$. The numerical study is limited to the solution of the linear system resulting from the first implicit time step using a Courant number CFL=100. In all these calculations, we consider a model problem: a flow normal to the interface (i.e. $M_t = 0$). In figures 1 we see an example of a theoretical and numerical estimation of the reduction factor of the error. We show here the level curves which represent the log of the precision after 20 iterations for different values of the parameters (b_1, b_2), the minimum being attained in this case for $b_1^{th} = 1.3$ and $b_2^{th} = -0.5$, $b_1^{num} = 1.4$ and $b_2^{num} = -0.6$. We see that we have good theoretical estimates of these parameters and we can therefore use them in the interface conditions of the Schwarz algorithm. Table 2 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 with the optimal parameters b_1^{num} and b_1^{num}. Here we denoted by IT_0^{num} and IT_{op}^{num} the observed (numerical) iteration number for classical and optimized interface conditions in order to achieve convergence with a threshlod $\varepsilon = 10^{-6}$. The same results are presented in the second picture of figure 2. In the first picture of figure 2 we compare the theoretically estimated iteration number in the classical and optimized case. Comparing the two pictures of figure 2 we see that the theoretical prediction are very close to the numerical tests. The conclusion of these numerical tests is, on one hand, that the theoretical prediction is very close to the numerical

Table 2. Overlapping Schwarz algorithm. Classical vs. optimized counts for different values of M_n.

M_n	IT_0^{num}	IT_{op}^{num}	M_n	IT_0^{num}	IT_{op}^{num}
0.1	48	19	0.5	22	18
0.2	41	20	0.7	20	16
0.3	32	20	0.8	22	15
0.4	26	19	0.9	18	12

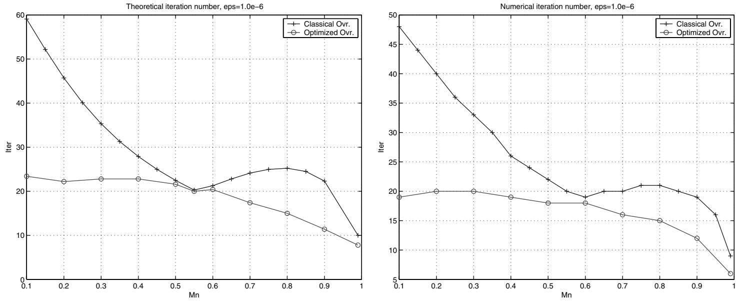


Fig. 2. Theoretical and numerical iteration number: classical vs. optimized conditions.

results i.e. by a numerical optimization (8) we can get a very good estimate of optimal parameters (b_1, b_2)). In addition, the gain, in the number of iterations, provided by the optimized interface conditions, is very promising for low Mach numbers, where the classical algorithm does not give optimal results. For larger Mach numbers, for instance, those close to 1, the classical algorithm already has a very good behavior so the optimization is less useful. At the same time we have studied here the zero order and therefore very simple transmission conditions. The use of higher order conditions could be further studied to obtain even better convergence results.

References

1. M. Bjørhus, *Semi-discrete subdomain iteration for hyperbolic systems*, Tech. Rep. 4, Norwegian University of Science and Technology, Norway, 1995.
2. X.-C. Cai, C. Farhat, and M. Sarkis, *A minimum overlap restricted additive Schwarz preconditioner and applications to 3D flow simulations*, Contemporary Mathematics, 218 (1998), pp. 479–485.

3. S. Clerc, *Non-overlapping Schwarz method for systems of first order equations*, Cont. Math., 218 (1998), pp. 408–416.
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. Num. Math., 49 (2004), pp. 153–186.
5. A. Quarteroni, *Domain decomposition methods for systems of conservation laws: spectral collocation approximation*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 1029–1052.
6. A. Quarteroni and L. Stolcis, *Homogeneous and heterogeneous domain decomposition methods for compressible flow at high Reynolds numbers*, Tech. Rep. 33, CRS4, 1996.
7. P. L. Roe, *Approximate Riemann solvers, parameter vectors and difference schemes*, J. Comput. Phys., 43 (1981), pp. 357–372.

Optimized Schwarz Methods with Robin Conditions for the Advection-Diffusion Equation

Olivier Dubois

McGill University, Department of Mathematics & Statistics, 805 Sherbrooke W.
Montréal, Québec, H3A 2K6, Canada. dubois@math.mcgill.ca

Summary. We study optimized Schwarz methods for the stationary advection-diffusion equation in two dimensions. We look at simple Robin transmission conditions, with one free parameter. In the nonoverlapping case, we solve exactly the associated min-max problem to get a direct formula for the optimized parameter. In the overlapping situation, we solve only an approximate min-max problem. The asymptotic performance of the resulting methods, for small mesh sizes, is derived. Numerical experiments illustrate the improved convergence compared to other Robin conditions.

1 Introduction

The classical Schwarz method, first devised as a tool to prove existence and uniqueness results, converges only when there is overlap between subdomains, and it converges very slowly for small overlap sizes. It was first proposed by Lions [8] to change the Dirichlet transmission conditions in the algorithm to other types of conditions, in order to obtain a convergent nonoverlapping variant. More recently, optimized Schwarz methods were introduced by Japhet [7]; using a Fourier analysis on a model problem, the convergence factor is uniformly minimized over a class of transmission conditions. The work of Japhet was originally carried out for the advection-diffusion equation in the plane, without overlap, and using second order transmission conditions. Optimized Schwarz methods are now well-studied for symmetric partial differential equations, for example for the Laplace and modified Helmholtz equations (see [4, 3] and references therein) and the Helmholtz equation (see [2, 5]).

The purpose of this work is to study optimized Robin transmission conditions for the advection-diffusion equation, both in the case of nonoverlapping and overlapping domain decompositions. We start, in Section 2, by introducing the model problem in the plane. In Section 3, we present a general Schwarz iteration and its convergence factor, from which optimal transmission conditions can be found. We also briefly describe the Taylor polynomial approximations of the optimal symbols, a way to obtain local transmission operators. In Section 4 and 5, we present optimized Robin conditions, in the nonoverlapping and overlapping cases respectively. We illustrate our results in Section 6 with numerical experiments.

2 The Model Problem

The derivation and analysis of optimized Schwarz methods is done for a model problem. Here we consider the advection-diffusion equation on the plane with constant coefficients

$$\begin{cases} \mathcal{L}u := -\nu\Delta u + \mathbf{a}\cdot\nabla u + cu = f \text{ in } \mathbb{R}^2, \\ \qquad\qquad\qquad u \text{ is bounded at infinity,} \end{cases}$$

where $\nu, c > 0$ and $\mathbf{a} = (a, b)$. For the convergence analysis of the algorithms presented subsequently, it will be sufficient, by linearity, to look at the homogeneous problem only, $f \equiv 0$. We decompose the plane into two subdomains Ω_1 and Ω_2 with an overlap of width L

$$\Omega_1 := \left(-\infty, \frac{L}{2}\right) \times \mathbb{R}, \quad \Omega_2 := \left(-\frac{L}{2}, \infty\right) \times \mathbb{R},$$

and we denote by u_i^n the approximate solution in subdomain Ω_i, at iteration n.

Our analysis is based on the Fourier transform in the y variable

$$\mathcal{F}_y[u(x,y)] = \hat{u}(x,k) := \frac{1}{\sqrt{2\pi}}\int_{-\infty}^{\infty} u(x,y)e^{-iyk}dy.$$

In Fourier space, the homogeneous advection-diffusion equation becomes

$$-\nu\frac{\partial^2\hat{u}}{\partial x^2} + a\frac{\partial\hat{u}}{\partial x} + (\nu k^2 - ibk + c)\hat{u} = 0.$$

This is a linear second order ODE in x that can be solved analytically. The roots to the corresponding characteristic equation are given by

$$\lambda^{\pm}(k) = \frac{a \pm \sqrt{a^2 + 4\nu c - 4i\nu bk + 4\nu^2k^2}}{2\nu}, \tag{1}$$

where $\mathrm{Re}(\lambda^+) > 0$ and $\mathrm{Re}(\lambda^-) < 0$. The two fundamental solutions are then

$$e^{\lambda^+(k)x}, \quad e^{\lambda^-(k)x}.$$

We introduce the convenient notation

$$z(k) := \sqrt{a^2 + 4\nu c - 4i\nu bk + 4\nu^2k^2}, \tag{2}$$

$$\xi(k) := \mathrm{Re}(z(k)), \quad \eta(k) := \mathrm{Im}(z(k)).$$

3 Optimal Conditions and Taylor Approximations

We first consider a general Schwarz iteration of the form

$$\begin{cases} \qquad\qquad \mathcal{L}u_1^{n+1} = 0 & \text{in } (-\infty, \frac{L}{2}) \times \mathbb{R}, \\ \frac{\partial u_1^{n+1}}{\partial x} - \mathcal{S}_1(u_1^{n+1}) = \frac{\partial u_2^n}{\partial x} - \mathcal{S}_1(u_2^n) & \text{at } x = \frac{L}{2}, \end{cases} \tag{3}$$

$$\begin{cases} \mathcal{L}u_2^{n+1} = 0 & \text{in } (-\frac{L}{2}, \infty) \times \mathbb{R}, \\ \frac{\partial u_2^{n+1}}{\partial x} - \mathcal{S}_2(u_2^{n+1}) = \frac{\partial u_1^n}{\partial x} - \mathcal{S}_2(u_1^n) & \text{at } x = -\frac{L}{2}. \end{cases} \tag{4}$$

where $\mathcal{S}_i$ are linear operators acting on the y variable only, with Fourier symbols σ_i

$$\mathcal{F}_y[\mathcal{S}_i(u)] = \sigma_i(k)\hat{u}(x,k).$$

Using the Fourier transform in y, we can solve each subproblem analytically, and find a convergence factor.

Proposition 1. *The convergence factor of the Schwarz iteration* (3)-(4) *in Fourier space is*

$$\rho(k, L, \sigma_1, \sigma_2) := \left| \frac{\hat{u}_1^{n+1}(\frac{L}{2}, k)}{\hat{u}_1^{n-1}(\frac{L}{2}, k)} \right| = \left| \frac{(\lambda^- - \sigma_1)(\lambda^+ - \sigma_2)}{(\lambda^+ - \sigma_1)(\lambda^- - \sigma_2)} e^{-L(\lambda^+ - \lambda^-)} \right|, \tag{5}$$

where $\lambda^\pm(k)$ *are defined by* (1).

By choosing $\sigma_1(k) = \lambda^-(k)$ and $\sigma_2(k) = \lambda^+(k)$, we can make the convergence factor vanish and hence obtain an optimal convergence *in* 2 *iterations only.* This gives optimal operators $\mathcal{S}_i^{opt}$ when transforming back to real space, which turn out to be Dirichlet-to-Neumann maps, see for example [9]. However these operators are nonlocal in y (their Fourier symbols $\lambda^\pm$ are not polynomials in k) and thus not convenient for practical implementation.

One way to find *local* conditions is to take, for σ_i, low order Taylor approximations of the optimal symbols $\lambda^\pm$. For example, zeroth order approximations give

$$\sigma_1 = \frac{a - \sqrt{a^2 + 4\nu c}}{2\nu}, \quad \sigma_2 = \frac{a + \sqrt{a^2 + 4\nu c}}{2\nu}, \tag{6}$$

which lead to a particular choice of Robin conditions. These methods work well only on small frequency components in y (the Taylor approximations are good only for small k). An analysis of these methods can be found in [6, 1].

4 Optimized Robin Conditions Without Overlap

We consider now a class of Robin transmission conditions by choosing

$$\mathcal{S}_1(u) = \frac{a - p}{2\nu} u, \quad \mathcal{S}_2(u) = \frac{a + p}{2\nu} u,$$

where p is a real number. Using the general formula (5), the convergence factor for this choice reduces to

$$\rho_{R1}(k, L, p) := \left| \frac{(p - z(k))^2}{(p + z(k))^2} e^{-\frac{Lz(k)}{\nu}} \right|, \tag{7}$$

where $z(k)$ is defined by (2). The idea of optimized Schwarz methods is, after fixing a class of conditions (Robin in this case), to minimize the convergence factor *uniformly* for all frequency components in a relevant range. This is formulated as a min-max problem. In our situation, a good value for the parameter p is the one solving the optimization problem

$$\min_{p\in\mathbb{R}}\left(\max_{k_{min}\leq k\leq k_{max}}|\rho_{R1}(k,L,p)|\right). \tag{8}$$

In the following results, we use the short-hand notation $\xi_{min}:=\xi(k_{min})$, $\xi_{max}:=\xi(k_{max})$ and similar notations for z_{min} and z_{max}.

Proposition 2 (Optimized Robin parameter, without overlap). *If there is no overlap ($L=0$), the unique minimizer p^* of problem* (8) *is given by*

$$p^*=\begin{cases}|z_{min}| & \textit{if } p_c<|z_{min}|,\\ p_c & \textit{if } |z_{min}|\leq p_c\leq|z_{max}|,\\ |z_{max}| & \textit{if } p_c>|z_{max}|,\end{cases}$$

$$\textit{where } p_c:=\sqrt{\frac{\xi_{min}|z_{max}|^2-\xi_{max}|z_{min}|^2}{\xi_{max}-\xi_{min}}}.$$

For symmetric equations, the optimized Robin parameter is given by an equioscillation property, namely $\rho_{R1}(k_{min},0,p^*)=\rho_{R1}(k_{max},0,p^*)$, see [3]. On the other hand, for the advection-diffusion equation, this characterization does not always hold. Indeed, Proposition 2 shows that this equioscillation happens only in the middle case, when $p^*=p_c$.

Proposition 3 (Optimized Robin asymptotics, without overlap). *For $L=0$ and $k_{max}=\frac{\pi}{h}$, the asymptotic performance for small h of the Schwarz method with optimized Robin transmission conditions is*

$$\max_{k_{min}\leq k\leq\frac{\pi}{h}}|\rho_{R1}(k,0,p^*)|=1-2\sqrt{\frac{2\xi_{min}}{\pi\nu}}h^{\frac{1}{2}}+O(h).$$

Note that the optimized Robin method has better asymptotic performance than the zeroth order Taylor approximation (6), which yields an expansion of the form $1-O(h)$ for small h. The proof of Proposition 2 and 3 can be found in [1].

Remark 1. We can also choose two different constants in the Robin conditions

$$\mathcal{S}_1(u)=\frac{a-p}{2\nu}u,\quad \mathcal{S}_2(u)=\frac{a+q}{2\nu}u,$$

and look for a good pair of parameters (p,q) by solving the min-max problem

$$\min_{p,q\in\mathbb{R}}\left(\max_{k_{min}\leq k\leq k_{max}}\left|\frac{(p-z)(q-z)}{(p+z)(q+z)}e^{-\frac{Lz}{\nu}}\right|\right).$$

This will be referred to as the ***optimized two-sided Robin conditions****. In this paper, when using these conditions, the parameters are computed by solving the min-max problem numerically; there are no complete analytical results yet.*

Fig. 1 shows, on the left, a comparison of the convergence factors for different nonoverlapping Schwarz methods using Robin conditions.

5 Optimized Robin Conditions With Overlap

We now consider the overlapping situation. The convergence factor (7) can be written as

$$|\rho_{R1}(k, L, p)| = \frac{(p - \xi(k))^2 + \eta(k)^2}{(p + \xi(k))^2 + \eta(k)^2} e^{-\frac{L\xi(k)}{\nu}}.$$

Instead of finding the exact solution to the min-max problem, we derive in this section an approximate parameter that works well asymptotically for small h. We observe that η remains bounded: $|\eta(k)| \leq |b|$, $\forall k$. Hence we have the upper bound

$$|\rho_{R1}(k, L, p)| \leq \frac{(p - \xi)^2 + b^2}{(p + \xi)^2 + b^2} e^{-\frac{L\xi}{\nu}} =: Q(\xi, p).$$

Instead of minimizing ρ, for simplicity we solve an approximate min-max problem using the upper bound

$$\min_{p \in \mathbb{R}} \left(\max_{\xi_{min} \leq \xi \leq \xi_{max}} Q(\xi, p) \right). \tag{9}$$

We take $k_{max} = \infty$ in this case to avoid extra complications. We expect that the parameter we obtain from this optimization will be close to the optimized parameter from (8), when $|b|$ and L are small.

Proposition 4 (Approximate Robin parameter, with overlap). *Let $L > 0$ and $k_{max} = \infty$. Define the critical value*

$$\xi_2(p) := \sqrt{\frac{2\nu p - Lb^2 + Lp^2 + 2\sqrt{\nu^2 p^2 - 2\nu L p b^2 - L^2 b^2 p^2}}{L}},$$

and let $p_{min} := \sqrt{\xi_{min}^2 + b^2}$. If $\xi_2(p_{min})$ is complex, or if $\xi_2(p_{min}) < \xi_{min}$, or if

$$Q(\xi_{min}, p_{min}) > Q(\xi_2(p_{min}), p_{min}),$$

then the unique minimizer p^* of problem (9) is $p^* = p_{min}$. Otherwise, the unique minimizer is given by the unique root p^* (greater than p_{min}) of the equation

$$Q(\xi_{min}, p^*) = Q(\xi_2(p^*), p^*).$$

Proposition 5 (Approximate Robin asymptotics, with overlap). *For $L = h$ and $k_{max} = \frac{\pi}{h}$, the asymptotic performance of the optimized Schwarz method, with the Robin parameter p^* obtained through Proposition 4, is given by*

$$\max_{\xi_{min} \leq \xi \leq \xi_{max}} |\rho_{R1}(k, h, p^*)| = 1 - 4 \left(\frac{\xi_{min}}{\nu} \right)^{\frac{1}{3}} h^{\frac{1}{3}} + O(h^{\frac{2}{3}}). \tag{10}$$

The proof of these results can also be found in [1]. In the special case when $b = 0$ (advection is normal to the interface), there is no approximation and our results above give the optimized Robin parameter. The asymptotic performance of the exact optimized Robin conditions (from solving (8)) is expected to be the same as (10) up to order $h^{1/3}$, with the same constant.

Fig. 1 shows, on the right, the convergence factors obtained for four different Robin transmission conditions, when overlap is used.

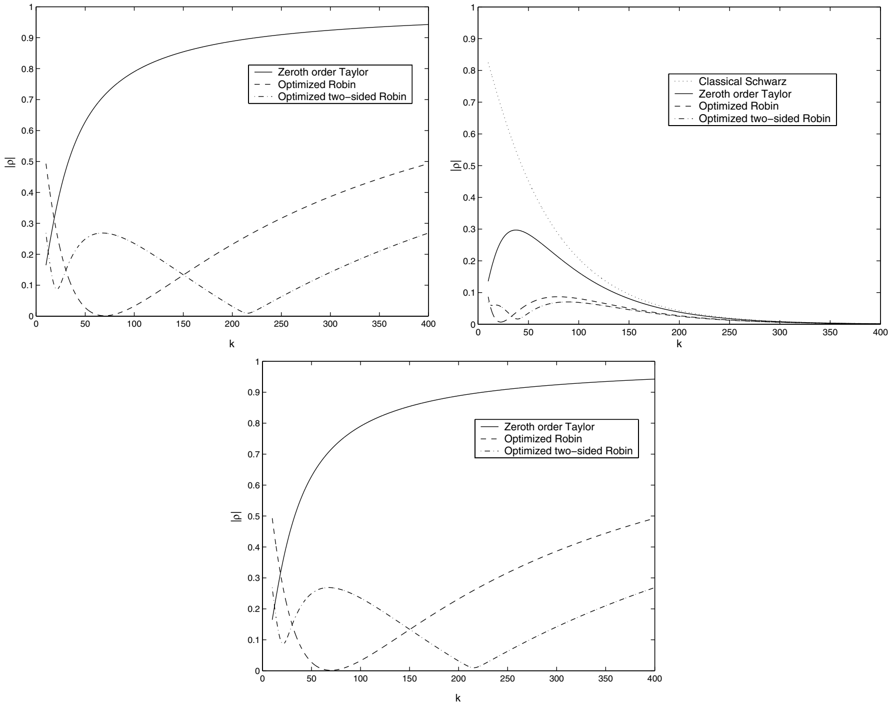


Fig. 1. Convergence factors for the values $\nu = 0.1$, $a = 1$, $b = 1$, $c = 1$, $[k_{min}, k_{max}] = [10, 400]$. The case without overlap is shown on the left, and with overlap $L = \pi/400$ on the right.

6 Numerical Experiments

We consider here an example with a varying advection $\mathbf{a}(x, y)$ obtained from a Navier-Stokes computation, see Fig. 2. The domain is the square $\Omega = (0, \pi)^2$, the viscosity is taken to be $\nu = 0.1$, and $c = 1$. The source term is given by $f(x, y) = \sin(5x)\sin(5y)$. The results were obtained using a finite difference solver, for rectangular domains. The original region is divided into two symmetric subdomains, with vertical interfaces. For the initial guess to start the Schwarz iteration, we use vectors of random values, to make sure the initial error contains a wide range of frequency components.

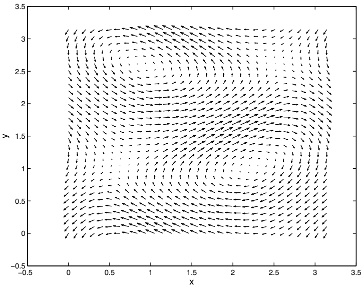


Fig. 2. The advection field.

The optimized Schwarz methods are constructed using model problems with constant coefficients. When the coefficients are varying (continuously) in the domain, we need to find optimized conditions at each mesh point on the interfaces separately. In our setting the optimized Robin parameters will depend on y, i.e. $p^* = p^*(y)$.

Note that the computation of the optimized conditions is done only once, before starting the Schwarz iteration.

Fig. 3 shows the convergence of the different Schwarz methods, using both nonoverlapping and overlapping decompositions. The effect of using overlap is significant; even with a small overlap of only two grid spaces, the number of iterations required to reach a tolerance is decreased by more than a factor 2.

We also looked at the effect of h on the convergence rate of the Schwarz iteration. Fig. 4 shows logarithmic plots of the number of iterations needed to achieve an error reduction of 10^{-6}, for different values of the mesh size h. The numerical results agree well with theory, both for what we have derived, and for what we expect for two-sided Robin conditions.

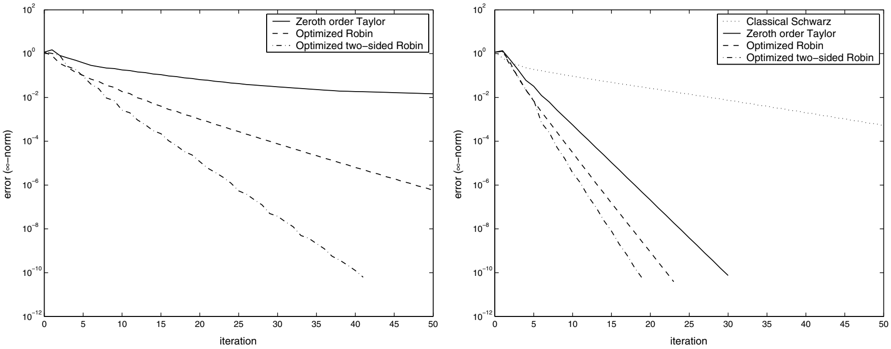


Fig. 3. Comparison of different transmission conditions for a varying advection, $\nu = 0.1$, $c = 1$, $h = \pi/300$. The case without overlap is shown on the left, and the case with overlap ($L = 2h$) on the right.

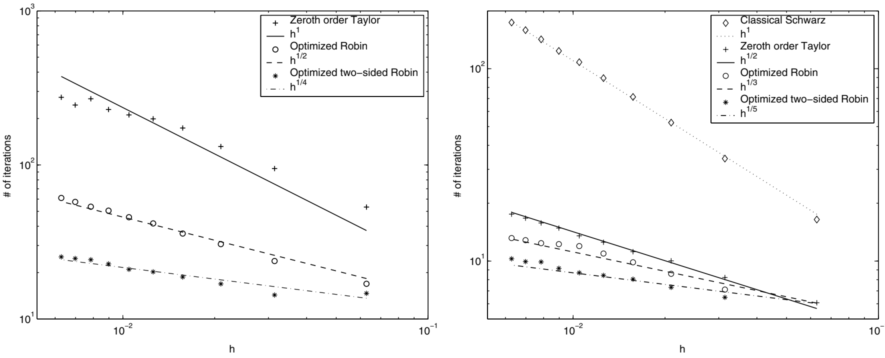


Fig. 4. Number of iterations needed to achieve an error of 10^{-6}, for different values of h, without overlap on the left and with overlap $L = 2h$ on the right.

7 Conclusion

We have computed optimized Robin transmission conditions in the Schwarz iteration for the advection-diffusion equation, by solving analytically the min-max problem. When the subdomains are not overlapping, the optimized parameter is given by an explicit formula. In the overlapping case, we have solved an approximate min-max problem only: computing the optimized parameter reduces to solving a nonlinear equation (in the worst case). The approximation we have made is good when the advection is not too strongly tangential to the interfaces, and for small mesh sizes h. The asymptotic performance of these optimized methods exhibits a weaker dependence on the mesh size than previously known Robin conditions.

References

1. O. Dubois, *Optimized Schwarz methods for the advection-diffusion equation*, Master's thesis, McGill University, 2003.
2. M. J. Gander, *Optimized Schwarz methods for Helmholtz problems*, in Thirteenth international conference on domain decomposition, N. Debit, M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, eds., 2001, pp. 245–252.
3. ———, *Optimized Schwarz methods*, Tech. Rep. 2003-01, Dept. of Mathematics and Statistics, McGill University, 2003. In revision for SIAM J. Numer. Anal.
4. M. J. Gander, L. Halpern, and F. Nataf, *Optimized Schwarz methods*, in Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan, T. Chan, T. Kako, H. Kawarada, and O. Pironneau, eds., Bergen, 2001, Domain Decomposition Press, pp. 15–28.
5. M. J. Gander, F. Magoulès, and F. Nataf, *Optimized Schwarz methods without overlap for the Helmholtz equation*, SIAM J. Sci. Comput., 24 (2002), pp. 38–60.
6. C. Japhet, *Conditions aux limites artificielles et décomposition de domaine: Méthode oo2 (optimisé d'ordre 2). Application à la résolution de problèmes en mécanique des fluides*, Tech. Rep. 373, CMAP (Ecole Polytechnique), 1997.
7. ———, *Optimized Krylov-Ventcell method. Application to convection-diffusion problems*, in Proceedings of the 9th international conference on domain decomposition methods, P. E. Bjørstad, M. S. Espedal, and D. E. Keyes, eds., ddm.org, 1998, pp. 382–389.
8. 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 , held in Houston, Texas, March 20-22, 1989, T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM.
9. F. Nataf and F. Rogier, *Factorization of the convection-diffusion operator and the Schwarz algorithm*, Math. Models Methods Appl. Sci., 5 (1995), pp. 67–93.

Optimized Algebraic Interface Conditions in Domain Decomposition Methods for Strongly Heterogeneous Unsymmetric Problems

Luca Gerardo-Giorda[1] and Frédéric Nataf[2]

[1] Dipartimento di Matematica, Università di Trento, Italy.
gerardo@science.unitn.it. (This author's work was supported by the HPMI-GH-99-00012-05 Marie Curie Industry Fellowship at IFP - France.)
[2] CNRS, UMR 7641, CMAP, École Polytechnique, France.
nataf@cmap.polytechnique.fr

1 Introduction

Let $\Omega = \mathbf{R} \times Q$, where Q is a bounded domain of $\mathbf{R}^2$, and consider the elliptic PDE of advection-diffusion-reaction type given by

$$\begin{array}{r} -\text{div }(c\nabla u) + \text{div }(\mathbf{b}u) + \eta u = f \quad \text{ in } \Omega \\ \mathcal{B}u = g \text{ on } \mathbf{R} \times \partial Q, \end{array} \tag{1}$$

with the additional requirement that the solutions be bounded at infinity. After a finite element, finite differences or finite volume discretization, we obtain a large sparse system of linear equations, given by

$$\mathbf{A}\,\mathbf{w} = \mathbf{f}. \tag{2}$$

Under classical assumptions on the coefficients of the problem (*e.g.* $\eta - \frac{1}{2}\text{div }\mathbf{b} > 0$ a.e. in Ω) the matrix $\mathbf{A}$ in (2) is definite positive.
We solve problem (2) by means of an Optimized Schwarz Method: such methods have been introduced at the continuous level in [4], and at the discrete level in [5]. We design optimized interface conditions directly at the algebraic level, in order to guarantee robustness with respect to heterogeneities in the coefficients.

2 LDU factorization and absorbing boundary conditions

In this section we illuminate the link between an LDU factorization of a matrix and the construction of absorbing conditions on the boundary of a domain (see [1]). As it is well known in domain decomposition literature, such conditions can provide exact interface transmission operators. Let then $\widetilde{\Omega} \in \mathbf{R}^3$ be a bounded polyedral domain. We assume that the underlying grid is obtained as a deformation of a Cartesian grid on the unit cube, so that for suitable integers N_x, N_y, and N_z, $\mathbf{w} \in \mathbf{R}^{N_x \times N_y \times N_z}$.

If the unknowns are numbered lexicographically, the vector $\mathbf{w}$ is a collection of N_x sub-vectors $w_i \in \mathbf{R}^{N_y \times N_z}$, *i.e.*

$$\mathbf{w} = (w_1^T, \dots, w_{N_x}^T)^T. \tag{3}$$

From (3), the discrete problem in $\widetilde{\Omega}$ reads

$$\mathbf{B}\,\mathbf{w} = \mathbf{g}, \tag{4}$$

where $\mathbf{g} = (g_1, .., g_{N_x})^T$, each g_i being a $N_y \times N_z$ vector, and where the matrix $\mathbf{B}$ of the discrete problem has a block tri-diagonal structure

$$\mathbf{B} = \begin{pmatrix} D_1 & U_1 & & \\ L_1 & D_2 & \ddots & \\ & \ddots & \ddots & U_{N_x-1} \\ & & L_{N_x-1} & D_{N_x} \end{pmatrix}, \tag{5}$$

where each block is a matrix of order $N_y \times N_z$.
An exact block factorization of the matrix $\mathbf{B}$ defined in (5) is given by

$$\mathbf{B} = (\mathbf{L} + \mathbf{T})\mathbf{T}^{-1}(\mathbf{U} + \mathbf{T}), \tag{6}$$

where

$$\mathbf{L} = \begin{pmatrix} 0 & & & \\ L_1 & \ddots & & \\ & \ddots & \ddots & \\ & & L_{N_x-1} & 0 \end{pmatrix} \qquad \mathbf{U} = \begin{pmatrix} 0 & U_1 & & \\ & \ddots & \ddots & \\ & & \ddots & U_{N_x-1} \\ & & & 0 \end{pmatrix},$$

while $\mathbf{T}$ is a block-diagonal matrix whose nonzero entries are the blocks T_i defined recursively as

$$T_i = \begin{cases} D_1 & \text{for } i = 1 \\ D_i - L_{i-1}T_{i-1}^{-1}U_{i-1} & \text{for } 1 < i \leq N_x. \end{cases}$$

At this time, we can give here the algebraic counterpart of absorbing boundary conditions. Assume $\mathbf{g} = (0, .., 0, g_{p+1}, .., g_{N_x})$, and let $N_p = N_x - p + 1$. To reduce the size of the problem, we look for a block matrix $\mathbf{K} \in (\mathbf{R}^{N_y \times N_z})^{N_p}$, each entry of which is a $N_y \times N_z$ matrix, such that the solution of $\mathbf{K}\mathbf{v} = \tilde{\mathbf{g}} = (0, g_{p+1}, .., g_{N_x})^T$ satisfies $v_k = w_{k+p-1}$ for $k = 1, ..N_p$. The rows 2 through N_p in the matrix $\mathbf{K}$ coincide with the last $N_p - 1$ rows of the original matrix $\mathbf{B}$. To identify the first row, which corresponds to the absorbing boundary condition, take as a right hand side in (4) the vector $\mathbf{g} = (0, .., 0, g_{p+1}, .., g_{N_x})$, and, owing to (6), consider the first p rows of the factorized problem

$$\begin{pmatrix} T_1 & & & \\ L_1 & T_2 & & \\ & \ddots & \ddots & \\ & & L_{p-1} & T_p \end{pmatrix} \begin{pmatrix} T_1^{-1} & & & \\ & T_2^{-1} & & \\ & & \ddots & \\ & & & T_p^{-1} \end{pmatrix} \begin{pmatrix} T_1 & U_1 & & & \\ & T_2 & U_2 & & \\ & & \ddots & \ddots & \\ & & & T_p & U_p \end{pmatrix} \begin{pmatrix} w_1 \\ \vdots \\ w_p \\ w_{p+1} \end{pmatrix} = \begin{pmatrix} 0 \\ \vdots \\ 0 \end{pmatrix}.$$

The first two are $p \times p$ square invertible block matrices, so we need to consider only the third one, a rectangular $p \times (p+1)$ matrix: from the last row we get

$$T_p w_p + U_p w_{p+1} = 0, \tag{7}$$

which, identifying $v_1 = w_p$ and $v_2 = w_{p+1}$, provides the first row in matrix $\mathbf{K}$.
Assume now that $\mathbf{g} = (g_1, .., g_{q-1}, 0, .., 0)^T$. A similar procedure can be developed to reduce the size of the problem, by starting the recurrence in the factorization (6) from D_{N_x}, as

$$\widetilde{T}_i = \begin{cases} D_i - U_i T_{i+1}^{-1} L_i & \text{for } 1 \le i < N_x \\ \\ D_{N_x} & \text{for } i = N_x, \end{cases}$$

and we can easily obtain the equation for the last row in the reduced equation as

$$L_q w_{q-1} + \widetilde{T}_q w_q = 0. \tag{8}$$

3 Optimal interface conditions for an infinite layered domain

In this section we go back to problem (1), where the domain Ω is infinite in the x direction. We consider a two domain decomposition $\bar{\Omega} = \bar{\Omega}_1 \cup \bar{\Omega}_2$, $\Omega_1 \cap \Omega_2 = \emptyset$, where

$$\Omega_1 = \mathbf{R}^- \times Q, \qquad \Omega_2 = \mathbf{R}^+ \times Q,$$

and we denote with $\Gamma = \partial\Omega_1 \cap \partial\Omega_2$ the common interface of the two subdomains. We assume that the viscosity coefficients are layered (*i.e.* they do not depend on the x variable), and consider a discretization on a uniform grid via a finite volume scheme with an upwind treatment of the advective flux.
The resulting linear system is given by

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{1\Gamma} & \mathbf{0} \\ \mathbf{A}_{\Gamma 1} & \mathbf{A}_{\Gamma\Gamma} & \mathbf{A}_{\Gamma 2} \\ \mathbf{0} & \mathbf{A}_{2\Gamma} & \mathbf{A}_{22} \end{pmatrix} \begin{pmatrix} \mathbf{w}_1 \\ \mathbf{w}_\Gamma \\ \mathbf{w}_2 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma \\ \mathbf{f}_2 \end{pmatrix} \tag{9}$$

where $\mathbf{w}_i$ is the vector of the internal unknowns in domain Ω_i $(i = 1, 2)$, and $\mathbf{w}_\Gamma$ is the vector of interface unknowns. In order to guarantee the conservativity of the finite volume scheme, the vector of interface unknown consists of two sets of variables, $\mathbf{w}_\Gamma = (w_\Gamma, w_\lambda)^T$, the first one expressing the continuity of the diffusive flux, the second expressing the continuity of the advective one.
If the unknowns are numbered lexicographically, the matrix $\mathbf{A}$ is given by

$$\mathbf{A} = \left(\begin{array}{cccc|c|cccc}
\ddots & \ddots & \ddots & & \vdots & & & & \\
& L_1 & D_1 & U_1 & 0 & & \mathbf{0} & & \\
& & L_1 & D_{1\Gamma} & \mathbf{U}_{1\Gamma} & & & & \\
\hline
\cdots & \cdots & 0 & \mathbf{L}_{1\Gamma} & \mathbf{D}_{\Gamma\Gamma} & \mathbf{U}_{2\Gamma} & 0 & \cdots & \cdots \\
\hline
& & & & \mathbf{L}_{2\Gamma} & D_{2\Gamma} & U_2 & & \\
& \mathbf{0} & & & 0 & L_2 & D_2 & U_2 & \\
& & & & \vdots & & \ddots & \ddots & \ddots
\end{array}\right), \tag{10}$$

where the block $\mathbf{D}_{\Gamma\Gamma}$ is square, whereas the blocks $\mathbf{L}_{i\Gamma}$, and $\mathbf{U}_{i\Gamma}$ $(i = 1, 2)$ are rectangular.
By duplicating the interface variables $\mathbf{w}_{\Gamma}$ into $\mathbf{w}_{\Gamma,1}$ and $\mathbf{w}_{\Gamma,2}$, we can define a Schwarz algorithm directly at the algebraic level, as

$$\begin{aligned}
&\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{1\Gamma} \\ \mathbf{A}_{\Gamma 1} & \mathbf{T}_1 \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^{k+1} \\ \mathbf{v}_{\Gamma,1}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma + (\mathbf{T}_1 - \mathbf{D}_{\Gamma\Gamma})\,\mathbf{v}_{\Gamma,2}^k - \mathbf{A}_{\Gamma 2}\mathbf{v}_2^k \end{pmatrix} \\
&\begin{pmatrix} \mathbf{A}_{22} & \mathbf{A}_{2\Gamma} \\ \mathbf{A}_{\Gamma 2} & \mathbf{T}_2 \end{pmatrix} \begin{pmatrix} \mathbf{v}_2^{k+1} \\ \mathbf{v}_{\Gamma,2}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_2 \\ \mathbf{f}_\Gamma + (\mathbf{T}_2 - \mathbf{D}_{\Gamma\Gamma})\,\mathbf{v}_{\Gamma,1}^k - \mathbf{A}_{\Gamma 1}\mathbf{v}_1^k \end{pmatrix}.
\end{aligned} \tag{11}$$

As it is well known in literature, if we take

$$\mathbf{T}_1 = \mathbf{A}_{\Gamma\Gamma} - \mathbf{A}_{\Gamma 2}\mathbf{A}_{22}^{-1}\mathbf{A}_{2\Gamma} \qquad \mathbf{T}_2 = \mathbf{A}_{\Gamma\Gamma} - \mathbf{A}_{\Gamma 1}\mathbf{A}_{11}^{-1}\mathbf{A}_{1\Gamma},$$

the algorithm (11) converges in two iterations. We are in the position to give the following result, the proof of which will be given in [3].

Lemma 1. *Let* $\mathbf{A}$ *be the matrix defined in (9), and let* $T_{1,\infty}$ *and* $T_{2\infty}$ *be such that* $T_{1,\infty} = D_1 - L_1 T_{1,\infty}^{-1} U_1$ *and* $T_{2,\infty} = D_2 - U_2 T_{2,\infty}^{-1} L_2$*. We have*

$$\mathbf{A}_{\Gamma 1}\mathbf{A}_{11}^{-1}\mathbf{A}_{1\Gamma} = \mathbf{L}_{1\Gamma}\left(D_{1\Gamma} - L_1\, T_{1,\infty}^{-1}\, U_1\right)^{-1}\mathbf{U}_{1\Gamma}$$

$$\mathbf{A}_{\Gamma 2}\mathbf{A}_{22}^{-1}\mathbf{A}_{2\Gamma} = \mathbf{U}_{2\Gamma}\left(D_{2\Gamma} - U_2\, T_{2,\infty}^{-1}\, L_2\right)^{-1}\mathbf{L}_{2\Gamma}.$$

■

Noticing that $\mathbf{A}_{\Gamma\Gamma} = \mathbf{D}_{\Gamma\Gamma}$, the optimal interface operators are given by

$$\begin{aligned}
\mathbf{T}_1^{\text{ex}} &= \mathbf{D}_{\Gamma\Gamma} - \mathbf{L}_{1\Gamma}\left[D_{1\Gamma} - L_1\, T_{1,\infty}^{-1}\, U_1\right]^{-1}\mathbf{U}_{1\Gamma} \\
\mathbf{T}_2^{\text{ex}} &= \mathbf{D}_{\Gamma\Gamma} - \mathbf{U}_{2\Gamma}\left[D_{2\Gamma} - U_2\, T_{2,\infty}^{-1}\, L_2\right]^{-1}\mathbf{L}_{2\Gamma}.
\end{aligned} \tag{12}$$

4 Optimized algebraic interface conditions for a non-overlapping Schwarz method

The lack of sparsity of the matrices $\mathbf{T}_1^{\text{ex}}$ and $\mathbf{T}_2^{\text{ex}}$ in (12), make them unsuitable in practice. Therefore we choose for $\mathbf{T}_1$ and $\mathbf{T}_2$ in (11) two suitable approximations of $\mathbf{T}_1^{\text{ex}}$ and $\mathbf{T}_2^{\text{ex}}$, respectively.
At the cost of enlarging the size of the interface problem, we choose $\mathbf{T}_1^{\text{app}}$ and $\mathbf{T}_2^{\text{app}}$ defined as follows:

$$\begin{aligned}
\mathbf{T}_1^{\text{app}} &= \mathbf{D}_{\Gamma\Gamma} - \mathbf{L}_{1\Gamma}\left[D_{1\Gamma} - L_1\,(T_{1,\infty}^{\text{app}})^{-1}\, U_1\right]^{-1}\mathbf{U}_{1\Gamma} \\
\mathbf{T}_2^{\text{app}} &= \mathbf{D}_{\Gamma\Gamma} - \mathbf{U}_{2\Gamma}\left[D_{2\Gamma} - U_2\,(T_{2,\infty}^{\text{app}})^{-1}\, L_2\right]^{-1}\mathbf{L}_{2\Gamma},
\end{aligned} \tag{13}$$

where $T_{1,\infty}^{\text{app}}$ and $T_{2,\infty}^{\text{app}}$ are suitable sparse approximations of $T_{1,\infty}$ and $T_{2,\infty}$, respectively. The most natural choice would be to take their diagonals, but, in order to have a usable condition, we wish to avoid the computation of both $T_{1,\infty}$ and $T_{2,\infty}$, which is too costly. Notice that if D_j, L_j, and U_j $(j = 1, 2)$ were all diagonal matrices the same would hold also for $T_{j,\infty}$. Moreover, if all the matrices involved commute, or if $L_j = U_j^T$, we would have

$$T_{1,\infty} = \frac{D_1}{2} + \sqrt{\frac{(-L_1)^{1/2} D_1 (-U_1)^{-1/2} (-L_1)^{-1/2} D_1 (-U_1)^{1/2}}{4} - L_1 U_1}.$$

and a similar formula holds for $T_{2,\infty}$, with the roles of L_2 and U_2 exchanged. These considerations have led us to consider the following approximations of $T_{1,\infty}$ and $T_{2,\infty}$.
Let d_j, l_j, and u_j be the diagonals of D_j, L_j and U_j, respectively.
Robin: We choose in (13)

$$T_{1,\infty}^{\text{app}} = \frac{D_1}{2} + \alpha_1^{opt} \mathcal{D}_1,$$

where $\mathcal{D}_1 = diag\left(\frac{\sqrt{d_1^2 - 4 l_1 u_1}}{2}\right)$. The optimized parameter is given by

$$(\alpha_1^{\text{opt}})^2 = \max\left\{\sqrt{r_1^2 + I_1^2}\ ,\ \sqrt{r_1 R_1 - I_1^2}\right\}, \tag{14}$$

where we have set $r_1 := \min \operatorname{Re}\lambda$, $R_1 := \max \operatorname{Re}\lambda$, and $I_1 := \max \operatorname{Im}\lambda$, $\lambda \in \sigma\left(\frac{(-L_1)^{1/2} D_1 (-U_1)^{-1/2} (-L_1)^{-1/2} D_1 (-U_1)^{1/2}}{4} - L_1 U_1\right) diag\left(\frac{\sqrt{d_1^2 - 4 l_1 u_1}}{2}\right)^{-2}$,
with a similar formula for $T_{2,\infty}^{\text{app}}$.

Order 2: This condition is obtained by blending together two first order approximations, and we have

$$T_{1,\infty}^{\text{app}} = L_1 \left([\widetilde{\mathcal{D}}_1, \mathcal{L}_1] + (\alpha_1 + \alpha_2)\mathcal{L}_1\right)^{-1} \left(\widetilde{\mathcal{D}}_1^2 + (\alpha_1 + \alpha_2)\widetilde{\mathcal{D}}_1 + \alpha_1 \alpha_2 Id - \mathcal{L}_1 \mathcal{U}_1\right)$$

where $[.,.]$ is the Lie bracket, where $\widetilde{\mathcal{D}}_1 = \frac{\mathcal{D}_1^{-1} D_1}{2}$, $\mathcal{L}_1 = \mathcal{D}_1^{-1} L_1$, $\mathcal{U}_1 = \mathcal{D}_1^{-1} U_1$, and where

$$(\alpha_1 \alpha_2)^2 = r_1\, R_1 \qquad (\alpha_1 + \alpha_2)^2 = \sqrt{2\,(r_1 + R_1)\sqrt{r_1\, R_1}}, \tag{15}$$

r_1 and R_1 being defined as before.

The tuning of the optimized parameters for both conditions can be found in [2], and a more exhaustive presentation of the construction of interface conditions and of the numerical tests will be given in a forthcoming paper [3]. The proposed interface conditions are built directly at the algebraic level, and are easy to implement. However, they rely heavily on the approximation of the Schur complement and, if on one hand the extension to a decomposition into strips appears quite straightforward, on the other hand further work needs to be done in order to analyse their scalability for an arbitrary decomposition of the computational domain.
Finally, it is easy to prove the following result (see [3]).

Lemma 2. *The Schwarz algorithm*

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{1\Gamma} \\ \mathbf{A}_{\Gamma 1} & \mathbf{T}_2^{\text{app}} \end{pmatrix} \begin{pmatrix} \mathbf{v}_1^{k+1} \\ \mathbf{v}_{\Gamma,1}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma + (\mathbf{T}_2^{\text{app}} - \mathbf{D}_{\Gamma\Gamma})\, \mathbf{v}_{\Gamma,2}^k - \mathbf{A}_{\Gamma 2} \mathbf{v}_2^k \end{pmatrix}$$

$$\begin{pmatrix} \mathbf{A}_{22} & \mathbf{A}_{2\Gamma} \\ \mathbf{A}_{\Gamma 2} & \mathbf{T}_1^{\text{app}} \end{pmatrix} \begin{pmatrix} \mathbf{v}_2^{k+1} \\ \mathbf{v}_{\Gamma,2}^{k+1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_2 \\ \mathbf{f}_\Gamma + (\mathbf{T}_1^{\text{app}} - \mathbf{D}_{\Gamma\Gamma})\, \mathbf{v}_{\Gamma,1}^k - \mathbf{A}_{\Gamma 1} \mathbf{v}_1^k \end{pmatrix}.$$

converges to the solution to problem (9). ■

4.1 Substructuring

The iterative method can be substructured in order to use a Krylov type method and speed up the convergence. We introduce the auxiliary variables

$$\mathbf{h}_1 = (\mathbf{T}_2^{\mathrm{app}} - \mathbf{D}_{\Gamma\Gamma})\, \mathbf{v}_{\Gamma,2} - \mathbf{A}_{\Gamma 2}\, \mathbf{v}_2, \qquad \mathbf{h}_2 = -\mathbf{A}_{\Gamma 1}\, \mathbf{v}_1 + (\mathbf{T}_1^{\mathrm{app}} - \mathbf{D}_{\Gamma\Gamma})\, \mathbf{v}_{\Gamma,1},$$

and we define the interface operator T_h

$$T_h : \begin{pmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \\ \mathbf{f} \end{pmatrix} \longmapsto \begin{pmatrix} -\mathbf{A}_{\Gamma 1}\mathbf{v}_1 + (\mathbf{T}_1^{\mathrm{app}} - \mathbf{D}_{\Gamma\Gamma})\, \mathbf{v}_{\Gamma,1} \\ \\ (\mathbf{T}_2^{\mathrm{app}} - \mathbf{D}_{\Gamma\Gamma})\, \mathbf{v}_{\Gamma,2} - \mathbf{A}_{\Gamma 2}\mathbf{v}_2 \end{pmatrix}$$

where $\mathbf{f} = (\mathbf{f}_1, \mathbf{f}_\Gamma, \mathbf{f}_2)^T$, whereas $(\mathbf{v}_1, \mathbf{v}_{\Gamma,1})$ and $(\mathbf{v}_2, \mathbf{v}_{\Gamma,2})$ are the solutions of

$$\begin{pmatrix} \mathbf{A}_{11} & \mathbf{A}_{1\Gamma} \\ \mathbf{A}_{\Gamma 1} & \mathbf{T}_2^{\mathrm{app}} \end{pmatrix} \begin{pmatrix} \mathbf{v}_1 \\ \mathbf{v}_{\Gamma,1} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_1 \\ \mathbf{f}_\Gamma + \mathbf{h}_1 \end{pmatrix}$$

and

$$\begin{pmatrix} \mathbf{A}_{22} & \mathbf{A}_{2\Gamma} \\ \mathbf{A}_{\Gamma 2} & \mathbf{T}_1^{\mathrm{app}} \end{pmatrix} \begin{pmatrix} \mathbf{v}_2 \\ \mathbf{v}_{\Gamma,2} \end{pmatrix} = \begin{pmatrix} \mathbf{f}_2 \\ \mathbf{f}_\Gamma + \mathbf{h}_2 \end{pmatrix}.$$

So far, the substructuring operator is obtained simply by matching the conditions on the interface, and reads in matrix form

$$\Big(\mathbf{Id} - \mathbf{\Pi T_h}\Big)\ (\mathbf{h}_1, \mathbf{h}_2)^T = \mathbf{F}, \tag{16}$$

where $\mathbf{\Pi}$ is the swap operator on the interface, where $\mathbf{F} = \mathbf{\Pi} T_h(0, 0, \mathbf{f})$, and where the matrix $\mathbf{T}_h$ is given in the following lemma (for a proof see [3]).

Lemma 3. *The matrix* $\mathbf{T}_h$ *in (16) is given by*

$$\begin{pmatrix} (\mathbf{T}_1^{\mathrm{app}} - \mathbf{T}_1^{\mathrm{ex}})\, (\mathbf{T}_1^{\mathrm{ex}} + \mathbf{T}_2^{\mathrm{app}} - \mathbf{D}_{\Gamma\Gamma})^{-1} & 0 \\ \\ 0 & (\mathbf{T}_2^{\mathrm{app}} - \mathbf{T}_2^{\mathrm{ex}})\, (\mathbf{T}_2^{\mathrm{ex}} + \mathbf{T}_1^{\mathrm{app}} - \mathbf{D}_{\Gamma\Gamma})^{-1} \end{pmatrix}.$$

5 Numerical Results

We consider problem (1) in $\Omega = \mathbf{R} \times (0, 1)$, with Dirichlet boundary conditions at the bottom and a Neumann boundary condition on the top. We use a finite volume discretization with an upwind scheme for the advective term. We build the matrices of the substructured problem for various interface conditions and we study their spectra. We give in the tables the iteration counts corresponding to the solution of the substructured problem by a GMRES algorithm with a random right hand side G, and the ratio of the largest modulus of the eigenvalues over the smallest real part. The stopping criterion for the GMRES algorithm is a reduction of the residual by a factor 10^{-10}. We consider both advection dominated and diffusion dominated flows, and different kind of heterogeneities. We report here the results for three different test cases.

Test 1: the flow is advection dominated, the viscosity coefficients are layered, and the subdomains are symmetric with respect to the interface.

Test 2: the flow is diffusion dominated, the viscosity coefficients are layered, but are not symmetric with respect to the interface.
Test 3: the flow is diffusion dominated, the viscosity coefficients are layered, non symmetric w.r.t. the interface, and anisotropic, with an anisotropy ratio up to order 10^4.
The velocity field is diagonal with respect to the interface and constant. The numerical tests are performed with MATLAB® 6.1. A more detailed description of the test cases as well as futher numerical results can be found in a forthcoming paper [3].

$p = q = 10$	ny		10	20	40	80	160	320
Test 1	iter	**Robin**	4	6	8	11	16	23
		Order 2	4	5	6	8	9	10
	cond	**Robin**	1.05	1.25	1.68	3.27	6.57	13.51
		Order 2	1.01	1.02	1.14	1.34	1.61	1.92
Test 2	iter	**Robin**	7	10	13	16	19	21
		Order 2	6	6	8	11	15	19
	cond	**Robin**	1.61	1.83	2.59	3.52	3.94	4.12
		Order 2	1.21	1.26	1.30	1.83	2.76	3.68
Test 3	iter	**Robin**	9	17	27	35	42	47
		Order 2	7	10	14	16	19	21
	cond	**Robin**	5.42	18.27	24.75	31.04	38.32	47.29
		Order 2	1.54	2.75	4.48	5.92	6.32	6.86

Table 1. Iteration counts and condition number for the substructured problem in Tests 1-3

Both conditions perform fairly well, in both terms of iteration counts and conditioning of the substructured problem, especially for the second order conditions, that show a good scalability with respect to the mesh size.

6 Conclusions

We have proposed two kinds of algebraic interface conditions for unsymmetric elliptic problem, which appear to be very efficient and robust in term of iteration counts and conditioning of the problem with respect to the mesh size and the heterogeneities in the viscosity coefficients.

References

1. B. Engquist and A. Majda, *Absorbing boundary conditions for the numerical simulation of waves*, Math. Comp., 31 (1977), pp. 629–651.

2. L. Gerardo Giorda and F. Nataf, *Optimized Schwarz Methods for unsymmetric layered problems with strongly discontinuous and anisotropic coefficients*, Tech. Rep. 561, CMAP (Ecole Polytechnique), December 2004.
3. ———, *Optimized Algebraic Schwarz Methods for strongly heterogeneous and anisotropic layered problems*, Tech. Rep. 575, CMAP (Ecole Polytechnique), June 2005.
4. 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 , held in Houston, Texas, March 20-22, 1989, T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM.
5. F.-X. Roux, F. Magoulès, S. Salmon, and L. Series, *Optimization of interface operator based on algebraic approach*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003.

Optimal and Optimized Domain Decomposition Methods on the Sphere

Sébastien Loisel

Department of Mathematics, Wachman Hall, 1805 North Broad Street, Temple University, Philadelphia, PA 19122, USA. loisel@temple.edu

1 Introduction

At the heart of numerical weather prediction algorithms lie a Laplace and positive definite Helmholtz problems on the sphere [12]. Recently, there has been interest in using finite elements [2] and domain decomposition methods [1, 10]. The Schwarz iteration [7, 8, 9] and its variants [9, 5, 6, 4, 3, 11] are popular domain decomposition methods.

In this paper, we introduce improved transmission operators for the Laplace problem on the sphere. In section 2, we review the case of the Laplace operator on the sphere and recall the Schwarz iteration and its convergence estimates, previously published in [1]; we also give a new semidiscrete estimate which is substantially similar to the continuous one. In section 3, we introduce the framework of the optimized Schwarz iteration and give optimized operators. In section 4, we present numerical results that agree with the theoretical predictions.

2 The Laplace operator on the sphere

We take the Laplace operator in $\mathbb{R}^3$, given by

$$\mathcal{L}u = u_{xx} + u_{yy} + u_{zz},$$

rephrase it in spherical coordinates and set $\dfrac{\partial u}{\partial r} = 0$ to obtain

$$\mathcal{L}u = \frac{1}{\sin^2\varphi}\frac{\partial^2 u}{\partial\theta^2} + \frac{1}{\sin\varphi}\frac{\partial}{\partial\varphi}\left(\sin\varphi\frac{\partial u}{\partial\varphi}\right),$$

where $\varphi \in [0, \pi]$ is the colatitude and $\theta \in [-\pi, \pi]$ the longitude.

2.1 The solution of the Laplace problem

We take a Fourier transform in θ but not in φ; this lets us analyze domain decompositions with latitudinal boundaries. The Laplacian becomes

$$\mathcal{L}\hat{u}(\varphi,m)=\frac{-m^2}{\sin^2\varphi}\hat{u}(\varphi,m)+\frac{1}{\sin\varphi}\frac{\partial}{\partial\varphi}\left(\sin\varphi\frac{\partial\hat{u}(\varphi,m)}{\partial\varphi}\right),\ \varphi\in[0,\pi],\ m\in\mathbb{Z}.$$

For boundary conditions, the periodicity in θ is taken care of by the Fourier decomposition. The poles impose that $u(0,\theta)$ and $u(\pi,\theta)$ do not vary in θ. For $m\neq 0$ this is equivalent to

$$\hat{u}(0,m)=\hat{u}(\pi,m)=0,\ m\in\mathbb{Z},\ m\neq 0.$$

For $m=0$, the relation $u_\varphi(0,\theta)=-u_\varphi(0,\theta+\pi)$ leads to $\int_0^{2\pi}u_\varphi(0,\theta)\,d\theta=-\int_0^{2\pi}u_\varphi(0,\theta)\,d\theta$, i.e.,

$$\hat{u}_\varphi(0,0)=\hat{u}_\varphi(\pi,0)=0.$$

If u is a solution of $\mathcal{L}u=f$ then so is $u+c$ ($c\in\mathbb{C}$), hence the ODE for $m=0$ is determined up to an additive constant.

With $m\neq 0$ fixed, the two independent solutions of $\mathcal{L}u=0$ are

$$g_\pm(\varphi,m)=\left(\frac{\sin(\varphi)}{\cos(\varphi)+1}\right)^{\pm|m|},\ m\in\mathbb{Z}\setminus\{0\}.$$

For $m=0$ the two independent solutions are

$$\hat{u}(\varphi,0)=C_1+C_2\log\left(\frac{1-\cos\varphi}{\sin\varphi}\right).$$

The solutions are defined on the domain $(0,\pi)$.

All the eigenvalues of $\mathcal{L}$ are of the form of $-n(n+1)$ for $n=0,1,...$; in particular, they are non-positive (and $\mathcal{L}$ is negative semi-definite.)

2.2 The Schwarz iteration for $\mathcal{L}$ with two latitudinal subdomains

Let $b<a$. Begin with random "candidate solutions" u_0 and v_0. Define u_{k+1} and v_{k+1} iteratively by:

$$\begin{cases}\mathcal{L}u_{k+1}=f & \text{in } \Omega_1=\{(\varphi,\theta)|0\leq\varphi<a\},\\ u_{k+1}(a,\theta)=v_k(a,\theta) & \theta\in[0,2\pi),\\ \mathcal{L}v_{k+1}=f & \text{in } \Omega_2=\{(\varphi,\theta)|b<\varphi\leq\pi\},\\ v_{k+1}(b,\theta)=u_k(b,\theta) & \theta\in[0,2\pi);\end{cases}\tag{1}$$

(see figure 1.)

We are interested in studying the error terms u_k-u and v_k-u where $\mathcal{L}u=f$ since they solve equations (1) with $f=0$. Hence for the remainder of this discussion, we will take $f=0$.

Using the Fourier transform in θ, we can write $\hat{u}_{k+2}(b,m)$ explicitly in terms of $\hat{u}_k(b,m)$. This allows us to obtain a convergence rate estimate, which we recall from [1].

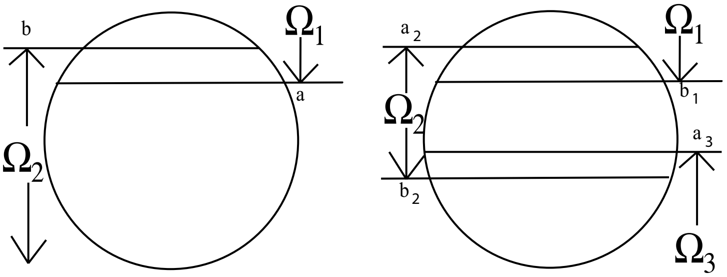


Fig. 1. Latitudinal domain decomposition. Left: two domains; right: multiple domains.

Theorem 1. *The Schwarz iteration on the sphere partitioned along two latitudes $b < a$ converges (except for the constant term.) The rate of convergence $|\hat{u}_{k+2}(b,m)/\hat{u}_k(b,m)|$ is*

$$C(m) = \left(\frac{\sin(b)}{\cos(b)+1}\right)^{2|m|} \left(\frac{\sin(a)}{\cos(a)+1}\right)^{-2|m|} < 1. \tag{2}$$

This convergence rate depends on the frequency m of u_k on the latitude b.

An analysis that is closer to the numerical algorithm would be to replace the continuous Fourier transform in θ by a discrete one.

Theorem 2. *(Semidiscrete analysis.) The Laplacian discretized in θ with n sample points:*

$$\mathcal{L}_n u = \frac{n^2}{4\pi^2 \sin^2 \varphi} \left(u\left(\varphi, \frac{j+1}{2\pi n}\right) - 2u(\varphi, j) + u\left(\varphi, \frac{j-1}{2\pi n}\right)\right) + \cot\varphi u_\varphi + u_{\varphi\varphi} \tag{3}$$

leads to a Schwarz iteration that converges with speed

$$\left(\frac{\sin(b)}{\cos(b)+1}\right)^{2|\tilde{m}|} \left(\frac{\sin(a)}{\cos(a)+1}\right)^{-2|\tilde{m}|} < 1$$

every two iterations, where

$$\tilde{m}^2 = \frac{n^2}{4\pi^2}(1 - \cos(2\pi k/n))$$

for the kth frequency.

The two contraction constants are very similar, and it is only possible to tell them apart on a logarithmic chart for the high frequencies (which converge quickly regardless.) For small values of m (ignoring $m = 0$ because that mode need not converge at all), the speed of convergence is very poor. The overall L^2 convergence rate (along the boundary) is given by $\sup_{m\geq 1} C(m) = C(1)$, and so the convergence rate of the Schwarz iteration deteriorates rapidly as $a - b$ vanishes.

While it is possible to prove that the Schwarz iteration converges regardless of subdomain shapes (so long as they are sufficiently "nice") and even regardless of the discretization (as long as it is sufficiently accurate) in the context of Sobolev spaces [7], it is difficult in general to obtain contraction constants as we have done here.

3 An optimized Schwarz iteration for $\mathcal{L}$ with latitudinal boundaries

We modify the transmission condition to obtain the following iteration:

$$\begin{cases} \mathcal{L}u_{k+1} = f & \text{in } \Omega_1 \\ \psi(\theta) * u_{k+1}(a,\theta) + \frac{\partial}{\partial\varphi}u_{k+1}(a,\theta) = \psi(\theta) * v_k(a,\theta) + \frac{\partial}{\partial\varphi}v_k(a,\theta) & \theta \in [0,2\pi), \\ \mathcal{L}v_{k+1} = f & \text{in } \Omega_2 \\ \xi(\theta) * v_{k+1}(b,\theta) + \frac{\partial}{\partial\varphi}v_{k+1}(b,\theta) = \xi(\theta) * u_k(b,\theta) + \frac{\partial}{\partial\varphi}u_k(b,\theta) & \theta \in [0,2\pi); \end{cases} \tag{4}$$

where ψ and ξ are distributions and Ω_1, Ω_2 are as previously defined. Choices include:

1. $(\psi * w)(\theta) = cw(\theta)$; that is, ψ is c times the point mass at $\theta = 0$. This results in a Robin transmission condition.
2. $(\psi * w)(\theta) = cw(\theta) + dw''(\theta)$. This results in a second order tangential transmission condition.
3. A nonlocal choice of ψ leading to an iteration that converges in two steps.

We have analyzed each case and obtained the following results.

Theorem 3. *(Nonlocal operator.) If, for each m, $\hat{\psi}(m) = |m|/\sin a$ and $\hat{\xi}(m) = -|m|/\sin b$, $\mathcal{L}u_0 = 0$ in $\Omega_1$and $\mathcal{L}v_0 = 0$ in Ω_2, then $u_1 = 0$ and $v_1 = 0$.*

Corollary 1. *The iteration (4) is convergent (modulo the constant mode) if $\hat{\psi}(m) > 0$ and $\hat{\xi}(m) < 0$ for all $m \neq 0$, regardless of overlap.*

The corollary follows from the calculations in the proof of the preceding theorem. We do not assume that $a \neq b$.

Theorem 4. *(Robin conditions.) Let $\psi * w = cw$ and $2N$ be the number of discretization points along the latitude $\varphi = \pi/2$. As long as $c > 0$, we have a convergent algorithm. The contraction constant is*

$$C_0(N) = \min_c \max_{m\in[1,N]} \kappa_1(m,c) = \min_c \max_{m\in[1,N]} \frac{(c-|m|)^2}{(c+|m|)^2}.$$

The minimum is obtained at $c = \sqrt{N}$, at which point the maximum contraction constant is

$$C_0(N) = \frac{(\sqrt{N}-1)^2}{(\sqrt{N}+1)^2}.$$

For the second order tangential operator, a continuous analysis leads to:

Theorem 5. *(Second order tangential transmission condition.) Let*

$$\psi * w = cw + d\frac{\partial^2}{\partial\varphi^2}w, \tag{5}$$

with $c \geq 0$ and $d \leq 0$, $cd \neq 0$. The best contraction constant is given by

$$C_2(N) = \min_{c,d} \max_{m\in[1,N]} \kappa_2(m,c,d) = \min_{c,d} \max_{m\in[1,N]} \frac{(c-dm^2-m)^2}{(c-dm^2+m)^2}.$$

Choosing c, d to obtain the smallest contraction gives

$$C_2(N) = \left(\frac{\sqrt{2}(N+1)^2\left(\frac{N}{(N+1)^2}\right)^{\frac{3}{4}} - 2N}{\sqrt{2}(N+1)^2\left(\frac{N}{(N+1)^2}\right)^{\frac{3}{4}} + 2N}\right)^2$$

for the parameters

$$c = -Nd = 2\left(\frac{N}{4N^2+8N+4}\right)^{\frac{3}{4}}(N+1). \tag{6}$$

We can use a semidiscrete analysis to obtain a similar result.

Theorem 6. *(Second order tangential transmission operator, semidiscrete.) A semidiscrete analysis leads to slightly different parameters c and d given by*

$$\begin{aligned}
\alpha &= \frac{N\pi^4 + 8N^3\pi^2 - N^2(8\pi^2+\pi^4) + N\pi^4}{4\pi^4 - 64\pi^2N^2 + 256N^4},\\
c' &= \frac{N(8n-\pi^2)}{2\alpha^{\frac{1}{4}}(8N^2-\pi^2)},\\
d' &= \frac{2\alpha^{\frac{3}{4}}(8N^2-\pi^2)}{N(8N-\pi^2)}.
\end{aligned}$$

In the presence of overlap, an extra trigonometric term appears that prevents exact analytic solutions. If we neglect such trigonometric terms, the optimization problem becomes to minimize the moduli of

$$\frac{\hat{\psi}(m)\sin a - |m|}{\hat{\psi}(m)\sin a + |m|} \quad \text{and} \quad \frac{\hat{\xi}(m)\sin b + |m|}{\hat{\xi}(m)\sin b - |m|}.$$

If $a = b$, this is a nonoverlapping problem with asymmetric subdomains, except if $a = \pi/2$. We adapt the preceding theorems.

Theorem 7. *To minimize the modulus of*

$$\frac{\hat{\psi}(m)\sin a - |m|}{\hat{\psi}(m)\sin a + |m|},$$

we can use $\psi = \sqrt{N}\csc a$ (Robin case) and $\psi = c\csc(a) + d\csc(a)\frac{\partial}{\partial\varphi}$ (with c, d given by either of the second order tangential choices.)

3.1 Multiple latitudinal subdomains

Let $l > 1$ and $u_{l+1}^{(k)}$, for $1 \leq k \leq n$, be the solutions of

$$\begin{cases} \mathcal{L}u_{l+1}^{(k)}(\varphi,\theta) = f & \text{in } \Omega_k \\ u_{l+1}^{(k)}(a_k,\theta) + \psi_k * \dfrac{\partial}{\partial\varphi}u_{l+1}^{(k)}(a_k,\theta) = u_l^{(k-1)}(a_k,\theta) + \psi_k * u_l^{(k-1)}(a_k,\theta) & \theta \in [0,2\pi) \text{ if } k > 1, \\ u_{l+1}^{(k)}(b_k,\theta) + \xi_k * \dfrac{\partial}{\partial\varphi}u_{l+1}^{(k)}(b_k,\theta) = u_l^{(k+1)}(b_k,\theta) + \xi_k * u_l^{(k+1)}(b_k,\theta) & \theta \in [0,2\pi) \text{ if } k < n; \end{cases}$$

where $0 = a_1 < a_2 < ... < a_n$, $b_1 < b_2 < ... < b_n = \pi$, $\Omega_k = \{(\varphi,\theta)|\varphi \in (a_k,b_k)\}$, $a_k < b_k$, $k = 1,...,n$ and $\cup_k[a_k,b_k] = [0,\pi]$ (see figure 1.) Once more using a Fourier transform in θ, one can show that the same optimal operators lead to convergence in n steps. The iteration leads to a matrix whose entries look like $\dfrac{\hat{\psi}(m)\sin a - |m|}{\hat{\psi}(m)\sin a + |m|}$ and one may heuristically use the same operators as in the two-subdomain case.

4 Numerical results

We have written a semispectral solver for the various transmission operators we have described and the numerical results are summarized in figure 2:

(a) We have computed 18 iterates of the Schwarz iteration and plotted the error at each even iteration to match with the analysis in the text. The transmission operators are Robin, second order tangential with coefficients (c,d) (dash-dot), second order tangential with coefficients (c',d') (dashed) and a discretized optimal operator (solid.) The slopes are the contraction constants. The bump at step 2 is because $\mathcal{L}u_0 \neq 0$.

(b) The decay of the contraction constant as the number of subdomains increases. The x axis is the number of subdomains and the y axis is the contraction constant. The x marks and diamonds are for the Robin and optimal operators, respectively, and the circles and squares are for the choices (c,d) and (c',d') of second order tangential operators. The truncation frequency is $N = 50$ in all cases; there are 101 points along the equator.

(c) Depiction of the behavior of the contraction-every-two-steps constant as we increase the discretization parameter N, two subdomains, no overlap. The number of points along the equator is $2N + 1$. The line with x marks is a Robin algorithm, the line with circles is with the second order operator and the diamonds is the optimal operator. The two circled lines are for the two choices (c,d) and (c',d') (slightly better) of the second order transmission parameters. Dotted lines are predictions from our analysis. The optimal operator does not lead to convergence in two steps due to the discretization.

(d) Same as (c), but with a single grid length of overlap. Since we have overlap, we include the Dirichlet operator as the * line. The optimal transmission operator behaved vastly better in the overlap case (exhibiting apparently superlinear convergence.)

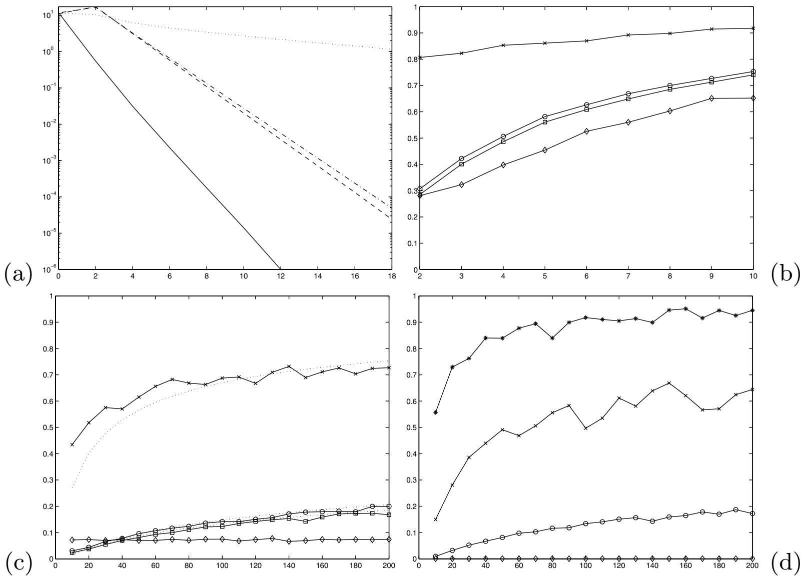


Fig. 2. (a): iterates of the various Schwarz algorithms (two subdomains, no overlap, semispectral code.) (b): contraction constants as a function of the number of subdomains (no overlap.) (c), (d): contraction constants as a function of the truncation frequency (two subdomains.) (c) is without overlap, (d) is one grid interval of overlap.

5 Conclusions

We have given optimal and optimized transmission operators for the Laplace problem on the sphere and have shown that they perform much better than the classical iteration with a Dirichlet condition. We have computed convergence rates for the Robin condition and two choices of second-order tangential operators, and compared them against the optimal nonlocal operator. A similar analysis for the positive definite Helmholtz problem will be detailed in a later paper.

References

1. J. Côté, M. J. Gander, L. Laayouni, and S. Loisel, *Comparison of the Dirichlet-Neumann and optimal Schwarz method on the sphere*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 235–242.

2. J. Côté and A. Staniforth, *An accurate and efficient finite-element global model of the shallow-water equations*, Monthly Weather Review, 118 (1990), pp. 2707–2717.
3. O. Dubois, *Optimized Schwarz methods for the advection-diffusion equation*, Master's thesis, McGill University, 2003.
4. M. J. Gander and G. H. Golub, *A non-overlapping optimized Schwarz method which converges with arbitrarily weak dependence on h*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003.
5. M. J. Gander, L. Halpern, and F. Nataf, *Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation*, in Eleventh international Conference of Domain Decomposition Methods, C.-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, eds., ddm.org, 1999.
6. ———, *Optimized Schwarz methods*, in 12th International Conference on Domain Decomposition Methods, T. Chan, T. Kako, H. Kawarada, and O. Pironneau, eds., ddm.org, 2001.
7. P.-L. Lions, *On the Schwarz alternating method. I.*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM, pp. 1–42.
8. ———, *On the Schwarz alternating method. II.*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM, pp. 47–70.
9. ———, *On the Schwarz alternating method. III: a variant for nonoverlapping subdomains*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989, T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM.
10. S. Loisel, *Optimal and optimized domain decomposition methods on the sphere*, PhD thesis, McGill University, 2005.
11. F. Nier, *Remarques sur les algorithmes de décomposition de domaines*, in Seminaire: Équations aux Dérivées Partielles, 1998–1999, École Polytech., 1999, pp. Exp. No. IX, 26.
12. A. Staniforth and J. Côté, *Semi-Lagrangian integration schemes for atmospheric models – a review*, Monthly Weather Review, 119 (1991), pp. 2206–2223.

Additive Schwarz Method for Scattering Problems Using the PML Method at Interfaces

Achim Schädle[1] and Lin Zschiedrich[2]

[1] Zuse Institute, Takustr. 7, 14195 Berlin, Germany. schaedle@zib.de
[2] Zuse Institute, Takustr. 7, 14195 Berlin, Germany. zschiedrich@zib.de

1 Introduction

The exterior Helmholtz problem is a basic model for wave propagation in the frequency domain on unbounded domains. As a rule of thumb, 10-20 grid points per wavelength are required. Hence if the modeling structures are a multiple of wavelengths in size, a discretization with finite elements results in large sparse indefinite and unsymmetric problems. There are no well established solvers, or preconditioners for these linear systems as there are for positive definite elliptic problems.

As a first basic step towards a solver for the class of linear systems described above we consider a non-overlapping Schwarz algorithm with only two subdomains, where the coupling among subdomains is done using the perfectly matched layer method.

We do not present a new idea here, and it is beyond the scope of the paper to do justice to previous work in this field. However we comment on a few references, that have been inspiring to us.

In [10] Toselli tried to use the Schwarz algorithm with perfectly matched layers (PML) at the interfaces, as a preconditioner. However we believe that the coupling of the incoming waves there, was done incorrectly; we comment on this in the concluding remark in Section 4. One may view the Ansatz by Després, see [1] and the references therein, and Shaidurov and Ogorodnikov [9], as a first order absorbing boundary condition. The use of Robin boundary conditions there is also motivated by the idea of equating energy fluxes over boundaries. Colino, Joly and Ghanemi [2] analyzed the Ansatz of Després and could prove convergence. Gander, Nataf and Magoulés [4] follow a slightly different Ansatz. They use local low order boundary conditions, that optimize transmission, based on an analysis of Fourier coefficients.

The PML method is in special cases one of the best approximations to the Dirichlet to Neumann (DtN) operator. With the DtN operator at hand the Schwarz algorithm would converge in a finite number of iteration steps.

2 Problem description

We consider time-harmonic electro-magnetic scattering problems in two space dimensions. Assuming that the electric field is polarized in the x, y-plane and that the obstacle is homogeneous in the z direction, the time-harmonic vectorial Maxwell's equations in 3D are reduced to equations in 2D. For the z component of the magnetic field we obtain the Helmholtz equation (1)

$$\nabla \cdot \epsilon^{-1} \nabla u + \omega^2 \mu u = 0 \text{ in } \tilde{\Omega}; \quad b(u, \partial_\nu u) = 0 \text{ on } \Gamma \tag{1}$$

Here ω is the frequency and μ and ϵ are the x, y-dependent relative permeability and conductivity respectively. $\tilde{\Omega}$ is typically the complement of a bounded set in $\mathbb{R}^2$, with boundary Γ, where the boundary condition b is given. The boundary condition b, if there is an interior boundary at all, is typically of the form $b(u, \partial_\nu u) = u$, $b(u, \partial_\nu u) = \partial_\nu u$ or $b(u, \partial_\nu u) = \partial_\nu u + cu$. The Helmholtz equation has to be completed by the Sommerfeld radiation boundary condition for the scattered field.

For simplicity we assume that $\epsilon = 1$, and set $k^2 = \omega^2 \mu$. The total field u can be written as the sum of the known incoming and the scattered field $u = u_{in} + u_{sc}$. The scattered field is a solution of (1) and satisfies the Sommerfeld radiation boundary conditions for $|(x, y)| \to \infty$ given by:

$$\lim_{|(x,y)| \to \infty} \partial_\nu u_{sc} = iku_{sc} \ , \tag{2}$$

where the limit is understood uniformly for all directions.

3 Coupling of incoming waves - DtN operator

The computation will be restricted to a bounded computational domain Ω. It is assumed that outside the computational domain ϵ and μ are constant along straight lines. In this case we can evaluate the Dirichlet to Neumann (DtN) operator using the perfectly matched layer method (PML) developed in [12].

Next we reformulate Problem (1) on the computational domain. This clearly shows how to couple incoming fields to the computational domain.

Setting $u = v \oplus w$ according to the decomposition $\tilde{\Omega} = \Omega \cup \Omega_{ext}$ we obtain the coupled system

$$\begin{aligned} \Delta v + k^2 v = 0 \ \text{ in } \Omega\,; \quad b(v, \partial_\nu v) = 0 \text{ on } \Gamma \cap \Omega \\ \partial_\nu v = \partial_\nu u_{in} + \partial_\nu w_{sc} \ \text{ on } \Gamma_{int} \end{aligned} \tag{3}$$

$$\begin{aligned} \Delta w_{sc} + k^2 w_{sc} = 0 \ \text{ in } \Omega_{ext}\,; \\ w_{sc} = v - u_{in} \ \text{ on } \Gamma_{int}\,; \ b(w_{sc} + u_{in}, \partial_\nu (w_{sc} + u_{in})) = 0 \text{ on } \Gamma \cap \Omega_{ext} \\ \lim_{|x| \to \infty} \partial_\nu w_{sc} - ikw_{sc} = 0 \end{aligned} \tag{4}$$

where the coupling is via the Dirichlet and Neumann data on the interface boundary Γ_{int}, connecting Ω and Ω_{ext}. From this we obtain the DtN operator, which is the operator that solves the exterior problem with given Dirichlet data Γ_{int} and returns the Neumann data. With the DtN-operator at hand one gets

$$\begin{aligned} \Delta v + k^2 v = 0 \ \text{ in } \Omega\,; \quad b(v, \partial_\nu v) = 0 \text{ on } \Gamma \cap \Omega \\ \partial_\nu v - \partial_\nu u_{in} = \mathrm{DtN}(v - u_{in}) \ \text{ on } \Gamma_{int}. \end{aligned} \tag{5}$$

In general the DtN operator is difficult to compute, but can be approximated using the PML, described briefly in Section 4. For more information on approximating the DtN operator, see the textbook [5], and the more recent review articles [11, 6].

4 Sketch of the perfectly matched layer method

We do not follow, the classical introduction of the perfectly matched layer method (PML) that is motivated by adding a layer of artifical absorbing material.

Our derivation of the PML method, described in detail in [7] is based on an analytic continuation, as in [8, 3]. Details of the implementation in 2D can by found in [12]. The basic idea is an analytic continuation of the solution in the exterior along a distance variable. We will only sketch the ideas here for the one-dimensional case.

Consider the Helmholtz equation in 1D on a semi-infinite interval for the scattered field.

$$\begin{aligned} \partial_{xx} u + k^2 u = 0 \quad x \in [-1, \infty) \\ u(-1) = 1\,; \quad \partial_\nu u = iku \text{ for } x \to \infty \end{aligned} \tag{6}$$

Our computational domain is the interval $[-1, 0]$. The solution in the exterior is analytic in x. Defining $\gamma(x) := (1+i\sigma)x$ and $\tilde{u}_{\text{PML}}(x) := u(\gamma(x))$, we have $\tilde{u}_{\text{PML}}(0) = u(0)$ and $\partial_\nu u(0) = \partial_\nu \tilde{u}_{\text{PML}}(0)/(1 + i\sigma)$. u_{PML} obeys

$$\begin{aligned} \partial_{xx} \tilde{u}_{\text{PML}} + k^2 (1 + i\sigma)^2 \tilde{u}_{\text{PML}} = 0 \quad x \in [0, \infty) \\ \tilde{u}_{\text{PML}}(0) = u(0)\,; \quad \partial_x \tilde{u}_{\text{PML}}(x) = ik\tilde{u}_{\text{PML}}(x)(1 + i\sigma) \end{aligned} \tag{7}$$

Fundamental solutions are $\exp(ik(1 + i\sigma)x)$ and $\exp(-ik(1 + i\sigma)x)$. The first one is called outgoing as it obeys the boundary condition, the second is called incoming as it does not. The first one decays exponentially, whereas the second grows exponentially; therefore it can be justified to replace $\tilde{u}_{\text{PML}}$ by u_{PML} given by Equation (9), and replace the infinite coupled system by the coupled system

$$\begin{aligned} \partial_{xx} v + k^2 v = 0 \quad x \in [-1, 0] \\ v(-1) = 1\,; \quad \partial_\nu v(0) = \partial_\nu u_{\text{PML}}(0)/(1 + i\sigma) \end{aligned} \tag{8}$$

$$\begin{aligned} \partial_{xx} u_{\text{PML}} + k^2 (1 + i\sigma)^2 u_{\text{PML}} = 0 \quad x \in [0, \rho] \\ u_{\text{PML}}(0) = v(0)\,; \quad \partial_x u_{\text{PML}}(\rho) = 0 \end{aligned} \tag{9}$$

Here ρ is the thickness of the PML. The error introduced by truncating the PML is analyzed in, e.g. [8, 7], where it is shown that the PML system is well-posed and the error decays exponentially with ρ.

Remark: Toselli [10] coupled the incoming field at the external boundary of the PML; this way the incoming field is damped in the PML and this might explain, why he concluded that it is best to use a very thin layer.

5 Two-domain decomposition

We now turn back to the two dimensional case. The idea for the Schwarz algorithm is to calculate the solution in every subdomain separately with transparent boundary conditions at the subdomain interfaces and add the scattered field of one domain to the incoming field for the neighboring domains, i.e. use a pseudo-DtN operator,

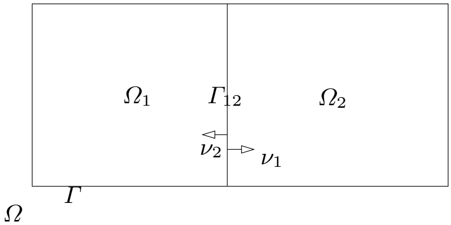


Fig. 1. Decomposition of Ω into two non-overlapping subdomains Ω_1 and Ω_2

where we assume that the exterior to each subdomain has a simple structure. If we are able to evaluate *the* DtN operator the Schwarz algorithm would converge in a finite number of steps.

For the simple two subdomain case the additive Schwarz algorithm is given in (10). Here u_j^n denotes the nth iterate on subdomain Ω_j, and Γ_{ij} the boundary between Ω_i and Ω_j.

$$\begin{aligned} \Delta u_j^{n+1} + k^2 u_j^{n+1} &= 0 \text{ in } \Omega_j \\ \partial_\nu u_j^{n+1} &= \mathrm{DtN}(u_j^{n+1} - u^{in}) + \partial_\nu u^{in} \text{ on } \bar{\Omega}_j \cap \Gamma \\ \partial_\nu u_j^{n+1} &= \mathrm{DtN}(u_j^{n+1} - u_i^n) + \partial_\nu u_i^n \text{ on } \Gamma_{ij} \end{aligned} \tag{10}$$

for $(i, j) = (1, 2), (2, 1)$, $n = 0, 1, \ldots$.

Denoting by ν_j the normal with respect to Ω_j we have $\partial_{\nu_j} u_i^n = -\partial_{\nu_i} u_i^n$.

We make the following assumptions: The subdomains are strips with homogenous Neumann, Dirichlet, or periodic boundary condition at non-interface boundaries, with transparent boundary condition at interfaces and are ordered linearly. This way we avoid crosspoints, which pose a problem. The incoming field is given on two neighboring domains with a common boundary, hence the incoming field may have a jump across this boundary and at the crosspoint, and is hence not a solution of the Helmholtz equation. This is also a problem from the computational point of view, as the Dirichlet data inserted in the DtN operator is assumed to be continuous. One idea to circumvent this difficulty is to add artificial outgoing waves, that compensate for the jump. Another one is to use a representation formula based on the Pole condition for the scattered field and evaluate it on the interface boundaries, but this is outside the scope of the present paper.

We assume that the boundary condition is a homogenous Neumann condition, i.e. $b(u, \partial_\nu u) = \partial_\nu u$, and set

$$a_\Omega(u, \varphi) = -\int_\Omega \nabla u \nabla \varphi + k^2 u \varphi \, dx \tag{11}$$

With this in the variational setting the solution u is the function $u \in H^1(\Omega)$ such that

$$a_\Omega(u, \varphi) + \int_{\Gamma_{int}} \partial_\nu u \varphi d\sigma(x) = 0 \quad \forall \varphi \in H^1(\Omega)$$

Inserting the boundary condition, we obtain

$$a_\Omega(u,\varphi) + \int_{\Gamma_{int}} \mathrm{DtN}(u - u^{in})\varphi + \partial_\nu u^{in} \varphi d\sigma(x) = 0 \quad \forall \varphi \in H^1(\Omega)$$

The Schwarz algorithm in variational form is given in (12) below. To avoid the evaluation of the Neumann data on the interface boundary we use a postprocessing step (13), so that the Neumann data is only given in weak form.

$$\begin{aligned} a_j(u_j^{n+1},\varphi) + &\underbrace{\int_{\Gamma_{ij}} \mathrm{DtN}(u_j^{n+1} - u_i^n)\varphi d\sigma(x) + \int_{\Gamma_{ij}} \partial_{\nu_j} u_j^n \varphi d\sigma(x)}_{\int_{\Gamma_{ij}} \partial_{\nu_1} u_1^{n+1} \varphi d\sigma(x)} \\ &+ \underbrace{\int_{\Gamma\cap\bar{\Omega}_j} \mathrm{DtN}(u_j^{n+1} - u^{in})\varphi + \partial_{\nu_1} u^{in} \varphi d\sigma(x)}_{\int_{\Gamma\cap\bar{\Omega}_j} \partial_{\nu_j} u_j^{n+1} \varphi d\sigma(x)} = 0 \quad \forall \varphi \in H^1(\Omega_1) \end{aligned} \tag{12}$$

$$\begin{aligned} \int_{\Gamma_{ij}} \partial_{\nu_j} u_j^{n+1} \varphi d\sigma(x) = &- a_j(u_j^{n+1},\varphi) \\ &- \int_{\Gamma\cap\bar{\Omega}_j} \mathrm{DtN}(u_j^{n+1} - u^{in})\varphi + \partial_{\nu_1} u^{in} \varphi d\sigma(x) \end{aligned} \tag{13}$$

6 Numerical experiments

We consider a very simple example. The computational domain is a $[-1,1]\times[0.5,0.5]$ rectangle, with periodic and transparent boundary conditions. To be precise, in Fig. 1 we take periodic boundary conditions at the top and bottom of Ω and transparent boundary conditions to the left and the right. The incoming field is a plane wave traveling from left to right. The computational domain is split in two squares along the y-axis. The function k depends on x and y and is a step function, k equals k_0 everywhere, except in two smaller squares of size $[0,0.5]\times[0,0.5]$ located in the center of the two subdomain, where it is $k_0/5$.

The calculation was done using the package *JCMfeetools* developed at the ZIB, with second order finite elements. The linear systems are solved using the sparse solver UMFPACK.

The thickness of the PML ρ is set to three wavelengths, the damping factor to $\sigma = 1$ and along the distance variable, we have chosen 12 grid-points on the coarse grid. The coarse grid including the PML has about 1100 unknowns on each subdomain. We plot the l_2 error versus number of Schwarz iteration steps for different ω for upto four uniform refinements of the initial grid. To this end the error is calculated with respect to a reference solution calculated on the whole domain with the same mesh on each subdomain. This is done for two settings. First for the algorithm described above, with the representation of the Neumann data in weak form and second evaluating the normal derivatives, via the gradient of the Ansatz function in the neighboring domain.

When we use the weak representation of the Neumann data, we obtain a convergent algorithm. The convergence rate depends strongly on the wavelength but only weakly on the discretization as can be seen in Fig 2.

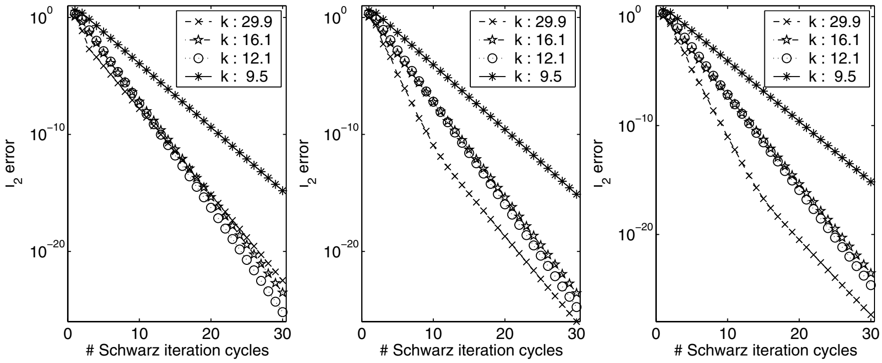


Fig. 2. Error of the Schwarz algorithm, for different wavenumbers k and different refinement levels using weak representation of the Neumann data. The left plot was calculated using 1168 unknowns, the middle one with 4000 unknowns and the right with 13120 unknowns.

In case we evaluate the Neumann data via the gradient of the Ansatz function the error of the domain decomposition method saturates as shown in the left and middle graph in Fig 3.

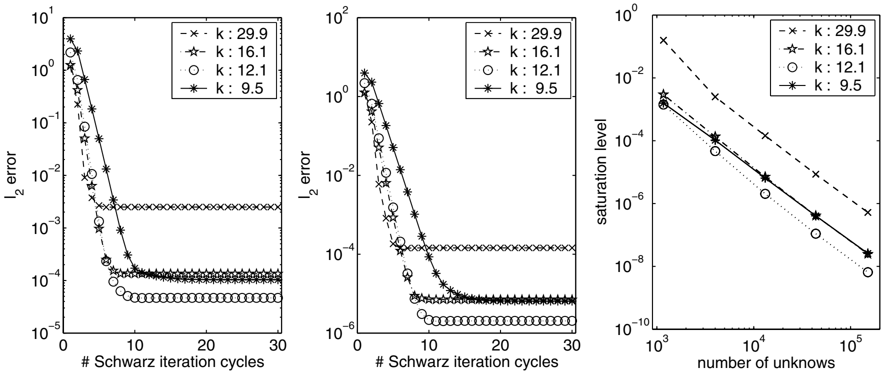


Fig. 3. (Left and middle): Error of the Schwarz algorithm, for different wavenumbers k and different refinement levels. The left plot was calculated using 4000 unknowns the middle one with 13120 unknowns in each subdomain. (Right): Decay of the level at which the error saturates, versus the number on unknowns.

Surprisingly, the level at which the error saturates, plotted in the rightmost graph of Fig 3 versus the number of unknowns, decays faster than might be expected, from the error estimate for the Neumann data. Recall that we use second order finite elements here.

Acknowledgment

We thank Ralf Kornhuber, Susanne Ertel and the members of the computational nano-optics group for fruitful discussions and give special thanks to Martin Gander and Ralf Hiptmair. The first author was supported by the DFG Research Center MATHEON "Mathematics for key technologies", Berlin. The second author was supported by the German Federal Ministry of Education and Research BMBF project under contract no 13N8252 (HiPhocs).

References

1. J.-D. BENAMOU AND B. DESPRÉS, *A domain decomposition method for the Helmholtz equation and related optimal control*, J. Comp. Phys., 136 (1997), pp. 68–82.
2. F. COLLINO, S. GHANEMI, AND P. JOLY, *Domain decomposition method for harmonic wave propagation: A general presentation*, Comput. Methods Appl. Mech. Eng., 184 (2000), pp. 171–211.
3. F. COLLINO AND P. MONK, *The perfectly matched layer in curvilinear coordinates*, SIAM J. Sci. Comput., 19 (1998), pp. 2061–2090.
4. M. J. GANDER, F. MAGOULÈS, AND F. NATAF, *Optimized Schwarz methods without overlap for the Helmholtz equation*, SIAM J. Sci. Comput., 24 (2002), pp. 38–60.
5. D. GIVOLI, *Non-reflecting boundary conditions*, J. Comput. Phys., 94 (1991), pp. 1–29.
6. T. HAGSTROM, *Radiation boundary conditions for numerical simulation of waves*, Acta Numerica, 8 (1999), pp. 47–106.
7. T. HOHAGE, F. SCHMIDT, AND L. ZSCHIEDRICH, *Solving time-harmonic scattering problems based on the pole condition II: Convergence of the PML method*, SIAM J. Math. Anal., 35 (2003), pp. 547–560.
8. M. LASSAS AND E. SOMERSALO, *On the existence and convergence of the solution of PML equations*, Computing, 60 (1998), pp. 229–241.
9. V. V. SHAIDUROV AND E. I. OGORODNIKOV, *Some numerical method of solving Helmholtz wave equation*, in Mathematical and numerical aspects of wave propagation phenomena, G. Cohen, L. Halpern, and P. Joly, eds., SIAM, 1991, pp. 73–79.
10. A. TOSELLI, *Some results on overlapping Schwarz methods for the Helmholtz equation employing perfectly matched layers*, Tech. Rep. 765, Courant Institute of Mathematical Sciences, New York University, New York, June 1998.
11. S. TSYNKOV, *Numerical solution of problems on unbounded domains. a review*, Appl. Numer. Math., 27 (1998), pp. 465–532.

12. L. Zschiedrich, R. Klose, A. Schdle, and F. Schmidt, *A new finite element realization of the perfectly matched layer method for helmholtz scattering problems on polygonal domains in 2D*, Tech. Rep. 03-44, Konrad-Zuse-Zentrum fur Informationstechnik Berlin, December 2003.

Optimized Restricted Additive Schwarz Methods

Amik St-Cyr[1], Martin J. Gander[2] and Stephen J. Thomas[3]

[1] National Center for Atmospheric Research, 1850 Table Mesa Drive, Boulder, CO 80305, USA. amik@ucar.edu
[2] University of Geneva, Switzerland. martin.gander@math.unige.ch
[3] National Center for Atmospheric Research, 1850 Table Mesa Drive, Boulder, CO 80305, USA. thomas@ucar.edu

Summary. A small modification of the restricted additive Schwarz (RAS) preconditioner at the algebraic level, motivated by continuous optimized Schwarz methods, leads to a greatly improved convergence rate of the iterative solver. The modification is only at the level of the subdomain matrices, and hence easy to do in an existing RAS implementation. Numerical experiments using finite difference and spectral element discretizations of the modified Helmholtz problem $u - \Delta u = f$ illustrate the effectiveness of the new approach.

1 Schwarz Methods at the Algebraic Level

The discretization of an elliptic partial differential equation

$$\mathcal{L}u = f \quad \text{in } \Omega, \quad \mathcal{B}u = g \quad \text{on } \partial\Omega, \tag{1}$$

where $\mathcal{L}$ is an elliptic differential operator, $\mathcal{B}$ is a boundary operator and Ω is a bounded domain, leads to a linear system of equations

$$A\mathbf{u} = \mathbf{f}. \tag{2}$$

A stationary iterative method for (2) is given by

$$\mathbf{u}^{n+1} = \mathbf{u}^n + M^{-1}(\mathbf{f} - A\mathbf{u}^n). \tag{3}$$

An initial guess $\mathbf{u}^0$ is required to start the iteration. Algebraic domain decomposition methods group the unknowns into subsets, $\mathbf{u}_j = R_j\mathbf{u}$, $j = 1, \ldots, J$, where R_j are rectangular matrices. Classical coefficient matrices for subdomain problems are defined by $A_j = R_j A R_j^T$. The additive Schwarz (AS) preconditioner [2], and the restricted additive Schwarz (RAS) preconditioner [1]) are defined by

$$M_{AS}^{-1} = \sum_{j=1}^{J} R_j^T A_j^{-1} R_j, \quad M_{RAS}^{-1} = \sum_{j=1}^{J} \tilde{R}_j^T A_j^{-1} R_j, \tag{4}$$

where the $\tilde{R}_j$ correspond to a non-overlapping decomposition, i.e. each entry u_l of the vector $\mathbf{u}$ occurs in $\tilde{R}_j\mathbf{u}$ for exactly one j.

The algebraic formulation of Schwarz methods has an important feature: a subdomain matrix A_j is not necessarily the restriction of A to a subdomain j. For example, if A represents a spectral element discretization of a differential operator, then A_j can be obtained from a finite element discretization at the collocation points. Furthermore, subdomain matrices A_j can be chosen to accelerate convergence and this is the focus of the next section.

2 Optimized Restricted Additive Schwarz Methods

Historically, domain decomposition methods were formulated at the continuous level. We consider a decomposition of the original domain Ω in (1) into two overlapping sub-domains Ω_1 and Ω_2, and we denote the interfaces by $\Gamma_{ij} = \partial\Omega_i \cap \Omega_j$, $i \neq j$, and the outer boundaries by $\partial\Omega_j = \partial\Omega \cap \bar{\Omega}_j$. In [5], a parallel Jacobi variant of the classical alternating Schwarz method was introduced for (1),

$$\begin{aligned} \mathcal{L}u_1^{n+1} &= f \quad \text{in } \Omega_1, & \mathcal{L}u_2^{n+1} &= f \quad \text{in } \Omega_2,\\ \mathcal{B}(u_1^{n+1}) &= g \quad \text{on } \partial\Omega_1, & \mathcal{B}(u_2^{n+1}) &= g \quad \text{on } \partial\Omega_2,\\ u_1^{n+1} &= u_2^n \text{ on } \Gamma_{12}, & u_2^{n+1} &= u_1^n \text{ on } \Gamma_{21}. \end{aligned} \tag{5}$$

It was shown in [3] that the discrete form of (5), namely

$$A_1\mathbf{u}_1^{n+1} = \mathbf{f}_1 + B_1\mathbf{u}_2^n, \quad A_2\mathbf{u}_2^{n+1} = \mathbf{f}_2 + B_2\mathbf{u}_1^n, \tag{6}$$

is equivalent to RAS in (4). In optimized algorithms, the Dirichlet transmission conditions in (5) are replaced by more effective transmission conditions, which corresponds to replacing the subdomain matrices A_j in (6) by $\tilde{A}_j$ and the transmission matrices B_j by $\tilde{B}_j$, corresponding to optimized transmission conditions, and leads to

$$\tilde{A}_1\mathbf{u}_1^{n+1} = \mathbf{f}_1 + \tilde{B}_1\mathbf{u}_2^n, \quad \tilde{A}_2\mathbf{u}_2^{n+1} = \mathbf{f}_2 + \tilde{B}_2\mathbf{u}_1^n, \tag{7}$$

see Sections 3 and 4 for how to choose $\tilde{A}_j$.

We now shown that, for sufficient overlap, the subdomain matrices A_j in the RAS algorithm (4) can be replaced by the optimized subdomain matrices $\tilde{A}_j$ from (7), to obtain an optimized RAS method (ORAS) equivalent to (7),

$$u^{n+1} = u^n + (\sum_{j=1}^{2} \tilde{R}_j^T \tilde{A}_j^{-1} R_j)(f - Au^n). \tag{8}$$

The additional interface matrices $\tilde{B}_j$ in (7) are not needed in the optimized RAS method (8), which greatly simplifies the transition from RAS to ORAS.

Definition 1 (Consistency). *Let R_j, $j = 1, 2$ be restriction matrices covering the entire discrete domain, and let $\mathbf{f}_j := R_j\mathbf{f}$. We call the matrix splitting R_j, $\tilde{A}_j$, $\tilde{B}_j$, $j = 1, 2$ in (7) consistent, if for all $\mathbf{f}$ and associated solution $\mathbf{u}$ of (2), $\mathbf{u}_1 = R_1\mathbf{u}$ and $\mathbf{u}_2 = R_2\mathbf{u}$ satisfy*

$$\tilde{A}_1\mathbf{u}_1 = \mathbf{f}_1 + \tilde{B}_1\mathbf{u}_2, \quad \tilde{A}_2\mathbf{u}_2 = \mathbf{f}_2 + \tilde{B}_2\mathbf{u}_1. \tag{9}$$

Lemma 1. *Let A in (2) have full rank. For a consistent matrix splitting R_j, $\tilde{A}_j$, $\tilde{B}_j$, $j = 1, 2$, we have the matrix identities*

$$\tilde{A}_1 R_1 - \tilde{B}_1 R_2 = R_1 A, \quad \tilde{A}_2 R_2 - \tilde{B}_2 R_1 = R_2 A. \tag{10}$$

Proof. We only prove the first identity, the second follows analogously. For an arbitrary $\mathbf{f}$, we apply R_1 to equation (2), and obtain, using consistency (9),

$$R_1 A\mathbf{u} = R_1 \mathbf{f} = \mathbf{f}_1 = \tilde{A}_1 \mathbf{u}_1 - \tilde{B}_1 \mathbf{u}_2.$$

Now using $\mathbf{u}_1 = R_1 \mathbf{u}$ and $\mathbf{u}_2 = R_2 \mathbf{u}$ on the right-hand side yields

$$(\tilde{A}_1 R_1 - \tilde{B}_1 R_2 - R_1 A)\mathbf{u} = 0.$$

Because $\mathbf{f}$ was arbitrary, the identity is true for all $\mathbf{u}$ and therefore the first identity in (10) is established.

While the definition of consistency is simple, it has important consequences: if the classical submatrices are used, i.e. $\tilde{A}_j = A_j = R_j A R_j^T$, $j = 1, 2$, then the restriction matrices R_j can be overlapping or non-overlapping, and with the associated B_j, we obtain a consistent splitting R_j, A_j, B_j, $j = 1, 2$. If however other subdomain matrices $\tilde{A}_j$ are employed, then the restriction matrices R_j must be such that the unknowns in u_1 affected by the change in $\tilde{A}_1$ are also available in u_2 to compensate via $\tilde{B}_1$ in equation (9), and similarly for u_2. Hence consistency implies for all non-classical splittings a condition on the overlap in the R_j in RAS. A strictly non-overlapping variant can be obtained when applying standard AS with non-overlapping R_j to the augmented system obtained from (7) at convergence,

$$\begin{bmatrix} \tilde{A}_1 & -\tilde{B}_1 \\ -\tilde{B}_2 & \tilde{A}_2 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \end{bmatrix}, \tag{11}$$

see the non-overlapping spectral element experiments in Section 4 and [9]. For optimized RAS, a further restriction on the overlap is necessary:

Lemma 2. *Let R_j, $j = 1, 2$, be restriction matrices covering the entire discrete domain, and let $\tilde{R}_j$ be the corresponding RAS versions of these matrices. If $\tilde{B}_1 R_2 \tilde{R}_1^T = 0$, then $\tilde{B}_1 R_2 \tilde{R}_2^T = \tilde{B}_1$, and if $\tilde{B}_2 R_1 \tilde{R}_2^T = 0$, then $\tilde{B}_2 R_1 \tilde{R}_1^T = \tilde{B}_2$.*

Proof. We first note that by the non-overlapping definition of $\tilde{R}_j$, $j = 1, 2$, the identity matrix I can be written as

$$I = \tilde{R}_1^T \tilde{R}_1 + \tilde{R}_2^T \tilde{R}_2. \tag{12}$$

Now multiplying $\tilde{B}_1 R_2 \tilde{R}_1^T = 0$ on the right by R_1 and substituting the term $\tilde{R}_1^T R_1$ using (12) leads to

$$(\tilde{B}_1 - \tilde{B}_1 R_2 \tilde{R}_2^T) R_2 = 0,$$

which completes the proof, since the fat restriction matrix R_2 has full rank. The second result follows analogously.

Theorem 1. *Let R_j, $\tilde{A}_j$, $\tilde{B}_j$, $j = 1, 2$ be a consistent matrix splitting, and let $\tilde{R}_j$ be the corresponding RAS versions of R_j. If the initial iterates $\mathbf{u}_j^0$, $j = 1, 2$, of the optimized Schwarz method (7) and the initial iterate $\mathbf{u}^0$ of the optimized RAS method (8) satisfy*

$$\mathbf{u}^0 = \tilde{R}_1^T \mathbf{u}_1^0 + \tilde{R}_2^T \mathbf{u}_2^0, \tag{13}$$

and if the overlap condition

$$\tilde{B}_1 R_2 \tilde{R}_1^T = 0, \quad \tilde{B}_2 R_1 \tilde{R}_2^T = 0 \tag{14}$$

is satisfied, then the two methods (7) and (8) generate an equivalent sequence of iterates,

$$\mathbf{u}^n = \tilde{R}_1^T \mathbf{u}_1^n + \tilde{R}_2^T \mathbf{u}_2^n. \tag{15}$$

Proof. The proof is by induction. For $n = 0$, we have (15) by assumption (13) on the initial iterates. We now assume that $\mathbf{u}^n = \tilde{R}_1^T \mathbf{u}_1^n + \tilde{R}_2^T \mathbf{u}_2^n$, and show that the identity (15) holds for $n+1$. Applying Lemma 1 to the first term of the sum in (8), we obtain

$$\begin{aligned}\tilde{R}_1^T \tilde{A}_1^{-1} R_1(\mathbf{f} - A\mathbf{u}^n) &= \tilde{R}_1^T \tilde{A}_1^{-1}(\mathbf{f}_1 - R_1 A\mathbf{u}^n) \\ &= \tilde{R}_1^T \tilde{A}_1^{-1}(\mathbf{f}_1 - (\tilde{A}_1 R_1 - \tilde{B}_1 R_2)\mathbf{u}^n) \\ &= \tilde{R}_1^T (\tilde{A}_1^{-1}\mathbf{f}_1 - R_1\mathbf{u}^n + \tilde{A}_1^{-1}\tilde{B}_1 R_2 \mathbf{u}^n),\end{aligned} \tag{16}$$

and similarly for the second term of the sum,

$$\tilde{R}_2^T \tilde{A}_2^{-1} R_2(\mathbf{f} - A\mathbf{u}^n) = \tilde{R}_2^T (\tilde{A}_2^{-1}\mathbf{f}_2 - R_2\mathbf{u}^n + \tilde{A}_2^{-1}\tilde{B}_2 R_1 \mathbf{u}^n). \tag{17}$$

Substituting these two expressions into (8), and using (12) leads to

$$\mathbf{u}^{n+1} = \tilde{R}_1^T (\tilde{A}_1^{-1}(\mathbf{f}_1 + \tilde{B}_1 R_2 \mathbf{u}^n)) + \tilde{R}_2^T (\tilde{A}_2^{-1}(\mathbf{f}_2 + \tilde{B}_2 R_1 \mathbf{u}^n)).$$

Now replacing by induction hypothesis $\mathbf{u}^n$ by $\tilde{R}_1^T \mathbf{u}_1^n + \tilde{R}_2^T \mathbf{u}_2^n$ on the right hand side and applying Lemma 2, we find together with (14)

$$\begin{aligned}\mathbf{u}^{n+1} &= \tilde{R}_1^T (\tilde{A}_1^{-1}(\mathbf{f}_1 + \tilde{B}_1 R_2(\tilde{R}_1^T \mathbf{u}_1^n + \tilde{R}_2^T \mathbf{u}_2^n))) \\ &\quad + \tilde{R}_2^T (\tilde{A}_2^{-1}(\mathbf{f}_2 + \tilde{B}_2 R_1(\tilde{R}_1^T \mathbf{u}_1^n + \tilde{R}_2^T \mathbf{u}_2^n))) \\ &= \tilde{R}_1^T (\tilde{A}_1^{-1}(\mathbf{f}_1 + \tilde{B}_1 \mathbf{u}_2^n)) + \tilde{R}_2^T (\tilde{A}_2^{-1}(\mathbf{f}_2 + \tilde{B}_2 \mathbf{u}_1^n)),\end{aligned}$$

which together with (7) implies $\mathbf{u}^{n+1} = \tilde{R}_1^T \mathbf{u}_1^{n+1} + \tilde{R}_2^T \mathbf{u}_2^{n+1}$.

3 The Schur Complement as Optimal Choice for $\tilde{A}_j$

We show now algebraically what the best choice of $\tilde{A}_j$ is: we partition A from (2) into two blocks with a common interface,

$$A\mathbf{u} = \begin{bmatrix} A_{1i} & C_1 & \\ B_2 & A_\Gamma & B_1 \\ & C_2 & A_{2i} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1i} \\ \mathbf{u}_\Gamma \\ \mathbf{u}_{2i} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{1i} \\ \mathbf{f}_\Gamma \\ \mathbf{f}_{2i} \end{bmatrix},$$

where $\mathbf{u}_{1i}$ and $\mathbf{u}_{2i}$ correspond to the interior unknowns and $\mathbf{u}_\Gamma$ corresponds to the interface unknowns. The classical Schwarz subdomain matrices are in this case

$$A_1 = \begin{bmatrix} A_{1i} & C_1 \\ B_2 & A_\Gamma \end{bmatrix}, \quad A_2 = \begin{bmatrix} A_\Gamma & B_1 \\ C_2 & A_{2i} \end{bmatrix},$$

and the subdomain solution vectors and the right hand side vectors are

$$\mathbf{u}_1 = \begin{bmatrix} \mathbf{u}_{1i} \\ \mathbf{u}_\Gamma \end{bmatrix}, \ \mathbf{u}_2 = \begin{bmatrix} \mathbf{u}_\Gamma \\ \mathbf{u}_{2i} \end{bmatrix}, \quad \mathbf{f}_1 = \begin{bmatrix} \mathbf{f}_{1i} \\ \mathbf{f}_\Gamma \end{bmatrix}, \ \mathbf{f}_2 = \begin{bmatrix} \mathbf{f}_\Gamma \\ \mathbf{f}_{2i} \end{bmatrix}.$$

The classical Schwarz iteration (6) would thus be

$$\begin{aligned} \begin{bmatrix} A_{1i} & C_1 \\ B_2 & A_\Gamma \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1i}^{n+1} \\ \mathbf{u}_{1\Gamma}^{n+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{f}_{1i} \\ \mathbf{f}_\Gamma - B_1 \mathbf{u}_{2i}^n \end{bmatrix}, \\ \begin{bmatrix} A_\Gamma & B_1 \\ C_2 & A_{2i} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{2\Gamma}^{n+1} \\ \mathbf{u}_{2i}^{n+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{f}_\Gamma - B_2 \mathbf{u}_{1i}^n \\ \mathbf{f}_{2i} \end{bmatrix}. \end{aligned} \tag{18}$$

Using a Schur complement to eliminate the unknowns $\mathbf{u}_{2i}$ on the first subdomain at the fixed point, we obtain

$$\begin{bmatrix} A_{1i} & C_1 \\ B_2 & A_\Gamma - B_1 A_{2i}^{-1} C_2 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1i} \\ \mathbf{u}_{1\Gamma} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_{1i} \\ \mathbf{f}_\Gamma - B_1 A_{2i}^{-1} \mathbf{f}_{2i} \end{bmatrix},$$

and $\mathbf{f}_{2i}$ can be expressed again using the unknowns of subdomain 2,

$$\mathbf{f}_{2i} = C_2 \mathbf{u}_{2\Gamma} + A_{2i} \mathbf{u}_{2i}.$$

Doing the same on the other subdomain, we obtain the new Schwarz method

$$\begin{aligned} \begin{bmatrix} A_{1i} & C_1 \\ B_2 & A_\Gamma - B_1 A_{2i}^{-1} C_2 \end{bmatrix} \begin{bmatrix} \mathbf{u}_{1i}^{n+1} \\ \mathbf{u}_{1\Gamma}^{n+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{f}_{1i} \\ \mathbf{f}_\Gamma - B_1 \mathbf{u}_{2i}^n - B_1 A_{2i}^{-1} C_2 \mathbf{u}_{2\Gamma}^n \end{bmatrix}, \\ \begin{bmatrix} A_\Gamma - B_2 A_{1i}^{-1} C_1 & B_1 \\ C_2 & A_{2i} \end{bmatrix} \begin{bmatrix} \mathbf{u}_{2\Gamma}^{n+1} \\ \mathbf{u}_{2i}^{n+1} \end{bmatrix} &= \begin{bmatrix} \mathbf{f}_\Gamma - B_2 \mathbf{u}_{1i}^n - B_2 A_{1i}^{-1} C_1 \mathbf{u}_{1\Gamma}^n \\ \mathbf{f}_{2i} \end{bmatrix}. \end{aligned} \tag{19}$$

This method converges in two steps, since after one solve, the right hand side in both subdomains is the right hand side of the Schur complement system, which is then solved in the next step. The optimal choice for the new subdomain matrices $\tilde{A}_j$, $j = 1, 2$, is therefore to subtract in A_1 from the last diagonal block the Schur complement $B_1 A_{2i}^{-1} C_2$, and from the first diagonal block in A_2 the Schur complement $B_2 A_{1i}^{-1} C_1$. Since these Schur complements are dense, using them significantly increases the cost per iteration. Any approximation of these Schur complements with the same sparsity structure as A_Γ however leads to an optimized Schwarz method with identical cost to the classical Schwarz method (18) per iteration. Approximation of the Schur complement at the algebraic level was extensively studied in [7]. We show in the next section an approximation based on the PDE which is discretized.

4 Numerical Results

As test problems, we use finite difference and spectral element discretizations of the modified Helmholtz problem in two spatial dimensions with appropriate boundary conditions,

$$\mathcal{L}u = (\eta - \Delta)u = f, \quad \text{in } \Omega. \tag{20}$$

Discretization of (20) using a standard five point finite difference stencil on an equidistant grid on the domain $\Omega = (0,1) \times (0,1)$ with homogeneous Dirichlet boundary conditions leads to the matrix problem

$$A^{FD}\mathbf{u}=\mathbf{f},\quad A^{FD}=\frac{1}{h^2}\begin{bmatrix} T_\eta & -I & \\ -I & T_\eta & \ddots \\ & \ddots & \ddots \end{bmatrix},\quad T_\eta=\begin{bmatrix} \eta h^2+4 & -1 & \\ -1 & \eta h^2+4 & \ddots \\ & \ddots & \ddots \end{bmatrix}.$$

The subdomain matrices A_j, $j=1,2$ of a classical Schwarz method are of the same form as A^{FD}, just smaller. To obtain the optimized subdomain matrices $\tilde{A}_j$, it suffices according to Section 3 to replace the last diagonal block T_η in A_1 and the first one in A_2 by an approximation of the Schur complements. Based on the discretized PDE, we use here the matrix [4]

$$\tilde{T}=\frac{1}{2}T_\eta+phI+\frac{q}{h}(T_0-2I),\quad T_0:=T_\eta|_{\eta=0}, \tag{21}$$

which corresponds to a general optimized transmission condition of order 2 with the two parameters p and q. The optimal choice of the parameters p and q in the new block $\tilde{T}$ depends on the problem parameter η, the overlap in the method, the mesh parameter h and the lowest frequency along the interface, $k_{\min}$. Using the results in [4], one can derive the hierarchy of choices in Table 1 for h small.

	p	q
T0	$\sqrt{\eta}$	0
T2	$\sqrt{\eta}$	$\frac{1}{2\sqrt{\eta}}$
O0, no overlap	$\sqrt{\pi}(k_{\min}^2+\eta)^{1/4}h^{-1/2}$	0
O0, overlap Ch	$2^{-1/3}(k_{\min}^2+\eta)^{1/3}(Ch)^{-1/3}$	0
O2, no overlap	$2^{-1/2}\pi^{1/4}(k_{\min}^2+\eta)^{3/8}h^{-1/4}$	$2^{-1/2}\pi^{-3/4}(k_{\min}^2+\eta)^{-1/8}h^{3/4}$
O2, overlap Ch	$2^{-3/5}(k_{\min}^2+\eta)^{2/5}(Ch)^{-1/5}$	$2^{-1/5}(k_{\min}^2+\eta)^{-1/5}(Ch)^{3/5}$

Table 1. Choices for the parameters p and q in the new interface blocks $\tilde{T}$ in (21). Tj stands for Taylor of order j, and Oj stands for optimized of order j.

Figure 1 illustrates the effect of replacing the interface blocks on the performance of the RAS iteration for the model problem on the unit square with $\eta=1$ and $h=1/30$. The asymptotic formulas from [4] were employed for the various choices of the parameters in (21). Clearly, the convergence of RAS is greatly accelerated and the number of operations per iteration is identical.

In a nodal spectral element discretization, the computational domain Ω is partitioned into K elements Ω_k in which u is expanded in terms of the N–th degree Lagrangian interpolants h_i defined in Ronquist [6]. A weak variational problem is obtained by integrating the equation with respect to test functions and directly evaluating inner products using Gaussian quadrature.

The model problem (20) is discretized on the domain $\Omega=(0,2)\times(0,4)$ with periodic boundary conditions and 32 spectral elements. The right hand side is constructed to be C^0 along element boundaries as displayed in Figure 2. Non-overlapping Schwarz methods are well-suited to spectral element discretizations. Here, a zero-th order optimized transmission condition is employed in AS applied to the augmented system. The resulting optimized Schwarz iteration is accelerated by a generalized

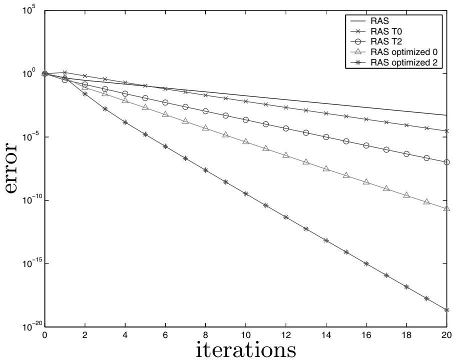


Fig. 1. Convergence curves of classical RAS, compared to the hierarchy of optimized RAS methods: Taylor optimized zero-th order (T0) and second order (T2), and RAS optimized zero-th (O0) and second order (O2).

minimal residual (GMRES) Krylov method [8]. Figure 2 also contains a plot of the residual error versus the number of GMRES iterations for diagonal (the inverse mass matrix) and optimized Schwarz preconditioning.

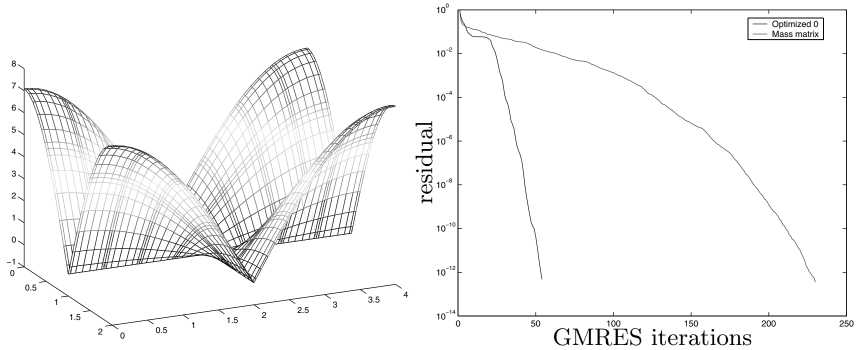


Fig. 2. Left panel: Right hand side of modified Helmholtz problem. Right panel: Residual error versus GMRES iterations.

References

1. X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797.
2. M. Dryja and O. B. Widlund, *An additive variant of the Schwarz alternating method in the case of many subregions*, Tech. Rep. 339, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 1987.

3. E. Efstathiou and M. J. Gander, *RAS: Understanding restricted additive Schwarz*, Tech. Rep. 6, McGill University, 2002.
4. M. J. Gander, *Optimized Schwarz methods*, SIAM J. Numer. Anal., 44 (2006), pp. 699–731.
5. P.-L. Lions, *On the Schwarz alternating method. I.*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM, pp. 1–42.
6. E. M. Ronquist, *Optimal Spectral Element Methods for the Unsteady Three-Dimensional Incompressible Navier-Stokes Equations*, PhD thesis, Massachusetts Institute of Technology, Department of Mechanical Engineering, 1988.
7. F.-X. Roux, F. Magoulès, S. Salmon, and L. Series, *Optimization of interface operator based on algebraic approach*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003.
8. Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.
9. A. St-Cyr, M. J. Gander, and S. J. Thomas, *Optimized multiplicative, additive and restricted additive Schwarz preconditioning.* In preparation, 2006.

MINISYMPOSIUM 3: Domain Decomposition Methods Applied to Challenging Engineering Problems

Organizers: Daniel Rixen[1], Christian Rey[2], and Pierre Gosselet[2]

[1] Technical University of Delft. D.J.Rixen@wbmt.tudelft.nl
[2] LMT Cachan. {rey,gosselet}@lmt.ens-cachan.fr

Domain decomposition solvers are popular for engineering analysis of applications such as car bodies, tires, oil reservoirs, or aerospace structures. These methods have shown to be very efficient in exploiting high performance computing capabilities. Nevertheless a significant gap remains between their optimality as predicted from idealized mathematical analysis and the lack of robustness experienced in many applications.

Defining efficient domain decomposition strategies remains very challenging due for instance to:

- the nature of engineering problems (heterogeneous, quasi-incompressible, involving multiphysics)
- the complexity of the analysis (nonlinear, dynamics, optimization, multiscale)
- the model quality (aspect ratio and interface smoothness of subdomains, non-matching meshes)

Although solvers can be tuned using specific preconditioners, scalings, and coarse grids, expertise is often required to apply domain decomposition methods judiciously. The minisymposium intends on one hand to pinpoint the difficulties encountered in practice when applying domain decomposition methods as “black box” tools and, on the other hand, to exhibit new advances that enhance their robustness. Providing a forum for theory, computation and application related discussions, the minisymposium will contribute to defining essential research directions for the future.

An Overview of Scalable FETI–DP Algorithms for Variational Inequalities

Zdeněk Dostál[1], David Horák[1] and Dan Stefanica[2]

[1] FEI VŠB-Technical University Ostrava, CZ-70833 Ostrava, Czech Republic.
zdenek.dostal@vsb.cz, david.horak@vsb.cz

[2] Baruch College, City University of New York, NY 10010, USA.
Dan_Stefanica@baruch.cuny.edu

Summary. We review our recent results concerning optimal algorithms for the numerical solution of both coercive and semi-coercive variational inequalities by combining dual-primal FETI algorithms with recent results for bound and equality constrained quadratic programming problems. The convergence bounds that guarantee the scalability of the algorithms are presented. These results are confirmed by numerical experiments.

1 Introduction

The Finite Element Tearing and Interconnecting (FETI) method was originally proposed by Farhat and Roux [14] as a parallel solver for problems described by elliptic partial differential equations. After introducing a so–called "natural coarse grid", Farhat, Mandel and Roux [13] modified the basic FETI method to obtain a numerically scalable algorithm. A similar result was achieved by the Dual-Primal FETI method (FETI–DP) introduced by Farhat et al. [12]; see also [15]. In this paper, we use the FETI–DP method to develop scalable algorithms for the numerical solution of elliptic variational inequalities. The FETI–DP methodology is first applied to the variational inequality to obtain either a strictly convex quadratic programming problem with non-negativity constraints, or a convex quadratic programming problem with bound and equality constraints. These problems are then solved efficiently by recently proposed improvements [4, 11] of the active set based proportioning algorithm [3], possibly combined with a semimonotonic augmented Lagrangian algorithm [5, 6]. The rate of convergence of these algorithms can be bounded in terms of the spectral condition number of the quadratic problem, and therefore the scalability of the resulting algorithm can be established provided that suitable bounds on the condition number of the Hessian of the quadratic cost function exist. We present such estimates in terms of the decomposition parameter H and the discretization parameter h. These bounds are independent of both the decomposition of the computational domain and the discretization, provided that we keep the ratio H/h fixed. We report numerical results that are in agreement with the theory and confirm the numerical scalability of our algorithm. Let us recall that an algorithm based on FETI–DP and on active set strategies with additional planning steps, FETI–C, was introduced by Farhat et al. [1]. The scalability of FETI–C was established experimentally.

2 Model problem

To simplify our exposition, we restrict our attention to a simple model problem. The computational domain is $\Omega = \Omega^1 \cup \Omega^2$, where $\Omega^1 = (0,1) \times (0,1)$ and $\Omega^2 = (1,2) \times (0,1)$, with boundaries Γ^1 and Γ^2, respectively. We denote by Γ_u^i, Γ_f^i, and Γ_c^i the fixed, free, and potential contact parts of Γ^i, $i = 1,2$. We assume that Γ_u^1 has non-zero measure, i.e., $\Gamma_u^1 \neq \emptyset$. For a coercive model problem, $\Gamma_u^2 \neq \emptyset$, while for a semicoercive model problem, $\Gamma_u^2 = \emptyset$; see Figure 1a. Let $\Gamma_c = \Gamma_c^1 \cup \Gamma_c^2$. The Sobolev space of the first order on Ω^i is denoted by $H^1(\Omega^i)$ and the space of Lebesgue square integrable functions is denoted by $L^2(\Omega^i)$. Let $V = V^1 \times V^2$, with

$$V^i = \left\{ v^i \in H^1(\Omega^i) : v^i = 0 \quad \text{on} \quad \Gamma_u^i \right\}, \quad i = 1,2.$$

Let $\mathcal{K} \subset V$ be a closed convex subset of $\mathcal{H} = H^1(\Omega^1) \times H^1(\Omega^2)$ defined by

$$\mathcal{K} \ = \ \left\{ (v^1, v^2) \in V : v^2 - v^1 \geq 0 \quad \text{on} \quad \Gamma_c \right\}.$$

We define the symmetric bilinear form $a(\cdot,\cdot) : \mathcal{H} \times \mathcal{H} \to R$ by

$$a(u,v) = \sum_{i=1}^{2} \int_{\Omega^i} \left(\frac{\partial u^i}{\partial x_1} \frac{\partial v^i}{\partial x_1} + \frac{\partial u^i}{\partial x_2} \frac{\partial v^i}{\partial x_2} \right) dx.$$

Let $f \in L^2(\Omega)$ be a given function and $f^i \in L^2(\Omega^i)$, $i = 1,2$, be the restrictions of f to Ω^i, $i = 1,2$. We define the linear form $l(\cdot) : \mathcal{H} \to R$ by

$$\ell(v) = \sum_{i=1}^{2} \int_{\Omega^i} f^i v^i dx$$

and consider the following problem:

$$\text{Find} \quad \min \ \frac{1}{2} a(u,u) - \ell(u) \quad \text{subject to} \quad u \in \mathcal{K}. \tag{1}$$

The solution of the model problem may be interpreted as the displacement of two membranes under the traction f. The left membrane Ω^1 is fixed at the left edge as in Figure 1a and the left edge of Ω^2 is not allowed to penetrate below the right edge of Ω^1. For the model problem to be well defined, we either require that the right edge of the right membrane Ω^2 is fixed, for the coercive problem, or, for the semicoercive problem, that the traction function f satisfies

$$\int_{\Omega^2} f \, dx < 0.$$

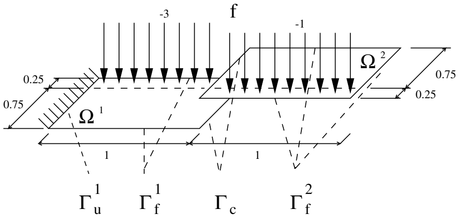


Fig. 1a: Semi–coercive model problem.

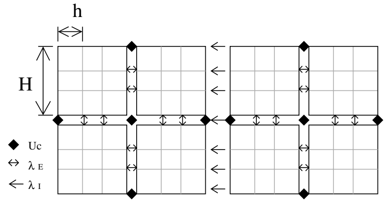


Fig. 1b: Decomposition: $H = .5, H/h = 3$.

3 A FETI–DP discretization of the problem

The first step in our domain decomposition method is to partition each domain Ω^i, $i = 1, 2$, using a rectangular grid into subdomains of diameter of order H. Let W be the finite element space whose restrictions to Ω^1 and Ω^2 are Q_1 finite element spaces of comparable mesh sizes of order h, corresponding to the subdomain grids in Ω^1 and Ω^2. We call a crosspoint either a corner that belongs to four subdomains, or a corner that belongs to two subdomains and is located on $\partial\Omega^1 \setminus \Gamma_u^1$ or on $\partial\Omega^2 \setminus \Gamma_u^2$. The nodes corresponding to the end points of Γ_c are not regarded as crosspoints; see Figure 1b. An important feature for developing FETI–DP type algorithms is that a single global degree of freedom is used at each crosspoint, while two degrees of freedom are introduced at all the other matching nodes across subdomain edges. Let $v \in W$. The continuity of v in Ω^1 and Ω^2 is enforced at every interface node that is not a crosspoint. For simplicity, we also denote by v the nodal values vector of $v \in W$. The discretized version of problem (1) with the auxiliary domain decomposition has the form

$$\min \frac{1}{2} v^T K v - v^T f \quad \text{subject to} \quad B_I v \leq 0 \quad \text{and} \quad B_E v = 0, \tag{2}$$

where the full rank matrices B_I and B_E describe the non-penetration (inequality) conditions and the gluing (equality) conditions, respectively, and f represents the discrete analog of the linear form $\ell(\cdot)$. In (2), $K = \operatorname{diag}(K_1, K_2)$ is the block diagonal stiffness matrix corresponding to the model problem (1). The block K_1 corresponding to Ω^1 is nonsingular, due to the Dirichlet boundary conditions on Γ_u^1. The block K_2 corresponding to Ω^2 is nonsingular for a coercive problem, and is singular, with the kernel made of a vector e with all entries equal to 1, for a semicoercive problem. The kernel of K is spanned by the matrix R defined by

$$R = \begin{bmatrix} 0 \\ e \end{bmatrix}.$$

Even though R is a column vector for our model problem, we will regard R as a matrix whose columns span the kernel of K. We partition the nodal values of $v \in W$ into crosspoint nodal values, denoted by v_c, and remainder nodal values, denoted by v_r. The continuity of v at crosspoints is enforced by using a global vector of degrees of freedom v_c^g and a global-to-local map L_c with one nonzero entry equal to 1 in each row, i.e., we require that $v_c = L_c v_c^g$. Therefore,

$$v = \begin{bmatrix} v_r \\ v_c \end{bmatrix} = \begin{bmatrix} v_r \\ L_c v_c^g \end{bmatrix}.$$

Let f_c and f_r be the parts of the right hand side f corresponding to the corner and remainder nodes, respectively. Let $B_{I,r}$ and $B_{I,c}$ be the matrices made of the columns of B_I corresponding to v_r and v_c, respectively; define $B_{E,r}$ and $B_{E,c}$ similarly. Let

$$B_r = \begin{bmatrix} B_{I,r} \\ B_{E,r} \end{bmatrix}, \quad B_c = \begin{bmatrix} B_{I,c} \\ B_{E,c} \end{bmatrix}, \quad B = [B_r \ B_c].$$

Let K_{rr}, K_{rc}, and K_{cc} denote the blocks of K corresponding to the decomposition of v into v_r and v_c. Consider the shortened vectors

$$\overline{v} = \begin{bmatrix} v_r \\ v_c^g \end{bmatrix} \in \overline{W}.$$

Let λ_I and λ_E be Lagrange multipliers enforcing the inequality and redundancy conditions. The Lagrangian $L(v, \lambda) = 1/2 \ v^T K v - v^T f + v^T B^T \lambda$ associated with problem (2) can be expressed as follows:

$$L(\overline{v}, \lambda) = \frac{1}{2} \overline{v}^T \overline{K} \overline{v} - \overline{v}^T \overline{f} + \overline{v}^T \overline{B}^T \lambda, \tag{3}$$

where

$$\lambda = \begin{bmatrix} \lambda_I \\ \lambda_E \end{bmatrix}, \quad \overline{K} = \begin{bmatrix} K_{rr} & K_{rc} L_c \\ L_c^T K_{rc}^T & L_c^T K_{cc} L_c \end{bmatrix}, \quad \overline{B} = [B_r \ B_c L_c], \quad \overline{f} = \begin{bmatrix} f_r \\ L_c^T f_c \end{bmatrix}.$$

Using duality theory [2], we can eliminate the primal variables v from the mixed formulation of (2). For a coercive problem, K is nonsingular and we obtain the problem of finding

$$\min \Theta(\lambda) = \min \frac{1}{2} \lambda^T F \lambda - \lambda^T \widetilde{d} \quad \text{s.t.} \quad \lambda_I \geq 0, \tag{4}$$

with $F = \overline{B} \ \overline{K}^{-1} \overline{B}^T$ and $\widetilde{d} = \overline{B} \ \overline{K}^{-1} \overline{f}$. For an efficient implementation of F it is important to exploit the structure of K; see [9, 10] for more details.
For a semicoercive problem, we obtain the problem of finding

$$\min \Theta(\lambda) = \min \frac{1}{2} \lambda^T F \lambda - \lambda^T \widetilde{d} \quad \text{s.t.} \quad \lambda_I \geq 0 \quad \text{and} \quad \widetilde{G} \lambda = \widetilde{e}, \tag{5}$$

where $F = \overline{B} \ \overline{K}^{\dagger} \overline{B}^T$, $\widetilde{d} = \overline{B} \ \overline{K}^{\dagger} \overline{f}$, $\widetilde{G} = R^T \overline{B}^T$, $\widetilde{e} = R^T \overline{f}$. Here, $\overline{K}^{\dagger}$ denotes a suitable generalized inverse that satisfies $\overline{K} \ \overline{K}^{\dagger} \ \overline{K} = \overline{K}$. Even though problem (5) is much more suitable for computations than (1) and was used for solving discretized variational inequalities efficiently [7], further improvement may be achieved as follows. Let $\widetilde{T}$ denote a nonsingular matrix that defines the orthonormalization of the rows of $\widetilde{G}$ such that the matrix $G = \widetilde{T} \widetilde{G}$ has orthonormal rows. Let $e = \widetilde{T} \widetilde{e}$. Then, problem (5) reads

$$\min \quad \frac{1}{2} \lambda^T F \lambda - \lambda^T \widetilde{d} \quad \text{s.t} \quad \lambda_I \geq 0 \quad \text{and} \quad G \lambda = e. \tag{6}$$

Next, we transform the problem of minimization on the subset of the affine space to a minimization problem on a subset of a vector space. Let $\widetilde{\lambda}$ be an arbitrary feasible vector such that $G\widetilde{\lambda} = e$. We look for the solution λ of (5) in the form $\lambda = \mu + \widetilde{\lambda}$. After returning to the old notation by replacing μ by λ, it is easy to see that (6) is equivalent to

$$\min \quad \frac{1}{2}\lambda^T F\lambda - d^T\lambda \quad \text{s.t} \quad G\lambda = 0 \quad \text{and} \quad \lambda_I \geq -\widetilde{\lambda_I}, \tag{7}$$

with $d = \widetilde{d} - F\widetilde{\lambda}$. Our final step is based on the observation that the augmented Lagrangian for problem (7) may be decomposed by the orthogonal projectors

$$Q = G^T G \qquad \text{and} \qquad P = I - Q$$

on the image space of G^T and on the kernel of G, respectively. Since $P\lambda = \lambda$ for any feasible λ, problem (7) is equivalent to

$$\min \quad \frac{1}{2}\lambda^T PFP\lambda - \lambda^T Pd \quad \text{s.t} \quad G\lambda = 0 \quad \text{and} \quad \lambda_I \geq -\widetilde{\lambda_I}. \tag{8}$$

4 Optimality

To solve the discretized variational inequality, we use our recently proposed algorithms [9, 10]. To solve the bound constrained quadratic programming problem (4), we use active set based algorithms with proportioning and gradient projections [4, 11]. The rate of convergence of the resulting algorithm can be estimated in terms of bounds on the spectrum of the Hessian of Θ. To solve the bound and equality constrained quadratic programming problem (8), we use semimonotonic augmented Lagrangian algorithms [5, 6]. The equality constraints are enforced by Lagrange multipliers generated in the outer loop, while the bound constrained problems are solved in the inner loop by the above mentioned algorithms. The rate of convergence of this algorithm may again be described in terms of bounds on the spectrum of the Hessian of Θ. Summing up, the optimality of our algorithms is guaranteed, provided that we establish optimal bounds on the spectrum of the Hessian of Θ. Such bounds on the spectrum of the operator F, possibly restricted to ImP, are given in the following theorem:

Theorem 1. *If F denotes the Hessian matrix of Θ in (4), the following spectral bounds hold:*

$$\lambda_{\max}(F) = ||F|| \;\leq\; C\left(\frac{H}{h}\right)^2; \quad \lambda_{\min}(F) \;\geq\; C.$$

If F denotes the Hessian matrix of Θ in (5), the following spectral bounds hold:

$$\lambda_{\max}(F|\mathrm{Im}P) \leq ||F|| \;\leq\; C\left(\frac{H}{h}\right)^2; \quad \lambda_{\min}(F|\mathrm{Im}P) \;\geq\; C.$$

Proof: See [9, 10].

5 Numerical experiments

We report some results for the numerical solutions of a coercive contact problem and of a semicoercive contact problem, in order to illustrate the performance and numerical scalability of our FETI–DP algorithms. In our experiments, we used a function f vanishing on $(0,1) \times [0,0.75) \cup (1,2) \times [0.25,1)$. For the coercive problem, f was equal to -1 on $(0,1) \times [0.75,1)$ and to -3 on $(1,2) \times [0,0.25)$, while for the semicoercive problem, f was equal to -5 on $(0,1) \times [0.75,1)$ and to -1 on $(1,2) \times [0,0.25)$. Each domain Ω^i was partitioned into identical squares with sides $H = 1/2, 1/4, 1/8, 1/16$. These squares were then discretized by a regular grid with the stepsize h. For each partition, the number of nodes on each edge, H/h, was taken to be 4, 8, and 16. The meshes matched across the interface for every neighboring subdomains. All experiments were performed in MATLAB. The solution of both the coercive and semicoercive model problems for $H = 1/4$ and $h = 1/4$ are presented in Figure 2. Selected results of the computations for varying values of H and H/h are given in Table 1, for the coercive problem, and in Table 2 for the semicoercive problem. The primal dimension/dual dimension/number of corners are recorded in the upper row in each field of the table, while the number of the conjugate gradient iterations required for the convergence of the solution to the given precision is recorded in the lower row. The key point is that the number of the conjugate gradient iterations for a fixed ratio H/h varies very moderately with the increasing number of subdomains.

Table 1. Convergence results for the FETI–DP algorithm - coercive problem.

H	1	1/2	1/4	1/8
$H/h = 16$	578/17/0	2312/153/10	9248/785/42	36992/3489/154
	16	27	48	51
$H/h = 8$	162/9/0	648/73/10	2592/369/42	10365/1633/154
	11	22	36	38
$H/h = 4$	50/5/0	200/33/10	800/161/42	3200/705/154
	7	17	21	27

Table 2. Convergence results for the FETI–DP algorithm - semicoercive problem.

H	1/2	1/4	1/8
$H/h = 16$	2312/155/8	9248/791/36	36992/3503/140
	61	51	53
$H/h = 8$	648/75/8	2592/375/36	10368/1647/140
	38	36	46
$H/h = 4$	200/35/8	800/167/36	3200/719/140
	29	28	35

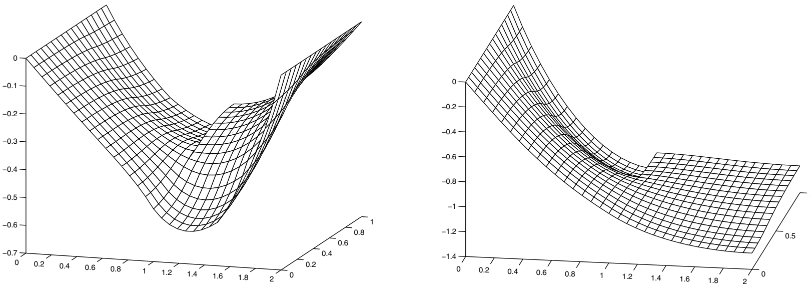


Fig. 2a: Solution of coercive problem. Fig. 2b: Solution of semi-coercive problem.

6 Comments and conclusions

We have applied the FETI–DP methodology to the numerical solution of a variational inequality. Theoretical arguments and results of numerical experiments show that the scalability of the FETI–DP method which has been established earlier for linear problems may be preserved even in the presence of nonlinear conditions on the contact boundary. The results are supported by numerical experiments. Similar results were obtained also for non-matching contact interfaces discretized by mortars [8].

Acknowledgments. The work of the first two authors was supported by Grant 101/04/1145 of the GA CR, by Grant S3086102 of GA CAS, and by Projects 1ET400300415 and ME641 of the Ministry of Education of the Czech Republic. The third author was supported by the Research Foundation of the City University of New York Awards PSC-CUNY 665463-00 34 and 66529-00 35.

References

1. P. Avery, G. Rebel, M. Lesoinne, and C. Farhat, *A numerically scalable dual–primal substructuring method for the solution of contact problems - part I: the frictionless case*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 2403–2426.
2. D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, New Hampshire, second ed., 1999.
3. Z. Dostál, *Box constrained quadratic programming with proportioning and projections*, SIAM J. Optim., 7 (1997), pp. 871–887.
4. ———, *A proportioning based algorithm for bound constrained quadratic programming with the rate of convergence*, Numer. Algorithms, 34 (2003), pp. 293–302.
5. ———, *Inexact semimonotonic augmented Lagrangians with optimal feasibility convergence for convex bound and equality constrained quadratic programming*, SIAM J. Num. Anal., 43 (2006), pp. 96–115.

6. ———, *An optimal algorithm for bound and equality constrained quadratic programming problems with bounded spectrum.* Submitted to Computing, 2006.
7. Z. Dostál, A. Friedlander, and S. A. Santos, *Solution of contact problems of elasticity by FETI domain decomposition*, Contemporary Mathematics, 218 (1998), pp. 82–93.
8. Z. Dostál, D. Horák, and D. Stefanica, *A scalable FETI–DP algorithm with non–penetration mortar conditions on contact interface.* Submitted, 2004.
9. Z. Dostál, D. Horák, and D. Stefanica, *A scalable FETI–DP algorithm for a coercive variational inequality*, J. Appl. Numer. Math., 54 (2005), pp. 378–390.
10. Z. Dostál, D. Horák, and D. Stefanica, *A scalable FETI–DP algorithm for a semi–coercive variational inequality.* Submitted, 2005.
11. Z. Dostál and J. Schöberl, *Minimizing quadratic functions over non-negative cone with the rate of convergence and finite termination*, Comput. Optim. Appl., 30 (2005), pp. 23–43.
12. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.
13. C. Farhat, J. Mandel, and F.-X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Methods Appl. Mech. Engrg., 115 (1994), pp. 365–385.
14. C. Farhat and F.-X. Roux, *An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems*, SIAM J. Sc. Stat. Comput., 13 (1992), pp. 379–396.
15. 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 (2002), pp. 159–179.

Performance Evaluation of a Multilevel Sub-structuring Method for Sparse Eigenvalue Problems*

Weiguo Gao[1], Xiaoye S. Li[1], Chao Yang[1], and Zhaojun Bai[2]

[1] Computational Research Division, Lawrence Berkeley National Laboratory, Berkeley, CA 94720, USA. {wggao, xsli, cyang}@lbl.gov

[2] Department of Computer Science, University of California, Davis, CA 95616, USA. bai@cs.ucdavis.edu

1 Introduction

The automated multilevel sub-structuring (AMLS) method [2, 7, 3] is an extension of a simple sub-structuring method called *component mode synthesis* (CMS) [6, 4] originally developed in the 1960s. The recent work by Bennighof and Lehoucq [3] provides a high level mathematical description of the AMLS method in a continuous variational setting, as well as a framework for describing AMLS in matrix algebra notations. The AMLS approach has been successfully used in vibration and acoustic analysis of very large scale finite element models of automobile bodies [7]. In this paper, we evaluate the performance of AMLS on other types of applications.

Similar to the domain decomposition techniques used in solving linear systems, AMLS reduces a large-scale eigenvalue problem to a sequence of smaller problems that are easier to solve. The method is amenable to an efficient parallel implementation. However, a few questions regarding the accuracy and computational efficiency of the method remain to be carefully examined. Our earlier paper [12] addressed some of these questions for a single-level algorithm. We developed a simple criterion for choosing spectral components from each sub-structure, performed algebraic analysis based on this mode selection criterion, and derived error bounds for the approximate eigenpair associated with the smallest eigenvalue. This paper focuses on the performance of the multilevel algorithm.

2 The algorithmic view of AMLS

We are concerned with solving the following algebraic eigenvalue problem

$$Kx = \lambda Mx, \tag{1}$$

*This work was supported by the Director, Office of Advanced Scientific Computing Research, Division of Mathematical, Information, and Computational Sciences of the U.S. Department of Energy under contract number DE-AC03-76SF00098.

where K is symmetric, M is symmetric positive definite, and both are sparse. Using a graph partitioning software package such as METIS [8] we can permute the matrix pencil (K, M) into a multilevel nested block structure shown below:

$$K = \begin{pmatrix} K_{11} & & & & & & \\ & K_{22} & & & sym. & & \\ K_{31} & K_{32} & K_{33} & & & & \\ & & & K_{44} & & & \\ & & & & K_{55} & & \\ & & & K_{64} & K_{65} & K_{66} & \\ K_{71} & K_{72} & K_{73} & K_{74} & K_{75} & K_{76} & K_{77} \end{pmatrix}, M = \begin{pmatrix} M_{11} & & & & & & \\ & M_{22} & & & sym. & & \\ M_{31} & M_{32} & M_{33} & & & & \\ & & & M_{44} & & & \\ & & & & M_{55} & & \\ & & & M_{64} & M_{65} & M_{66} & \\ M_{71} & M_{72} & M_{73} & M_{74} & M_{75} & M_{76} & M_{77} \end{pmatrix} \quad (2)$$

The blocks K_{ij} and M_{ij} are of size n_i-by-n_j. A byproduct of this partitioning and reordering algorithm is a *separator tree* depicted in Figure 1. The separator tree can be used to succinctly describe the matrix structure (2), the computational tasks and their dependencies in the AMLS algorithm. The internal tree nodes (marked by □) represent the separators (also known as the *interface* blocks, e.g. K_{33}, K_{66} and K_{77}), and the bottom leaf nodes (marked by ○) represent the substructures (e.g. K_{11}, K_{22}, K_{44} and K_{55}). The permutation of the pencil (K, M) is followed by a block factorization of the K matrix, i.e., $K = LDL^T$, where

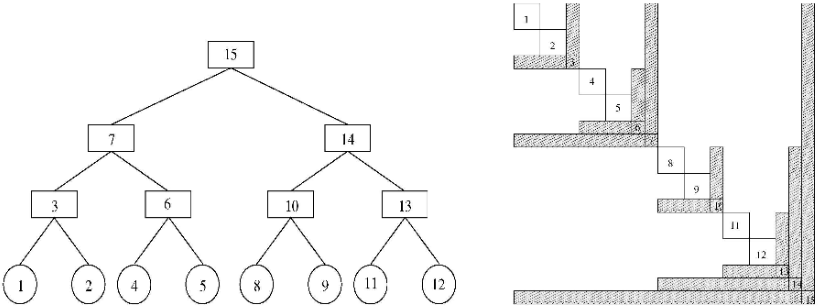


Fig. 1. Separator tree (left) and the reordered matrix (right) for a three-level dissection.

$$D = L^{-1}KL^{-T} = \operatorname{diag}(K_{11}, K_{22}, \widehat{K}_{33}, K_{44}, K_{55}, \widehat{K}_{66}, \widehat{K}_{77}) \overset{def}{=} \widehat{K}\ . \quad (3)$$

and L is given by:

$$L = \begin{pmatrix} I_{n_1} & & & & & & \\ & I_{n_2} & & & & & sym. \\ K_{31}K_{11}^{-1} & K_{32}K_{22}^{-1} & I_{n_3} & & & & \\ & & & I_{n_4} & & & \\ & & & & I_{n_5} & & \\ & & & K_{64}K_{44}^{-1} & K_{65}K_{55}^{-1} & I_{n_6} & \\ K_{71}K_{11}^{-1} & K_{72}K_{22}^{-1} & K_{73}\widehat{K}_{33}^{-1} & K_{74}K_{44}^{-1} & K_{75}K_{55}^{-1} & K_{76}\widehat{K}_{66}^{-1} & I_{n_7} \end{pmatrix} \quad (4)$$

Applying the same congruence transformation defined by L^{-1} to M yields:

$$L^{-1}ML^{-T} \overset{def}{=} \widehat{M} = \begin{pmatrix} M_{11} & & & & & & \\ & M_{22} & & & & sym. & \\ \widehat{M}_{31} & \widehat{M}_{32} & \widehat{M}_{33} & & & & \\ & & & M_{44} & & & \\ & & & & M_{55} & & \\ & & & \widehat{M}_{64} & \widehat{M}_{65} & \widehat{M}_{66} & \\ \widehat{M}_{71} & \widehat{M}_{72} & \widehat{M}_{73} & \widehat{M}_{74} & \widehat{M}_{75} & \widehat{M}_{76} & \widehat{M}_{77} \end{pmatrix}. \quad (5)$$

Note that $\widehat{M}$ has the same block structure as M, and only the diagonal blocks associated with the leaves of the separator tree are not altered; all the other blocks are modified. Moreover, the altered blocks of $\widehat{M}$ typically contain more non-zero elements than those in M. The eigenvalues of $(\widehat{K}, \widehat{M})$ are identical to those of (K, M), and the corresponding eigenvectors $\widehat{x}$ are related to those of the original problem (1) through $\widehat{x} = L^T x$.

Instead of computing eigenvalues of (K, M) directly, AMLS solves a number of subproblems defined by the diagonal blocks of $\widehat{K}$ and $\widehat{M}$. Suppose S_i contains eigenvectors associated with k_i desired eigenvectors of (K_{ii}, M_{ii}) (or $(\widehat{K}_{ii}, \widehat{M}_{ii})$), then, AMLS constructs a subspace in the form of

$$S = \mathrm{diag}(S_1, S_2, \ldots, S_N) \ . \tag{6}$$

The eigenvectors associated with (K_{ii}, M_{ii}) will be referred to as the *sub-structure modes*, and those associated with $(\widehat{K}_{ii}, \widehat{M}_{ii})$ will be referred to as the *coupling modes*. The approximation to the desired eigenpairs of the pencil $(\widehat{K}, \widehat{M})$ are obtained by projecting the pencil $(\widehat{K}, \widehat{M})$ onto the subspace spanned by S, i.e., we seek θ and $q \in \mathbb{R}^{\bar{k}}$, where $\bar{k} = \sum_{i=1}^{N} k_i$, such that

$$(S^T \widehat{K} S) q = \theta (S^T \widehat{M} S) q. \tag{7}$$

It follows from the Rayleigh-Ritz theory [11, page 213] that θ serves as an approximation to an eigenvalue of (K, M), and the vector formed by $z = L^{-T} S q$ is the approximation to the corresponding eigenvector. Algorithm 1 summarizes the major steps of the AMLS algorithm.

Note that when the interface blocks are much smaller than the sub-structures, we can include all the coupling modes by replacing S_i with I_{n_i} in (6). As a result, the projected problem (7) is simplified while its dimension is still kept small.

A straightforward implementation of Algorithm 1 is not very cost-effective. The amount of memory required to store the block eliminator L and the matrix $\widehat{M} = L^{-1} M L^{-T}$ is typically high due to fill-in. We used the following strategies to reduce this cost: (1) Since computing the desired eigenvalues and the eigenvectors does not require $\widehat{M}$ explicitly, we project M into the subspace spanned by the columns of $L^{-T} S$ incrementally as L and S are being computed in an order defined by a bottom-up traversal of the separator tree. In another word, we *interleave* Steps (2) to (5) of Algorithm 1; (2) We use a *semi-implicit* scheme to store L. We only explicitly compute and store the blocks in the columns associated with the separator nodes. The blocks in the columns associated with the leaf nodes are not computed explicitly. Whenever needed, $K_{ji} K_{ii}^{-1}$ is applied to a matrix block directly through a sequence of sparse triangular solves and matrix-matrix multiplications.

More implementation details can be found in our longer report [5].

Algorithm 1 *Algebraic Multilevel Sub-structuring (AMLS)*

```
Input:   A matrix pencil $(K, M)$, where $K$ is symmetric and nonsingular and
         $M$ is symmetric positive definite
Output:  $\theta_j \in R^1$ and $z_j \in R^n$, $(j = 1, 2, ..., k)$ such that $Kz_j \approx \theta_j M z_j$
(1)      Partition and reorder $K$ and $M$ to be in the form of (2)
(2)      Perform block factorization $K = LDL^T$
(3)      Apply the congruence transformation defined by $L^{-1}$ to $(K, M)$ to
         obtain $(\widehat{K}, \widehat{M})$ defined by (3) and (5)
(4)      Compute a subset of the eigenpairs of interest for the subproblems
         $(K_{ii}, M_{ii})$ (or $(\widehat{K}_{ii}, \widehat{M}_{ii})$). Then, form the matrix $S$ in (6)
(5)      Project the matrix pencil $(\widehat{K}, \widehat{M})$ into the subspace $span\{S\}$
(6)      Compute $k$ desired eigenpairs $(\theta_j, q_j)$ from $(S^T \widehat{K} S)q = \theta(S^T \widehat{M} S)q$,
         and set $z_j = L^{-T} S q_j$ for $j = 1, 2, ..., k$
```

3 Performance evaluation

We evaluate the performance of AMLS on two applications. Our first problem arises from a finite element model of a six-cell damped detuned accelerator structure [9]. The eigenvalues of this generalized eigenvalue problem correspond to the cavity resonance frequencies and the eigenvectors represent the electromagnetic accelerating field. We will refer to this problem as DDS6. Our second problem arises from the normal mode vibrational analysis of a 3000-atom polyethylene (PE) particle [13]. In this application, we are interested in the low frequency vibrations of the PE molecule. We will refer to this problem as PE3K.

Our platform is a single Power3 processor with a clock speed of 375Mhz and 2 MB of level-2 cache. We use *nev* to denote the number of wanted eigenvalues. The accuracy tolerance for each subproblem is denoted by τ_{sub}, and the accuracy tolerance for the projected problem is denoted by τ_{proj}. We use *nmodes* to denote the number of modes chosen from each sub-structure.

DDS6

The dimension of this problem is 65740, and the number of nonzero entries in $K+M$ is 1455772. Table 1 shows the AMLS timing and memory usage measurements. We experimented with different partitioning levels. For a single level partitioning, we set *nmodes* to 100. When we increase the number of levels by one, we reduce *nmodes* by half to keep the total number of sub-structure modes roughly constant. Since the separators in this problem are small, all the coupling modes are included in the subspace (6). Column 3 shows that the total memory usage does not increase too much with an increasing number of levels. By using the semi-implicit representation for L, we save some memory but need extra time for recomputing some off-diagonal blocks. This tradeoff between memory reduction and extra runtime is shown in Columns 4 and 5, which indicate that we save up to 50% of the memory with only 10-15% extra runtime. This is very attractive when memory is at a premium. Column 6 shows the time spent in the first phase of AMLS, which consists of various transformations (Steps (2)-(5) of Algorithm 1). The time spent in the second phase of the algorithm, Step (6), is reported in Column 7. The total time is reported in the last column. As the number of levels increases, the transformation time decreases, whereas the projected problem becomes larger and hence requires more time to solve. The variation of the total CPU time is small with respect to the number of levels.

Table 1. Problem DDS6, $nev = 100$, $\tau_{sub} = 10^{-10}$, $\tau_{proj} = 10^{-5}$.

levels	nmodes	mem	mem-saved	recompute	phase 1	phase 2	total
		(MB)	(MB)	(sec)	(sec)	(sec)	(sec)
2	100	319	199 (38.4%)	9.2 (1.5%)	457.7	137.2	594.8
3	50	263	263 (50.0%)	51.5 (11.0%)	287.7	178.8	466.5
4	25	325	248 (43.3%)	60.7 (13.3%)	220.2	235.4	455.6
5	12	392	228 (36.8%)	64.0 (13.2%)	194.0	291.9	485.9
6	6	480	192 (28.6%)	55.3 (10.9%)	151.9	352.4	504.2

As a comparison, it took about 407 seconds and 308 Megabytes memory to compute the smallest 100 eigenpairs by a shift-and-invert Lanczos (SIL) method (using ARPACK and SuperLLT packages [10] with MeTiS reordering.) Thus when $nev = 100$, AMLS and SIL are comparable in both speed and memory usage. However, Figure 2 shows that AMLS is more efficient than SIL when more eigenvalues are needed. In AMLS, the time consumed by phase 1 (transformations) is roughly the same for different *nev*s. The increase in the total CPU time for a larger *nev* is mainly due to the increased cost associated with solving a larger projected problem (labeled as "AMLS-Ritz" in Figure 2), but this increase is far below linear. Linear increase in total CPU time is expected in SIL because multiple shifts may be required to compute eigenvalues that are far part. In our experiment, we set the number of eigenvalues to be computed by a single-shift SIL run to 100. Since the cost associated with each single-shift SIL run is roughly the same for each shift, the total cost for a multi-shift SIL run increases linearly with respect to *nev*.

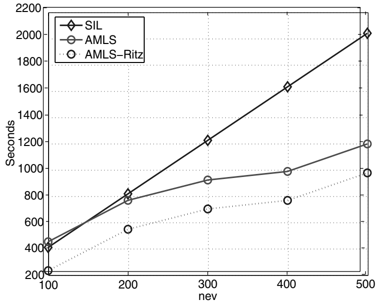


Fig. 2. Runtime of AMLS and SIL with increasing *nev*. Problem DDS6, $levels = 4$, $nmodes = 25$.

Figure 3 shows the relative error of the smallest 100 eigenvalues returned from the AMLS calculation. As shown in the left figure, the accuracy deteriorates with increasing number of levels, which is true even for the first few eigenvalues. This is due to the limited number of modes selected in the sub-structures. In the right figure, we show the results with fixed number of levels (5 here) but different *nmodes*. Although the accuracy increases with more modes selected, as expected, this increase is very gradual. For example, the bottom curve is only about 1 digit more accurate than the top one, but the size of the projected problem (see (7)) for the bottom curve is almost twice as large as that of the top curve.

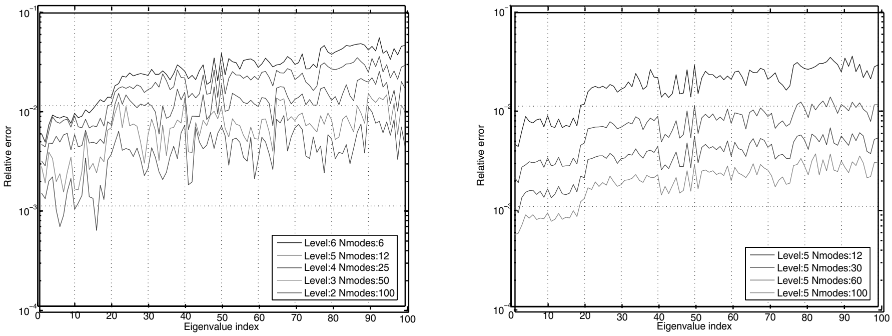


Fig. 3. Eigenvalue accuracy of DDS6. Left: increasing levels. Right: Fixed level, increasing *nmodes*.

PE3K

The low frequency vibrational modes of the PE molecule can be solved by computing the eigenvalues and eigenvectors associated with the Hessian of a potential function that describes the interaction between different atoms. For a 3000-atom molecule, the dimension of the Hessian matrix is 9000. Figure 4 shows the molecular structure of the PE particle and the sparsity pattern of the Hessian matrix after it is permuted by MeTiS. We observe that PE3K contains separators of large dimensions, resulting in excessive fills. This makes the SIL calculation memory intensive [13]. Our semi-

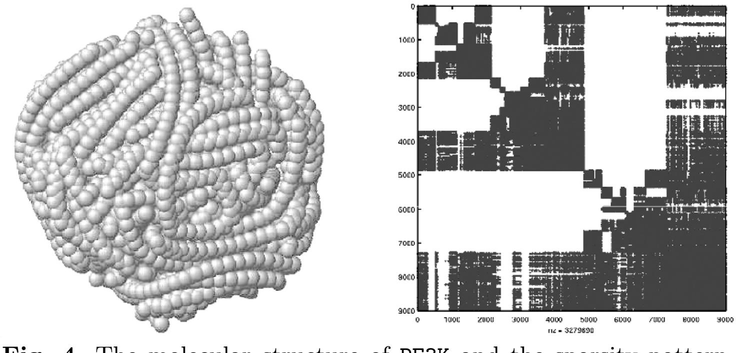


Fig. 4. The molecular structure of PE3K and the sparsity pattern of the Hessian after it is permuted by MeTiS.

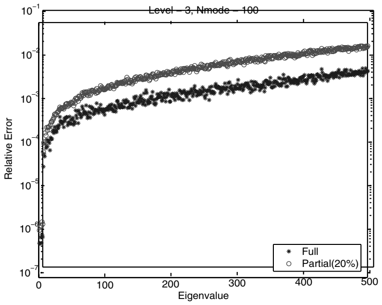


Fig. 5. Eigenvalue accuracy of PE3K, full or partial selection of interface modes.

implicit representation of L greatly reduced the memory required in the AMLS calculation (saving 35% of memory). By choosing only a fraction of the coupling modes from each separator, we also reduced the dimension of the projected problem (7). In Figure 5, we compared the accuracy of a 3-level AMLS calculation in which 20% of coupling modes are computed and chosen from each separator with a 3-level calculation in which all coupling modes are selected. Both calculations used $nmodes = 100$ for each sub-structure. Figure 5 shows that the partial selection of the coupling modes does not affect the accuracy of the AMLS calculation significantly for this problem. It is important to note that choosing 20% of coupling modes enables us to reduce the AMLS runtime from 1776 to 581 seconds.

4 Conclusions and related work

When a large number of eigenvalues with a few digits of accuracy are wanted, the multilevel sub-structuring method is computationally more advantageous than the conventional shift-and-invert Lanczos algorithm. This is due to the fact that AMLS does not have the bottlenecks associated with the reorthognalization and triangular solve. However, when the accuracy requirement is high, AMLS becomes less appealing. Some research is under way to address the accuracy issue. We are developing better mode selection criteria so that the projected subspace retains better spectral information from (K, M) while its size is still restricted. Bekas and Saad [1] suggests to enhance the algorithm by using spectral Schur complements with higher order approximations. Further evaluation is needed to determine the effectiveness of these strategies.

References

1. C. Bekas and Y. Saad, *Computation of smallest Eigenvalues using spectral Schur complements*, SIAM J. Sci. Comput., 27 (2005), pp. 458–481.
2. J. K. Bennighof, *Adaptive multi-level substructuring for acoustic radiation and scattering from complex structures*, in Computational methods for Fluid/Structure Interaction, A. J. Kalinowski, ed., vol. 178, New York, November 1993, American Society of Mechanical Engineers (ASME), pp. 25–38.

3. J. K. Bennighof and R. B. Lehoucq, *An automated multilevel substructuring method for Eigenspace computation in linear elastodynamics*, SIAM J. Sci. Comput., 25 (2004), pp. 2084–2106.
4. R. R. Craig, Jr. and M. C. C. Bampton, *Coupling of substructures for dynamic analysis*, AIAA Journal, 6 (1968), pp. 1313–1319.
5. W. Gao, X. S. Li, C. Yang, and Z. Bai, *An implementation and evaluation of the AMLS method for sparse Eigenvalue problems*, Tech. Rep. LBNL-57438, Lawrence Berkeley National Laboratory, February 2006. Submitted to ACM Trans. Math. Software.
6. W. C. Hurty, *Vibrations of structural systems by component-mode synthesis*, Journal of the Engineering Mechanics Division, ASCE, 86 (1960), pp. 51–69.
7. M. F. Kaplan, *Implementation of Automated Multilevel Substructuring for Frequency Response Analysis of Structures*, PhD thesis, University of Texas at Austin, Austin, TX, December 2001.
8. G. Karypis and V. Kumar, *MeTiS, A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Ordering of Sparse Matricies. Version 4.0*, University of Minnesota, Department of Computer Science, Minneapolis, MN, September 1998.
9. K. Ko, N. Folwell, L. Ge, A. Guetz, V. Ivanov, R. Lee, Z. Li, I. Malik, W. Mi, C.-K. Ng, and M. Wolf, *Electromagnetic systems simulation - "from simulation to fabrication"*, tech. rep., Stanford Linear Accelerator Center, Menlo Park, CA, 2003. SciDAC Report.
10. E. G. Ng and B. W. Peyton, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Stat. Comput., 14 (1993), pp. 1034–1056.
11. B. N. Parlett, *The Symmetric Eigenvalue Problem*, Prentice-Hall, 1980.
12. C. Yang, W. Gao, Z. Bai, X. S. Li, L.-Q. Lee, P. Husbands, and E. G. Ng, *An algebraic sub-structuring method for large-scale Eigenvalue calculation*, SIAM J. Sci. Comput., 27 (2006), pp. 873–892.
13. C. Yang, B. W. Peyton, D. W. Noid, B. G. Sumpter, and R. E. Tuzun, *Large-scale normal coordinate analysis for molecular structures*, SIAM J. Sci. Comput., 23 (2001), pp. 563–582.

Advection Diffusion Problems with Pure Advection Approximation in Subregions

Martin J. Gander[1], Laurence Halpern[2], Caroline Japhet[2], and Véronique Martin[3]

[1] Université de Genève, 2-4 rue du Lièvre, CP 64, CH-1211 Genève, Switzerland. martin.gander@math.unige.ch
[2] LAGA, Université Paris XIII, 99 Avenue J.-B. Clément, 93430 Villetaneuse, France. {halpern,japhet}@math.univ-paris13.fr
[3] LAMFA UMR 6140, Université Picardie Jules Verne, 33 rue Saint-Leu 80039 Amiens Cedex 1, France. veronique.martin@u-picardie.fr

Summary. We study in this paper a model problem of advection diffusion type on a region which contains a subregion where it is sufficient to approximate the problem by the pure advection equation. We define coupling conditions at the interface between the two regions which lead to a coupled solution which approximates the fully viscous solution more accurately than other conditions from the literature, and we develop a fast algorithm to solve the coupled problem.

1 Introduction

There are two main reasons for coupling different models in different regions: the first are problems where the physics is different in different regions, and hence different models need to be used, for example in fluid-structure coupling. The second are problems where one is in principle interested in the full physical model, but the full model is too expensive computationally over the entire region, and hence one would like to use a simpler model in most of the region, and the full one only where it is essential to capture the physical phenomena. We are interested in the latter case here. In our context of advection diffusion, coupling conditions for the stationary case were developed in [4]; they are obtained by a limiting process where the viscosity goes to zero in one subregion and is fixed in the other. Other coupling conditions were studied in [1] to obtain a coupled solution which is closer to the fully viscous one.

One is also interested in efficient algorithms to solve the coupled problems. These algorithms are naturally iterative substructuring algorithms. While an algorithm was proposed in [4], no algorithm was proposed in [1] for the coupling conditions approximating the fully viscous solution.

We propose here coupling conditions for the case of the fully viscous solution of the time dependent advection diffusion equation, and we develop an effective iterative substructuring algorithm for the coupled problem. After introducing our model problem in Section 2 together with the subproblems, we present the two coupling strategies from [4] and [1] in Section 3, and we introduce a new set of coupling conditions. We then compare the approximation properties of the three sets of coupling conditions to the fully viscous solution in Section 4. In Section 5, we present an iterative substructuring algorithm from [4], and introduce new algorithmic transmission conditions which imply our new coupling conditions at convergence and lead to an efficient iterative substructuring algorithm. We show numerical experiments in one and two spatial dimensions in Section 6.

2 Model Problem

We consider the non-stationary advection diffusion equation

$$\begin{aligned} \mathcal{L}_{ad}u &= f, && \text{in } \Omega \times (0,T), \\ u(\cdot,0) &= u_0 && \text{in } \Omega, \\ \mathcal{B}u &= g && \text{on } \partial\Omega, \end{aligned} \tag{1}$$

where Ω is a bounded open subset of $\mathbb{R}^2$, $\mathcal{L}_{ad} := \partial_t + \mathbf{a}\cdot\nabla - \nu\Delta + c$ is the advection diffusion operator, $\nu > 0$ is the viscosity, $c > 0$ is a constant, $\mathbf{a} = (a, b)$ is the velocity field, and $\mathcal{B}$ is some boundary operator leading to a well posed problem. In the following we call u the viscous solution. We now assume that the viscous effects are not important for the physical phenomena in a subregion $\Omega_2 \subset \Omega$, and hence we would like to use the pure advection operator $\mathcal{L}_a := \partial_t + \mathbf{a}\cdot\nabla + c$ in that subregion. With $\Omega_1 = \Omega\backslash\overline{\Omega}_2$, see Figure 1, this leads to the two subproblems

$$\left\{\begin{aligned} \mathcal{L}_{ad}u_1 &= f \;\; \text{in } \Omega_1 \times (0,T), \\ u_1(\cdot,0) &= u_0 \;\; \text{in } \Omega_1, \\ \mathcal{B}u_1 &= g \;\; \text{on } \partial\Omega \cap \partial\Omega_1, \end{aligned}\right. \qquad \left\{\begin{aligned} \mathcal{L}_{a}u_2 &= f \;\; \text{in } \Omega_2 \times (0,T), \\ u_2(\cdot,0) &= u_0 \;\; \text{in } \Omega_2, \\ \mathcal{B}u_2 &= g \;\; \text{on } \partial\Omega \cap \partial\Omega_2, \end{aligned}\right. \tag{2}$$

which need to be completed by coupling conditions on Γ, the common boundary between Ω_1 and Ω_2. Since the advection operator $\mathcal{L}_a$ is of order 1, it is necessary to know on which part of the interface $\mathbf{a}\cdot\mathbf{n}$ is positive or negative ($\mathbf{n}$ is the unit outward normal of Ω_1). We thus introduce $\Gamma_{in} = \{x \in \Gamma, \, \mathbf{a}\cdot\mathbf{n} > \mathbf{0}\}$ and $\Gamma_{out} = \{x \in \Gamma, \, \mathbf{a}\cdot\mathbf{n} \leq \mathbf{0}\}$, where $\Gamma = \Gamma_{in} \cup \Gamma_{out}$, see Figure 1.

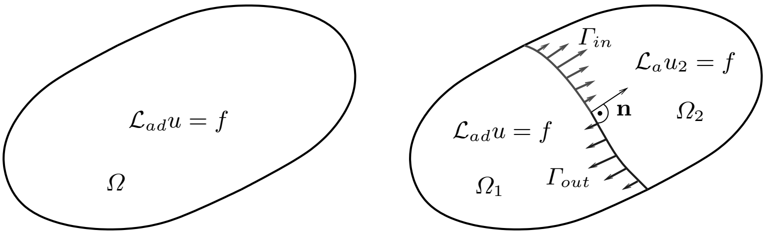


Fig. 1. Fully viscous problem on the left, and coupled subproblems on the right.

3 Coupling Conditions

If we solve the advection diffusion equation in Ω by a domain decomposition method, it is well known that the solution as well as its normal derivative must be continuous across Γ, and the only issue is to define algorithms which converge rapidly to the solution of the global problem, see [6] for a review of classical algorithms, and [2, 5] for optimized ones.

But if the equations are different in each subdomain, there are two issues: first, one has to define coupling conditions so that (2) define together with the coupling conditions a global solution close to the fully viscous one, and second one needs to find an efficient iterative substructuring algorithm to compute this solution. This algorithm can use arbitrary transmission conditions which are good for its convergence, as long as they imply at convergence the coupling conditions defining the coupled solution.

A first approach to obtain coupling conditions was introduced in [4] through a limiting process in the viscosity (singular perturbation method). With a variational formulation for the global viscous problem, and letting the viscosity tend to 0 in a subregion, it has been shown in [4] that the solution of this limiting process satisfies

$$\begin{aligned} -\nu\frac{\partial u_1}{\partial \mathbf{n}} + \mathbf{a}\cdot\mathbf{n}\,u_1 &= \mathbf{a}\cdot\mathbf{n}\,u_2 \quad && \text{on } \Gamma = \Gamma_{in}\cup\Gamma_{out}, \\ u_1 &= u_2 && \text{on } \Gamma_{in}, \end{aligned} \tag{3}$$

which is equivalent to the coupling conditions

$$\begin{aligned} u_1 &= u_2 && \text{on } \Gamma_{in}, \\ -\nu\frac{\partial u_1}{\partial \mathbf{n}} &= 0 && \text{on } \Gamma_{in}, \\ -\nu\frac{\partial u_1}{\partial \mathbf{n}} + \mathbf{a}\cdot\mathbf{n}\,u_1 &= \mathbf{a}\cdot\mathbf{n}\,u_2 \quad && \text{on } \Gamma_{out}. \end{aligned} \tag{4}$$

A second set of coupling conditions based on absorbing boundary condition theory was proposed in [1],

$$\begin{aligned} u_1 &= u_2 && \text{on } \Gamma_{in}, \\ \frac{\partial u_1}{\partial \mathbf{n}} &= \frac{\partial u_2}{\partial \mathbf{n}} && \text{on } \Gamma_{in}, \\ -\nu\frac{\partial u_1}{\partial \mathbf{n}} + \mathbf{a}\cdot\mathbf{n}\,u_1 &= \mathbf{a}\cdot\mathbf{n}\,u_2 \quad && \text{on } \Gamma_{out}. \end{aligned} \tag{5}$$

Both coupling conditions (4) and (5) imply that on Γ_{out} neither the solution nor its derivative are continuous. Since this is in contradiction with the solution of the fully viscous problem, in which we are interested, we propose a third set of coupling conditions by modifying the conditions (5) to obtain at least continuity of u on the interface,

$$\begin{aligned} u_1 &= u_2 && \text{on } \Gamma_{in}, \\ \frac{\partial u_1}{\partial \mathbf{n}} &= \frac{\partial u_2}{\partial \mathbf{n}} && \text{on } \Gamma_{in}, \\ u_1 &= u_2 && \text{on } \Gamma_{out}. \end{aligned} \tag{6}$$

In the next section, we show that if $\Gamma \equiv \Gamma_{in}$ the coupling conditions (5) and (6) give more accurate approximations to the fully viscous solution than the coupling conditions (4).

4 Error Estimates with Respect to the Viscous Solution

We consider the stationary case of (2) on the domain $\Omega = \mathbb{R}^2$, with subdomains $\Omega_1 = (-\infty, 0) \times \mathbb{R}$ and $\Omega_2 = (0, +\infty) \times \mathbb{R}$, and we estimate the error between the viscous solution and the coupled solution for each of the coupling conditions (4), (5) and (6) when the velocity field $\mathbf{a}$ is constant.

Using Fourier analysis and energy estimates, the details of which are beyond the scope of this short paper, we obtain for ν small the asymptotic results in Table 1, where $\|\cdot\|_{\Omega_i}$ denotes the L^2 norm in Ω_i. These results show that if $\mathbf{a}\cdot\mathbf{n} > 0$, then the approximation of the viscous solution by the coupled solution through conditions (5) and (6) is better in the viscous subregion Ω_1 than with the conditions (4). In fact, conditions (5) and (6) are not based on the limiting process in the viscosity, and hence retain in some sense the viscous character of the entire problem. In Ω_2 the error is $\mathcal{O}(\nu)$ independently of the coupling conditions, since we solve the advection equation instead of the advection-diffusion equation. Note also that in this case with the coupling conditions (5) and (6) we have continuity of the solution and of its normal derivative, whereas with the coupling conditions (4), we have continuity of the solution only.

	Case $\mathbf{a}\cdot\mathbf{n} > 0$ $(\Gamma \equiv \Gamma_{in})$	
	Conditions (4)	Conditions (5) and (6)
$\|u - u_1\|_{\Omega_1}$	$\mathcal{O}(\nu^{3/2})$	$\mathcal{O}(\nu^{5/2})$
$\|u - u_2\|_{\Omega_2}$	$\mathcal{O}(\nu)$	$\mathcal{O}(\nu)$

	Case $\mathbf{a}\cdot\mathbf{n} \leq 0$ $(\Gamma \equiv \Gamma_{out})$	
	Conditions (4) and (5)	Conditions (6)
$\|u - u_1\|_{\Omega_1}$	$\mathcal{O}(\nu)$	$\mathcal{O}(\nu)$
$\|u - u_2\|_{\Omega_2}$	$\mathcal{O}(\nu)$	$\mathcal{O}(\nu)$

Table 1. Asymptotic approximation quality of the coupled solution to the viscous solution through different coupling conditions.

If $\mathbf{a}\cdot\mathbf{n} \leq 0$, the solution in Ω_2 does not depend on the transmission conditions, and since we solve the advection equation in this domain, the error is $\mathcal{O}(\nu)$. Then the error is propagated into Ω_1, so we cannot have an error better than $\mathcal{O}(\nu)$ in Ω_1 independently of the coupling conditions. Note however that now only conditions (6) lead to continuity of the coupled solution.

5 Algorithmic Transmission Conditions

We now turn our attention to algorithms to compute the coupled subproblem solution. In [4], the following algorithm based on the coupling conditions (4) was proposed for the steady case (θ is a relaxation parameter):

$$\begin{cases} \mathcal{L}_{ad} u_1^{k+1} = f & \text{in } \Omega_1, \\ -\nu \dfrac{\partial u_1^{k+1}}{\partial \mathbf{n}} = 0 & \text{on } \Gamma_{in}, \\ -\nu \dfrac{\partial u_1^{k+1}}{\partial \mathbf{n}} + \mathbf{a}\cdot\mathbf{n}\, u_1^{k+1} = \mathbf{a}\cdot\mathbf{n}\, u_2^k & \text{on } \Gamma_{out}, \end{cases}$$
$$\begin{cases} \mathcal{L}_a u_2^{k+1} = f & \text{in } \Omega_2, \\ u_2^{k+1} = \theta u_1^k + (1-\theta) u_2^k & \text{on } \Gamma_{in}, \end{cases} \tag{7}$$

and it was shown that the algorithm is well posed and convergent.

In [3] an algorithm was proposed for the conditions (6) in the steady state case. This algorithm does not use the coupling conditions, but better suited transmission conditions which imply the coupling conditions at convergence. We generalize this approach here to the unsteady case, which leads to an optimized Schwarz waveform relaxation method. We first consider the case of a constant velocity field. If $\mathbf{a}\cdot\mathbf{n} \leq 0$, i.e. $\Gamma \equiv \Gamma_{out}$, the solution in Ω_2 does not depend on the conditions on Γ, and to obtain (6), Dirichlet conditions must be used for Ω_1. Now if $\mathbf{a}\cdot\mathbf{n} > 0$, i.e. $\Gamma \equiv \Gamma_{in}$, then we use the theory of absorbing boundary conditions to obtain optimal transmission conditions $\mathcal{B}_1$ and $\mathcal{B}_2$ for the algorithm:

$$\begin{cases} \mathcal{L}_{ad} u_1^{k+1} = f & \text{in } \Omega_1 \times (0,T), \\ u_1^{k+1}(\cdot,0) = u_0 & \text{in } \Omega_1, \\ \mathcal{B} u_1^{k+1} = g & \text{on } \partial\Omega \cap \partial\Omega_1, \\ \mathcal{B}_1 u_1^{k+1} = \mathcal{B}_1 u_2^k & \text{on } \Gamma \times (0,T), \end{cases} \quad \begin{cases} \mathcal{L}_a u_2^{k+1} = f & \text{in } \Omega_2 \times (0,T), \\ u_2^{k+1}(\cdot,0) = u_0 & \text{in } \Omega_2, \\ \mathcal{B} u_2^{k+1} = g & \text{on } \partial\Omega \cap \partial\Omega_2, \\ \mathcal{B}_2 u_2^{k+1} = \mathcal{B}_2 u_1^k & \text{on } \Gamma \times (0,T). \end{cases}$$

Using the error equations, one can show that if $\mathcal{B}_1$ is the advection operator, then we have convergence of the algorithm in two steps.

In the case of a non-constant velocity field, we propose precisely the same strategy, which leads to the algorithm:

$$\begin{cases} \mathcal{L}_{ad} u_1^{k+1} = f & \text{in } \Omega_1 \times (0,T), \\ u_1^{k+1}(\cdot,0) = u_0 & \text{in } \Omega_1, \\ \mathcal{B} u_1^{k+1} = g & \text{on } \partial\Omega \cap \partial\Omega_1, \\ \mathcal{L}_a u_1^{k+1} = \mathcal{L}_a u_2^k & \text{on } \Gamma_{in} \times (0,T), \\ u_1^{k+1} = u_2^k & \text{on } \Gamma_{out} \times (0,T). \end{cases} \quad \begin{cases} \mathcal{L}_a u_2^{k+1} = f & \text{in } \Omega_2 \times (0,T), \\ u_2^{k+1}(\cdot,0) = u_0 & \text{in } \Omega_2, \\ \mathcal{B} u_2^{k+1} = g & \text{on } \partial\Omega \cap \partial\Omega_2, \\ u_2^{k+1} = u_1^k & \text{on } \Gamma_{in} \times (0,T). \end{cases} \tag{8}$$

Note that if the sign of $\mathbf{a}\cdot\mathbf{n}$ is constant, provided you use the relation $\mathcal{L}_a u_2^k = f$ on $\Gamma_{in} \times (0,T)$, then algorithm (8) converges in two steps like algorithm (7). If not, our numerical results in the next section suggest that the algorithm has good convergence properties also, but it remains to prove convergence of the new algorithm in that case.

6 Numerical Results

We first consider the stationary case in 1d, with parameters $\nu = 0.1$, $c = 1$ and $f(x) = \sin(x) + \cos(x)$. In Figure 2, we show on the left the viscous and coupled solutions for $a = 1$, and on the right for $a = -1$. The interface Γ is at $x = 0$, and in each case the boundary conditions are chosen such that there is no boundary layer. One can clearly see that for $a > 0$, conditions (4) lead to a jump in the derivative at the interface, whereas with conditions (5), (6) the coupled solution and its derivative

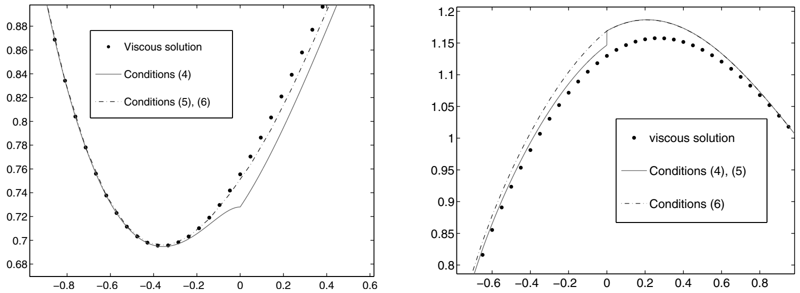


Fig. 2. Viscous and coupled solutions for $a > 0$ on the left and for $a < 0$ on the right.

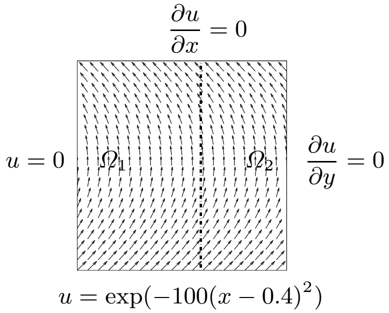

$\frac{\partial u}{\partial x} = 0$
$u = 0$
Ω_1
Ω_2
$\frac{\partial u}{\partial y} = 0$
$u = \exp(-100(x - 0.4)^2)$

Fig. 3. Domain Ω.

are continuous. For $a < 0$, conditions (4) lead to a discontinuity at the interface, whereas conditions (5), (6) lead to a continuous coupled solution. Note that the jump is proportional to ν, see [4].

In Figure 4, we compare the viscous and the coupled solutions for several values of ν in the L^2 norm in Ω_1 and Ω_2 when $a = 1$ and $a = -1$. The numerical results agree well with the theoretical results given in Section 4.

We next consider the time dependent case in two dimensions with a rotating velocity, as shown in Figure 3. The viscosity is $\nu = 0.001$, we work on the homogeneous equation $f \equiv 0$, and the rotating velocity is given by $a(x, y) = 0.5 - y$, $b(x, y) = 0.5$, such that $\mathbf{a} \cdot \mathbf{n}$ is positive on the first half of the interface and negative on the other half.

Figure 5 shows cross sections of the solution at $y = 0.3$ and $y = 0.5$ where $\mathbf{a} \cdot \mathbf{n} > 0$, and the information goes from Ω_1 to Ω_2 and stops diffusing after reaching the interface, and then cross sections at $y = 0.7$ and $y = 0.9$, where $\mathbf{a} \cdot \mathbf{n} < 0$, and diffusion sets in again after crossing the interface.

7 Conclusions

We have proposed a new set of coupling conditions which permits the replacement of the advection diffusion operator by the pure advection operator in regions where the viscosity is not very important. These new conditions retain better asymptotic

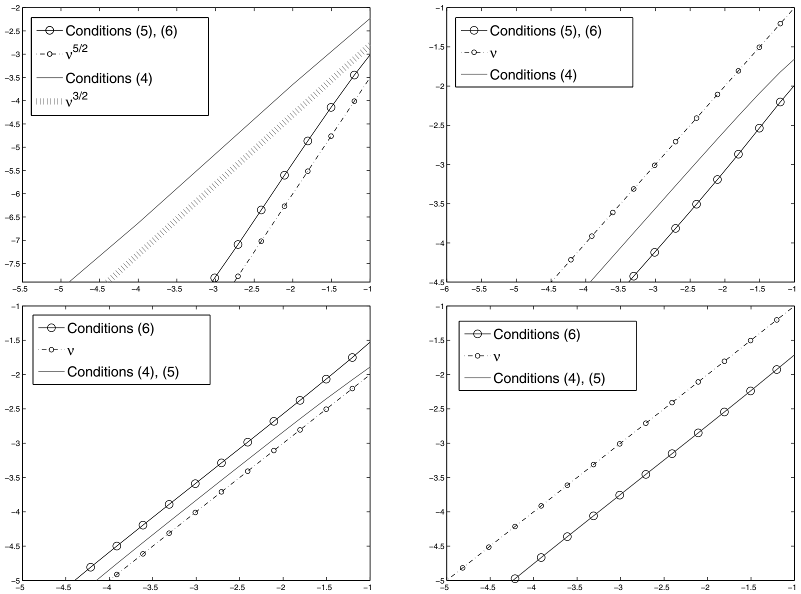


Fig. 4. 1 d case : L^2-error for $a = 1$ in Ω_1 on top left and in Ω_2 on top right, and for $a = -1$ in Ω_1 at the bottom left and in Ω_2 at the bottom right, versus ν.

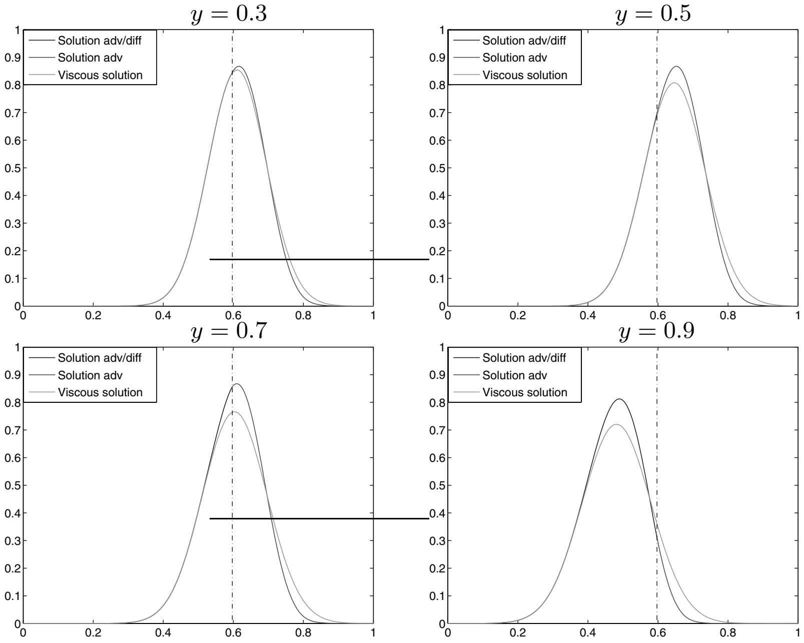


Fig. 5. Cross sections of the viscous and coupled solutions with conditions (6) at the final time for various positions y on the interface Γ.

approximation properties with respect to the fully viscous solution than earlier coupling conditions in the literature. We have also defined a rapidly converging iterative substructuring algorithm that uses computational transmission conditions which at convergence imply the new coupling conditions. While numerical experiments show good convergence properties of this new algorithm, it remains to prove convergence of the new algorithm.

References

1. E. Dubach, *Contribution à la résolution des equations fluides en domaine non borné*, PhD thesis, Universite Paris 13, Fevrier 1993.
2. O. Dubois, *Optimized Schwarz methods for the advection-diffusion equation*, Master's thesis, McGill University, 2003.
3. M. J. Gander, L. Halpern, and C. Japhet, *Optimized Schwarz algorithms for coupling convection and convection-diffusion problems*, in Thirteenth international conference on domain decomposition, N. Debit, M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, eds., ddm.org, 2001, pp. 253–260.
4. F. Gastaldi, A. Quarteroni, and G. S. Landriani, *On the coupling of two-dimensional hyperbolic and elliptic equations: analytical and numerical approach*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989, T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM, pp. 22–63.
5. C. Japhet, *Méthode de décomposition de domaine et conditions aux limites artificielles en mécanique des fluides: méthode optimisée d'ordre 2*, PhD thesis, Université Paris 13, 1998.
6. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.

Construction of a New Domain Decomposition Method for the Stokes Equations

Frédéric Nataf[1] and Gerd Rapin[2]

[1] CMAP, CNRS; UMR7641, Ecole Polytechnique, 91128 Palaiseau Cedex, France.
[2] Math. Dep., NAM, Georg-August-Universität Göttingen, D-37083, Germany.

Summary. We propose a new domain decomposition method for the Stokes equations in two and three dimensions. The algorithm, we propose, is very similar to an algorithm which is obtained by a Richardson iteration for the Schur complement equation using a Neumann-Neumann preconditioner. A comparison of both methods with the help of a Fourier analysis shows clearly the advantage of the new approach. This has also been validated by numerical experiments.

1 Introduction

In this paper we study a Neumann-Neumann type algorithm for the Stokes equations. The last decade has shown, that these kind of domain decomposition methods are very efficient. Most of the theoretical and numerical work has been carried out for symmetric second order problems, see [6]. Then the method has been extended to other problems, like the advection-diffusion equations ([1]) or recently the Stokes equations, cf. [5, 7].

In the case of two domains consisting of the two half planes it is well known, that the Neumann-Neumann preconditioner is an exact preconditioner for the Schur complement equation for scalar equations like the Laplace problem (cf. [6]). As we will show, this property could not be transfered to the vector valued Stokes problem due to the incompressibility constraint.

We will construct a method, which preserves this property. The first preliminary numerical results clearly indicate a better convergence behavior.

2 The preconditioned Schur Complement equation

In order to make the presentation as simple as possible we restrict ourselves to the two dimensional case. But the extension to the three dimensional case is straightforward.

Let $\Omega \subset \mathbb{R}^2$ be a bounded polygonal domain. The Stokes problem is a simple model for incompressible flows and is defined as follows: Find a velocity $\mathbf{u}$ and a pressure p, such that

$$\begin{aligned} -\nu\triangle\mathbf{u} + \nabla p = \mathbf{f}, \qquad \nabla\cdot\mathbf{u} = 0 \qquad \text{in } \Omega \\ \mathbf{u} \,=\, 0 \quad \text{on } \partial\Omega. \end{aligned} \tag{1}$$

$\mathbf{f} \in [L^2(\Omega)]^d$ is a source term and ν is the viscosity. In what follows, we denote the Stokes operator by $\mathcal{A}_{Stokes}(\mathbf{v}, q) := (-\nu\triangle\mathbf{v} + \nabla p, \nabla\cdot\mathbf{v})$.

2.1 Schur complement equation

Most of the domain decomposition methods for the Stokes equations use the classical sub-structuring or static condensation procedure. This means, that they end up with a Schur complement equation. Since the corresponding Steklov-Poincaré operator is badly conditioned, the application of suitable preconditioners is mandatory. One of the best-known preconditioner is the Neumann-Neumann preconditioner (cf. [7, 2, 5]).

Assume a bounded Lipschitz domain $\Omega \subset \mathbb{R}^2$ divided into two nonoverlapping subdomains Ω_1 and Ω_2. The interface is denoted by $\Gamma := \partial\Omega_1 \cap \partial\Omega_2$.

In the case of the Stokes equations an additional problem occurs. If we assume, that $\mathbf{u}_i \in [H^1(\Omega_i)]^2$ satisfies the incompressibility constraint, i.e. $\nabla\cdot\mathbf{u}_i = 0$, then the Green's formula yields $\int_{\partial\Omega_i} \mathbf{u}_i \cdot \mathbf{n}_i ds = 0$ for the trace of $\mathbf{u}_i$ where $\mathbf{n}_i$ is the outward normal of Ω_i. Therefore we have to consider the subspace

$$H_*^{\frac{1}{2}}(\Gamma) := \{\boldsymbol{\varphi} \in [H_{00}^{\frac{1}{2}}(\Gamma)]^2 \mid \int_\Gamma \boldsymbol{\varphi}\cdot\mathbf{n}_i ds = 0\}$$

of the trace space taking into account the homogeneous boundary conditions on $\partial\Omega_i \cap \partial\Omega$. We consider the operator

$$\begin{aligned} \Sigma : H_*^{\frac{1}{2}}(\Gamma) \;\times\; [L^2(\Omega)]^2 \quad &\rightarrow \quad [H^{-\frac{1}{2}}(\Gamma)]^2 \\ (\mathbf{u}_\Gamma, \mathbf{f}) \quad &\mapsto \quad \frac{1}{2}\left(\nu\frac{\partial\mathbf{u}_1}{\partial\mathbf{n}_1} - p_1\mathbf{n}_1\right)\Big|_\Gamma + \frac{1}{2}\left(\nu\frac{\partial\mathbf{u}_2}{\partial\mathbf{n}_2} - p_2\mathbf{n}_2\right)\Big|_\Gamma \end{aligned}$$

where $(\mathbf{u}_i, p_i) \in [H^1(\Omega_i)]^2 \times L_0^2(\Omega_i)$ are the unique solutions of the local Stokes problems

$$\begin{aligned} \mathcal{A}_{Stokes}(\mathbf{u}_i, p_i) \;&=\; (\mathbf{f}, 0) \quad \text{in } \Omega_i \\ \mathbf{u}_i = 0 \quad \text{on } \partial\Omega_i \cap \partial\Omega, \qquad & \mathbf{u}_i = \mathbf{u}_\Gamma \quad \text{on } \Gamma. \end{aligned}$$

It is clear, that the problem

$$\text{Find } \boldsymbol{\phi} \in H_*^{\frac{1}{2}}(\Gamma) \text{ such that } \quad \langle \Sigma(\boldsymbol{\phi}, 0), \psi\rangle = \langle -\Sigma(0, \mathbf{f}), \psi\rangle, \quad \forall\psi \in H_*^{\frac{1}{2}}(\Gamma) \tag{2}$$

is satisfied by the restriction of the continuous solution (1) to the interface Γ. $\langle\cdot,\cdot\rangle$ denotes the dual product $\langle\cdot,\cdot\rangle_{H^{-\frac{1}{2}}(\Gamma)\times H_{00}^{\frac{1}{2}}(\Gamma)}$.

2.2 Neumann-Neumann preconditioner

The Neumann-Neumann preconditioner of the Steklov-Poincaré operator $\mathbb{S} := \Sigma(\cdot, 0)$ is defined by

$$\mathcal{T} : (H^{-\frac{1}{2}}(\Gamma))^2 \;\rightarrow\; H_*^{\frac{1}{2}}(\Gamma), \qquad \boldsymbol{\phi} \;\mapsto\; \left(\frac{1}{2}(\mathbf{v}_{1,j} + \mathbf{v}_{2,j})|_\Gamma\right)_{j=1}^{2} .$$

where $\mathbf{v}_i = (v_{i,1}, v_{i,2}) \in [H^1(\Omega_i)]^2$ satisfies

$$\begin{array}{c} \mathcal{A}_{Stokes}(\mathbf{v}_i, q_i) \;=\; 0 \quad \text{in } \Omega_i \\ \mathbf{v}_i = 0 \quad \text{on } \partial\Omega_i \cap \partial\Omega, \qquad \dfrac{\partial \mathbf{v}_i}{\partial \mathbf{n}_i} - q_i \mathbf{n}_i = \boldsymbol{\phi} \quad \text{on } \Gamma. \end{array}$$

In order to keep the presentation simple we consider the following Richardson iteration for equation (2): Starting with an initial guess $\boldsymbol{\varphi}^0 \in H_*^{\frac{1}{2}}(\Gamma)$ we obtain

$$\boldsymbol{\varphi}_{k+1} = \boldsymbol{\varphi}_k - \mathcal{T}(\mathbb{S}\boldsymbol{\varphi}_k + \Sigma(0, f)), \qquad k = 0, 1, 2, \ldots. \tag{3}$$

Please notice that all $\boldsymbol{\varphi}_{k+1}$, $k \in \mathbb{N}$, satisfy $\displaystyle\int_{\partial\Omega_i} \boldsymbol{\varphi}_{k+1} \cdot \mathbf{n}_i ds = 0$. Thus after a proper initialization all iterations $\boldsymbol{\varphi}_k$ are elements of $H_*^{\frac{1}{2}}(\Gamma)$. Of course, in a practical implementation the Richardson iteration (3) would be replaced by a suitable Krylov method.

3 Smith Factorization

We first recall the definition of the Smith factorization of a matrix with polynomial entries and apply it to the Stokes system.

Theorem 1. *Let n be an integer and A an invertible $n \times n$ matrix with polynomial entries with respect to the variable λ: $A = (a_{ij}(\lambda))_{1 \le i,j \le n}$.*
Then, there exist matrices E, F and a diagonal matrix D with polynomial entries satisfying $A = EDF$.

More details can be found in [8]. We first formally take the Fourier transform of system (1) with respect to y (dual variable is k). We keep the partial derivatives in x since in the sequel we shall consider a model problem where the interface between the subdomains is orthogonal to the x direction. We note that

$$\hat{A}_{Stokes} = \begin{pmatrix} -\nu(\partial_{xx} - k^2) & 0 & \partial_x \\ 0 & -\nu(\partial_{xx} - k^2) & ik \\ \partial_x & ik & 0 \end{pmatrix}. \tag{4}$$

We perform the Smith factorization of $\hat{A}_{Stokes}$ by considering it as a matrix with polynomials in ∂_x. Applying the inverse Fourier transform yields

$$A_{Stokes} = EDF \tag{5}$$

where $D_{11} = D_{22} = 1$ and $D_{33} = -\nu\triangle^2$ and

$$E := T_2^{-1}\begin{pmatrix} -\nu\triangle\partial_y & \nu\partial_{xxx} & -\nu\partial_x \\ 0 & T_2 & 0 \\ \partial_{xy} & -\partial_{xx} & 1 \end{pmatrix}, \qquad F := \begin{pmatrix} \nu\partial_{yy} & \nu\partial_{yx} & \partial_x \\ 0 & -\nu\triangle & \partial_y \\ 0 & 1 & 0 \end{pmatrix}$$

where T_2 is a differential operator in the y-direction whose symbol is $i\nu k^3$.

This suggests that the derivation of a DDM for the bi-Laplacian is a key ingredient for a DDM for the Stokes system. One should note that a stream function formulation gives the same differential equation for the stream function.

4 The new algorithm

Using the Smith factorization (5) the new algorithm can be derived from a standard Neumann-Neumann algorithm for the Bi-Laplacian, which converges in two steps in the case of the plane divided into the two half planes. For details we refer to [3, 4].

The new algorithm is very similar to the algorithm given by (3). Again, each iteration step requires the solution of two local boundary value problems with Dirichlet and Neumann boundary conditions. But this time we distinguish between tangential parts and normal parts of the velocity and impose different boundary conditions for each part.

In order to write the resulting algorithm in an intrinsic form, we introduce the stress $\boldsymbol{\sigma}(\mathbf{u}, p) = \nu\dfrac{\partial\mathbf{u}}{\partial\mathbf{n}} - p\mathbf{n}$ on the interface for a velocity $\mathbf{u}$ and a pressure p. For any vector $\mathbf{u}$ its normal (resp. tangential) component on the interface is $\mathbf{u}_n$ (resp. $\mathbf{u}_\tau$). We denote by $\boldsymbol{\sigma}_n$ and $\boldsymbol{\sigma}_\tau$ the normal and tangential parts of $\boldsymbol{\sigma}$, respectively. We consider a decomposition of the domain 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 1. Starting with an initial guess satisfying $\mathbf{u}^0_{i,\tau_i} = \mathbf{u}^0_{j,\tau_j}$ and $\boldsymbol{\sigma}^0_{i,n_i} = -\boldsymbol{\sigma}^0_{j,n_j}$ on Γ_{ij}, the correction step is defined as follows for $1 \le i \le N$:

$$\begin{aligned} A_{Stokes}(\tilde{\mathbf{u}}_i^{n+1}, \tilde{p}_i^{n+1})^T &= 0 \text{ in } \Omega_i, \qquad \tilde{\mathbf{u}}_i^{n+1} = 0 \text{ on } \partial\Omega_i \cap \partial\Omega \\ \tilde{\mathbf{u}}_{i,n_i}^{n+1} &= -(\mathbf{u}^n_{i,n_i} - \mathbf{u}^n_{j,n_j})/2 \text{ on } \Gamma_{ij} \\ \sigma_{\tau_i}(\tilde{\mathbf{u}}_i^{n+1}, \tilde{p}_i^{n+1}) &= -(\sigma_{\tau_i}(\tilde{\mathbf{u}}_i^n, \tilde{p}_i^n) + \sigma_{\tau_j}(\tilde{\mathbf{u}}_j^n, \tilde{p}_j^n))/2 \text{ on } \Gamma_{ij} \end{aligned}$$

followed by an updating step:

$$\begin{aligned} A_{Stokes}(\mathbf{u}_i^{n+1}, p_i^{n+1})^T &= f \text{ in } \Omega_i \qquad \mathbf{u}_i^{n+1} = 0 \text{ on } \partial\Omega_i \cap \partial\Omega \\ \mathbf{u}_{i,\tau_i}^{n+1} &= \mathbf{u}^n_{i,\tau_i} + (\tilde{\mathbf{u}}^{n+1}_{i,\tau_i} + \tilde{\mathbf{u}}^{n+1}_{j,\tau_j})/2 \text{ on } \Gamma_{ij} \\ \boldsymbol{\sigma}_{n_i}(\mathbf{u}_i^{n+1}, p_i^{n+1}) &= \boldsymbol{\sigma}_{n_i}(\mathbf{u}_i^n, p_i^n) \\ &\quad + (\boldsymbol{\sigma}_{n_i}(\tilde{\mathbf{u}}_i^{n+1}, \tilde{p}_i^{n+1}) - \boldsymbol{\sigma}_{n_j}(\tilde{\mathbf{u}}_j^{n+1}, \tilde{p}_j^{n+1}))/2 \text{ on } \Gamma_{ij}. \end{aligned}$$

The boundary conditions in the correction step involve the normal velocity and the tangential stress whereas in the updating step they involve the tangential velocity and the normal stress. In 3D, the algorithm has the same definition. By construction, it converges in two steps.

Theorem 2. *For a domain $\Omega = \mathbb{R}^2$ divided into two non overlapping half planes, the algorithm 1 converges in two iterations.*

5 Analysis of the Neumann-Neumann Algorithm

Here we focus on the Neumann-Neumann algorithm and we will use the Smith factorization in order prove that the Neumann-Neumann algorithm (3) does not converge in only two steps in the case of the plane $\Omega = \mathbb{R}^2$ divided into the two half planes $\Omega_1 := (-\infty, 0) \times \mathbb{R}$ and $\Omega_1 := (0, \infty) \times \mathbb{R}$. Therefore the Neumann-Neumann preconditioner is not an exact preconditioner.

5.1 Reformulation of the algorithm

For the above decomposition the Smith factorization enables us to formulate the Neumann-Neumann algorithm (3) of the Stokes equations solely in terms of the second velocity components. The third row of equation of (5) gives $-\triangle^2 z = g$ with $z = (F(\mathbf{u}, p))_3 = u_2$ and $g = (E^{-1}(\mathbf{f}, 0))_3$. Then the first velocity and the pressure component can be eliminated in the interface conditions using the Stokes equations. Let us define $\mathcal{L}u := -\nu \triangle u$.

We end up with the following algorithm: Starting with an initial guess

$$u_1^n = u_2^n, \quad \frac{\partial}{\partial \mathbf{n}_1}(\mathcal{L} - \nu\partial_{yy})u_1^n = -\frac{\partial}{\partial \mathbf{n}_2}(\mathcal{L} - \nu\partial_{yy})u_2^n \quad \text{on } \Gamma$$

the correction step for $n = 1, 2, \ldots$ is given by

$$\begin{aligned} -\nu\triangle^2 v_i^n &= 0 \quad \text{in } \Omega_i & (6)\\ \frac{\partial v_i^n}{\partial \mathbf{n}_i} &= -\frac{1}{2}\left(\frac{\partial u_1^{n-1}}{\partial \mathbf{n}_1} + \frac{\partial u_2^{n-1}}{\partial \mathbf{n}_2}\right) \quad \text{on } \Gamma & (7)\\ (\mathcal{L} - \nu\partial_{yy})v_i^n &= -\frac{1}{2}\left(\mathcal{L}u_i^{n-1} - \mathcal{L}u_{3-i}^{n-1}\right) \quad \text{on } \Gamma & (8) \end{aligned}$$

for $i = 1, 2$. The updating step is defined by

$$\begin{aligned} -\nu\triangle^2 u_i^n &= g \quad \text{in } \Omega_i, & (9)\\ u_i^n &= u_i^{n-1} + \frac{1}{2}\left(v_1^n + v_2^n\right) \quad \text{on } \Gamma & (10)\\ \frac{\partial}{\partial \mathbf{n}_i}(\mathcal{L} - \nu\partial_{yy})u_i^n &= \frac{\partial}{\partial \mathbf{n}_i}(\mathcal{L} - \nu\partial_{yy})u_i^{n-1} \\ &\quad + \frac{1}{2}\frac{\partial}{\partial \mathbf{n}_i}(\mathcal{L} - \nu\partial_{yy})\left(v_1^n + v_2^n\right) \quad \text{on } \Gamma & (11) \end{aligned}$$

with $g = (E^{-1}(\mathbf{f}, 0))_3$ and $i = 1, 2$.

5.2 A Fourier Analysis

We start with the reformulated algorithm (6)-(8), (9)-(11). Again, using the linearity of the scheme, we obtain for the error $\tilde{e}_i^n$ in the n-th iteration step in subdomain Ω_i the update formula $\tilde{e}_i^n = \tilde{e}_i^{n-1} + \tilde{z}_i^n$ where $\tilde{z}_i^n$ satisfies

$$\begin{aligned} -\nu\triangle^2 \tilde{z}_i^n &= 0 \quad \text{in } \Omega_i & (12)\\ \tilde{z}_i^n &= \frac{1}{2}(v_1^n + v_2^n) \quad \text{on } \Gamma & (13)\\ \partial_x(-\nu\partial_{xx} - 2\nu\partial_{yy})\tilde{z}_i^n &= \frac{1}{2}\partial_x(-\nu\partial_{xx} - 2\nu\partial_{yy})(v_1^n + v_2^n) \quad \text{on } \Gamma. & (14) \end{aligned}$$

v_1^n, v_2^n are the solutions of the correction step (6)-(8) with right hand side

$$H_{NN}^n := -\frac{1}{2}\nu\left(\frac{\partial\triangle\tilde{e}_1^n}{\partial\mathbf{n}_1}+\frac{\partial\triangle\tilde{e}_2^n}{\partial\mathbf{n}_2}\right)\bigg|_{x=0}, \quad K_{NN}^n := -\frac{1}{2}\left(\frac{\partial\tilde{e}_1^n}{\partial\mathbf{n}_1}+\frac{\partial\tilde{e}_2^n}{\partial\mathbf{n}_2}\right)\bigg|_{x=0}.$$

Let us start with the correction step. After a Fourier transform we obtain

$$\nu(-\partial_{xxxx}+2k^2\partial_{xx}-k^4)\hat{v}_i^n(x,k)=0.$$

For a fixed k these are ordinary differential equations in x with solutions

$$\hat{v}_1^n(x,k) = C_{11}^n\exp(|k|x)+C_{12}^n x\exp(|k|x) \quad (15)$$

$$\hat{v}_2^n(x,k) = C_{21}^n\exp(-|k|x)+C_{22}^n x\exp(-|k|x). \quad (16)$$

Using the interface conditions (7) we get

$$\hat{K}_{NN}^{n-1} = |k|C_{11}^n + C_{12}^n, \qquad -\hat{K}_{NN}^{n-1} = -|k|C_{21}^n + C_{22}^n.$$

The second interface condition (8) yields

$$\hat{H}_{NN}^{n-1} = -\nu|k|^2C_{11}^n - 2\nu|k|C_{12}^n, \quad -\hat{H}_{NN}^{n-1} = \nu|k|^2C_{21}^n + 2\nu|k|C_{22}.$$

Thus, we have four linear equations for the four unknowns C_{11}^n, C_{12}^n, C_{21}^n, and C_{22}^n. After simple computations we obtain

$$C_{11}^n = \frac{2}{3}\frac{1}{|k|}\hat{K}_{NN}^{n-1}+\frac{\hat{H}_{NN}^{n-1}}{3\nu|k|^2}, \quad C_{12}^n = \frac{1}{3}\hat{K}_{NN}^{n-1}-\frac{\hat{H}_{NN}^{n-1}}{3\nu|k|}$$

$$C_{21}^n = \frac{2}{3}\frac{1}{|k|}\hat{K}_{NN}^{n-1}-\frac{\hat{H}_{NN}^{n-1}}{3\nu|k|^2}, \quad C_{22}^n = -\frac{1}{3}\hat{K}_{NN}^{n-1}-\frac{\hat{H}_{NN}^{n-1}}{3\nu|k|}.$$

Next, we use the solutions of the correction step in order to compute the right hand side of the updating step

$$\tilde{f}^n := \frac{1}{2}\left(\hat{v}_1^n+\hat{v}_2^n\right)|_{x=0} = \frac{1}{2}(C_{11}^n+C_{21}^n) = \frac{2}{3}\frac{\hat{K}_{NN}^{n-1}}{|k|}$$

$$\tilde{g}^n := \left(\frac{1}{2}\partial x(-\nu\partial_{xx}-2\nu\partial_{yy})(\hat{v}_1^n+\hat{v}_2^n)\right)\bigg|_{x=0} = \frac{2}{3}|k|\hat{H}_{NN}^{n-1}.$$

Again, after Fourier transform the solutions of (12) are given by

$$\hat{z}_1^n(x,k) = D_{11}^n\exp(|k|x)+D_{12}^n x\exp(|k|x),$$

$$\hat{z}_2^n(x,k) = D_{21}^n\exp(-|k|x)+D_{22}^n x\exp(-|k|x)$$

using that the solutions vanish at infinity. Inserting the boundary condition (13) yields $D_{11}^n = D_{21}^n = \tilde{f}^n = \frac{2}{3}\frac{\hat{K}_{NN}^n}{|k|}$. Now, we consider the second transmission condition (14). Then we can derive

$$D_{12}^n = -\frac{2}{3}\frac{1}{\nu|k|}\hat{H}_{NN}^{n-1}+\frac{2}{3}\hat{K}_{NN}^{n-1}, \quad D_{22}^n = -\frac{2}{3}\frac{\hat{H}_{NN}^{n-1}}{\nu|k|}-\frac{2}{3}\hat{K}_{NN}^{n-1}.$$

This result can be used to compute $\hat{H}_{NN}^n$ and $\hat{K}_{NN}^n$. They are given by

$$\hat{K}^n_{NN} = \hat{K}^{n-1}_{NN} - \frac{1}{2}\left(\frac{\partial \hat{z}^n_1}{\partial x} - \frac{\partial \hat{z}^n_2}{\partial x}\right)\bigg|_{x=0} = -\frac{1}{3}\hat{K}^{n-1}_{NN}$$

resp.

$$\hat{H}^n_{NN} = \hat{H}^{n-1}_{NN} - \frac{1}{2}\left(-\nu\partial_{xx}(\hat{z}^n_1 - \hat{z}^n_2)\right)|_{x=0} = -\frac{1}{3}\hat{H}^{n-1}_{NN}.$$

Let us summarize the result

Theorem 3. *Consider the case $\Omega = \mathbb{R}^2$. If the domain Ω is divided into the two half planes, the preconditioned Richardson iteration (3) of the Schur complement equation converges. Moreover, the error is reduced by the factor* 3 *in each iteration step.*

6 Preliminary Numerical Results

The domain $\Omega = (-A, B) \times (0, 1)$ is decomposed into two subdomains $\Omega_1 = (-A, 0) \times (0, 1)$ and $\Omega_2 = (0, B) \times (0, 1)$. We compare the new algorithm to the iterative version of the Neumann-Neumann algorithm. The stopping criteria is that the jumps of the normal derivative of the tangential component of the velocity has been reduced by the factor 10^{-4}. In table 1 (left) $A = B = 1$, we see that both algorithms are insensitive with respect to the mesh size. Of course, due to the discrete approximation we cannot expect the optimal convergence in two steps. But we only need one more step to achieve the error bound. We have also varied the width of the subdomains (middle table). As expected the convergence of the Neumann-Neumann method deteriorates. For large aspect ratios, the method diverges (– in the table), since there exists an eigenvalue of the operator corresponding to the Richardson iteration with a modulus larger than 1. But in this case, the convergence can still be enforced by its use as a preconditioner in a Krylov method as it is usually the case. Our new algorithm seems to be surprisingly robust with respect to the subdomain widths. For moderate variations we always need 3 iterations steps. If we choose very thin subdomains, for instance $A = 1$, $B = 20$, the stopping criterion is met in only 7 steps. In table 1 (right), we have added a reaction term $c > 0$ to the first two equations of the Stokes system. For instance c might be the inverse of the time step in a time-dependent computation. We see that the new algorithm is fairly stable.

h	new algo	N-N
0.02	3	10
0.025	3	12
0.05	3	11
0.5	3	11
0.1	3	11
0.2	3	10

B	new algo	N-N
1	3	11
2	3	12
3	3	11
5	3	15
10	3	–
20	7	–

c	new algo	N-N
0.001	3	11
0.01	3	16
0.1	3	19
1	3	19
10	3	16
100	3	10

Table 1. Number of iterations for different mesh sizes (left), aspect ratio (middle) and different reaction terms (right).

References

1. Y. Achdou, P. L. Tallec, F. Nataf, and M. Vidrascu, *A domain decoposition preconditioner for an advection-diffusion problem*, Comp. Meth. Appl. Mech. Engrg, 184 (2000), pp. 145–170.
2. M. Ainsworth and S. Sherwin, *Domain decomposition preconditioners for p and hp finite element approximations of Stokes equations*, Comput. Methods Appl. Mech. Engrg., 175 (1999), pp. 243–266.
3. V. Dolean, F. Nataf, and G. Rapin, *New constructions of domain decomposition methods for systems of PDEs*, C.R. Math. Acad. Sci. Paris, 340 (2005), pp. 693–696.
4. ———, *Deriving a new domain decomposition method for the Stokes equations unsing the Smith factorization*, tech. rep., Georg-August University Göttingen, 2006. Submitted.
5. L. F. Pavarino and O. B. Widlund, *Balancing Neumann-Neumann methods for incompressible Stokes equations*, Comm. Pure Appl. Math., 55 (2002), pp. 302–335.
6. Y.-H. D. Roeck and P. L. Tallec, *Analysis and test of a local domain decomposition preconditioner*, in Proceedings of the Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., Philadelphia, PA, 1991, SIAM, pp. 112–128.
7. P. L. 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 (1997), pp. 361–379.
8. J. T. Wloka, B. Rowley, and B. Lawruk, *Boundary Value Problems for Elliptic Systems*, Cambridge University Press, 1995.

MINISYMPOSIUM 4: Domain Decomposition Methods for Electromagnetic Field Problems

Organizers: Ronald H. W. Hoppe[1] and Jin-Fa Lee[2]

[1] University of Houston. rohop@math.uh.edu
[2] Ohio State University. jinlee@ece.osu.edu

During the last couple of years, domain decomposition techniques have been developed, analyzed and implemented for the numerical solution of Maxwell's equations in both time and frequency domains. Moreover, these methods have been successfully applied to various technologically relevant problems ranging from antenna design to high power electronics. This minisymposium brings together scientists from mathematical and electroengineering communities to present the latest scientific results on theoretical and algorithmic aspects, as well as on innovative applications.

A Domain Decomposition Approach for Non-conformal Couplings between Finite and Boundary Elements for Electromagnetic Scattering Problems in $\mathcal{R}^3$ *

Marinos Vouvakis and Jin-Fa Lee

ElectroScience Laboratory, Electrical and Computer Engineering Department,
Ohio State University, 1320 Kinnear Rd., Columbus, OH 43212, USA.
vouvakis.1@osu.edu, lee.1863@osu.edu

1 Introduction

To solve electromagetic scattering problems in $\mathcal{R}^3$, the popular approach is to combine and couple finite and boundary elements. Common engineering practises in coupling finite and boundary elements usually result in non-symmetric and non-variational formulations [5, 8]. The symmetric coupling between finite and boundary elements was first proposed by Costabel [2] in 1987. Since then, quite a few papers have been published on the topic of symmetric couplings. Among them, we list references [3, 4, 12, 7]. In particular, references [4, 12, 7] deal with variational formulations for solving electromagnetic wave radiation and scattering problems. Although the formulations detailed in [4, 12, 7] result in symmetric couplings between finite and boundary elements, they still suffer the notorious internal resonances. The purpose of this chapter is to present a variational formulation, which couples finite and boundary elements through non-conformal meshes. The formulation results in matrix equations that are symmetric, coercive, and free of internal resonances.

Our plan for this chapter is as follows. Section 2 details the proposed variational formulation for non-conformal couplings between finite and boundary elements. In section 3, we show that, through a box-shaped computational domain, the proposed formulation is free of internal resonances and it satisfies the C.B.S inequality [1]. Moreover, in section 3 we validate the accuracy of the proposed formulation by a complex scattering problem. A brief conclusion is provided in section 4.

2 Formulation

2.1 Boundary Value Problems

This chapter considers the solution of an electromagnetic scattering problem in $\mathcal{R}^3$. A finite computational domain, $\Omega \subset \mathcal{R}^3$, encloses all the scatterers inside. The

*This project is supported by AFOSR MURI Grant #FA9550-04-1-0359

exterior region, $\Omega^c = \mathcal{R}^3/\overline{\Omega}$, is then homogeneous and assumed to be free space. Let $\mathbf{E}$ denotes the scattered electric field in the exterior region Ω^c and the total electric field inside Ω. It is then the solution of the transmission problem [4]:

$$\begin{aligned}
\nabla \times \nabla \times \mathbf{E} - k^2\mathbf{E} = &\quad 0 \quad \text{in } \Omega^c \\
\nabla \times \frac{1}{\mu_r}\nabla \times \mathbf{E} - k^2\epsilon_r\mathbf{E} = &\quad 0 \quad \text{in } \Omega \\
[\gamma_t\mathbf{E}]_\Gamma = \gamma_t\mathbf{E}^{inc}, [\frac{1}{\mu_r}\gamma_N\mathbf{E}]_\Gamma = &\ \gamma_N\mathbf{E}^{inc} \text{ on } \Gamma \\
\lim_{|\mathbf{x}|\to\infty} \nabla \times \mathbf{E} \times \mathbf{x} - ik|\mathbf{x}|\mathbf{E} = &\quad 0
\end{aligned} \tag{1}$$

In Eq. (1), k is the wavenumber in free space, the two surface trace operators are $\gamma_t\mathbf{E} = \mathbf{n} \times \mathbf{E} \times \mathbf{n}$ for the tangential components of $\mathbf{E}$ on Γ and $\gamma_N\mathbf{E} = \nabla \times \mathbf{E} \times \mathbf{n}$ for the "magnetic trace" on Γ. The surface unit normal $\mathbf{n}$ points from Ω towards the exterior region Ω^c. Finally, $[\gamma\phi]_\Gamma = \gamma\phi|_\Omega - \gamma\phi|_{\Omega^c}$ denotes the jump of a function ϕ across Γ.

The current formulation starts first by introducing two "cement" variables [6], $\mathbf{j}^-$ and $\mathbf{j}^+$, on the boundary Γ. These two cement variables are related to the electric currents on Γ in Ω and Ω^c, respectively. Subsequently, the original transmission problem Eq. (1) can be stated alternatively as:

$$\begin{aligned}
&\textbf{in } \Omega \\
\nabla \times \frac{1}{\mu_r}\nabla \times \mathbf{E} - k^2\epsilon_r\mathbf{E} &= 0 \\
\frac{1}{\mu_r}\gamma_N\mathbf{E} &= \mathbf{j}^-
\end{aligned} \tag{2}$$

$$\begin{aligned}
&\textbf{in } \Omega^c \\
\nabla \times \nabla \times \mathbf{E} - k^2\mathbf{E} &= 0 \\
\lim_{|\mathbf{x}|\to\infty} \nabla \times \mathbf{E} \times \mathbf{x} - ik|\mathbf{x}|\mathbf{E} &= 0 \\
-\gamma_N\mathbf{E} &= \mathbf{j}^+
\end{aligned} \tag{3}$$

$$\begin{aligned}
&\textbf{Transmission Conditions on } \Gamma \\
\mathbf{e}^- - \mathbf{e}^+ &= \gamma_t\mathbf{E}^{inc} \\
\mathbf{j}^- + \mathbf{j}^+ &= \gamma_N\mathbf{E}^{inc}
\end{aligned} \tag{4}$$

However, direct numerical implementation based on the transmission conditions (4) is not desirable since they are closely related to the Dirichlet-to-Neumann mappings, which usually subject the sub-domains to the "internal resonances" during the solution process. Taking our cue from the domain decomposition literature, we simply replace (4) by the Robin transmission conditions [6]. Namely,

$$\begin{aligned}
&\textbf{Robin Transmission Conditions on } \Gamma \\
-ik\mathbf{e}^- + \mathbf{j}^- &= -ik\mathbf{e}^+ - \mathbf{j}^+ - \mathbf{f}^{inc} \\
-ik\mathbf{e}^+ + \mathbf{j}^+ &= -ik\mathbf{e}^- - \mathbf{j}^- + \mathbf{g}^{inc}
\end{aligned} \tag{5}$$

where $\mathbf{f}^{inc} = ik\gamma_t\mathbf{E}^{inc} + \gamma_N\mathbf{E}^{inc}$ and $\mathbf{g}^{inc} = ik\gamma_t\mathbf{E}^{inc} - \gamma_N\mathbf{E}^{inc}$.

2.2 Galerkin Variational Formulation

From the physical consideration that both the electric and magnetic energies of the system need be finite, it is transparent to see that the vector field $\mathbf{E}$ in Eq. (1) resides in the product space $\mathbf{H}_0(\mathbf{curl};\Omega) \times \mathbf{H}_{loc}(\mathbf{curl};\Omega^c)$ [4]. To establish the proper spaces of the tangential traces $\mathbf{e}^-$, $\mathbf{e}^+$ as well as the cement variables $\mathbf{j}^-$ and $\mathbf{j}^+$, we borrow heavily from [4] the following results:

Theorem 1. *The trace mappings* $\gamma_t^+ : \boldsymbol{H}_{loc}(\boldsymbol{curl};\Omega^c) \mapsto \boldsymbol{H}^{-1/2}\left(\boldsymbol{curl}_\Gamma,\Gamma^+\right)$, $\gamma_t^- : \boldsymbol{H}_0(\boldsymbol{curl};\Omega) \mapsto \boldsymbol{H}^{-1/2}\left(\boldsymbol{curl}_\Gamma,\Gamma^-\right)$ *are continuous and surjective. Moreover, the traces* $\gamma_N^\pm$ *furnish continuous mappings:* $\gamma_N^+ : \boldsymbol{H}_{loc}\left(\boldsymbol{curl}^2;\Omega^c\right) \mapsto \boldsymbol{H}^{-1/2}\left(div_\Gamma,\Gamma^+\right)$ *and* $\gamma_N^- : \boldsymbol{H}\left(\boldsymbol{curl}^2;\Omega\right) \mapsto \boldsymbol{H}^{-1/2}\left(div_\Gamma,\Gamma^-\right)$.

Now we are ready to state the variational formulation which couples finite and boundary elements on non-conformal meshes. By non-conformity, we refer to the fact that the triangulation on Γ^- needs not be the same as the triangulation on Γ^+. This non-conformity feature provides two major benefits: (a) different orders of polynomial approximations can be employed separately for finite elements and boundary elements. Subsequently, the triangulations on Γ^- and Γ^+ would require drastically different spatial resolutions; and, (b) in the process of goal-oriented adaptive mesh refinements [11], the triangulation on Γ^- often become un-necessary fine in certain regions for the boundary elements. The non-conformal coupling approach allows for a more uniform triangulation on Γ^+ and hence can greatly reduce the computational burden.

In Ω, the variational formulation for the finite elements can be stated as

$$\begin{gathered}\text{Given a } \mathbf{j}^- \in \mathbf{H}^{-1/2}\left(div_\Gamma,\Gamma^-\right)\text{, find } \mathbf{E} \in \mathbf{H}_0(\mathbf{curl};\Omega) \text{ such that}\\ a(\mathbf{v},\mathbf{E}) - \left\langle \gamma_t\mathbf{v},\mathbf{j}^-\right\rangle_{\Gamma^-} = 0 \qquad (6)\\ \forall \mathbf{v} \in \mathbf{H}_0(\mathbf{curl};\Omega)\end{gathered}$$

with $a(\mathbf{v},\mathbf{E}) = \int_\Omega \left[\nabla\times\mathbf{v}\cdot\frac{1}{\mu_r}\nabla\times\mathbf{E} - k^2\mathbf{v}\cdot\epsilon_r\mathbf{E}\right] dV$ and $\langle\beta,\lambda\rangle_{\Gamma\pm} = \int_{\Gamma\pm}(\beta\cdot\lambda)\,dS$.

As for the exterior region Ω^c, we start with the Stratton-Chu representation formula [4]

$$\mathbf{E}(\mathbf{x}) = \Psi_M\left(\mathbf{e}^+\right)(\mathbf{x}) - \Psi_A\left(\mathbf{j}^+\right)(\mathbf{x}) - \frac{1}{k^2}\nabla\Psi_V\left(\nabla_\Gamma\cdot\mathbf{j}^+\right)(\mathbf{x}) \quad \mathbf{x}\notin\Gamma \tag{7}$$

Here $\Psi_M(\cdot), \Psi_A(\cdot)$, and $\Psi_V(\cdot)$ are potentials. Ψ_V is the scalar single layer potential given by

$$\Psi_V(\phi)(\mathbf{x}) = \int_{\Gamma+} G(\mathbf{x},\mathbf{y})\,\phi(\mathbf{y})\,dS(\mathbf{y}) \quad \mathbf{x}\notin\Gamma \tag{8}$$

with the Helmholtz kernel $G(\mathbf{x},\mathbf{y}) = \dfrac{\exp(ik|\mathbf{x}-\mathbf{y}|)}{4\pi|\mathbf{x}-\mathbf{y}|}$, $\mathbf{x}\neq\mathbf{y}$. Ψ_A is the vector version of the single layer potential; and, Ψ_M is the vector double layer potential given by

$$\Psi_M(\mathbf{v})(\mathbf{x}) = \int_{\Gamma+} (\nabla_{\mathbf{y}} G(\mathbf{x},\mathbf{y}) \times \mathbf{v})\, dS(\mathbf{y}) \tag{9}$$

The variational formulation for the surface traces, $\mathbf{e}^+$ and $\mathbf{j}^+$, can be obtained using the exterior Calderon projector [4]. We write:

$$\begin{aligned} &\text{Find } \mathbf{e}^+ \in \mathbf{H}^{-1/2}\left(\mathbf{curl}_\Gamma, \Gamma^+\right) \text{ and } \mathbf{j}^+ \in \mathbf{H}^{-1/2}\left(div_\Gamma, \Gamma^+\right) \text{ such that} \\ &\left\langle \lambda^+, \mathbf{e}^+ \right\rangle_{\Gamma+} = \left\langle \lambda^+, \left(\frac{1}{2}\mathcal{I} + \mathcal{C}\right)\left(\mathbf{e}^+\right) \right\rangle_{\Gamma+} - \left\langle \lambda^+, \mathcal{S}\left(\mathbf{j}^+\right) \right\rangle_{\Gamma+} \\ &\left\langle \beta^+, \mathbf{j}^+ \right\rangle_{\Gamma+} = \left\langle \beta^+, \mathcal{N}\left(\mathbf{e}^+\right) \right\rangle_{\Gamma+} + \left\langle \beta^+, \left(\frac{1}{2}\mathcal{I} - \mathcal{B}\right)\left(\mathbf{j}^+\right) \right\rangle_{\Gamma+} \\ &\forall \beta^+ \in \mathbf{H}^{-1/2}\left(\mathbf{curl}_\Gamma, \Gamma^+\right) \text{ and } \lambda^+ \in \mathbf{H}^{-1/2}\left(div_\Gamma, \Gamma^+\right). \end{aligned} \tag{10}$$

where the operators are:

$$\begin{aligned} \mathcal{S} &:= \gamma_t \Psi_S && : \mathbf{H}^{-1/2}(div_\Gamma, \Gamma) \mapsto \mathbf{H}^{-1/2}(\mathbf{curl}_\Gamma, \Gamma) \\ \mathcal{B} &:= \frac{1}{2}\left(\gamma_N^- + \gamma_N^+\right)\Psi_A && : \mathbf{H}^{-1/2}(div_\Gamma, \Gamma) \mapsto \mathbf{H}^{-1/2}(div_\Gamma, \Gamma) \\ \mathcal{C} &:= \frac{1}{2}\left(\gamma_t^- + \gamma_t^+\right)\Psi_M && : \mathbf{H}^{-1/2}(\mathbf{curl}_\Gamma, \Gamma) \mapsto \mathbf{H}^{-1/2}(\mathbf{curl}_\Gamma, \Gamma) \\ \mathcal{N} &:= \gamma_N \Psi_M && : \mathbf{H}^{-1/2}(\mathbf{curl}_\Gamma, \Gamma) \mapsto \mathbf{H}^{-1/2}(div_\Gamma, \Gamma) \end{aligned} \tag{11}$$

where $\Psi_S(\mathbf{j}) = \Psi_A(\mathbf{j}) + \frac{1}{k^2}\nabla\Psi_V(\nabla_\Gamma \cdot \mathbf{j})$.

Moreover, the corresponding variational statement for the transmission conditions described in Eq. (5) is

$$\begin{aligned} &\text{Find } \left(\mathbf{e}^-, \mathbf{e}^+\right) \in \mathbf{H}^{-1/2}\left(\mathbf{curl}_\Gamma, \Gamma^-\right) \times \mathbf{H}^{-1/2}\left(\mathbf{curl}_\Gamma, \Gamma^+\right) \text{ and} \\ &\left(\mathbf{j}^-, \mathbf{j}^+\right) \in \mathbf{H}^{-1/2}\left(div_\Gamma, \Gamma^-\right) \times \mathbf{H}^{-1/2}\left(div_\Gamma, \Gamma^+\right) \text{ such that} \end{aligned}$$

$$\begin{aligned} \left\langle \lambda^-, \mathbf{e}^- \right\rangle_{\Gamma-} + \frac{i}{k}\left\langle \lambda^-, \mathbf{j}^- \right\rangle_{\Gamma-} &= \left\langle \lambda^-, \mathbf{e}^+ \right\rangle_{\Gamma-} - \frac{i}{k}\left\langle \lambda^-, \mathbf{j}^+ \right\rangle_{\Gamma-} - \frac{i}{k}\left\langle \lambda^-, \mathbf{f}^{inc} \right\rangle \Gamma^- \\ -ik\left\langle \beta^-, \mathbf{e}^- \right\rangle_{\Gamma-} + \left\langle \beta^-, \mathbf{j}^- \right\rangle_{\Gamma-} &= -ik\left\langle \beta^-, \mathbf{e}^+ \right\rangle_{\Gamma-} - \left\langle \beta^-, \mathbf{j}^+ \right\rangle_{\Gamma-} - \left\langle \beta^-, \mathbf{f}^{inc} \right\rangle_{\Gamma-} \end{aligned} \tag{12}$$

$$\begin{aligned} \left\langle \lambda^+, \mathbf{e}^+ \right\rangle_{\Gamma+} + \frac{i}{k}\left\langle \lambda^+, \mathbf{j}^+ \right\rangle_{\Gamma+} &= \left\langle \lambda^+, \mathbf{e}^- \right\rangle_{\Gamma+} - \frac{i}{k}\left\langle \lambda^+, \mathbf{j}^- \right\rangle_{\Gamma+} + \frac{i}{k}\left\langle \lambda^+, \mathbf{g}^{inc} \right\rangle_{\Gamma+} \\ -ik\left\langle \beta^+, \mathbf{e}^+ \right\rangle_{\Gamma+} + \left\langle \beta^+, \mathbf{j}^+ \right\rangle_{\Gamma+} &= -ik\left\langle \beta^+, \mathbf{e}^- \right\rangle_{\Gamma+} - \left\langle \beta^+, \mathbf{j}^- \right\rangle_{\Gamma+} + \left\langle \beta^+, \mathbf{g}^{inc} \right\rangle_{\Gamma+} \end{aligned} \tag{13}$$

$$\begin{aligned} &\forall \left(\beta^-, \beta^+\right) \in \mathbf{H}^{-1/2}\left(\mathbf{curl}_\Gamma, \Gamma^-\right) \times \mathbf{H}^{-1/2}\left(\mathbf{curl}_\Gamma, \Gamma^+\right) \text{ and} \\ &\left(\lambda^-, \lambda^+\right) \in \mathbf{H}^{-1/2}\left(div_\Gamma, \Gamma^-\right) \times \mathbf{H}^{-1/2}\left(div_\Gamma, \Gamma^+\right) \end{aligned}$$

Substituting Eq. (10) into Eq. (13) results in

$$\begin{aligned}
&\left\langle \lambda^{+}, \left(\frac{1}{2}\mathcal{I}+\mathcal{C}\right)\left(\mathbf{e}^{+}\right)\right\rangle_{\mathit{\Gamma}+} - \left\langle \lambda^{+}, \mathcal{S}\left(\mathbf{j}^{+}\right)\right\rangle_{\mathit{\Gamma}+} + \frac{i}{k}\left\langle \lambda^{+}, \mathbf{j}^{+}\right\rangle_{\mathit{\Gamma}+} \\
&\qquad = \left\langle \lambda^{+}, \mathbf{e}^{-}\right\rangle_{\mathit{\Gamma}+} - \frac{i}{k}\left\langle \lambda^{+}, \mathbf{j}^{-}\right\rangle_{\mathit{\Gamma}+} + \frac{i}{k}\left\langle \lambda^{+}, \mathbf{g}^{inc}\right\rangle_{\mathit{\Gamma}+} \\
&-ik\left\langle \beta^{+}, \mathbf{e}^{+}\right\rangle_{\mathit{\Gamma}+} + \left\langle \beta^{+}, \mathcal{N}\left(\mathbf{e}^{+}\right)\right\rangle_{\mathit{\Gamma}+} + \left\langle \beta^{+}, \left(\frac{1}{2}\mathcal{I}-\mathcal{B}\right)\left(\mathbf{j}^{+}\right)\right\rangle_{\mathit{\Gamma}+} \\
&\qquad = -ik\left\langle \beta^{+}, \mathbf{e}^{-}\right\rangle_{\mathit{\Gamma}+} - \left\langle \beta^{+}, \mathbf{j}^{-}\right\rangle_{\mathit{\Gamma}+} + \left\langle \beta^{+}, \mathbf{g}^{inc}\right\rangle_{\mathit{\Gamma}+}
\end{aligned} \tag{14}$$

Finally, we state the overall variational formulation for the proposed non-conformal coupling between finite and boundary elements:

Find $\mathbf{E} \in \mathbf{H}_0\left(\mathbf{curl}; \mathit{\Omega}\right)$, $\mathbf{j}^{-} \in \mathbf{H}^{-1/2}\left(div_{\mathit{\Gamma}}, \mathit{\Gamma}^{-}\right)$, $\mathbf{e}^{+} \in \mathbf{H}^{-1/2}\left(\mathbf{curl}_{\mathit{\Gamma}}, \mathit{\Gamma}^{+}\right)$, and $\mathbf{j}^{+} \in \mathbf{H}^{-1/2}\left(div_{\mathit{\Gamma}}, \mathit{\Gamma}^{+}\right)$ such that

$$\begin{aligned}
&a\ \left(\mathbf{v}, \mathbf{E}\right) - \frac{1}{2}\left\langle \gamma_t \mathbf{v}, \mathbf{j}^{-}\right\rangle_{\mathit{\Gamma}-} - \frac{ik}{2}\left\langle \gamma_t \mathbf{v}, \mathbf{e}^{-}\right\rangle_{\mathit{\Gamma}-} + \frac{ik}{2}\left\langle \gamma_t \mathbf{v}, \mathbf{e}^{+}\right\rangle_{\mathit{\Gamma}-} + \frac{1}{2}\left\langle \gamma_t \mathbf{v}, \mathbf{j}^{+}\right\rangle_{\mathit{\Gamma}-} \\
&= -\frac{1}{2}\left\langle \gamma_t \mathbf{v}, \mathbf{f}^{inc}\right\rangle_{\mathit{\Gamma}-} \\
&-\frac{1}{2}\left\langle \lambda^{-}, \mathbf{e}^{-}\right\rangle_{\mathit{\Gamma}-} - \frac{i}{2k}\left\langle \lambda^{-}, \mathbf{j}^{-}\right\rangle_{\mathit{\Gamma}-} + \frac{1}{2}\left\langle \lambda^{+}, \mathbf{e}^{+}\right\rangle_{\mathit{\Gamma}-} - \frac{i}{2k}\left\langle \lambda^{-}, \mathbf{j}^{+}\right\rangle_{\mathit{\Gamma}-} \\
&= \frac{i}{2k}\left\langle \lambda^{-}, \mathbf{f}^{inc}\right\rangle_{\mathit{\Gamma}-} \\
&-\frac{ik}{2}\left\langle \beta^{+}, \mathbf{e}^{+}\right\rangle_{\mathit{\Gamma}+} + \frac{1}{2}\left\langle \beta^{+}, \mathcal{N}\left(\mathbf{e}^{+}\right)\right\rangle_{\mathit{\Gamma}+} + \frac{1}{2}\left\langle \beta^{+}, \left(\frac{1}{2}\mathcal{I}-\mathcal{B}\right)\left(\mathbf{j}^{+}\right)\right\rangle_{\mathit{\Gamma}+} \\
&+\frac{ik}{2}\left\langle \beta^{+}, \mathbf{e}^{-}\right\rangle_{\mathit{\Gamma}+} + \frac{1}{2}\left\langle \beta^{+}, \mathbf{j}^{-}\right\rangle_{\mathit{\Gamma}+} = \frac{1}{2}\left\langle \beta^{+}, \mathbf{g}^{inc}\right\rangle_{\mathit{\Gamma}+} \\
&-\frac{1}{2}\left\langle \lambda^{+}, \left(\frac{1}{2}\mathcal{I}+\mathcal{C}\right)\left(\mathbf{e}^{+}\right)\right\rangle_{\mathit{\Gamma}+} + \frac{1}{2}\left\langle \lambda^{+}, \mathcal{S}\left(\mathbf{j}^{+}\right)\right\rangle_{\mathit{\Gamma}+} - \frac{i}{2k}\left\langle \lambda^{+}, \mathbf{j}^{+}\right\rangle_{\mathit{\Gamma}+} \\
&-\frac{i}{2k}\left\langle \lambda^{+}, \mathbf{j}^{-}\right\rangle_{\mathit{\Gamma}+} + \frac{1}{2}\left\langle \lambda^{+}, \mathbf{e}^{-}\right\rangle_{\mathit{\Gamma}+} = -\frac{i}{2k}\left\langle \lambda^{+}, \mathbf{g}^{inc}\right\rangle_{\mathit{\Gamma}+}
\end{aligned} \tag{15}$$

$\forall \mathbf{v} \in \mathbf{H}_0\left(\mathbf{curl}; \mathit{\Omega}\right)$, $\lambda^{-} \in \mathbf{H}^{-1/2}\left(div_{\mathit{\Gamma}}, \mathit{\Gamma}^{-}\right)$, $\beta^{+} \in \mathbf{H}^{-1/2}\left(\mathbf{curl}_{\mathit{\Gamma}}, \mathit{\Gamma}^{+}\right)$, and $\lambda^{+} \in \mathbf{H}^{-1/2}\left(div_{\mathit{\Gamma}}, \mathit{\Gamma}^{+}\right)$.

2.3 Matrix Equation for the Nonconformal Coupling Between Finite and Boundary Elements

In the finite dimensional discretization, we have employed the following approximations in tetrahedra and on triangles for the variables:

$\mathbf{E}$: second order Nédélec elements of the 1^{st} kind [9] in Ω_h

$\mathbf{e}^-$: $\gamma_t \mathbf{E}$ on Γ_h^-

$\mathbf{j}^-$: second order Raviart-Thomas elements [10] on Γ_h^-

$\mathbf{e}^+$: edge elements on Γ_h^+

$\mathbf{j}^+$: first order Raviart-Thomas elements [10] on Γ_h^+

Subsequently, the final matrix equation corresponds to the variational formulation (15) is of the form

$$\begin{bmatrix} A_{II} & A_{I\Gamma} & 0 & 0 & 0 \\ A_{\Gamma I} & A_{\Gamma\Gamma} - \frac{ik}{2} T_{\Gamma^-\Gamma^-} & \frac{1}{2} D_{\Gamma^-\Gamma^-} & \frac{ik}{2} T_{\Gamma^-\Gamma^-} & \frac{1}{2} D_{\Gamma^-\Gamma^+} \\ 0 & -\frac{1}{2} D^t_{\Gamma^-\Gamma^-} & \frac{i}{2k} T_{\Gamma^-\Gamma^-} & \frac{1}{2} D^t_{\Gamma^-\Gamma^+} & -\frac{i}{2k} T_{\Gamma^-\Gamma^+} \\ 0 & \frac{ik}{2} T^t_{\Gamma^-\Gamma^+} & \frac{1}{2} D^t_{\Gamma^-\Gamma^+} & \frac{1}{2} Q_e - \frac{ik}{2} T_{\Gamma^+\Gamma^+} & \frac{1}{2} P \\ 0 & \frac{1}{2} D^t_{\Gamma^-\Gamma^+} & -\frac{i}{2k} T^t_{\Gamma^-\Gamma^+} & \frac{1}{2} U \left(\equiv P^t\right) & \frac{1}{2} Q_j - \frac{i}{2k} T_{\Gamma^+\Gamma^+} \end{bmatrix} \begin{bmatrix} \mathbf{E}_{int} \\ \mathbf{e}^- \\ \mathbf{j}^- \\ \mathbf{e}^+ \\ \mathbf{j}^+ \end{bmatrix}$$

$$= \begin{bmatrix} 0 & \mathbf{f}_e^{inc} & \mathbf{f}_j^{inc} & \mathbf{g}_e^{inc} & \mathbf{g}_j^{inc} \end{bmatrix}^t \qquad (16)$$

Note that in Eq. (16), we have partitioned the unknown coefficients of $\mathbf{E}$ into $\mathbf{E}_{int}$ and $\mathbf{e}^-$ for the interior and surface unknowns, respectively. The submatrices and their corresponding bilinear forms are summarized below.

$\left[\begin{smallmatrix} A_{II} & A_{I\Gamma} \\ A_{\Gamma I} & A_{\Gamma\Gamma} \end{smallmatrix}\right] : a(\mathbf{v}, \mathbf{E})$ $\quad T_{\Gamma^-\Gamma^-} : \langle \gamma_t \mathbf{v}, \mathbf{e}^- \rangle_{\Gamma^-}$ $\quad T_{\Gamma^+\Gamma^+} : \langle \beta^+, \mathbf{e}^+ \rangle_{\Gamma^+}$

$T_{\Gamma^-\Gamma^+} : \langle \gamma_t \mathbf{v}, \mathbf{e}^+ \rangle_{\Gamma^-}$ $\quad D_{\Gamma^-\Gamma^-} : \langle \gamma_t \mathbf{v}, \mathbf{j}^- \rangle_{\Gamma^-}$ $\quad D_{\Gamma^-\Gamma^+} : \langle \gamma_t \mathbf{v}, \mathbf{j}^+ \rangle_{\Gamma^-}$

$Q_e : \langle \beta^+, \mathcal{N}(\mathbf{e}^+) \rangle_{\Gamma^+}$ $\quad Q_j : \langle \lambda^+, \mathcal{S}(\mathbf{j}^+) \rangle_{\Gamma^+}$ $\quad P : \left\langle \beta^+, \left(\frac{1}{2}\mathcal{I} - \mathcal{B}\right)(\mathbf{j}^+) \right\rangle_{\Gamma^+}$

$U: \left\langle \lambda^+, \left(\frac{1}{2}\mathcal{I} + \mathcal{C}\right)(\mathbf{e}^+) \right\rangle_{\Gamma^+}$

3 Numerical Results

In Figure 1, we show the condition numbers of the final matrix equations resulting from the symmetric couplings based on the Costabel approach [12, 4] and the new proposed non-conformal coupling for a box-shaped computational domain. Note that Figure 1(a) and (b) clear indicate that the previous symmetric formulations suffer the notorious internal resonances, whereas the new proposed approach does not. Moreover, in Figure 1(c), we plot the eigenvalues distribution of the same matrix (from the proposed method) of the off-diagonal blocks after applying the block diagonal preconditioner [1]. All the eigenvalues are within the unit circle, and clearly observe the C.B.S. inequality. In Figure 2, the bistatic radar cross section (RCS) computed using the proposed method for a metallic generic battle ship are compared with those obtained by a fast boundary element code, based on electric field integral equation (EFIE). The agreement is excellent between the two results and hence validate the accuracy of the proposed approach.

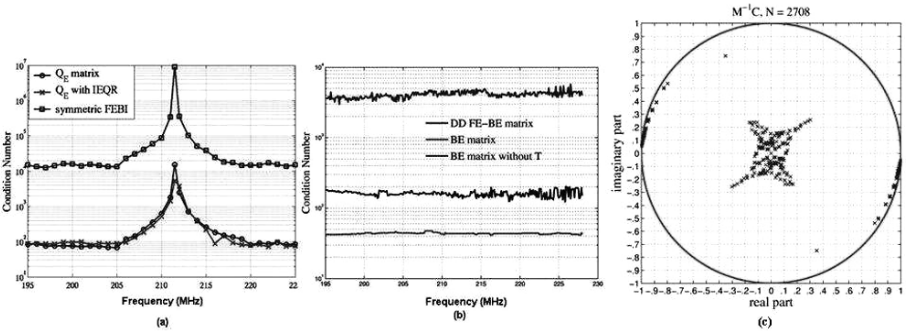


Fig. 1. Condition numbers and eigenvalue distributions of the coupled finite elements and boundary elements formulations for a box domain. (a) The symmetric formulation based on Costabel approach [12, 4]; (b) The currently proposed approach; and, (c) Eigenvalues distribution of the off-diagonal blocks after preconditioned. Note that all the eigenvalues are within the unit circle and thus satisfied the C.B.S inequality [1].

4 Conclusions

This chapter describes a variational formulation for non-conformal couplings between finite and boundary elements for electromagnetic scattering problems in $\mathcal{R}^3$. Numerical examples demonstrate that the proposed DD-FE-BE formulation does not suffer the notorious internal resonances and results in matrix equations that satisfy the C.B.S. inequality after applying the block diagonal preconditioner.

References

1. O. Axelsson, *Iterative Solution Methods*, Cambridge University Press, New York, 1994.
2. M. Costabel, *Symmetric methods for the coupling of finite elements and boundary elements*, in Boundary Elements IX, C. A. Brebbia, W. L. Wendland, and G. Kuhn, eds., vol. 1, Springer-Verlang, 1987, pp. 411–420.

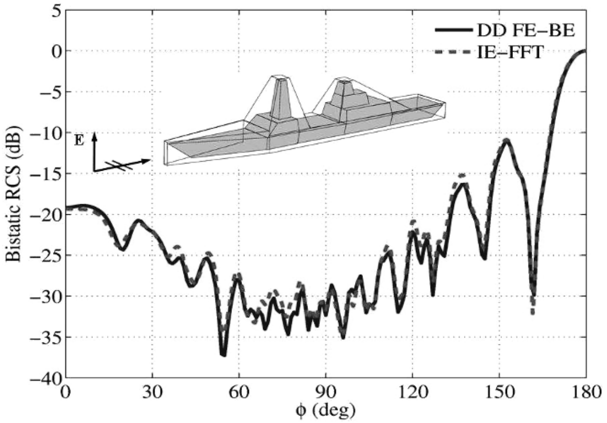


Fig. 2. Comparisons of the computed bi-static RCS using the proposed DD-FE-BE method and the IE-FFT accelerated boundary element method.

3. R. Hiptmair, *Symmetric coupling for eddy current problems*, SIAM J. Numer. Anal., 40 (2002), pp. 41–65.
4. ———, *Coupling of finite elements and boundary elements in electromagnetic scattering*, SIAM J. Numer. Anal., 41 (2003), pp. 919–944.
5. J.-M. Jin, J. L. Volakis, and J. D. Collins, *A finite-element-boundary integral method for scattering and radiation by two and three-dimensional structures*, IEEE Antennas and Propagation Magazine, 33 (1991), pp. 22–32.
6. S.-C. Lee, M. N. Vouvakis, and J.-F. Lee, *A non-overlapping domain decomposition method with non-matching grids for modeling large finite antenna arrays*, J. Comput. Phys., 203 (2005), pp. 1–21.
7. S.-C. Lee, M. N. Vouvakis, K. Zhao, and J.-F. Lee, *Analysing microwave devices using a symmetric coupling of finite and boundary elements*, Internat. J. Numer. Methods Engrg., 64 (2005), pp. 528–546.
8. J. Liu and J.-M. Jin, *A novel hybridization of higher order finite element and boundary integral methods for electromagnetic scattering and radiation problems*, IEEE Trans. Antennas Propagat., 49 (2001), pp. 1794–1806.
9. J.-C. Nédélec, *Mixed finite elements in R^3*, Numer. Math., 35 (1980), pp. 315–341.
10. P. A. Raviart and J. M. Thomas, *A mixed finite element method for 2^{nd} order elliptic problems*, in Mathematical Aspects of Finite Element Methods, A. Dold and B. Eckmann, eds., vol. 606 of Lecture Notes of Mathematics, Springer, 1975.
11. D. K. Sun, Z. Cendes, and J.-F. Lee, *Adaptive mesh refinement, h-version, for solving multiport microwave devices in three dimensions*, IEEE Trans. Magn., 36 (2000), pp. 1596–1599.
12. M. N. Vouvakis, S.-C. Lee, K. Zhao, and J.-F. Lee, *A symmetric FEM-IE formulation with a single-level IE-QR algorithm for solving electromagnetic radiation and scattering problems*, IEEE Trans. Antennas Propagat., 52 (2004), pp. 3060–3070.

MINISYMPOSIUM 5: Space-time Parallel Methods for Partial Differential Equations

Organizers: Martin Gander[1] and Laurence Halpern[2]

[1] Swiss Federal Institute of Technology, Geneva. martin.gander@math.unige.ch
[2] University of Paris XIII. halpern@math.univ-paris13.fr

Space-time parallel methods had a second youth with the introduction of the parareal algorithm in 2001. While the convergence properties of this algorithm are not yet fully understood, there are several other space-time parallel algorithms which are actively researched, notably algorithms of Schwarz waveform relaxation type and space-time multigrid methods.

This minisymposium includes a historical introduction to space-time parallel methods, links them to the parareal algorithm, and presents new results for parareal and optimized Schwarz waveform relaxation methods.

Optimized Schwarz Waveform Relaxation Algorithms with Nonconforming Time Discretization for Coupling Convection-diffusion Problems with Discontinuous Coefficients

Eric Blayo[1], Laurence Halpern[2], and Caroline Japhet[2]

[1] LMC, Université Joseph Fourier, B.P. 53, 38041 Grenoble Cedex 9, France. Eric.Blayo@imag.fr

[2] LAGA, Université Paris XIII, 99 Avenue J-B Clément, 93430 Villetaneuse, France. halpern@math.univ-paris13.fr,japhet@math.univ-paris13.fr

Summary. We present and study an optimized Schwarz Waveform Relaxation algorithm for convection-diffusion problems with discontinuous coefficients. Such analysis is a first step towards the coupling of heterogeneous climatic models. The SWR algorithms are global in time, and thus allow for the use of non conforming space-time discretizations. They are therefore well adapted to coupling models with very different spatial and time scales, as in ocean-atmosphere coupling. As the cost per iteration can be very high, we introduce new transmission conditions in the algorithm which optimize the convergence speed. In order to get higher order schemes in time, we use in each subdomain a discontinuous Galerkin method for the time-discretization. We present numerical results to illustrate this approach, and we analyse numerically the time-discretization error.

1 Introduction

We present an optimized Schwarz Waveform Relaxation algorithm for convection-diffusion problems with discontinuous coefficients. Such methods have proven to provide an efficient approach in the case of the wave equation with discontinuous wave speed [3], and convection-difusion problems in one [1] and two dimensions [5] with constant coefficients. Our final objective is to propose efficient algorithms for coupling heterogeneous models (e.g. ocean-atmosphere) in the context of climate modelling. The SWR algorithms are global in time, and therefore are well adapted to coupling model; they lead, at convergence, to a model with the physical transmission conditions, they reduce the exchange of information between codes, and they permit the use of non conforming discretizations in space-time. This last point is crucial in climate modelling, where very different scales in time and space are present.

As a first step, we consider the domain decomposition problem for a convection-diffusion equation with discontinuous coefficients. After introducing our model problem in Section 2, we present in Section 3 a classical strategy for coupling ocean and atmosphere models, which consists in realizing one additive Schwarz iteration with physical transmission conditions, in each time window [6]. In order to get a more efficient method which improves the converged solution, we introduce in Section 4 a Schwarz Waveform Relaxation method with optimized transmission conditions of order 1. This method allows for the use of non conforming space-time discretizations. As our objective is to get higher order schemes in time, we introduce a discontinuous Galerkin method [4]. The formulation is given in Section 5. As the grids in time are different in each subdomain, the projection between arbitrary grids is performed by an efficient algorithm introduced in [3]. Numerical results illustrate the validity of our approach in Section 6.

2 Model problem

We consider the one dimensional convection diffusion equation

$$\begin{aligned} \mathcal{L}u &= f, && \text{in } \Omega \times (0,T),\\ u(x,0) &= u_0(x), && \forall x \in \Omega,\\ u(x_0,t) &= u(x_1,t) = 0, && t \in (0,T), \end{aligned}$$

where $\Omega =]x_0, x_1[$ is a bounded open subset of $\mathbb{R}$ (containing zero), $\mathcal{L}$ is the convection diffusion operator

$$\mathcal{L}u := \frac{\partial u}{\partial t} + \frac{\partial}{\partial x}(a(x)u) - \frac{\partial}{\partial x}(\nu(x)\frac{\partial u}{\partial x}),$$

and the velocity a and the viscosity ν are supposed to be constant in the two nonoverlapping subregions $\Omega_1 =]x_0, 0[$ and $\Omega_2 =]0, x_1[$ of Ω, but can be discontinuous at zero:

$$a(x) = \begin{cases} a_1, & x \in \Omega_1 \\ a_2, & x \in \Omega_2 \end{cases}, \quad \nu(x) = \begin{cases} \nu_1, & x \in \Omega_1 \\ \nu_2, & x \in \Omega_2 \end{cases},$$

with $\nu_i > 0,\ i = 1, 2$. Without loss of generality, we can assume that a is nonnegative. This problem is equivalent to the following subproblems:

$$\begin{cases} \mathcal{L}_1 u_1 := \dfrac{\partial u_1}{\partial t} + a_1 \dfrac{\partial u_1}{\partial x} - \nu_1 \dfrac{\partial^2 u_1}{\partial x^2} = f, & \text{in } \Omega_1 \times (0,T),\\ \qquad\qquad\qquad\qquad u_1(x,0) = u_0(x), & \forall x \in \Omega_1,\\ \qquad\qquad\qquad\qquad u_1(x_0,t) = 0, & t \in (0,T), \end{cases}$$

$$\begin{cases} \mathcal{L}_2 u_2 := \dfrac{\partial u_2}{\partial t} + a_2 \dfrac{\partial u_2}{\partial x} - \nu_2 \dfrac{\partial^2 u_2}{\partial x^2} = f, & \text{in } \Omega_2 \times (0,T),\\ \qquad\qquad\qquad\qquad u_2(x,0) = u_0(x), & \forall x \in \Omega_2,\\ \qquad\qquad\qquad\qquad u_2(t,x_1) = 0, & t \in (0,T), \end{cases}$$

with the physical transmission conditions at $x = 0$:

$$\begin{cases} u_1(0,t) = u_2(0,t), & t \in (0,T),\\ (a_1 - \nu_1 \dfrac{\partial}{\partial x})u_1(0,t) = (a_2 - \nu_2 \dfrac{\partial}{\partial x})u_2(0,t), & t \in (0,T). \end{cases} \tag{1}$$

To solve this problem numerically, it is natural to use an algorithm where the transmission conditions are the physical conditions (in our case, conditions (1)), and it is especially the case when coupling heterogeneous climate component models.

3 Algorithm for ocean-atmosphere coupling

A commonly used strategy for solving ocean-atmosphere coupling consists in decomposing the time interval $(0,T)$ into windows, $[0,T]=\cup_{n=0}^{N}[T_n,T_{n+1}]$, and to use one additive Schwarz iteration with the physical transmission conditions, in each time window [6]. Let $u_{i,n}$ be a discrete approximation of u_i in Ω^i in the window $[T_{n-1},T_n]$. Then, $u_{i,n+1},\ i=1,2$, is the solution of

$$\left\{\begin{aligned}\mathcal{L}_1 u_{1,n+1} &= f, && \text{in } \Omega_1\times(T_n,T_{n+1}),\\ u_{1,n+1}(x,T_n) &= u_{1,n}(x,T_n), && \forall x\in\Omega_1,\\ u_{1,n+1}(x_0,t) &= 0, && t\in(T_n,T_{n+1}),\\ u_{1,n+1}(0,t) &= u_{2,n}(0,T_n), && t\in(T_n,T_{n+1}),\end{aligned}\right. \tag{2}$$

$$\left\{\begin{aligned}\mathcal{L}_2 u_{2,n+1} &= f, && \text{in } \Omega_2\times(T_n,T_{n+1}),\\ u_{2,n+1}(x,T_n) &= u_{2,n}(x,T_n), && \forall x\in\Omega_2,\\ u_{2,n+1}(t,x_1) &= 0, && t\in(T_n,T_{n+1}),\\ (a_2-\nu_2\frac{\partial}{\partial x})u_{2,n+1}(0,t) &= (a_1-\nu_1\frac{\partial}{\partial x})u_{1,n}(0,T_n), && t\in(T_n,T_{n+1}),\end{aligned}\right. \tag{3}$$

Remark 1. It is important to notice that in the previous algorithm the transmission conditions are constant in time, on each time window (T_i,T_{i+1}).

In ocean-atmosphere coupling, the use of very few iteration (one iteration here) in each time window is motivated by the fact that the computation time per iteration is very high. In order to improve the numerical solution, with very few iteration per time window, we propose to use in each time window an Optimized Schwarz Waveform Relaxation with transmission conditions based on a differential in time.

4 Optimized Schwarz Waveform Relaxation

The general Schwarz Waveform Relaxation, in one time window, for example in the whole window $(0,T)$ is written as follows:

$$\left\{\begin{aligned}\mathcal{L}_1 u_1^{k+1} &= f, && \text{in } \Omega_1\times(0,T),\\ u_1^{k+1}(x,0) &= u_0(x), && \forall x\in\Omega_1,\\ u_1^{k+1}(x_0,t) &= 0, && t\in(0,T),\\ (\nu_1\frac{\partial}{\partial x}-a_1+\Lambda_1)u_1^{k+1}(0,t) &= (\nu_2\frac{\partial}{\partial x}-a_2+\Lambda_1)u_2^{k}(0,t), && t\in(0,T),\end{aligned}\right.$$

$$\left\{\begin{aligned}\mathcal{L}_2 u_2^{k+1} &= f, && \text{in } \Omega_2\times(0,T),\\ u_2^{k+1}(x,0) &= u_0(x), && \forall x\in\Omega_2,\\ u_2^{k+1}(t,x_1) &= 0, && t\in(0,T),\\ (\nu_2\frac{\partial}{\partial x}-a_2+\Lambda_2)u_2^{k+1}(0,t) &= (\nu_1\frac{\partial}{\partial x}-a_1+\Lambda_2)u_1^{k}(0,t), && t\in(0,T),\end{aligned}\right.$$

where Λ_1 and Λ_2 are linear operators, involving derivatives in time.

4.1 Optimized transmission conditions

The optimal transmission conditions can be derived from a Fourier analysis in the case $\Omega = \mathbb{R}$. Using the error equations and a Fourier transform with parameter ω,

$$\rho(\omega) := \left(\frac{\lambda_2(\omega) - r_1^-(\omega)}{\lambda_1(\omega) - r_1^-(\omega)}\right)\left(\frac{\lambda_1(\omega) - r_2^+(\omega)}{\lambda_2(\omega) - r_2^+(\omega)}\right)$$

with $r_1^-(\omega) = \dfrac{a_1 - \sqrt{a_1^2 + 4\nu_1 i\omega}}{2}$, $r_2^+(\omega) = \dfrac{a_2 + \sqrt{a_2^2 + 4\nu_2 i\omega}}{2}$ and $\lambda_i, i = 1, 2$ the symbol of Λ_i. The optimal choice, which gives a convergence in 2 iterations, is $\lambda_2 = r_1^-(\omega)$ and $\lambda_1 = r_2^+(\omega)$. The calculations are straightforward extensions to those in [1]. As the optimal corresponding transfer operators Λ_1, Λ_2 are nonlocal in time and thus more costly than local transfers, we propose to use the following transfer operators

$$\Lambda_1 := \frac{a_2 + p_2}{2} + \frac{q_2}{2}\frac{\partial}{\partial t}, \quad \Lambda_2 := \frac{a_1 - p_1}{2} - \frac{q_1}{2}\frac{\partial}{\partial t}$$

where the parameters p_1, p_2, q_1, q_2 minimize the convergence rate.The condition on the parameters p_1, p_2, q_1, q_2 for the local subdomain problems to be well-posed are $q_j \geq 0$ (due to energy estimates as in [1]). The question of convergence of the algorithm remains open, even though there are numerical evidences for a positive answer (see [2] for theoretical results using Robin transmission conditions).

4.2 Optimized Schwarz Waveform Relaxation with time windows

We now define the algorithm with many time windows: Let $[0, T] = \cup_{n=0}^{N} [T_n, T_{n+1}]$, and let $p \geq 1$ be an integer, that we will take small (typically $p \leq 3$) in order to make very few iterations in each time window. Let $u_{i,n}^k$ be a discrete approximation of u_i in Ω^i in the window (T_{n-1}, T_n) at step k of the SWR method. Then, the next time window's solution $u_{i,n+1}$ in Ω^i is obtained after p SWR iterations:
for $k = 0, ..., p-1$:

$$\left\{\begin{array}{rcll} \mathcal{L}_1 u_{1,n}^{k+1} &=& f, & \text{in } \Omega_1 \times (T_n, T_{n+1}), \\ u_{1,n}^{k+1}(x, T_n) &=& u_{1,n}(x, T_n), & \forall x \in \Omega_1, \\ u_{1,n}^{k+1}(x_0, t) &=& 0, & t \in (T_n, T_{n+1}), \\ (\nu_1 \dfrac{\partial}{\partial x} - a_1 + \Lambda_1) u_{1,n}^{k+1}(0, t) &=& (\nu_2 \dfrac{\partial}{\partial x} - a_2 + \Lambda_1) u_{2,n}^k(0, t), & t \in (T_n, T_{n+1}), \end{array}\right. \tag{4}$$

$$\left\{\begin{array}{rcll} \mathcal{L}_2 u_{2,n}^{k+1} &=& f, & \text{in } \Omega_2 \times (T_n, T_{n+1}), \\ u_{2,n}^{k+1}(x, T_n) &=& u_{2,n}(x, T_n), & \forall x \in \Omega_2, \\ u_{2,n}^{k+1}(t, x_1) &=& 0, & t \in (T_n, T_{n+1}), \\ (\nu_2 \dfrac{\partial}{\partial x} - a_2 + \Lambda_2) u_{2,n}^{k+1}(0, t) &=& (\nu_1 \dfrac{\partial}{\partial x} - a_1 + \Lambda_2) u_{1,n}^k(0, t), & t \in (T_n, T_{n+1}), \end{array}\right. \tag{5}$$

and $u_{1,n+1} := u_{1,n}^p$, $u_{2,n+1} := u_{2,n}^p$.

5 Time discretization with a discontinuous Galerkin Method

Let us introduce the discretization of the subproblems in a time window $I = (T_n, T_{n+1})$. We consider, for example, the subproblem in Ω_1 at step k of the SWR procedure. It can be written in the form

$$\begin{cases} \mathcal{L}_1 u = f & \text{in } \Omega_1 \times I, \\ u(x_0, \cdot) = 0 & \text{in } I, \\ (\nu_1 \dfrac{\partial u}{\partial x} + \beta u + \gamma \dfrac{\partial u}{\partial t})(0, \cdot) = g & \text{in } I, \\ u(\cdot, 0) = u_0, & \text{in } \Omega_1, \end{cases}$$

with $\beta = -a_1 + \dfrac{a_2 + p_2}{2}$, $\gamma = \dfrac{q_2}{2}$, and $g(t) = (\nu_2 \dfrac{\partial}{\partial x} - a_2 + \Lambda_1) u_{2,n}^k(0, t)$.
This problem is equivalent to the weak formulation: Find $u(t) \in V = H^1(\Omega_1)$ such that $u(0) = u_0$ and

$$((\dot{u}(t), v)) + \tilde{a}(u(t), v) = \ell_t(v), \quad \forall v \in V$$

with $(\cdot, \cdot)$ the scalar product in $L^2(\Omega_1)$, and for $u \in V$:

$$\begin{cases} ((u, v)) := (u, v) + \gamma u(0) v(0) \\ \tilde{a}(u, v) := b(u, v) + \beta u(0) v(0), \quad \text{with} \quad b(u, v) = \nu_1 (\dfrac{\partial u}{\partial x}, \dfrac{\partial v}{\partial x}) + a_1 (\dfrac{\partial u}{\partial x}, v) \\ \ell_t(v) := (f(t), v) + g(t) v(0) \end{cases}$$

The discontinuous Galerkin Method [4] is based on the use of a discontinuous finite element formulation in time. Let $I = \prod_{k=1}^{K} I_k$ with $I_k = [t_{k-1}, t_k]$, and let $v_+^k = \lim_{s \to 0^+} v(t_k + s)$ and $v_-^k = \lim_{s \to 0^-} v(t_k + s)$. Let V_h be a finite-dimensional subspace of V, and

$$\mathbb{P}_q(I_k) = \{v : I_k \longrightarrow V_h : v(t) = \sum_{i=0}^{q} v_i t^i \text{ with } v_i \in V_h\}$$

The discontinuous Galerkin Method can now be formulated as follows:

$$\begin{cases} U_-^0 = u_0 \\ \text{For } k = 1, \cdots, K, \text{ given } U_-^{k-1}, \text{ find } U \equiv U_{|I_k} \in \mathbb{P}_q(I_k) \text{ such that} \\ \displaystyle\int_{I_k} [((\dot{U}, v)) + \tilde{a}(U, v)] dt + ((U_+^{k-1}, v_+^{k-1})) = \\ \qquad\qquad \displaystyle\int_{I_k} \ell_t(v) dt + ((U_-^{k-1}, v_+^{k-1})), \quad \forall v \in \mathbb{P}_q(I_k) \end{cases} \tag{6}$$

For $q = 0$, using the notation $U^k \equiv U_-^k \equiv U_+^{k-1}$ and $\Delta t_k = t_k - t_{k-1}$, the method reduces to

$$\begin{cases} U^0 = u_0 \\ \text{For } k = 1, \cdots, K, \text{ find } U^k \in V_h \text{ such that} \\ ((\dfrac{U^k - U^{k-1}}{\Delta t_k}, v)) + \tilde{a}(U^k, v) = \dfrac{1}{\Delta t_k} \displaystyle\int_{I_k} \ell_t(v), \quad \forall v \in V_h \end{cases}$$

This method is a simple modification of the backward Euler scheme in that case. For $q = 1$, (6) is equivalent to the following system with, for $t \in I_k$, $U(t) = U_0 + \dfrac{t - t_{k-1}}{\Delta t_k} U_1$, $U_i \in V_h$,

$$\left\{\begin{aligned}
&(U_0, v) + \Delta t_k\, b(U_0, v) + (\Delta t_k\, \beta + \gamma) U_0(0) v(0) + (U_1, v) + \frac{1}{2} \Delta t_k\, b(U_1, v) \\
&\qquad + \Delta t_k\, (\frac{\beta}{2} + \gamma) U_1(0) v(0) = (U_-^{k-1}, v) + \gamma U_-^{k-1}(0)\ v(0) \\
&\qquad\qquad + \int_{I_k} (f(s), v) ds + v(0) \int_{I_k} g(s) ds, \quad \forall v \in V_h \\
&\frac{1}{2} \Delta t_k\, b(U_0, v) + \frac{\beta}{2} \Delta t_k\, U_0(0) v(0) + \frac{1}{2} (U_1, v) + \frac{1}{3} \Delta t_k\, b(U_1,\, v) \\
&\qquad + (\frac{\gamma}{2} + \frac{\beta}{3} \Delta t_k) U_1(0) v(0) = \frac{1}{\Delta t_k} \int_{I_k} (s - t_{k-1}) (f(s),\, v) ds \\
&\qquad\qquad + \frac{1}{\nu_1} v(0) \int_{I_k} (s - t_{k-1})\ g(s) ds, \quad \forall v \in V_h
\end{aligned}\right.$$

6 Numerical results

In this presentation, we take $q = 0$ in the discontinuous Galerkin method.

6.1 Relative L^2 error versus the time step

In this part, we consider the case with one time window only, with different grids in time in each subdomain, and we observe the relative L^2 error between the SWR converged solution and the continuous solution, versus the number of refinements of the time grid. We choose $a_1 = a_2 = 1$, $\nu_1 = \nu_2 = 1$, and $u(x,t) = sin(x)cos(t)$, in $[0, 2\pi] \times [0, 2.5]$ as exact solution. The space domain $[0, 2\pi]$ is decomposed in two subdomains $\bar{\Omega}_1 = [0, 2]$ and $\bar{\Omega}_2 = [2, 2\pi]$. The mesh size is $h_1 = 0.01$ for Ω_1 and $h_2 = (2\pi - 2)/200$ for Ω_2. In order to compare the L^2 relative error on the nonconforming time grids case to the error obtained on a uniform conforming time grid, we consider four initial meshes in time (see figure 1):

- a uniform finner conforming mesh (mesh 1) with $\Delta t = 2.5/24$,
- a nonconforming mesh (mesh 2) with $\Delta t = 2.5/24$ in Ω_1 and $\Delta t = 2.5/16$ in Ω_2,
- a nonconforming mesh (mesh 3) with $\Delta t = 2.5/16$ in Ω_1 and $\Delta t = 2.5/24$ in Ω_2,
- a uniform coarser conforming mesh (mesh 4) with $\Delta t = 2.5/16$.

Figure 2 shows the relative L^2 error versus the number of refinement for these four meshes, and the time step Δt versus the number of refinement, in logarithmic scale. At each refinement, the time step is divided by two. The results of Figure 2 show that the relative L^2 error tends to zero at the same rate than the time step, and this fits with the error estimates in [4]. On the other hand, we observe that the two curves corresponding to the nonconforming meshes (mesh 2 and mesh 3) are between the curves of the conforming meshes (mesh 1 and mesh 4).

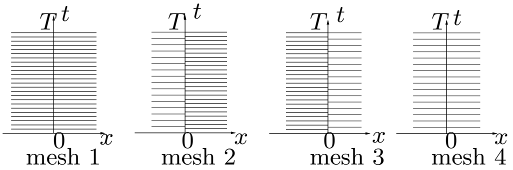


Fig. 1. Uniform conforming time grids (mesh 1 and mesh 4) and nonconforming time grids (mesh 2 and mesh 3).

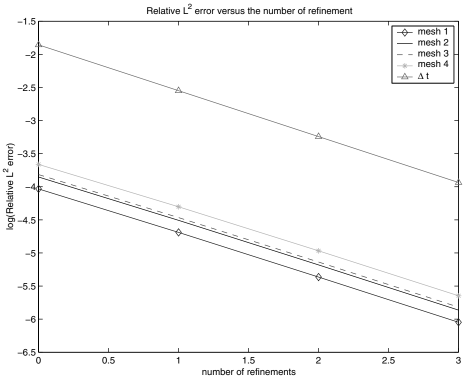


Fig. 2. Relative L^2 error versus the number of refinements for the initial meshes: mesh 1 (diamond line), mesh 2 (solid line), mesh 3 (dashed line), and mesh 4 (star line). The triangle line is the time step Δt versus the number of refinements, in logarithmic scale.

6.2 Comparison of the two algorithms

In this part, we consider the problem

$$\begin{cases} \mathcal{L}u = 0 \ \text{ in }]0,6[\times[0,3] \\ u(0,t) = u(6,t) = 0 \ , t \in [0,3], \quad u(x,0) = e^{-3(1.2-x).^2}, x \in [0,6] \end{cases}$$

In order to compare algorithm (2)-(3) to the SWR algorithm (4)-(5), we decompose the time interval into three windows: $[0,3] = [0,1] \cup [1,2] \cup [2,3]$ and we compare the computed solutions obtained from each method. We take $a_1 = 0.1$, $\nu_1 = 0.2$, $a_2 = \nu_2 = 1$. The space domain $[0,6]$ is decomposed into two subdomains $\bar{\Omega}_1 = [0,3]$ and $\bar{\Omega}_2 = [3,6]$. The mesh size is $h_1 = 0.01$ for Ω_1 and $h_2 = 0.06$ for Ω_2. The time step in each window is $\Delta t_1 = 0.01$ for Ω_1 and $\Delta t_2 = 0.02$ for Ω_2. In figure 3 on the right, we observe that the 3-windows computed solution with the SWR algorithm (4)-(5) is close to the one-window solution. Moreover it more precise than the 3-windows computed solution of figure 3 which is obtained with the algorithm (2)-(3) (figure 3 on the left).

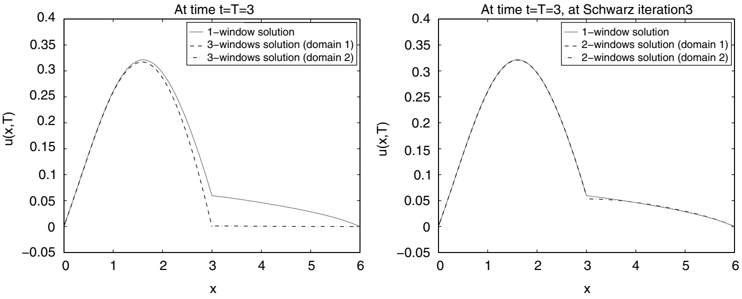


Fig. 3. One time window solution (solid line) and 3-windows solutions (dashed line for Ω_1 and dashdot line for Ω_2), with algorithm (2)-(3) on the left, at time t=T=3, and with the SWR method on the right, at time t=T=3, and at SWR iteration 3.

7 Conclusions

We have introduced a Schwarz Waveform Relaxation Algorithm for the convection-diffusion equation with discontinuous coefficients. The transmission conditions involve normal derivatives and derivatives in time as well. These have been used in the computations, together with a zero-order discontinuous Galerkin method and a projection between the time grids. We have shown numerically that the discretization order is preserved. We now intend to extend the strategy to higher order Galerkin methods, and to write projection steps that maintain, for the whole process, the order of the scheme in each subdomain.

References

1. D. Bennequin, M. J. Gander, and L. Halpern, *Optimized Schwarz waveform relaxation methods for convection reaction diffusion problems*, Tech. Rep. 24, Institut Galilée, Paris XIII, 2004.
2. M. J. Gander, L. Halpern, and M. Kern, *A Schwarz Waveform Relaxation method for advection–diffusion–reaction problems with discontinuous coefficients and non-matching grids*, in Proceedings of the 16th International Conference on Domain Decomposition Methods, O. B. Widlund and D. E. Keyes, eds., Springer, 2006. these proceedings.
3. M. J. Gander, L. Halpern, and F. Nataf, *Optimal Schwarz waveform relaxation for the one dimensional wave equation*, SIAM J. Numer. Anal., 41 (2003), pp. 1643–1681.
4. C. Johnson, *Numerical Solutions of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, 1987.
5. V. Martin, *An optimized Schwarz Waveform Relaxation method for the unsteady convection diffusion equation in two dimensions*, Appl. Numer. Math., 52 (2005), pp. 401–428.
6. P. Pellerin, H. Ritchie, F. J. Saucier, F. Roy, S. Desjardins, M. Valin, and V. Lee, *Impact of a two-way coupling between an atmospheric and an ocean-ice model over the gulf of St. Lawrence*, Monthly Weather Review, 32 (2004), pp. 1379–1398.

Stability of the Parareal Time Discretization for Parabolic Inverse Problems

Daoud S. Daoud[1]

Department of Mathematics, Eastern Mediterranean University, Famagusta, North Cyprus, Via Mersin 10, Turkey. daoud.daoud@emu.edu.tr

Summary. The practical aspect of the parareal algorithm that consists of using two solvers the coarse and fine over different time stepping to produce a rapid convergent iterative method for multi processors computations. The coarse solver solve the equation sequentially on the coarse time step while the fine solver use the information from the coarse solution to solve, in parallel, over the fine time steps. In this work we discuss the stability of the parareal-inverse problem algorithm for solving the parabolic inverse problem given by

$$\begin{array}{lll} u_t = u_{xx} + p(t)u + \phi(x,t), & & 0 < x < 1,\ 0 < t \le T, \\ u(x,0) = f(x), & 0 \le x \le 1, & \\ u(0,t) = g_0(t), & 0 < t \le T, & \\ u(1,t) = g_1(t), & 0 < t \le T, & \end{array}$$

and subject to the over specification of a condition at a point x_0 in the spatial domain $u(x_0,t) = E(t)$. We derive a stability amplification factor for the parareal-inverse algorithm and present a stability analysis in terms of the relation between the coarse and fine time steps and the value of $p(t)$. Some model problems are considered to demonstrate necessary conditions for stability.

1 Introduction

The parallelization with respect to the time variable is not an entirely new approach; the first research article in this area was an article by Nievergelt on the solution of ordinary differential equations [10] and an article by Miranker and Liniger [9] on the numerical integration of ordinary differential equations.

Recently after the development of the initial algorithms a new form of the algorithms has been proposed which consists of discretizing the problem over an interval of time using fine and coarse time steps to allow a combination of accuracy improvement, through an iterative process, and parallelization over slices of coarse time interval. The algorithm has been re derived and then named as *Parareal Algorithm* by Lion's et al. [7], also further modified by Bal and Maday [3] to solve unsteady state problem and evidently establishing a relation between the coarse and fine time step in order to define the time gain in the parallelization procedure.

The stability and the convergence of the algorithm has been further studied by Bal [2] mainly concluding that the algorithm replaces a coarse discretization method of order m by a higher order dicsretization method. Staff and Ronquist [11] also presented necessary conditions for the stability of the parareal algorithm. For further detailed views of the method and further applications we refer to Baffico et al. [1], Farhat and Chandersis [6], and Maday and Turinici [8].

In this article we will focus on the stability of the parareal algorithm for solving the following inverse problem for determining a control function $p(t)$ in a parabolic equation. Find $u = u(x,t)$ and $p = p(t)$ which satisfy

$$\begin{array}{ll} u_t = u_{xx} + p(t)u + \phi(x,t), & 0 < x < 1,\ 0 < t \leq T, \\ u(x,0) = f(x), & 0 \leq x \leq 1, \\ u(0,t) = g_0(t), & 0 < t \leq T, \\ u(1,t) = g_1(t), & 0 < t \leq T, \end{array} \tag{1}$$

subject to the over specification condition at the point x_0 in the spatial domain $u(x_0,t) = E(t)$. Here f, g_0, g_1, E and ϕ are known functions while the functions u and p are unknown, with $-1 < p(t) < 0$ for $t \in [0,T]$. The model problem given by (1) is used to describe a heat transfer process with a source parameter present and the over specification condition represents the temperature at a given point x_0 in the spatial domain at time t. Thus the purpose of solving this inverse problem is to identify the source control parameter that produces at any given time a desired temperature at a given point x_0 in the spatial domain.

2 The Parareal-Inverse Problem Algorithm

The main aspect of the parareal algorithms is to allow a parallelization in time over slices of coarse time interval using coarse time solver in combination with accuracy improvements through an iterative method (predictor-corrector form) using fine and coarse time solvers over each coarse time interval Δt ($\Delta t = T/N$).

In this article the coarse and fine time step solvers will be denoted by $G_{\Delta t}$, and $F_{\delta t}$, respectively, where $\delta t = \dfrac{\Delta t}{s}$, and s is the number of fine time steps over the coarse interval $[t_n, t_{n+1}] = [t_n, t_n + s\delta t]$, for $n = 0, 1, \ldots N-1$.

Through this work we will consider the parareal algorithm scheme in the form presented by Bal [2] and also later considered by Staff and Ronquist [11], given by

$$u_{k+1}^{n+1} = G_{\Delta t}(u_{k+1}^{n}) + F_{s\delta t}(u_k^n) - G_{\Delta t}(u_k^n). \tag{2}$$

The solution algorithm of the inverse problem (1) by an implicit type of methods, the backward Euler's method, possess an updating of the control function $p(t)$ and $u(x,t)$, or in another words correction steps at each time level prior to proceeding to the advanced time level (cf. e.g.[4], [5]). On the other hand the solution by the forward Euler's scheme does not require any correction for the control function $p(t)$, but in order to apply the parareal algorithm the updating of the value of $p(t)$ for the fine propagator is required for the advanced fine solution step using the over specification condition $u(x_0, t) = E(t)$.

Since the parareal algorithm posses a correction step over each coarse time interval it was observed that, through the coarse solution propagator, for the correction of the $p(t)$ it is sufficient to perform one iteration only, internally, over the time step $[t_n, t_{n+1}]$ that is because of the further iterations and correction of the solution by the parareal algorithm. The generic form of the parareal algorithm for the solution of the inverse problem is given as follows.

Algorithm 2.1 *Parareal Inverse Problem Algorithm*

(a) Over the domain $\Omega \times [t_n, t_{n+1}]$ and for $k = 1$, consider the coarse propagator i.e.

$$\frac{u_1^{n+1} - u_1^n}{\Delta t} = (u_{xx})_1^{n+1} + p(t_n)u^{n+1} \quad n = 1, \dots N-1,$$

the solution u_1^{n+1} denoted by $G_{\Delta t}(u_1^n)$.
$p(t^{n+1})$ correction: Consider the correction of $p(t)$ by the following relation

$$p(t^{n+1}) = \frac{E'(t_{n+1}) - (u_{xx})_1|_{(x_0, t_{n+1})} - \phi(x_0, t_{n+1})}{E(t_{n+1})}.$$

(b) For $k+1 > 1$ and over the domain $\Omega \times [t_n, t_{n+1}]$.
a) Consider the coarse propagator i.e.

$$\frac{u_{k+1}^{n+1} - u_{k+1}^n}{\Delta t} = (u_{xx})_{k+1}^{n+1} + p(t_n)u_{k+1}^{n+1} \quad n = 1, \dots N-1,$$

$p(t^{n+1})$ correction: Consider the correction of $p(t)$ by the following relation

$$p(t_{n+1}) = \frac{E'(t_{n+1}) - (u_{xx})_{k+1}|_{(x_0, t_{n+1})} - f(x_0, t_{n+1})}{E(t_{n+1})},$$

the solution u_{k+1}^{n+1} is denoted by $G_{\Delta t}(u_{k+1}^n)$.
b) Consider the fine propagator solution over $\Omega \times [t_n, t_{n+l}]$, $\quad l = 1, s-1$. Solve for

$$\frac{u_k^{n+l} - u_k^{n+l-1}}{\delta t} = (u_{xx})_k^{n+l-1} + p(t_{n+l-1})u_k^{n+l-1}.$$

The solution $u_k^{n+s} = u_k^{n+1}$ is denoted by $F_{s\delta t}(u_k^n)$, and

$$p^{n+l} = \frac{E'(t_{n+l}) - u_{xx,k}|_{(x_0, t_{n+l})} - \phi(x_0, t_{n+l})}{E(t_{n+l})}, \quad \text{for } l = 1, \dots s-1,$$

where $s = \dfrac{\Delta t}{\delta t}$.

Then the solution u_{k+1}^{n+1} is given by

$$u_{k+1}^{n+1} = G_{\Delta t}(u_{k+1}^n) + F_{s\delta t}(u_k^n) - G_{\Delta t}(u_k^n). \tag{3}$$

3 Stability of The Parareal-Inverse Algorithm

Let $u(x,t)$ be the solution of the model problem

$$u_t = u_{xx} + p(t)u(t), \tag{4}$$

subject to the following initial and boundary conditions

$$u(0,t) = 0, u(1,t) = 0 \text{ and } u(x_0,t) = u_0, \tag{5}$$

and with the specified condition $u(x_0,t) = E(t)$.

The spatial derivative operator is approximated by the second order central difference approximation given by

$$(u_{xx})_{(x_i,t)} \simeq h^{-2}[u(x_{i+1},t) - 2u(x_i,t) + u(x_{i-1},t)] + \mathcal{O}(h^2). \tag{6}$$

For the stability analysis we will consider the Fourier transform of the discrete problem, and in the Fourier domain the problem corresponding to (4) is given by

$$\widehat{u_t} = Q(\xi,t)\widehat{u}(\xi,t), \tag{7}$$

where $Q(\xi,t) = q(\xi) + \widehat{p}(t)$, such that $Q(u) = \widehat{Q(\xi)\widehat{u}(\xi)}$ and $q(\xi) = -2h^{-2}\sin^2(\xi/2)$. Then

$$Q(\xi) = q(\xi) + \widehat{p}(t) = -2h^{-2}\sin^2(\frac{\xi}{2}) + \widehat{p}(t). \tag{8}$$

The forward and backward Euler's schemes are considered to be the fine and coarse solvers for the parareal-inverse algorithm, respectively. The amplification factor of the backward Euler's scheme in the Fourier domain is given by

$$\rho(\xi,t_n)_{G_{\Delta t}} = (1 - Q(\xi,t_n)\Delta t)^{-1} = (1 + (2h^{-2}\sin^2(\frac{\xi}{2}) - \widehat{p}(t_n))\Delta t)^{-1}.$$

This scheme is unconditional stable for $p(t) < 0$ [12], and the corresponding amplification factor for the solution by the forward Euler's scheme over the time interval $[t_n, t_n + s\delta t]$, is given by

$$\rho(\xi,tn)_{F_{s\delta t}} = \prod_{i=1}^{s}(1 + Q(\xi,t_{n+i-1})\delta t) = \prod_{i=1}^{s}(1 + (-2h^{-2}sin^2(\frac{\xi}{2}) + \widehat{p}(t_{n+i-1}))\delta t),$$

and it is a conditional stable scheme according to stability condition for the forward Euler's scheme for any $p(t)$ [12].

For the stability analysis we will consider the approach by Staff and Ronquist [11] and we will present the stability studies for the following cases
case 1: $\Delta t = s\delta t$ $(s > 1)$,
case 2: $\Delta t = \delta t$ $(s = 1)$.

3.1 Case I $\Delta t = s\delta t$ $(s > 1)$

For this case of the stability analysis the coarse time step Δt is divided into s fine subintervals $(s > 1)$ and the iterative solution of (7) by the parareal-inverse algorithm 2.1 is given by

$$\widehat{u}_{k+1}^{n+1} = (1-Q(\xi,t_n)\Delta t)^{-1}\widehat{u}_{k+1}^{n} + \prod_{i=1}^{s}(1+Q(\xi,t_{n+i-1})\delta t)\widehat{u}_{k}^{n} - (1-Q(\xi,t_n)\Delta t)^{-1}\widehat{u}_{k}^{n}. \tag{9}$$

Following the stability analysis by [11] then the stability function, the amplification factor for (9) is given by

$$\begin{aligned}\rho(\xi,t_n) &= 2(1-Q(\xi,t_n)\Delta t)^{-1} - \prod_{i=1}^{s}(1+Q(\xi,t_n)\delta t,\\ &= (1-Q(\xi,t_n)\Delta t)^{-1}\left[2-(1-Q(\xi,t_n)\Delta t)\prod_{i=1}^{s}(1+Q(\xi,t_{n+i-1})\delta t)\right]\\ &= (1-Q(\xi,t_n)\Delta t)^{-1}\tau(\xi,t_n)\end{aligned} \tag{10}$$

For the second term,$\tau(\xi,t_n)$, in (10) if we perform the multiplication we then conclude that

$$\tau(\xi,t_n) = \left[2-(1-Q(\xi,t_n)\Delta t)\left[1+\delta t\sum_{i=1}^{s}Q(\xi,t_{n+i-1})+\mathcal{O}(\delta t^2)\right]\right].$$

Therefore

$$\begin{aligned}\tau(\xi,t_n) &= 2-1+Q(\xi,t_n)\Delta t - \delta t(1-Q(\xi,t_n)\Delta t)\sum_{i=1}^{s}Q(\xi,t_{n+i-1})+\mathcal{O}(\delta t^2),\\ \tau(\xi,t_n) &\simeq 1+Q(\xi,t_n)\Delta t - \delta t\sum_{i=1}^{s}(-2h^{-2}\sin^2(\xi/2)+p(t_{n+i-1}))\\ &\le 1-2r_c\sin^2(\xi/2)+\Delta t p(t_n)+\sum_{i=1}^{s}\left(2r_f\sin^2(\xi/2)-\delta t p(t_{n+i-1})\right),\end{aligned}$$

where $r_c = \dfrac{\Delta t}{h^2}$, $r_f = \dfrac{\delta t}{h^2}$ corresponds to the coarse and fine propagator respectively, and $\dfrac{\Delta t}{\delta t} = s$. Hence for $-1 < p(t_n) < 0$, we conclude that $|\rho(\xi,t_n)| < |(1-Q(\xi,t_n)\Delta t)^{-1}||\tau(\xi,t_n)| < 1$. These conditions for the stability of the first case are summarized in the following theorem.

Theorem 1. *Consider the inverse model problem (1) solved by the parareal algorithm 2.1,*

$$u_{k+1}^{n+1} = G_{\Delta t}(u_{k+1}^{n}) + F_{s\delta t}(u_k^n) - G_{\Delta t}(u_k^n), \tag{11}$$

where $G_{\Delta t}$ and $F_{s\delta t}$ are the coarse and fine solvers respectively, and for $s = \Delta t/\delta t > 1$. If $r_f = \delta t/h^2$ satisfy the fine solver stability condition and $p(t) \in [-1,0]$ then the stability function $\rho(\xi,t_n)$, corresponding to (9) and defined by (10), satisfy

$$|\rho(\xi,t_n)| < 1,$$

for all $r_c = \Delta t/h^2$.

3.2 Case II, $\Delta t = \delta t$ $(s = 1)$

For the case when $s = 1$ the stability amplification factor is given by

$$\rho(\xi, t_n) = (1 + Q(\xi, t_n)\Delta t) - 2(1 - Q(\xi, t_n)\Delta t)^{-1}.$$

Because of the page limit the main conclusion will be summarized by the following theorem.

Theorem 2. *Consider the inverse model problem (1) solved by the parareal algorithm 2.1*

$$u_{k+1}^{n+1} = G_{\Delta t}(u_{k+1}^n) + F_{\delta t}(u_k^n) - G_{\Delta t}(u_k^n), \tag{12}$$

where $G_{\Delta t}$ and $F_{\delta t}$ are the coarse and fine solvers, respectively. Then

$$|\rho(\xi, t_n)| < 1,$$

for all $\dfrac{\delta t}{h^2} = \dfrac{\Delta t}{h^2} < \dfrac{1}{4}$ and $-1 < p(t) < 0$, where $\rho(\xi, t_n)$ is the amplification factor corresponding to (9) for $s = 1$ i.e. $\Delta t = \delta t$.

4 Model problem

For the validation of the necessary stability conditions of the presented in previous section we considered the model problems defined by

$$u_t = u_{xx} + p(t)u + \phi(x,t) \quad \text{over } \Omega = [0,1] \times (0,1),$$

with exact solution $u(x,t) = e^{-t^2}(\cos \pi x + \sin \pi x)$, and $\phi(x,t)$ defined in accordance to different definitions of $p(t)$. We considered $p(t) = -1 - t^2 < 0$ and $p(t) = 1 + 2t > 0$ for $t \in (0,1)$ respectively. The initial and boundary conditions and $E(t) = u(x_0, t)$ at $x_0 = 0.5$ are defined by the exact solution.

The stability functions (i.e. the amplification factors) are plotted using polar graphics for different values of r_c and r_f.

For the case when $s > 1$ the plots are presented in figure 1 for different values of $p(t)$, r_c and r_f values as well. Figure 1 show how the amplification factor given by (10) exceeded the desired stability bound for $r_f > 0.5$ and we also have the same conclusion for $-1 < p(t) < 0$ and positive values of $p(t)$.

For the case when $s = 1$ the plots of the amplification factor given by $\rho(\xi, t_n)$ in (10) are presented in figure 2. We consider different values for $r = \Delta t/h^2$ and $p(t)$, the plots shows how the stability amplification factor comply with the necessary conditions as stated in theorem 2.

References

1. L. Baffico, S. Bernard, Y. Maday, G. Turinci, and G. Zérah, *Parallel in time molecular dynamics simulations*, Phys. Rev. E, 66 (2002).
2. G. Bal, *On the convergence and the stability of the parareal algorithm to solve partial differential equations*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., vol. 40 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 425–432.

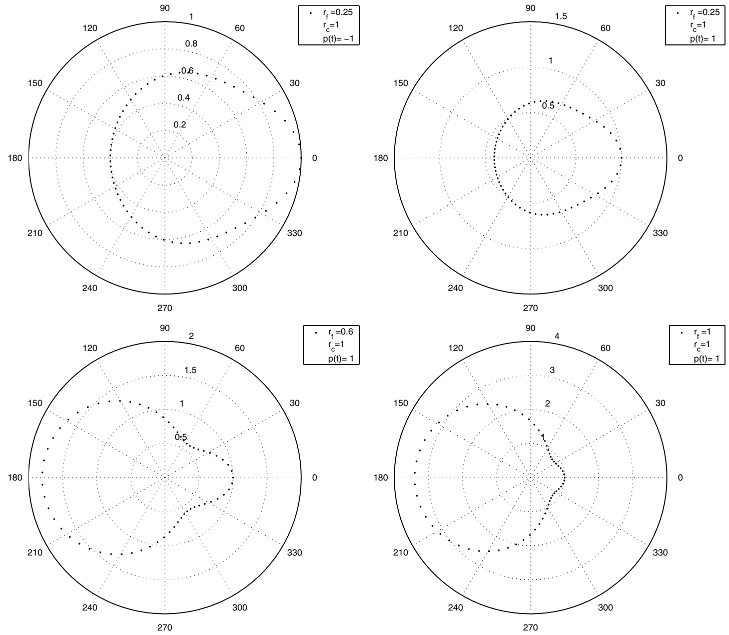


Fig. 1. The stability region for case 1 using different values of r_f, r_c and $p(t)$

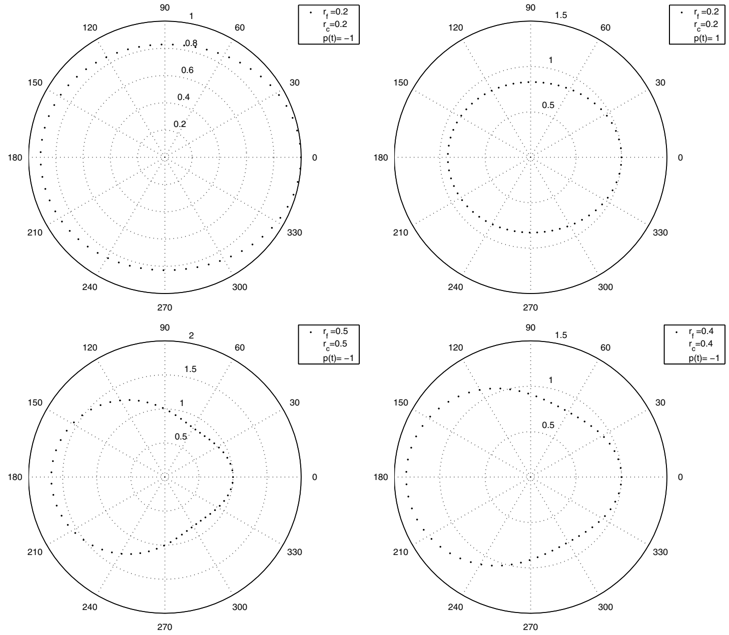


Fig. 2. The stability region for case 2 using different values of the ratio r and $p(t)$

3. G. Bal and Y. Maday, *A "parareal" time discretization for non-linear pdes with application to the pricing of an american put*, in Recent Developments in Domain Decomposition Methods. Proceedings of the Workshop on Domain Decomposition, Zürich, Switzerland, L. F. Pavarino and A. Toselli, eds., vol. 23 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2002, pp. 189–202.
4. J. R. Cannon, Y. Lin, and S. Wang, *Determination of source parameter in parabolic equation*, Mecanica, 3 (1992), pp. 85–94.
5. D. S. Daoud and D. Subasi, *A splitting up algorithm for the determination of the control parameter in multi dimensional parabolic problem*, Appl. Math. Comput., 166 (2005), pp. 584–595.
6. 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 (2003), pp. 1397–1434.
7. J.-L. Lions, Y. Maday, and G. Turinici, *A parareal in time discretization of pdes*, C.R. Acad. Sci. Paris, Serie I, 332 (2001), pp. 661–668.
8. Y. Maday and G. Turininci, *Parallel in time algorithms for quantum control: Parareal time discretization scheme*, Int. J. Quant. Chem., 93 (2003), pp. 223–228.
9. W. L. Miranker and W. Liniger, *Parallel methods for the numerical integration of ordinary differential equations*, Math. Comp., 21 (1967), pp. 303–320.
10. J. Nievergelt, *Parallel methods for integration ordinary differential equations*, Comm. ACM, 7 (1964), pp. 731–733.
11. G. A. Staff and E. M. Ronquist, *Stability of the parareal algorithm*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., vol. 40 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 449–456.
12. J. W. Thomas, *Numerical partial differential equations: Finite difference methods*, vol. 22 of Texts in Applied Mathematics, Springer Verlag, 1995.

A Schwarz Waveform Relaxation Method for Advection–Diffusion–Reaction Problems with Discontinuous Coefficients and Non-matching Grids

Martin J. Gander[1], Laurence Halpern[2], and Michel Kern[3]

[1] Section de Mathématiques, Université de Genève, Suisse.
Martin.Gander@math.unige.ch
[2] LAGA, Institut Galilée, Université Paris XIII, France.
halpern@math.univ-paris13.fr
[3] INRIA, Rocquencourt, France. Michel.Kern@inria.fr

Summary. We present a non-overlapping Schwarz waveform relaxation method for solving advection-reaction-diffusion problems in heterogeneous media. The domain decomposition method is global in time, which permits the use of different time steps in different subdomains. We determine optimal non-local, and optimized Robin transmission conditions. We also present a space-time finite volume scheme especially designed to handle such transmission conditions. We show the performance of the method on an example inspired from nuclear waste disposal simulations.

1 Motivation and Problem Setting

What to do with nuclear waste is a question being addressed by several organizations worldwide. Long term storage within a deep geological formation is one of the possible strategies, and Andra, the French Agency for Nuclear Waste Management, is currently carrying out feasibility studies for building such a repository. Given the time span involved (several hundreds of thousands, even millions, of years), physical experiments are at best difficult, and one must resort to numerical simulations to evaluate the safety of a proposed design.

Deep disposal of nuclear waste raises a number of challenges for numerical simulations: widely differing lengths and time-scales, highly variable coefficients and stringent accuracy requirements. In the site under consideration by Andra, the repository would be located in a highly impermeable geological layer, whereas the layers just above and below have very different physical properties. In the clay layer, the radionuclides move essentially because of diffusion, whereas in the dogger layer that is above the main phenomenon is advection (see [2] and the other publications in the same issue for a detailed discussion of numerical methods that can be applied to a simplified, though relevant, situation).

It is then natural to use different time steps in the various layers, so as to match the time step with the physics. To do this, we propose to adapt a global in time domain decomposition method proposed by Gander and Halpern in [1] (see also [4], and [6] for a different application) to the case of a model with discontinuous coefficients. The main advantage of the method is that it allows us to take different time steps in the subdomains, while only synchronizing at the end of the time simulation.

Our model problem is the one dimensional advection–diffusion–reaction equation

$$\begin{aligned} \mathcal{L}u := \frac{\partial u}{\partial t} - \frac{\partial}{\partial x}\left(D\frac{\partial u}{\partial x} - au\right) + bu = f, \qquad &\text{on } \mathbf{R}\times[0,T], \\ u(x,0) = u_0(x), \quad &x \in \mathbf{R}, \end{aligned} \tag{1}$$

where the reaction coefficient b is taken constant and the coefficients a and D are assumed constant on each half line $\mathbf{R}^+$ and $\mathbf{R}^-$, but may be discontinuous at 0,

$$a = \begin{cases} a^+ & x \in \mathbf{R}^+, \\ a^- & x \in \mathbf{R}^-, \end{cases} \qquad D = \begin{cases} D^+ & x \in \mathbf{R}^+, \\ D^- & x \in \mathbf{R}^-. \end{cases} \tag{2}$$

If $u_0 \in L^2(\mathbf{R})$ and $f \in L^2(0,T;L^2(\mathbf{R}))$, then problem (1) has a unique weak solution $u \in L^\infty(0,T;L^2(\mathbf{R}))\bigcap L^2(]0,T[;H^1(\mathbf{R}))$, see [5]. In the sequel, it will be convenient to use the notation

$$\begin{aligned} \mathcal{L}^\pm v &:= \frac{\partial v}{\partial t} - \frac{\partial}{\partial x}\left(D^\pm\frac{\partial v}{\partial x} - a^\pm v\right) + bv, \qquad x \in \mathbf{R}^\pm,\ t>0, \\ \mathcal{B}^\pm v &:= \mp D^\pm\frac{\partial v}{\partial x} \pm a^\pm v, \qquad\qquad\qquad x=0,\ t>0. \end{aligned} \tag{3}$$

One can show that (1), (2) is equivalent to the decomposed problem

$$\begin{aligned} \mathcal{L}^-u^- &= f, & &\text{on } \mathbf{R}^-\times[0,T], & \mathcal{L}^+u^+ &= f, & &\text{on } \mathbf{R}^+\times[0,T], \\ u^-(x,0) &= u_0(x), & &x\in\mathbf{R}^-, & u^+(x,0) &= u_0(x), & &x\in\mathbf{R}^+, \end{aligned} \tag{4}$$

together with the *coupling conditions*

$$u^+(0,t) = u^-(0,t), \quad \mathcal{B}^+u^+(0,t) = -\mathcal{B}^-u^-(0,t), \quad t\in[0,T]. \tag{5}$$

2 Domain Decomposition Algorithm

A simple algorithm based on relaxation of the coupling conditions (5) does not converge in general, not even in the simpler cases, see for example [7]. Instead of introducing a relaxation parameter, as in the classical Dirichlet-Neumann method, we introduce *transmission conditions* which imply the coupling conditions in (5) at convergence, and lead at the same time to an effective iterative method. We introduce two operators Λ^+ and Λ^- acting on functions defined on $[0,T]$, such that

$$\forall g \in L^2(\mathbf{R}),\ \widehat{\Lambda^\pm g}(\omega) = \lambda^\pm(\omega)\widehat{g}(\omega),\ \forall\omega\in\mathbf{R},$$

where $\widehat{g}$ is the Fourier transform of the function g, and $\lambda^\pm$ is the *symbol* of $\Lambda^\pm$. For $k = 0,1,2,\ldots$, we consider the Schwarz waveform relaxation algorithm

$$\begin{aligned}
\mathcal{L}^+ u_{k+1}^+ &= f, && \text{on } \mathbf{R}^+ \times [0,T],\\
u_{k+1}^+(x,0) &= u_0(x), && x \in \mathbf{R}^+,\\
(\mathcal{B}^+ + \Lambda^+)u_{k+1}^+(0,t) &= (-\mathcal{B}^- + \Lambda^+)u_k^-(0,t), && t \in [0,T],\\
\mathcal{L}^- u_{k+1}^- &= f, && \text{on } \mathbf{R}^- \times [0,T],\\
u_{k+1}^-(x,0) &= u_0(x), && x \in \mathbf{R}^-,\\
(\mathcal{B}^- + \Lambda^-)u_{k+1}^-(0,t) &= (-\mathcal{B}^+ + \Lambda^-)u_k^+(0,t), && t \in [0,T].
\end{aligned} \tag{6}$$

If this algorithm converges, then, provided $\Lambda^+ - \Lambda^-$ has a null kernel, the limit is a solution of the coupled problem (4), (5), and hence of the original problem (1).

2.1 Optimal Transmission Conditions

In order to choose the transmission operators Λ^+ and Λ^-, we first determine the convergence factor of the algorithm. Since the problem is linear, the error equations coincide with the homogeneous equations, that is we may take $f = 0$ and $u_0 = 0$ in algorithm (6) above. In order to use Fourier transforms in time, we assume that all functions are extended by 0 for $t < 0$. Denoting the errors in $\mathbf{R}^\pm$ by $e_k^\pm$, we see that the Fourier transforms of e_k^+ and e_k^- are given by

$$\begin{aligned}
\widehat{e_k^-}(x,\omega) &= \beta_k(\omega)\,\mathrm{e}^{r^+(a^-,D^-,\omega)x}, \quad (x,\omega) \in \mathbf{R}^- \times \mathbf{R},\\
\widehat{e_k^+}(x,\omega) &= \alpha_k(\omega)\,\mathrm{e}^{r^-(a^+,D^+,\omega)x}, \quad (x,\omega) \in \mathbf{R}^+ \times \mathbf{R},
\end{aligned} \tag{7}$$

where α_k and β_k are determined by the transmission conditions, and $r^+(a, D, \omega)$ and $r^-(a, D, \omega)$ are the roots with positive and negative real parts of the characteristic equation

$$Dr^2 - ar - (b + \mathrm{i}\omega) = 0. \tag{8}$$

If we substitute (7) into the transmission conditions of algorithm (6), we obtain over a double step of the algorithm

$$\alpha_{k+1}(\omega) = \rho(\omega)\alpha_{k-1}(\omega), \quad \beta_{k+1}(\omega) = \rho(\omega)\beta_{k-1}(\omega) \tag{9}$$

with the convergence factor $\rho(\omega)$ for each $\omega \in \mathbf{R}$ given by

$$\rho(\omega) = \frac{a^- - D^- r^+(a^-,D^-,\omega) + \lambda^+(\omega)}{a^+ - D^+ r^-(a^+,D^+,\omega) + \lambda^+(\omega)} \cdot \frac{a^+ - D^+ r^-(a^+,D^+,\omega) - \lambda^-(\omega)}{a^- - D^- r^+(a^-,D^-,\omega) - \lambda^-(\omega)}. \tag{10}$$

Remark 1. The previous equation shows that there is a choice for $\lambda^\pm$ that leads to convergence in two iterations. However, the corresponding operators are non-local in time (because of the square-root in $r^\pm(a, D, \omega)$. In the next Subsection, we therefore approximate the optimal operators by local ones.

2.2 Local Transmission Conditions

We approximate the square roots in the roots of (8) by parameters $p^\pm$ which leads to

$$\lambda_{\mathrm{app}}^+(\omega) = \frac{p^- - a^-}{2} \quad \text{and} \quad \lambda_{\mathrm{app}}^-(\omega) = \frac{p^+ + a^+}{2}, \quad \forall \omega \in \mathbf{R}, \tag{11}$$

and hence leads to Robin transmission conditions in algorithm (6).

We call the left *subdomain problem* the system formed by the first two equations of (4), together with the boundary condition

$$(\mathcal{B}^- + \lambda^-_{\text{app}})u^-(0,t) = g^-, \quad \text{for } t > 0,$$

and similarly for the right subdomain problem. As the coefficients are constants in each subdomain, we can prove the following result exactly as in [1] (see Theorem 5.3, and also [5] for the definition of the anisotropic Sobolev space $H^{2,1}(\mathbf{R}^- \times (0,T))$).

Theorem 1 (Well Posedness of Subdomain Problems). *Let $u_0 \in H^1(\mathbf{R})$, $f \in L^2(0,T;\mathbf{R})$, and $g^\pm \in H^{1/4}(0,T)$. Then, for any real numbers $\lambda^\pm_{app}$, the subdomain problems have unique solutions $u^\pm \in H^{2,1}(\mathbf{R}^- \times (0,T))$.*

Therefore the subdomain solutions are smooth enough to apply the transmission operators and this proves by induction that algorithm (6) with the Robin transmission conditions (11) is well defined (see also Theorem 5.4 in [1]).

Theorem 2 (Well Posedness of the Algorithm). *Let $f \in L^2(0,T;\mathbf{R})$, $u_0 \in H^1(\mathbf{R})$, and the initial guesses $u_0^\pm \in H^{2,1}(\mathbf{R}^- \times (0,T)) \times H^{2,1}(\mathbf{R}^+ \times (0,T))$. Then, for any real numbers $p^\pm$, algorithm* (6) *with Robin transmission conditions (11) is well defined in $H^{2,1}(\mathbf{R}^- \times (0,T)) \times H^{2,1}(\mathbf{R}^+ \times (0,T))$.*

Convergence of the algorithm follows from energy estimates similar to the ones in [1], where however the additional difficulty due to the discontinuities leads to additional constraints on the parameters.

Theorem 3 (Convergence of the Algorithm). *If the three following constraints are satisfied: $\lambda^-_{app} + \lambda^+_{app} > 0$, $\lambda^-_{app} - \lambda^+_{app} + \frac{a^+}{2} \geq 0$, $\lambda^-_{app} - \lambda^+_{app} + \frac{a^-}{2} \leq 0$, then algorithm* (6)*, with Robin transmission conditions (11), is convergent.*

Note that in the case of constant coefficients, and $p^+ = p^- = p$, the constraints reduce to $p > 0$, which is consistent with results in [1].

How should the parameters $p^\pm$ be chosen? A simple approach is to use a low frequency approximation, obtained by a Taylor expansion of the square roots in the roots of (8), which leads to

$$p^+ = \sqrt{(a^+)^2 + 4D^+b}, \quad p^- = \sqrt{(a^-)^2 + 4D^-b}. \tag{12}$$

Such transmission conditions are however not very effective for high frequencies. A better approach is to minimize the convergence factor, i.e. to solve the min-max problem

$$\min_{p^+,p^-}\left(\max_{0\leq\omega\leq\omega_{\max}} |\rho(\omega,p^+,p^-,a^+,a^-,D^+,D^-,b)|\right), \tag{13}$$

where ρ is given in (10). As we are working with a numerical scheme, the frequencies cannot be arbitrarily high, but can be restricted to $\omega_{\max} = \pi/\Delta t$.

Theorem 4. *If $p^+ = p^- = p$, then for $a^+, a^- > 0$ the solution of the min-max problem (13) is for Δt small given by*

$$p \approx \frac{\left(2^3\pi(D^+D^-)\left(\sqrt{D^+}+\sqrt{D^-}\right)^2\left(a^+-a^-+\sqrt{(a^+)^2+4D^+b}+\sqrt{(a^-)^2+4D^-b}\right)^2\right)^{\frac{1}{4}}}{\sqrt{D^+}+\sqrt{D^-}}\Delta t^{-\frac{1}{4}}, \tag{14}$$

which leads to the asymptotic bound on the convergence factor

$$|\rho| \leq 1 - \left(\frac{2^5\left(\sqrt{D^+}+\sqrt{D^-}\right)^2\left(a^+-a^-+\sqrt{(a^+)^2+4D^+b}+\sqrt{(a^-)^2+4D^-b}\right)^2}{D^+D^-\pi} \right)^{\frac{1}{4}} \Delta t^{\frac{1}{4}}. \tag{15}$$

Theorem 5. *If $D^+ = D^- = D$, then for $a^+, a^- > 0$ the solution of the min-max problem (13) is for Δt small given by*

$$\begin{aligned} p^+ &\approx (2^9\pi^3 D^3(a^+ - a^- + \sqrt{(a^+)^2 + 4Db} + \sqrt{(a^-)^2 + 4Db})^2)^{\frac{1}{8}} \Delta t^{-\frac{3}{8}}, \\ p^- &\approx (2^{-5}\pi D(a^+ - a^- + \sqrt{(a^+)^2 + 4Db} + \sqrt{(a^-)^2 + 4Db})^6)^{\frac{1}{8}} \Delta t^{-\frac{1}{8}}, \end{aligned} \tag{16}$$

which leads to the asymptotic bound on the convergence factor

$$|\rho| \leq 1 - \left(\frac{2^{13}(a^+-a^-+\sqrt{(a^+)^2+4Db}+\sqrt{(a^-)^2+4Db})^2}{D\pi} \right)^{\frac{1}{8}} \Delta t^{\frac{1}{8}}. \tag{17}$$

The most general case where $p^+ \neq p^-$ and $D^\pm$ are arbitrary is asymptotically the most interesting one, since the discontinuity in D changes the exponent in the asymptotically optimal parameter and hence in the convergence factor. This case is currently under investigation.

3 Finite Volume Discretization of the Algorithm

We discretize the subdomain problem by a space-time finite volume method, implicit in time and upwind for the advective part. We denote the space and time steps by Δx, Δt, the grid points by $x_j = j\Delta x$, $j = 0, \ldots, N_x$ (with $N_x \Delta x = L$), and $t^n = n\Delta t$, $n = 0, \ldots, N_t$, (with $N_t \Delta t = T$). We also let $u_h = (u_j^n)_{(j,n)}$ be the approximate solution, with $u_j^n \approx u(x_j, t^n)$. We consider u_h as a constant function on each rectangle $R_j^n = (x_{j-1/2}, x_{j+1/2}) \times (t^{n-1/2}, t^{n+1/2})$ (the fully shaded rectangle in Figure 1). The discrete derivatives are defined by the difference quotient, and

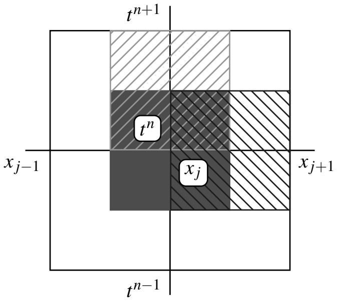


Fig. 1. Finite volume grid. Function is constant on solid rectangle, x -derivative on right-hashed rectangle, t-derivative on left-hashed rectangle.

are constant on staggered grids, as indicated in Figure 1. Last, we let $u_j^{n+1/2} = \dfrac{u_j^n + u_j^{n+1}}{2}$.

The discrete scheme for interior points in each subdomain is obtained by integrating the partial differential equation in (6) over the rectangle R_j^n and then using standard finite volume approximations, which leads to

$$\frac{u_j^{n+1}-u_j^n}{\Delta t} - D\frac{u_{j+1}^{n+1/2}-2u_j^{n+1/2}+u_{j-1}^{n+1/2}}{\Delta x^2} + a\frac{u_j^{n+1/2}-u_{j-1}^{n+1/2}}{\Delta x} + bu_j^{n+1/2} = f_j^{n+1/2}. \quad (18)$$

The scheme can be shown to be unconditionally stable, and first order accurate [3].

The main interest of the finite volume method is that we can handle the transmission conditions in (6) in a natural way. Now we just integrate over half the cell, for example on the right subdomain, and use the transmission condition on the cell boundary on the left, to obtain

$$\frac{\Delta x}{2}\frac{u_0^{n+1}-u_0^n}{\Delta t} - D\frac{u_1^{n+1/2}-u_0^{n+1/2}}{\Delta x} + au_0^{n+1/2} + \frac{\Delta x}{2}b\,u_0^{n+1} + \lambda_{\text{app}}\,u_0^{n+1/2} = g^{n+1/2}, \quad (19)$$

and similarly over the left subdomain. In the same way, we obtain an expression for the operator on the right hand side of the transmission condition. One can show that if the entire domain is homogeneous, then the scheme with the discrete boundary conditions coincides with the interior scheme applied at the interface node [3].

Since the space and time steps will usually be different on the two sides of the interface, we introduce an L^2 projection operator on the boundary (acting on step functions defined in the time domain), as was done in [4].

4 Numerical Experiments

We present an example of the behavior of our algorithm, with discontinuous coefficients, and different time and space steps in the two subdomains. The parameters for the two subdomains are shown in Table 1. Several snapshots of the solution, at 3 different times, and for two different iterations are shown in Figure 2.

	D	a	p	Δx	Δt
Left subdomain $\mathbf{R}^-$	$4\ 10^{-2}$	4	18.5	10^{-2}	$4\ 10^{-3}$
Right subdomain, $\mathbf{R}^+$	$12\ 10^{-2}$	2	6.4	$2\ 10^{-2}$	$2\ 10^{-3}$

Table 1. Physical and numerical parameters for an example.

Last, to illustrate Theorem 5, we show in Figure 3 the number of iterations needed to reduce the residual by 10^6 when running the algorithm on the discretized problem, for various values of the parameters p^+ and p^-. The parameters corresponding to Theorem 5 and to the values found by minimizing the continuous convergence factor (10) are both shown in the figure (we use the same values as in Table 1 above, except that now $D^+ = D^- = 4\,10^{-2}$).

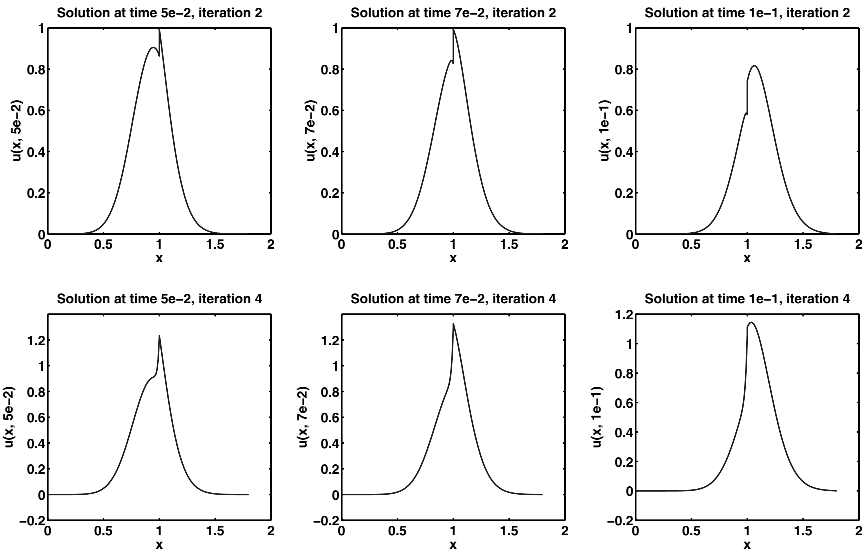


Fig. 2. Evolution of the solution at two different iterations. Top row: iteration 2, bottom row: iteration 4. Left column: $t = 0.05$, middle column $t = 0.07$, right column $t = 0.1$.

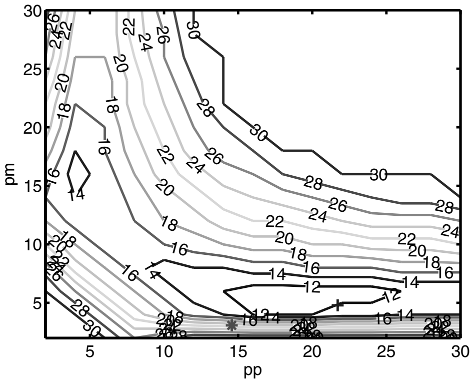


Fig. 3. Level curves for the number of iterations needed to reach convergence for various values of the parameters p^- and p^+. The lower left star marks the parameters derived from Theorem 5, whereas the upper right cross shows the "optimal" parameters, as found by numerically minimizing the continuous convergence rate.

References

1. D. Bennequin, M. J. Gander, and L. Halpern, *Optimized Schwarz waveform relaxation methods for convection reaction diffusion problems*, Tech. Rep. 24, Institut Galilée, Paris XIII, 2004.
2. A. Bourgeat, M. Kern, S. Schumacher, and J. Talandier, *The* Couplex *test cases: Nuclear waste disposal simulation*, Computational Geosciences, 8 (2004), pp. 83–98.
3. M. J. Gander, L. Halpern, and M. Kern, *A Schwarz Waveform Relaxation method for advection–diffusion–reaction problems with discontinuous coefficients and non-matching grids*, in Proceedings of the 16th International Conference on Domain Decomposition Methods, O. B. Widlund and D. E. Keyes, eds., Springer, 2006. these proceedings.
4. M. J. Gander, L. Halpern, and F. Nataf, *Optimal Schwarz waveform relaxation for the one dimensional wave equation*, SIAM J. Numer. Anal., 41 (2003), pp. 1643–1681.
5. J.-L. Lions and E. Magenes, *Nonhomogeneous Boundary Value Problems and Applications*, vol. II of Die Grundlehren der mathematischen Wissenschaften Band 182, Springer, 1972. Translated from the French by P. Kenneth.
6. V. Martin, *Schwarz waveform relaxation method for the viscous shallow water equations*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 653–660.
7. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.

On the Superlinear and Linear Convergence of the Parareal Algorithm

Martin J. Gander[1] and Stefan Vandewalle[2]

[1] Section de Mathématiques, University of Geneva, 1211 Geneva 4, Switzerland. Martin.Gander@math.unige.ch

[2] Department of Computer Science, Katholieke Universiteit Leuven, 3001 Leuven, Belgium. Stefan.Vandewalle@cs.kuleuven.be

Summary. The parareal algorithm is a method to solve time dependent problems parallel in time: it approximates parts of the solution later in time simultaneously to parts of the solution earlier in time. In this paper the relation of the parareal algorithm to space-time multigrid and multiple shooting methods is first briefly discussed. The focus of the paper is on some new convergence results that show superlinear convergence of the algorithm when used on bounded time intervals, and linear convergence for unbounded intervals.

1 Introduction

The parareal algorithm was first presented in [8] to solve evolution problems in parallel. The name was chosen to indicate that the algorithm is well suited for parallel real time computations of evolution problems whose solution cannot be obtained in real time using one processor only. The method approximates successfully the solution later in time before having fully accurate approximations from earlier times. The algorithm has received a lot of attention over the past few years; for extensive experiments and studies of convergence and stability issues we refer to [9, 3] and the contributions in the 15th Domain Decomposition Conference Proceedings [7].

Parareal is not the first algorithm to propose the solution of evolution problems in a time-parallel fashion. Already in 1964, Nievergelt suggested a parallel time integration algorithm [11], which led to multiple shooting methods. The idea is to decompose the time integration interval into subintervals, to solve an initial value problem on each subinterval concurrently, and to force continuity of the solution branches on successive intervals by means of a Newton procedure. Since then, many variants of the method have been developed and used for the time-parallel integration of evolution problems, see e.g. [1, 2]. In [4], we show that the parareal algorithm can be interpreted as a particular multiple shooting method, where the Jacobian matrix is approximated in a finite difference way on the coarse mesh in time.

In 1967, Miranker and Liniger [10] proposed a family of predictor-corrector methods, in which the prediction and correction steps can be performed in parallel over a number of time-steps. Their idea was to "widen the computational front", i.e., to allow processors to compute solution values on several time-steps concurrently. A similar motivation led to the block time integration methods by Shampine and Watts [13]. More recently, [12] and [15] considered the time-parallel application of iterative methods to the system of equations derived with implicit time-integration schemes. Instead of iterating until convergence over each time step before moving on to the next, they showed that it is possible to iterate over a number of time steps at once. Thus a different processor can be assigned to each time step and they all iterate simultaneously. The acceleration of such methods by means of a multigrid technique led to the class of parabolic multigrid methods, as introduced in [5]. The multigrid waveform relaxation and space-time multigrid methods also belong to that class. In [14], a time-parallel variant was shown to achieve excellent speedups on a computer with 512 processors; while run as sequential algorithm the method is comparable to the best classical time marching schemes. Experiments with time-parallel methods on 2^{14} processors are reported in [6]. In [4], it is shown that the parareal algorithm can also be cast into the parabolic multigrid framework. In particular, the parareal algorithm can be identified with a two level multigrid Full Approximation Scheme, with a special Jacobi-type smoother, with strong semi-coarsening in time, and selection and extension operators for restriction and interpolation.

2 A Review of the Parareal Algorithm

The parareal algorithm for the system of ordinary differential equations

$$\boldsymbol{u}' = \boldsymbol{f}(\boldsymbol{u}), \quad \boldsymbol{u}(0) = \boldsymbol{u}_0, \quad t \in [0,T], \tag{1}$$

is defined using two propagation operators. The operator $G(t_2, t_1, \boldsymbol{u}_1)$ provides a rough approximation to $\boldsymbol{u}(t_2)$ of the solution of (1) with initial condition $\boldsymbol{u}(t_1) = \boldsymbol{u}_1$, whereas the operator $F(t_2, t_1, \boldsymbol{u}_1)$ provides a more accurate approximation of $\boldsymbol{u}(t_2)$. The algorithm starts with an initial approximation $\boldsymbol{U}_n^0$, $n = 0, 1, \ldots, N$ at time $t_0, t_1, \ldots, t_N$ given for example by the sequential computation of $\boldsymbol{U}_{n+1}^0 = G(t_{n+1}, t_n, \boldsymbol{U}_n^0)$, with $\boldsymbol{U}_0^0 = \boldsymbol{u}_0$, and then performs for $k = 0, 1, 2, \ldots$ the correction iteration

$$\boldsymbol{U}_{n+1}^{k+1} = G(t_{n+1}, t_n, \boldsymbol{U}_n^{k+1}) + F(t_{n+1}, t_n, \boldsymbol{U}_n^k) - G(t_{n+1}, t_n, \boldsymbol{U}_n^k). \tag{2}$$

Note that, for $k \to \infty$, the method will upon convergence generate a series of values $\boldsymbol{U}_n$ that satisfy $\boldsymbol{U}_{n+1} = F(t_{n+1}, t_n, \boldsymbol{U}_n)$. That is, the approximation at time t_n will have achieved the accuracy of the F-propagator. Alternatively, one can restrict the number of iterations of (2) to a finite value. In that case, (2) defines a new time-integration scheme. The accuracy of the $\boldsymbol{U}_n^k$ values is characterized by a theorem from [8]. The theorem applies for a scalar linear problem of the form

$$u' = -au, \quad u(0) = u_0, \quad t \in [0,T]. \tag{3}$$

Theorem 1. *Let $\Delta T = T/N$, $t_n = n\Delta T$ for $n = 0, 1, \ldots, N$. Let $F(t_{n+1}, t_n, U_n^k)$ be the exact solution at t_{n+1} of (3) with $u(t_n) = U_n^k$, and $G(t_{n+1}, t_n, U_n^k)$ the corresponding backward Euler approximation with time step ΔT. Then,*

$$\max_{1\leq n\leq N} |u(t_n) - U_n^k| \leq C_k \Delta T^{k+1}, \tag{4}$$

where the constant C_k is independent of ΔT.

Hence, for a fixed iteration step k, the algorithm behaves like an $O(\Delta T^{k+1})$ method. Note that the convergence of the algorithm for a fixed ΔT and increasing number of iterations k is not covered by the above theorem, because the constant C_k grows with k in the estimate of the proof in [8].

3 Convergence analysis for a scalar ODE

We show two new convergence result for fixed ΔT when k becomes large. The first result is valid on bounded time intervals, $T < \infty$, whereas the second one also holds for unbounded time intervals. The results apply for an arbitrary explicit or implicit one step method applied to (3) with $a \in \mathbb{C}$, i.e., $U_{n+1} = \beta U_n$, in the region of absolute stability of the method, i.e., $|\beta| \leq 1$.

In our analysis an important role will be played by a strictly upper triangular Toeplitz matrix M of size N. Its elements are defined as follows,

$$M_{ij} = \begin{cases} \beta^{j-i-1} & \text{if } j > i, \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

A key property of M, whose proof we omit here, is that

$$|\beta| \leq 1 \quad \Longrightarrow \quad ||M^k||_\infty \leq \binom{N-1}{k}. \tag{6}$$

Theorem 2 (Superlinear convergence on bounded intervals). *Let $T < \infty$, $\Delta T = T/N$, and $t_n = n\Delta T$ for $n = 0, 1, \ldots, N$. Let $F(t_{n+1}, t_n, U_n^k)$ be the exact solution at t_{n+1} of (3) with $u(t_n) = U_n^k$, and let $G(t_{n+1}, t_n, U_n^k) = \beta U_n^k$ be a one step method in its region of absolute stability, i.e., $|\beta| \leq 1$. Then,*

$$\max_{1\leq n\leq N} |u(t_n) - U_n^k| \leq \frac{|e^{-a\Delta T} - \beta|^k}{k!} \prod_{j=1}^{k} (N - j) \max_{1\leq n\leq N} |u(t_n) - U_n^0|. \tag{7}$$

If the local truncation error of G is bounded by $C\Delta T^{p+1}$, then

$$\max_{1\leq n\leq N} |u(t_n) - U_n^k| \leq \frac{(CT)^k}{k!} \Delta T^{pk} \max_{1\leq n\leq N} |u(t_n) - U_n^0|. \tag{8}$$

Proof. We denote by e_n^k the error at iteration step n of the parareal algorithm at time t_n, i.e., $e_n^k := u(t_n) - U_n^k$. With (2) and an induction argument on n, it is easy to see that this error satisfies

$$e_n^{k+1} = \beta e_{n-1}^{k+1} + (e^{-a\Delta T} - \beta) e_{n-1}^k = (e^{-a\Delta T} - \beta) \sum_{j=1}^{n-1} \beta^{n-j-1} e_j^k.$$

This relation can be written in matrix form by collecting e_n^k in the vector $\boldsymbol{e}^k = (e_N^k, e_{N-1}^k, \ldots, e_1^k)^T$, which leads to

$$\boldsymbol{e}^{k+1} = (e^{-a\Delta T} - \beta) M \boldsymbol{e}^k, \tag{9}$$

where the matrix M is given in (5). By induction on (9), we obtain

$$||\boldsymbol{e}^k||_\infty \leq |(e^{-a\Delta T} - \beta)|^k ||M^k||_\infty ||\boldsymbol{e}^0||_\infty, \tag{10}$$

which together with (6) implies (7). The bound (8) follows from the bound on the local truncation error together with a simple estimate of the product,

$$\frac{|e^{-a\Delta T} - \beta|^k}{k!} \prod_{j=1}^{k} (N - j) \leq \frac{C^k \Delta T^{(p+1)k}}{k!} N^k = \frac{(CT)^k}{k!} \Delta T^{pk}.$$

Remark 1. The product term in (7) shows that the parareal algorithm converges for any ΔT on any bounded time interval in at most $N-1$ steps. Furthermore the algorithm converges superlinearly, as the division by $k!$ in (7) shows. Finally, if instead of an exact solution on the subintervals a fine grid approximation is used, the proof remains valid with some minor modifications.

Theorem 3 (Linear convergence on long time intervals). *Let ΔT be given, and $t_n = n\Delta T$ for $n = 0, 1, \ldots$. Let $F(t_{n+1}, t_n, U_n^k)$ be the exact solution at t_{n+1} of (3) with $u(t_n) = U_n^k$, and let $G(t_{n+1}, t_n, U_n^k) = \beta U_n^k$ be a one step method in its region of absolute stability, with $|\beta| < 1$. Then,*

$$\sup_{n>0} |u(t_n) - U_n^k| \leq \left(\frac{|e^{-a\Delta T} - \beta|}{1 - |\beta|} \right)^k \sup_{n>0} |u(t_n) - U_n^0|. \tag{11}$$

If the local truncation error of G is bounded by $C\Delta T^{p+1}$, then

$$\sup_{n>0} |u(t_n) - U_n^k| \leq \left(\frac{C\Delta T^p}{\Re(a) + O(\Delta T)} \right)^k \sup_{n>0} |u(t_n) - U_n^0|. \tag{12}$$

Proof. In the present case M, as defined in (5), is an infinite dimensional Toeplitz operator. Its infinity norm is given by

$$||M||_\infty = \sum_{j=0}^{\infty} |\beta|^j = \frac{1}{1 - |\beta|}.$$

Using (9), we obtain for the error vectors $\boldsymbol{e}^k$ of infinite length the relation

$$||\boldsymbol{e}^k||_\infty \leq |(e^{-a\Delta T} - \beta)|^k ||M||_\infty^k ||\boldsymbol{e}^0||_\infty = \left(\frac{|(e^{-a\Delta T} - \beta)|}{1 - |\beta|} \right)^k ||\boldsymbol{e}^0||_\infty, \tag{13}$$

which proves the first result. For the second result, the bound on the local truncation error, $|e^{-a\Delta T} - \beta| \leq C\Delta T^{p+1}$, implies for $p > 0$ that $\beta = 1 - a\Delta T + O(\Delta T^2)$, and hence $1 - |\beta| = \Re(a)\Delta T + O(\Delta T^2)$, which implies (12).

4 Convergence analysis for partial differential equations

We now use the results derived in Section 3 to investigate the performance of the parareal algorithm for partial differential equations. We consider two model problems, a diffusion problem and an advection problem. For the diffusion case, we consider the heat equation, without loss of generality in one dimension,

$$u_t = u_{xx}, \quad \text{in } \Omega = \mathbb{R}, \quad u(0,x) \in L^2(\Omega). \tag{14}$$

Using a Fourier transform in space, this equation becomes a system of decoupled ordinary differential equations for each Fourier mode ω,

$$\hat{u}_t = -\omega^2 \hat{u}, \tag{15}$$

and hence the convergence results of Theorems 2 and 3 can be directly applied. If we discretize the heat equation in time using the backward Euler method, then we have the following convergence result for the parareal algorithm.

Theorem 4 (Heat Equation Convergence Result). *Under the conditions of Theorem 2, with $a = \omega^2$, and $G(t_{n+1}, t_n, U_n^k) = \beta U_n^k$ with $\beta = \frac{1}{1+\omega^2 \Delta T}$, from the backward Euler method, the parareal algorithm has a superlinear bound on the convergence rate on bounded time intervals,*

$$\max_{1\le n\le N} ||u(t_n) - U_n^k||_2 \le \frac{\gamma_s^k}{k!} \prod_{j=1}^{k} (N-j) \max_{1\le n\le N} ||u(t_n) - U_n^0||_2, \tag{16}$$

where $||\cdot||_2$ denotes the spectral norm in space and the constant γ_s is universal, $\gamma_s = 0.2036321888$. On unbounded time intervals, we have

$$\sup_{n>0} ||u(t_n) - U_n^k||_2 \le \gamma_l^k \sup_{n>0} ||u(t_n) - U_n^0||_2, \tag{17}$$

where the universal constant $\gamma_l = 0.2984256075$.

Proof. A simple calculation shows that the numerator in the superlinear bound (7) is uniformly bounded for the backward Euler method by

$$|e^{-\omega^2 \Delta T} - \frac{1}{1+\omega^2 \Delta T}| \le \gamma_s,$$

where the maximum γ_s is attained at $\omega^2 \Delta T = \bar{x}_s := 2.512862417$. This leads to (16) by using the Parseval-Plancherel identity.

The convergence factor in the linear bound (12) is also bounded by

$$\frac{|e^{-\omega^2 \Delta T} - \frac{1}{1+\omega^2 \Delta T}|}{1 - \frac{1}{1+\omega^2 \Delta T}} \le \gamma_l,$$

where the maximum γ_l is attained at $\omega^2 \Delta T = \bar{x}_l := 1.793282133$, which leads to (17) using the Parseval-Plancherel identity.

Next, we consider a pure advection problem

$$u_t = u_x, \quad \text{in } \Omega = \mathbb{R}, \quad u(0,x) \in L^2(\Omega). \tag{18}$$

Using a Fourier transform in time, this equation becomes

$$\hat{u}_t = -i\omega\hat{u}. \tag{19}$$

The convergence results of Theorems 2 and 3 can be directly applied. If we discretize the advection equation in time using the backward Euler method, then we have the following convergence result for the parareal algorithm.

Theorem 5 (Advection Equation Convergence Result). *Under the conditions of Theorem 2, with $a = -i\omega$, and $G(t_{n+1}, t_n, U_n^k) = \beta U_n^k$ with $\beta = \frac{1}{1+i\omega\Delta T}$, from the backward Euler method, the parareal algorithm has a superlinear bound on the convergence rate on bounded time intervals,*

$$\max_{1\le n\le N} ||u(t_n) - U_n^k||_2 \le \frac{\alpha_s^k}{k!}\prod_{j=1}^{k}(N-j) \max_{1\le n\le N} ||u(t_n) - U_n^0||_2, \tag{20}$$

where the constant α_s is universal, $\alpha_s = 1.224353426$.

Proof. A simple calculation shows that the numerator in the superlinear bound (7) is uniformly bounded, for the backward Euler method, by

$$|e^{-i\omega\Delta T} - \frac{1}{1+i\omega\Delta T}| \le \alpha_s,$$

which leads to (20) using the Parseval-Plancherel identity.

Remark 2. There is no long term convergence result for (18). The convergence factor in (11) is not bounded by a quantity less than one.

5 Numerical Experiments

In order to verify the theoretical results, we first show some numerical experiments for the scalar model problem (3) with $f = 0$, $a = 1$, $u_0 = 1$. The Backward Euler method is chosen for both the coarse approximation and the fine approximation, with time step ΔT and $\Delta T/m$ respectively. We show in Figure 1 the convergence results obtained for $T = 1$, $T = 10$ and $T = 50$, using $N = 10$ and $m = 20$ in each case. One can clearly see that the parareal algorithm has two different convergence regimes: for $T = 1$, the algorithm converges superlinearly, and the superlinear bound from Theorem 2 is quite sharp. For $T = 10$, the convergence rate is initially linear, and then a transition occurs to the superlinear convergence regime. Finally, for $T = 50$, the algorithm is in the linear convergence regime and the bound from Theorem 3 is quite sharp. Note also that the bound from Theorem 1 indicates stagnation for $T = 10$, since $\Delta T = 1$, and divergence for $T = 50$, since then $\Delta T > 1$. The parareal algorithm does however also converge for $\Delta T \ge 1$.

We now turn our attention to the PDE case and show some experiments for the heat equation $u_t = u_{xx} + f$, in $(0, L) \times (0, T]$ with homogeneous initial and boundary

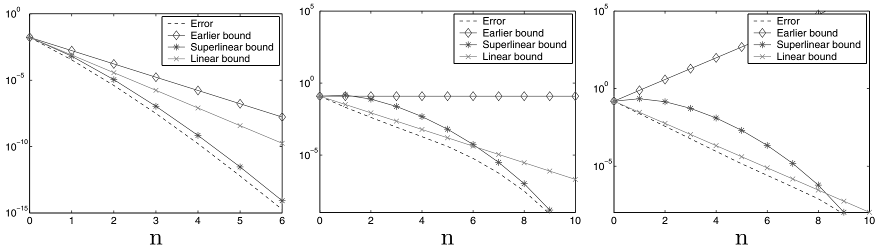


Fig. 1. Convergence of the parareal algorithm for (3) on a short, medium and long time interval.

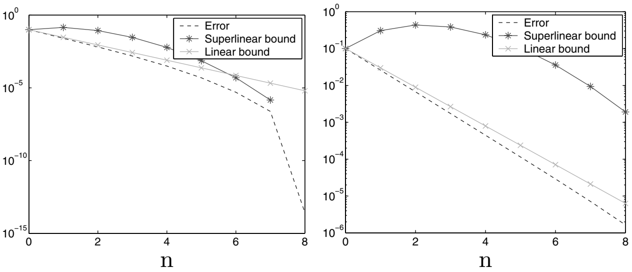


Fig. 2. Error in the L^∞ norm in time and L^2 norm in space for the parareal algorithm applied to the heat equation, on a short (left) and long (right) interval.

conditions and with $f = x^4(1-x)+t^2$. The domain length L is chosen such that the linear bound in (17) of Theorem 4 is attained, which implies that $L = \pi\sqrt{\Delta T/\bar{x}_s}$. With $\Delta T = 1/2$ and $m = 10$, we obtain the results shown in Figure 2. On the left, results are shown for $T = 4$, where the algorithm with $\Delta T = 1/2$ will converge in 8 steps. One can see that this is clearly the case. Before that, the algorithm is in the superlinear convergence regime, as predicted by the superlinear bound. Note that the latter bound indicates zero as the error at the eighth step, and thus cannot be plotted on the logarithmic scale. On the right, the error is shown for $T = 8$, and the algorithm is clearly in the linear convergence regime.

References

1. A. Bellen and M. Zennaro, *Parallel algorithms for initial-value problems for difference and differential equations*, J. Comput. Appl. Math., 25 (1989), pp. 341–350.
2. P. Chartier and B. Philippe, *A parallel shooting technique for solving dissipative ODEs*, Computing, 51 (1993), pp. 209–236.
3. 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 (2003), pp. 1397–1434.
4. M. J. Gander and S. Vandewalle, *Analysis of the parareal time-parallel time-integration method*, Technical Report TW 443, K.U. Leuven, Department of Computer Science, November 2005.

5. W. Hackbusch, *Parabolic multi-grid methods*, in Computing Methods in Applied Sciences and Engineering, VI, R. Glowinski and J.-L. Lions, eds., North-Holland, 1984, pp. 189–197.
6. G. Horton, S. Vandewalle, and P. Worley, *An algorithm with polylog parallel complexity for solving parabolic partial differential equations*, SIAM J. Sci. Comput., 16 (1995), pp. 531–541.
7. R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., *Proceedings of the 15th international domain decomposition conference*, Springer LNCSE, 2003.
8. J.-L. Lions, Y. Maday, and G. Turinici, *A parareal in time discretization of pde's*, C.R. Acad. Sci. Paris, Serie I, 332 (2001), pp. 661–668.
9. Y. Maday and G. Turinici, *A parareal in time procedure for thecontrol of partial differential equations*, C.R.A.S. Sér. I Math, 335 (2002), pp. 387–391.
10. W. L. Miranker and W. Liniger, *Parallel methods for the numerical integration of ordinary differential equations*, Math. Comp., 91 (1967), pp. 303–320.
11. J. Nievergelt, *Parallel methods for integration ordinary differential equations*, Comm. ACM, 7 (1964), pp. 731–733.
12. J. H. Saltz and V. K. Naik, *Towards developing robust algorithms for solving partial differential equations on mimd machines*, Parallel Comput., 6 (1988), pp. 19–44.
13. L. F. Shampine and H. A. Watts, *Block implicit one-step methods*, Math. Comp., 23 (1969), pp. 731–740.
14. S. Vandewalle and E. V. de Velde, *Space-time concurrent multigrid waveform relaxation*, Ann. Numer. Math., 1 (1994), pp. 347–363.
15. D. E. Womble, *A time-stepping algorithm for parallel computers*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 824–837.

Optimized Sponge Layers, Optimized Schwarz Waveform Relaxation Algorithms for Convection-diffusion Problems and Best Approximation

Laurence Halpern[1]

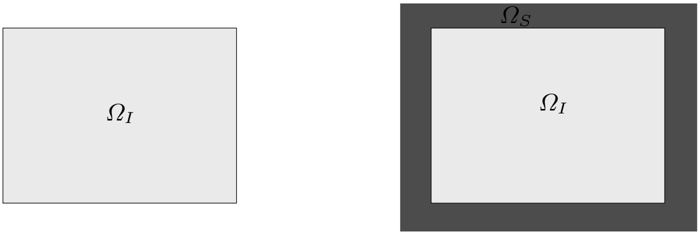
[1] Laboratoire Analyse, Géométrie et Applications, Université Paris XIII, 99 Avenue J.-B. Clément, 93430 Villetaneuse, France. halpern@math.univ-paris13.fr

Summary. When solving an evolution equation in an unbounded domain, various strategies have to be applied, aiming at reducing the number of unknowns and the computational cost, from infinite to a finite and not too large number. Among them are truncation of the domain with a sponge boundary, and Schwarz Waveform Relaxation algorithm with overlap. These problems are closely related, as they both use the Dirichlet-to-Neumann map as a starting point for transparent boundary condition on the one hand, and optimal algorithms on the other hand. Differential boundary conditions can then be obtained by minimization of the reflection coefficients or the convergence rate. In the case of unsteady convection-diffusion problems, this leads to a non standard complex best approximation problem that we present and solve.

1 Problems settings

1.1 Absorbing boundary conditions with a sponge

When computing the flow passed an airfoil, or the diffraction by an object, the mathematical problem is set on an unbounded domain, while the domain of interest (*i.e.* where the knowledge of the solution is relevant), Ω_I, is bounded and sometimes small . Then a computational domain is needed, called Ω_C, on which the problem is actually solved. The problem must be complemented with boundary conditions on $\partial\Omega_C$. It is desirable to introduce a *sponge boundary* Ω_S which absorbs the spurious reflexion, see Figure 1. The question we address here is the following: how to design boundary conditions on $\partial\Omega_C$ such that, for a given sponge layer of size L, the error in Ω_I be minimized. The issue is somewhat different from those used in the usual absorbing boundary condition setting, where there is no layer (see [1, 4, 6]), or in the classical sponge layer [7] or PML setting [3], where the equation is modified in the layer.

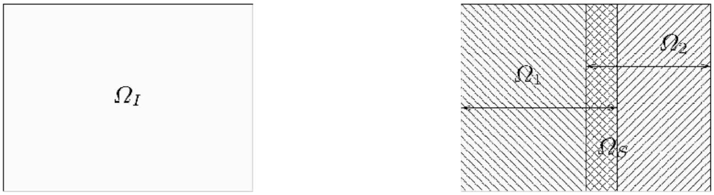


Domain of interest Ω_I Domain of computation $\Omega_C = \Omega_I \cup \Omega_S$

Fig. 1. sponge boundary

1.2 Domain decomposition with overlap

Suppose now that the domain of interest Ω_I be too large to be treated by a single computer (like for instance in combustion problems, climate modeling, etc.). Then one can divide the domain into several parts, which overlap or not. In each domain the original problem is solved, whereas one has to supplement with transmission conditions between the subdomains. A model geometry is described in Figure 2.



Domain of interest Ω_I Decomposed Domain

Fig. 2. Domain decomposition with overlap

In this case, given the size of the overlap, the transmission conditions are designed so as to minimize the convergence rate of the Schwarz algorithm. As we shall see in the next two sections, the two procedures previously described lead to the same optimization problem. For the wave equation, an explicit answer was given in [5] for low degrees. We present here the case of the unsteady reaction convection diffusion equation in $\mathbb{R}^{n+1}$

$$\begin{gathered}\mathcal{L}(u) := u_t - \nu\Delta u + a\partial_x u + \boldsymbol{b}\cdot\boldsymbol{\nabla} u + cu = F \text{ in } \mathbb{R}^{n+1}\times(0,T),\\ u(\cdot,0) = u_0 \text{ in } \mathbb{R}^{n+1},\end{gathered} \tag{1}$$

where the coefficients satisfy $\nu > 0,\ a > 0, \boldsymbol{b} \in \mathbb{R}^n, c > 0$. The operator $\boldsymbol{\nabla}$ operates only in the $\boldsymbol{y}$ direction in $\mathbb{R}^n$. The simpler problem of designing absorbing boundary conditions, without a sponge, has been addressed in [6], introducing an expansion in continued fractions.

We first describe the methods in Sections 2 and 3, and we set the best approximation problem. In Section 4 we study this best approximation problem, which is defined in the complex plane, and involves a nonlinear functional. Therefore it is more involved than the standard one. In Section 5 we show numerical evidences for the optimality of the method.

2 Sponge boundaries for the convection-diffusion equation: the half-space case

A model problem is the following: the original domain is $\mathbb{R}^{n+1}$, the domain of interest is $\Omega_I = (-\infty, X) \times \mathbb{R}^n$ and the domain of computation is $\Omega_C = (-\infty, X+L) \times \mathbb{R}^n$. A key point is that the data are compactly supported in Ω_C.

2.1 The transparent boundary condition

As it is now classical, the transparent boundary condition on the boundary $\partial\Omega_C$ is obtained through a Fourier transform in time and in the transverse direction $\boldsymbol{y}$. Transforming the equation leads to

$$-\nu\partial_{xx}\hat{u} + a\partial_x\hat{u} + (i(\omega + \boldsymbol{b}\cdot\boldsymbol{k}) + \nu|k|^2 + c)\hat{u} = 0$$

where $\hat{u}(x, \boldsymbol{k}, \omega)$ is the Fourier transform in the variables $\boldsymbol{y}$ and t. The characteristic equation is

$$-\nu\lambda^2 + a\lambda + i(\omega + \boldsymbol{b}\cdot\boldsymbol{k}) + \nu|k|^2 + c = 0 \tag{2}$$

It has two roots, such that $Re\,\lambda^+ \geq a$, $Re\,\lambda^- \leq 0$. The solution in the exterior of Ω_C can be written as

$$\hat{u}(x) = \hat{u}(X+L)e^{\lambda^-(x-(X+L))}$$

and the transparent boundary condition is given by

$$\partial_x\hat{u}(X+L, \boldsymbol{k}, \omega) = \lambda^-\hat{u}(X+L, \boldsymbol{k}, \omega)$$

We call Λ^- the pseudo-differential operator in the variables $\boldsymbol{y}$ and t whose symbol is λ^-, and the original problem in $\mathbb{R}^{n+1}$ is equivalent to

$$\begin{aligned} &\mathcal{L}(u) = F \text{ in } \Omega_C \times (0,T), \\ &u(\cdot,0) = u_0 \text{ in } \Omega_C, \\ &\partial_x u(X+L, \boldsymbol{y}, t) = \Lambda^- u(X+L, \boldsymbol{y}, t) \end{aligned} \tag{3}$$

2.2 Sponge boundaries: reflection coefficient

Let now v be a solution of problem with an approximate boundary condition

$$\partial_x v(X+L, \boldsymbol{y}, t) = \Lambda_a^- v(X+L, \boldsymbol{y}, t),$$

where Λ_a^- is an operator in the variables $\boldsymbol{y}$ and t, whose symbol λ_a^- will have to be a rational fraction in $\boldsymbol{k}$ and ω. We introduce the reflection coefficient

$$R(\omega, \boldsymbol{k}, \lambda_a^-, L) = \frac{\lambda^- - \lambda_a^-}{\lambda^+ - \lambda_a^-} \; e^{(\lambda^- - \lambda^+)\, L}$$

An easy calculation shows that the error between u and v is given by

$$\|u - v\|^2_{L^2(\Omega_I)} = \int \frac{|R(\omega, \boldsymbol{k}, \lambda_a^-, L)|^2}{2 Re\, \lambda^+} |\hat{u}(X, \omega, \boldsymbol{k})|^2 d\omega\, dk$$

In [6], it was proposed in the case $c = 0$ to approximate λ^- by continued fractions, for $L = 0$, which produces a small error for small viscosity. For larger viscosities, another approach can be used, namely to search for λ_a^- in a class of rational fractions, which minimize the reflection coefficient. This will be done at the end of Section 3.

3 Overlapping Optimized Schwarz Waveform Relaxation methods for the convection-diffusion equation

The model problem is the same as in Section 2. All the results in the next three sections can be found in [2]. The general Schwarz Waveform Relaxation algorithm for two domains $\Omega_1 = (-\infty, L) \times \mathbb{R}^n$, $\Omega_2 = (0, \infty) \times \mathbb{R}^n$ writes:

$$\begin{cases} \mathcal{L}(u_1^{k+1}) = f & \text{in } \Omega_1 \times (0,T) \\ u_1^{k+1}(\cdot, 0) = u_0 & \text{in } \Omega_1 \\ \mathcal{B}_1 u_1^{k+1}(L, \cdot) = \mathcal{B}_1 u_2^k(L, \cdot) & \text{in } (0,T) \end{cases}$$
$$\begin{cases} \mathcal{L}(u_2^{k+1}) = f & \text{in } \Omega_2 \times (0,T) \\ u_2^{k+1}(\cdot, 0) = u_0 & \text{in } \Omega_2 \\ \mathcal{B}_2 u_2^{k+1}(0, \cdot) = \mathcal{B}_2 u_1^k(0, \cdot) & \text{in } (0,T) \end{cases}$$

A natural variant of the Schwarz algorithm would be to use $\mathcal{B}_1$ and $\mathcal{B}_2$ equal to identity. It can be proved to be convergent with overlap, with a convergence rate depending of the size of the overlap.

3.1 The optimal Schwarz algorithm

Theorem 1. *The Schwarz method converges in two iterations with or without overlap when the operators $\mathcal{B}_i$ are given by:*

$$\mathcal{B}_1 = \partial_x - \Lambda^-, \quad \mathcal{B}_2 = \partial_x - \Lambda^+,$$

where $\Lambda^\pm$ are the operators whose symbols are the roots of (2).

3.2 Approximations by polynomials

As in the case of absorbing boundary conditions, we choose approximate operators:

$$\mathcal{B}_1^a = \partial_x - \Lambda_a^-, \quad \mathcal{B}_2^a = \partial_x - \Lambda_a^+$$

Since Λ^- and Λ^+ are related by $\Lambda^- + \Lambda^+ = \frac{a}{\nu}$, we choose the approximations to be such that $\Lambda_a^- + \Lambda_a^+ = \frac{a}{\nu}$. We define the error in step k in domain Ω_j to be e_j^k. With the same notations as in the previous section, and by analogous computations, we find the recursive relation

$$\widehat{e_j^{k+2}}(\omega, 0, \boldsymbol{k}) = \rho(\omega, k, \lambda_a^-, L)\widehat{e_j^k}(\omega, 0, \boldsymbol{k})$$

where the *convergence rate* $\rho(\omega, \boldsymbol{k}, \lambda_a^-, L)$ is given by

$$\rho(\omega, \boldsymbol{k}, \lambda_a^-, L) = R^2(\omega, \boldsymbol{k}, \lambda_a^-, L/2).$$

It measures the speed of convergence of the algorithm. The smaller it is, the faster the algorithm is. We rewrite it slightly differently. Let

$$\delta(\omega, \boldsymbol{k}) = a^2 + 4\nu(i(\omega + \boldsymbol{b} \cdot \boldsymbol{k}) + \nu|\boldsymbol{k}|^2 + c). \tag{4}$$

We can write

$$\lambda^- = \frac{a - \delta^{1/2}}{2\nu},$$

and $\delta^{1/2}(\omega, \boldsymbol{k}) = f(i(\omega + \boldsymbol{b} \cdot \boldsymbol{k}) + \nu|\boldsymbol{k}|^2)$ is approximated by a polynomial P in the variable $i(\omega + \boldsymbol{b} \cdot \boldsymbol{k}) + \nu|\boldsymbol{k}|^2$, and

$$\lambda_a^- = \frac{a - P}{2\nu}.$$

Therefore the convergence rate takes the simple form

$$\rho(\omega, \boldsymbol{k}, \lambda_a^-, L) = \left(\frac{P - \delta^{1/2}}{P + \delta^{1/2}}\right)^2 e^{-\delta^{1/2}L/\nu}. \tag{5}$$

In any case, in order to produce a convergent algorithm, we must have, $|\rho| \leq 1$ *a.e.* and $|\rho| < 1$ on any compact set in $\mathbb{R} \times \mathbb{R}^n$. We notice that for a general polynomial P we can have

$$\lim_{(\omega, |\boldsymbol{k}|) \to +\infty} \left| \frac{P - \delta^{1/2}}{P + \delta^{1/2}} \right| = 1.$$

3.3 Approximate transmission conditions

We consider here approximations of order ≤ 1. If $P = p + qz \in \mathbb{P}_1$, then

$$\begin{aligned} \mathcal{B}_1 &\equiv \partial_x - \frac{a - p}{2\nu} + q(\partial_t + \boldsymbol{b} \cdot \boldsymbol{\nabla} - \nu\Delta_S + cI), \\ \mathcal{B}_2 &\equiv \partial_x - \frac{a + p}{2\nu} - q(\partial_t + \boldsymbol{b} \cdot \boldsymbol{\nabla} - \nu\Delta_S + cI). \end{aligned}$$

Theorem 2. *For $p > 0, q \geq 0, p > \dfrac{a^2}{4\nu}q$, the algorithm is well-posed and converges with and without overlap.*

The case $q = 0$ corresponds to a polynomial of degree zero. This theorem is actually a composite of several results: first the algorithm is well-defined in relevant anisotropic Sobolev spaces: the result relies on trace theorems and energy estimates. Second the algorithms are convergent: in the nonoverlapping case, it relies again on energy estimates in each domain, arranged in such clever way as to cancel out the terms on the boundary when summing up the estimates. In the overlapping case, the convergence rate is uniformly strictly bounded away from one. The one-dimensional results can be found in [2], the two-dimensional case without second order derivatives is treated in V. Martin's thesis and published in [8]. Her result extends to the case we present here without any particular effort.

4 The best approximation problems

The convergence rate has two factors: the overlap intervenes in the term $e^{-2\delta^{1/2}L}$. Thus, in presence of an overlap, high frequency are taken care of by the overlap. In any case, when numerical schemes are involved, only discrete frequencies are present, and they are bounded from below and above. Let Y_j be the maximum size of the domain in the y_j direction. If δt and $\{\delta y_1, \cdots, \delta y_n\}$ are the discrete steps in time and space, the frequencies can be only such that $\omega \in I_T, k_j \in I_j$, with $I_T = (\frac{\pi}{T}, \frac{\pi}{\delta t})$, and $I_j = (\frac{\pi}{Y_j}, \frac{\pi}{\delta y_j})$. The best approximation problem consists in, for a given n, finding P in P_n minimizing $\sup_{\omega \in I_T, k_j \in I_j} |\rho(\omega, k, \lambda_a^-, L)|$.

Using the forms in (4) and (5), we can rewrite it, for a given n, as finding P in P_n minimizing

$$\sup_{z \in K} \left| \frac{P(z) - f(z)}{P(z) + f(z)} e^{-Lf(z)/\nu} \right| \tag{6}$$

where K is a compact set in $\mathbb{C}+$, $K = \{i(\omega + \boldsymbol{b} \cdot \boldsymbol{k}) + \nu |\boldsymbol{k}|^2, \omega \in I_T, k_j \in I_j, 1 \leq j \leq n\}$.

4.1 A general result

Let K be a compact set in $\mathbb{C}$, f a continuous function on K, such that $f(K) \subset \{z \in \mathbb{C} : Re\, z > 0\}$. Define

$$\delta_n(l) = \inf_{p \in \mathbb{P}_n} \sup_{z \in K} \left| \frac{p(z) - f(z)}{p(z) + f(z)} e^{-lf(z)} \right|,$$

Problem (6) generalizes as:

$$\text{Find } p_n^* \text{ such that } \sup_{z \in K} \left| \frac{p_n^*(z) - f(z)}{p_n^*(z) + f(z)} e^{-lf(z)} \right| = \delta_n(l)$$

This is a non classical complex best approximation problem, for two reasons: first the cost function $\dfrac{p(z) - f(z)}{p(z) + f(z)}$ is non linear, second there is a weight $e^{-lf(z)}$ which decreases rapidly, and allows for large values of $\dfrac{p(z) - f(z)}{p(z) + f(z)}$. We have a fairly complete theory in the non overlapping case: existence, uniqueness, and an equioscillation property. Furthermore any local minimum is global. In the overlapping case, general results are more restrictive: for l sufficiently small, there is a solution, any solution equioscillates, and if $\delta_n(l) e^{l \sup_{z \in K} \Re f(z)} < 1$, then the solution is unique. In the symmetric case, i.e., if K is symmetric with respect to the real axis, and if for any z in K, $f(\bar{z}) = \overline{f(z)}$, then the polynomial of best approximation has real coefficients. Furthermore for odd n the number of equioscillations is larger than or equal to $n + 3$.

4.2 The 1-D case

In this case, the convergence rate actually equioscillates in 3 real points, and we can have explicit formulae to determine the best polynomial p_1^*. Furthermore the constraints on the coefficients for well-posedness are fulfilled. In 2-D, it is still an open question.When solving by a numerical scheme, the overlap is such that $L \approx C_1 \Delta x$ and the space and time meshes are related by $\Delta t \approx C_2 \Delta x^\beta$, $\beta \geq 1$ (in general β can be 1 or 2). With overlap, for $\beta = 1$, $sup|\rho| \approx 1 - \mathcal{O}(\Delta x^{1/8})$, while for $\beta = 2$, $sup|\rho| \approx 1 - \mathcal{O}(\Delta x^{1/5})$. Without overlap, in both cases, $sup|\rho| \approx 1 - \mathcal{O}(\Delta t^{1/8})$. Thus, if $\Delta t \approx \Delta x$, the performances with or without overlap are comparable, if $\Delta t \approx \Delta x^2$, the performance is better with overlap.

5 Numerical results for domain decomposition

In order to check the relevance of the theoretical best approximation, we run the case $\nu = 0.2, a = 1, c = 0$, $\Omega = (0, 6)$, $T = 2.5$. The initial data is $u(x, 0) = e^{-3(1.2-x)^2}$. The boudary conditions are $u(0, t) = 0$ and $u(6, t) = 0$. We choose $\Omega_1 = (0, 3.04)$, $\Omega_2 = (2.96, 6)$, which means $L = 0.08$. The scheme is upwind in space, backward Euler in time, with $\Delta x = 0.02, \Delta t = 0.005$. The initial guess is random. Figure 3 shows that the theoretical best value of p and q, coefficients of P (represented by the star), is very close to the one observed numerically.

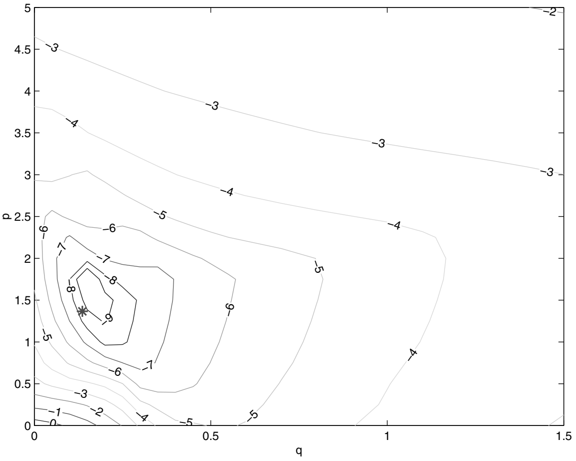


Fig. 3. Error after 5 iterations as a function of p and q.

6 Conclusion

We have proposed a complete theory based on a best approximation problem arising in sponge layers or SWR algorithms for parabolic equations. In one dimension it can be solved explicitely, thus providing the best answers to our questions. It remains to extend it in three directions: to rational fractions, to higher order, and to higher dimensions.

References

1. A. Bayliss and E. Turkel, *Radiation boundary conditions for wave-like equations*, Comm. Pure Appl. Math., 33 (1980), pp. 707–725.
2. D. Bennequin, M. J. Gander, and L. Halpern, *Optimized Schwarz waveform relaxation methods for convection reaction diffusion problems*, Tech. Rep. 24, Institut Galilée, Paris XIII, 2004.
3. J.-P. Berenger, *Three-dimensional perfectly matched layer for the absorption of electromagnetic waves*, J. Comput. Phys, 127 (1996), pp. 363–379.
4. B. Engquist and A. Majda, *Radiation boundary conditions for acoustic and elastic calculations*, Comm. Pure Appl. Math., 32 (1979), pp. 313–357.
5. M. J. Gander and L. Halpern, *Absorbing boundary conditions for the wave equation and parallel computing*, Math. Comp., 74 (2005), pp. 153–176.
6. L. Halpern, *Artificial boundary conditions for the linear advection-diffusion equation*, Math. Comp., 46 (1986), pp. 425–438.
7. M. Israeli and S. A. Orszag, *Approximation of radiation boundary conditions*, J. Comput. Phys., 41 (1981), pp. 115–135.
8. V. Martin, *An optimized Schwarz Waveform Relaxation method for the unsteady convection diffusion equation in two dimensions*, Appl. Numer. Math., 52 (2005), pp. 401–428.

MINISYMPOSIUM 6: Schwarz Preconditioners and Accelerators

Organizers: Marcus Sarkis[1] and Daniel Szyld[2]

[1] Instituto de Matemática Pura a Aplicada msarkis@fluid.impa.br
[2] Temple University szyld@math.temple.edu

Many recently proposed domain decomposition preconditioners do not easily fit within the classical convergence framework. Presentations in this mini-symposium will focus on some recent results on these preconditioners. Some of the topics to be covered include: Algebraic theory, nonlinear preconditioners, restricted Schwarz methods, alternative coarse spaces, hybrid preconditioners, and accelerators.

Numerical Implementation of Overlapping Balancing Domain Decomposition Methods on Unstructured Meshes

Jung-Han Kimn[1] and Blaise Bourdin[2]

[1] Department of Mathematics and the Center for Computation and Technology, Louisiana State University, Baton Rouge, LA 70803, USA. kimn@math.lsu.edu

[2] Department of Mathematics, Louisiana State University, Baton Rouge, LA 70803, USA. bourdin@math.lsu.edu

Summary. The Overlapping Balancing Domain Decomposition (OBDD) methods can be considered as an extension of the Balancing Domain Decomposition (BDD) methods to the case of overlapping subdomains. This new approach, has been proposed and studied in [5, 3]. In this paper, we will discuss its practical parallel implementation and present numerical experiments on large unstructured meshes.

1 Introduction

The Overlapping Balancing Domain Decomposition Methods (OBDD) is a two level overlapping Schwarz method. Its coarse space as well as the projection and restriction operators are based on partition of unity functions. This new algorithm has been presented in [5, 3]. More recently, it has also been extended to the Helmholtz problem (see [4, 3]).

The main goal of this paper is to present an efficient and scalable implementation on large unstructured meshes. The proposed algorithm does not require the construction of a coarse mesh and avoids expensive communication between coarse and fine levels. The implementation we present works on an arbitrary number of processors and does not requires an *a priori* manual decomposition of the domain into subdomains. It relies heavily on the construction of overlapping subdomains and associated partition of unity functions. These functions are used both as a communication mechanism between coarse and fine levels, and as the generating functions for the coarse space. More details on two level overlapping Schwarz methods with partition of unity–based coarse space can be found in [7, 8, 9].

1.1 Notations and presentation of the method

All along this paper, we focus on the implementation of the Poisson problem with Dirichlet boundary condition on a polygonal domain Ω. Given a function $f \in H^{-1}(\Omega)$, and $\partial\Omega_D \subset \partial\Omega$ with a finite number of connected components, we want to solve the problem

$$-\Delta u = f \text{ in } \Omega, u = u_0 \text{ on } \partial\Omega_D. \tag{1}$$

Let $\mathcal{T}$ be a conforming mesh partitioning of Ω with N_e elements and N_v vertices, partitioned into N parts $\mathcal{T}_i, 1 \leq i \leq N$ with N_e^i elements and N_v^i vertices. For any positive integer k , the overlapping mesh $\mathcal{T}_i^k$ is a sub-mesh of $\mathcal{T}$ whose vertices are either in $\mathcal{T}_i$ or linked to a vertex of $\mathcal{T}_i$ by at most k edges. We denote by Ω_i and Ω_i^k the domains associated with these meshes. Lastly, let A be the matrix associated to a discretization of (1). In our experiment, we have used a finite element method with linear elements, but this is not a requirement of the method.

The construction of the Overlapping Balancing Domain Decomposition method is similar to that of the well-known Balancing Domain Decomposition method. Its main ingredient is the construction of a partition of unity θ_i, $1 \leq i \leq N$, such that $\theta_i > 0$ on $\mathcal{T}_i^k$, and $\theta_i = 0$ on $\mathcal{T} \setminus \mathcal{T}_i^k$. Using the functions θ_i, we define N diagonal weight matrices D_i of size $N_v \times N_v$ whose diagonal elements are the θ_i.

In this method, the dimension of the coarse space is equal to the number of Ω_i^k, and the associated matrix A_c is given by

$$A_c(i,j) = \theta_i^T A \theta_j, \quad 1 \leq i,j \leq N. \tag{2}$$

On each subdomain, the local problems involve solving a local version of (1) with homogeneous Neumann interface conditions. Of course, this is a singular problem, however one can show that the partition of unity functions θ_i generate the null space of the associated local matrix A_i, from which one can easily derive compatibility conditions.

For more details on the theoretical aspects of the method, and a precise description, see [4] and [3].

2 Implementation of the OBDD method on Unstructured Meshes

The Overlapping Balancing Domain Decomposition Method has been implemented using an existing parallel finite element package previously written by the second author. The implementation, we describe in the sequel, is general enough that it should be fairly easy to reproduce in any other finite element code. However, some of the technical choices detailed later are dependent on the software packages we used. Namely the unstructured two and three dimensional meshed were generated using Cubit, developed at Sandia National Laboratories [6], and the internal mesh representation is based on the EXODUS II libraries, also from Sandia National Laboratories. The automatic domain decomposition was obtained using METIS and ParMETIS [2]. Lastly, we used PETSc [1] for all distributed linear algebra needs, and most communication operations.

The OBDD itself has also been implemented as a shell preconditioner in PETSc

2.1 Construction of the overlapping subdomains and the partition of unity functions

The first step toward the implementation of the OBDD method is to construct the overlapping subdomains and the partition of unity functions, using a non-overlapping domain decomposition computed with METIS. The following algorithm does that in a fully distributed and scalable way.

Let $\mathcal{T}$ be a part of the mesh of Ω. We say that a vertex (resp. an element) of Ω is *local* to $\mathcal{T}$ if it belongs to $\mathcal{T}$. We say that an element of Ω is a *near* element for $\mathcal{T}$ if one of its vertices is local to $\mathcal{T}$. Similarly, we say that a vertex $v \in \Omega$ is a near vertex for $\mathcal{T}$ if it belongs to a near element for $\mathcal{T}$, but is not local to $\mathcal{T}$. Lastly, any vertex or element that is neither local nor near is referred to as *distant*. With these notations, we note that Ω_i^k is simply the union of all local and near vertices and the elements of Ω_i^{k-1}. This is the essence of our iterative construction.

In the mesh representation system we used, we did not have access to the adjacency graph of the vertices, or a list of element neighbors. Our algorithm requires only each processor to store the entire connectivity table of the mesh.

In order to construct the partition of unity functions and the overlapping subdomains simultaneously, each processor uses a temporary counter d_i of size equal to the total number of vertices. At the initial stage, one sets $d_i(v) = 1$ if v is local to Ω, and 0 otherwise. Then, one repeats the following process for $0 \leq j \leq k$: for $1 \leq l \leq N_e$, the element l is near Ω_i^j if $d_i(v) > 0$ at any of its vertices v. Using the connectivity table, compute then the list of all near vertices to Ω. Lastly increment $d_i(v)$ for all v local or near to Ω_i^j. After the k iteration, $d_i(v) = k + 1$ if $v \in \Omega_i$, $d_i(v) = 0$ if $v \notin \Omega_i^k$. At this point, all that remains to do is to set $\theta_i(v) = d_i(v)/\sum_{j=1}^{N} d_i(v)$.

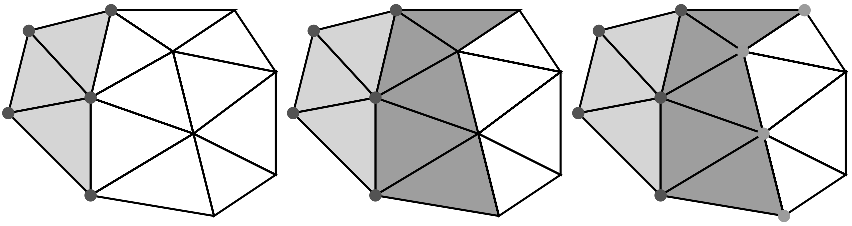


Fig. 1. Extension of the overlap in three steps.

Figure 1 illustrates the three step construction of Ω_i^{k+1} out of Ω_i^k. The leftmost figure highlights the local vertices and elements for Ω_i^k. In the middle figure, the near elements for Ω_i^k have been identified. From these near elements, it is now easy to identify the near vertices, as illustrated on the right. All local and near elements for Ω_i^k are the local elements for Ω_i^{k+i}, so that process can be iterated as many times as necessary.

Note that this algorithm is very similar in spirit to a fast marching method (see for instance [10]). Indeed, the functions d_k are the distance to the non-overlapping domains, in a metric where $d(v_i, v_j)$ is proportional to the smallest number of edges linking two vertex v_i and v_j.

Note also that the complexity of this algorithm is independent on the number of processor, and that it requires communication only at its very final stage. The complexity of this algorithm is on the order of $\mathcal{O}(kN_e)$ and grows linearly with the size of the overlap. As demonstrated in the sequel, a typical overlap choice is 3 to 5, so the construction of the θ_i is very efficient. However, should one have access to the list of edges of the meshes, or the list of neighboring element to a given one, this complexity would be greatly reduced.

2.2 Coarse problem

The coarse matrix is given by $A_c(i,j) = \theta_i^T A \theta_j$. However, its construction does not require the actual computation of these matrix-vector products. Also, it is easy to see that A_c has a sparsity structure, as $\mathrm{supp}(\theta_i) \cap \mathrm{supp}(\theta_j) \neq \emptyset$ only if $\Omega_i^k \cap \Omega_j^k \neq \emptyset$.

In our implementation, we first find all subdomains with non-empty intersection, which give the sparsity structure of A_c. Then, for each processor, $A\theta_j$ is obtained from computing $A_j\theta_j$. Then we communicate this vector to all neighboring subdomains so that each processor can assemble its own row in A_c. This algorithm is fully scalable since it involves only communications between neighboring processors, and no "all to all" message passing. As illustrated in the experiments in the next section, the OBDD perform best with relatively small overlap. In this case, it is enough to build the adjacency graph of the non-overlapping domains, which is slightly faster. However, this is not true with very large overlap.

Lastly, since the dimension of the coarse problem is relatively small (recall that it is equal to the number of processors), we store it in one of the processors, and coarse solve can be performed using a direct solver.

2.3 Local problems

Our implementation uses PETSc which does not have data structures dedicated to overlapping submatrices. Therefore, we chose to reassemble the local matrices A_j instead of extracting them from the global matrix A. Note that this has to be done only once, so it is not very expensive.

As we expect our algorithm to be very scalable, our goal is to use a large number of processors, which means relatively small local problems. For that reason, we use direct local solvers. The cost of the initial factorization is offset by the speed gain in solving the local problems multiple times.

Since we consider local problems with homogeneous Neumann interface conditions, the local matrices are singular. However, their null spaces are given by their associated partition of unity function (see [5, 3] for more details). In the implementation, we still have to add a small damping factor to the diagonal of the matrix, or the local factorization would sometimes fail. This damping factor is typically of order 10^{-10}.

3 Numerical Results

In the numerical experiment presented here, Ω is the square $[-5,5] \times [-5,5]$. We consider a homogeneous Dirichlet problem for two different right hand sides: $f(x,y) \equiv 1$

(Problem 1), or $f(x, y) = 1$ if $xy > 0$ and $f(x, y) = -1$ otherwise (Problem 2). The experiments are based on solving both problems for various overlap size k and various mesh sizes. The larger mesh has approximately 1,000,000 vertices and 2,000,000 elements (*i.e.* $h \sim .01$). The second one is made of 450,000 vertices and 890,000 elements ($h \sim .015$), and the last one of 250,000 vertices and 500,000 elements ($h \sim .02$). We ran our test implementation on many other problems, and got very similar behaviors.

Table 1 display the evolution of the number of iterations of OBDD of Problem 1 and Problem 2. Along the horizontal lines, the ratio between the geometric size of the overlap and the size of the subdomains remains constant while the mesh size varies. As expected, the number of iterations does not change significantly. Along vertical lines, the number of processor is increased. As expected, the number of iterations decreases as long as the number of nodes in the overlap region remains small compared to that in the actual subdomain.

Table 1. Number of iterations for Problem 1 (Problem 2).

♯ of CPUs	Mesh1 (ovlp=10)	Mesh2 (ovlp=7)	Mesh3 (ovlp=5)
32	55 (51)	51 (51)	52 (51)
64	50 (49)	47 (44)	46 (44)
80	45 (43)	47 (46)	50 (48)

Figure 2 represent the evolution of the computation time and of the number of iterations as a function of the overlap. The simulation is carried out for the smaller of the three meshes, and on 32 processors. The first curve corresponds to the total time spent in the solver. In the second one, we subtracted the factorization time for the local matrices. As expected, the total time increases slightly with very large overlaps. However, most of this time increase is due to the local matrices factorization. Indeed, as the overlap size increases, the number of iteration decreases steadily. For all practical purposes, we found no reason to use large overlap. Overlap sizes of 3-5 typically give the fastest convergence.

In Figure 3, we perform this experiment for various number of processors, on our largest mesh. The same conclusion holds in each case. In our implementation, we assume that the overlapping subdomains associated to disjoint subdomains were also disjoint, which is not necessarily true with very large overlaps and very small subdomains. For that reason, we are not able to use large overlap sizes with 64 processors.

In Figure 4, we demonstrate the scalability of the Overlapping Balancing Domain Decomposition method, and of our implementation. We solved again Problem 2 with various overlap sizes and up to 240 processors. As expected, both the total time and the number of iterations decrease with the number of processors. Note also that the gain from an increase of the overlap size is quite minimal.

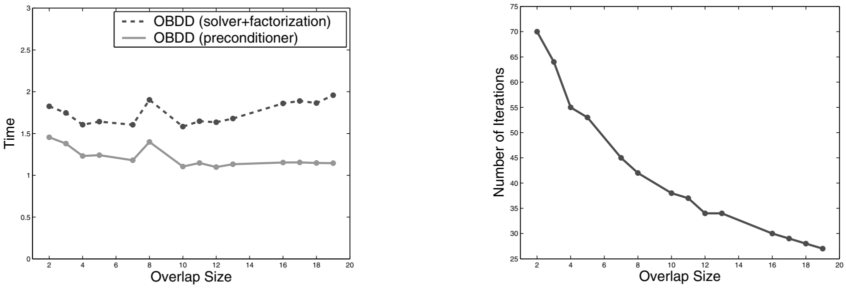


Fig. 2. Times and numbers of iteration versus the overlap size.

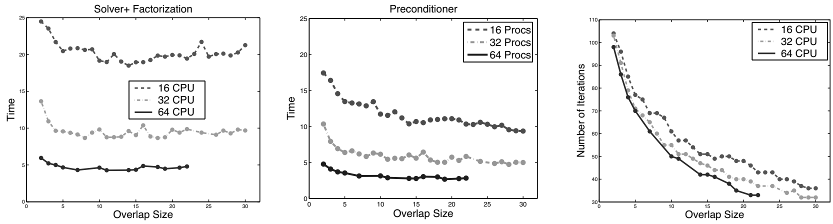


Fig. 3. Time and numbers of iteration versus the overlap size.

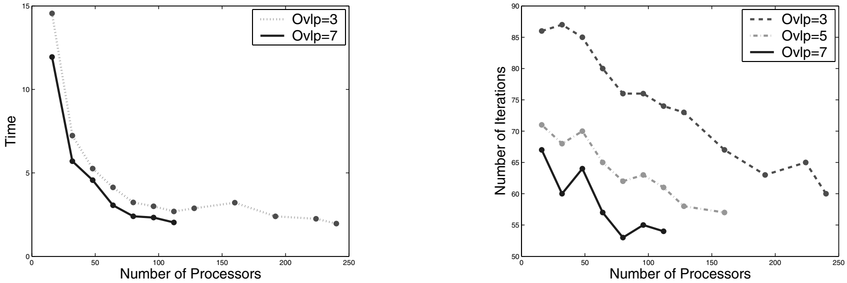


Fig. 4. Scalability of the algorithm and its implementation.

The last figure is perhaps the most important. Here, we compare our method with a widely available solver. For our problem, we found that the best combination of solvers and preconditioners in PETSc is the Conjugated Gradient with a block-Jacobi preconditioner, iterative local solvers, and incomplete LU local preconditioners. In Figure 5, we compare the performances of the OBDD and block-Jacobi preconditioners. Our algorithm performs significantly better than the best available solver in PETSc in all cases.

References

1. S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, L. C. McInnes, and B. F. Smith, *PETSc home page.* http://www.mcs.anl.gov/petsc, 2001.

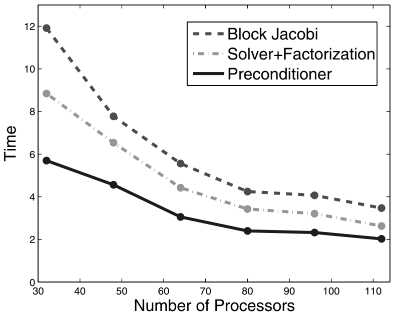


Fig. 5. Comparison of the OBDD and the block-Jacobi preconditioners.

2. G. Karypis, R. Aggarwal, K. Schoegel, V. Kumar, and S. Shekhar, *METIS home page*. http://glaros.dtc.umn.edu/gkhome/views/metis.
3. J.-H. Kimn and M. Sarkis, *OBDD: Overlapping balancing domain decomposition methods and generalizations to the Helmholtz equation*, in Proceedings of the 16th International Conference on Domain Decomposition Methods, O. B. Widlund and D. E. Keyes, eds., Springer, 2006. These proceedings.
4. J.-H. Kimn and M. Sarkis, *Restricted overlapping balancing domain decomposition methods and restricted coarse problem for the Helmholtz equation*, Comput. Methods Appl. Mech. Engrg., (2006). Submitted.
5. ———, *Theoretical analysis theory of overlapping balancing domain decomposition methods for elliptic problems*. In preparation, 2006.
6. S. J. Owen, *The CUBIT tool suite home page*. http://cubit.sandia.gov.
7. M. Sarkis, *Partition of unity coarse spaces*, in Fluid flow and transport in porous media: mathematical and numerical treatment, vol. 295 of Contemp. Math., AMS, Providence, RI, 2002, pp. 445–456.
8. ———, *A coarse space for elasticity: Partition of unity rigid body motions coarse space*, in Proceedings of the Applied Mathematics and Scientific Computing Dubrovnik, Croacia, June, 2001, Z. D. et. al., ed., Kluwer Academic Press, 2003, pp. 3–31.
9. ———, *Partition of unity coarse spaces: Enhanced versions, discontinuous coefficients and applications to elasticity*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003.
10. J. A. Sethian, *A fast marching level set method for monotonically advancing fronts*, Proc. Natl. Acad. Sci. U.S.A., 93 (1996), pp. 1591–1595.

OBDD: Overlapping Balancing Domain Decomposition Methods and Generalizations to the Helmholtz Equation

Jung-Han Kimn[1] and Marcus Sarkis[2]

[1] Department of Mathematics and the Center for Computation and Technology, Louisiana State University, Baton Rouge, LA, 70803, USA. kimn@math.lsu.edu

[2] Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, Brazil, and Worcester Polytechnic Institute, Worcester, MA 01609, USA. msarkis@fluid.impa.br. Research supported in part by CNPQ (Brazil) under grant 305539/2003-8 and by the U.S. NSF under grant CGR 9984404.

1 Introduction

Balancing Domain Decomposition (BDD) methods belong to the family of preconditioners based on nonoverlapping decomposition of subregions and they have been tested successfully on several challenging large scale applications. Here we extend the BDD algorithms to the case of overlapping subregions and we name them Overlapping Balancing Domain Decomposition (OBDD) algorithms. Like the BDD methods, coarse space and weighting matrices play crucial roles in making both the proposed algorithms scalable with respect to the number of subdomains as well as making balanced the local Neumann subproblems on the overlapping subregions on each iteration of the preconditioned system. The OBDD algorithms also differ from the standard overlapping additive Schwarz method (ASM) of hybrid form since those are based on Dirichlet local problems on the overlapping subregions. This difference motivated us to generalize the OBDD algorithms to the Helmholtz equation where we use the Sommerfeld boundary condition for the local problems and a combination of partition of unity and plane waves for the coarse problem.

1.1 Balancing Domain Decomposition Methods

To have a clear picture of the OBDD algorithms, we first provide a short review of two-level Balancing Domain Decomposition (BDD) methods introduced in [6, 10]. BDD methods are iterative substructuring algorithms, i.e. methods where the interior degrees of freedom of each of the nonoverlapping substructures are eliminated. Hence the discrete problem

$$Ax = f \tag{1}$$

obtained from a finite element discretization method applied to the domain Ω is reduced and posed on the interface $\Gamma = \cup_{i=1}^{N} \Gamma_i$. Here $\Gamma_i = \partial\Omega_i \backslash \partial\Omega$ are the local interfaces. The linear system is then reduced to the form

$$Su = g,$$

where

$$S = \sum_{i=1}^{N} R_i^T S_i R_i,$$

where the matrices S_i are the local Schur complements and R_i are the regular restriction operators from nodal values on Γ to Γ_i. To simplify the exposition, we assume that the matrix A comes from a finite element discretization of the Poisson problem and therefore, the Schur complement matrices S_i are symmetric positive semi-definite (the kernel consists of constant functions) when $\partial\Omega_i \cap \partial\Omega_D = \emptyset$, or positive definite otherwise. Here, $\partial\Omega_D$ is the Dirichlet part of $\partial\Omega$. To build the BDD preconditioner, weighting matrices D_i on the interface are constructed so that

$$\sum R_i^T R_i D_i = I_\Gamma \tag{2}$$

forms a partition of unity on the interface Γ. The weighting matrices D_i, for the Poisson problem with constant coefficient, can be chosen as the diagonal matrix defined as zero at the nodes on $\Gamma \backslash \Gamma_i$ and the reciprocal of the number of subdomains a node $x \in \Gamma_i$ is associated with. The preconditioner is of the hybrid type given by

$$T_{BDD} = P_0 + (I - P_0)(\sum_{i=1}^{N} T_i)(I - P_0), \tag{3}$$

where the coarse problem P_0 is simply the orthogonal projection (in the S-norm) onto the coarse space V_0. The coarse space V_0 is defined as the span of the basis functions $D_i R_i^T n_i$ where each column vector n_i, except for subdomains for which $\partial\Omega\ \partial\Omega_i \cap \partial\Omega_D \neq \emptyset$, is a vector that generates the null space of S_i, i.e. the column vector $[1, 1, 1, 1, \ldots, 1]^T$ on nodes of Γ_i. Hence,

$$P_0 = R_0^T (R_0 S R_0^T)^{-1} R_0 S, \tag{4}$$

where the columns of the matrix R_0^T are formed by all the columns of $D_i R_i^T n_i$.

The local operators T_i are defined as

$$T_i = D_i R_i^T S_i^+ R_i D_i S \tag{5}$$

where S_i^+ is the pseudo inverse of the local Schur complement S_i. We remark that each local Neumann problem S_i^+ is solved up to a constant when the $\partial\Omega_i \cap \partial\Omega_D = \emptyset$. The compatibility condition is guaranteed because a coarse problem is solved just prior; if y belongs to the range of $(I - P_0)$, and using the definition of P_0 (an orthogonal projection in the S-norm), we have $(D_i R_i^T n_i, Sy)_\Gamma = 0$ (inner product on Γ), and therefore $(n_i, R_i D_i Sy)_{\Gamma_i} = 0$ (inner product on Γ_i). Hence, $R_i D_i Sy$ is perpendicular to the null space of S_i and the local problem $S_i x_i = D_i R_i Sy$ satisfies the compatibility condition, and we say that the local problems are balanced.

1.2 Overlapping Balancing Domain Decomposition Methods

We generalize the nonoverlapping BDD method to the overlapping domain case. This is done by maintaining the BDD structure described above. We replace the Schur complement matrix S by the whole matrix A. We replace the restriction operator R_i

to Γ_i by a restriction operator R_i^δ to all nodes of the extended subdomain $\overline{\Omega}_i^\delta \backslash \partial\Omega_D$ (including also the boundary nodes on $\partial\Omega_i^\delta \backslash \partial\Omega_D$). We replace the Neumann problem S_i^+ by a Neumann problem $(A_i^\delta)^+$ on Ω_i^δ with a Neumann boundary condition on $\partial\Omega_i^\delta \backslash \partial\Omega_D$ and zero Dirichlet boundary condition on $\partial\Omega_i^\delta \cap \partial\Omega_D$. We replace the partition of unity (2) by a partition of unity on $\overline{\Omega}\backslash\partial\Omega_D$

$$\sum_{i=1}^{N} (R_i^\delta)^T R_i^\delta D_i^\delta = I_{\overline{\Omega}\backslash\partial\Omega_D}, \tag{6}$$

where the weighting matrix D_i^δ is a diagonal matrix with diagonal elements given by the regular partition of unity we find on the theory of Schwarz methods. Similarly, the coarse space V_0^δ is also based on this partition of unity (with some modification near $\partial\Omega_D$ to satisfy Dirichlet boundary conditions). The coarse problem P_0^δ is the orthogonal projection (in the A-norm) onto the space V_0^δ and the OBDD preconditioner is defined as

$$T_{OBDD} = P_0^\delta + (I - P_0^\delta)(\sum_{i=1}^{N} T_i^\delta)(I - P_0^\delta), \tag{7}$$

where the local problems are given by

$$T_i^\delta = D_i^\delta (R_i^\delta)^T (A_i^\delta)^+ R_i^\delta D_i^\delta A. \tag{8}$$

The same arguments about BDD compatibilities hold here: if y belongs to the range of $(I - P_0^\delta)$ we have $(D_i^\delta (R_i^\delta)^T n_i^\delta, Ay)_{\Omega\backslash\partial\Omega_D} = 0$, and so $(n_i^\delta, R_i^\delta D_i^\delta Ay)_{\Omega_i^\delta\backslash\partial\Omega_D} = 0)$. Hence, $R_i^\delta D_i^\delta Ay$ is perpendicular to the vector n_i^δ (a column vector of ones on the nodes of Ω_i^δ when $\Omega_i^\delta \cap \partial\Omega_D = \emptyset$). The vector n_i^δ spans a space that contains the kernel of A_i^δ, and so the local Neumann problems $A_i^\delta x_i = D_i^\delta R_i^\delta Ay$ satisfy the local compatibility condition.

1.3 Advantages and Disadvantages of BDD versus OBDD

We note that differently from BDD methods, the OBDD methods work on the whole finite element function space without eliminating any variables. Hence we solve $Ax = b$ instead of $Su = g$. As a first consequence, we avoid completely the local Dirichlet solvers required for the BDD methods to compute residuals as well as to build the coarse matrix. This is a considerable advantage for the OBDD methods since these BDD local Dirichlet solvers require exact solvers in each iteration with the preconditioned system, and more dramatically, specially in three dimensional problems, a large number of preprocessing exact local Dirichlet solvers are required to build the coarse matrix. We note also that the coarse matrix of the proposed OBDD methods are of the same size as those of BDD methods, i.e. one degree of freedom per subdomain. However, the OBDD coarse matrices are more sparse than those of BDD since results in connectivity only among the neighboring subdomains. Another advantage of using OBDD methods is that they are less sensitive to the roughness of the boundary of the subdomains (in general boundaries of extended subdomains are smoother than nonoverlapping subdomains).

The proposed OBDD algorithms also have disadvantages. The first one is the extra cost when working with extended subdomains. Hence for effective performance in terms of CPU time and memory allocation, small overlap is a common practice. The second disadvantage is that the condition number obtained by the OBDD methods are $O(1 + H/(\delta h))$ while the BDD methods are $O(1 + \log(H/h)^2)$. Numerically we show that for the minimum overlap case, the preconditioned systems associated to OBDD results in small condition numbers, so the linear bound is comparable to the two log factors for the BDD. For three dimensional problems, the ratio H/h would be relatively small and therefore, the linear bound of the OBDD would get closer to the two logs bound of the BDD. The third disadvantage is that the inner products and the vector sums inside the PCG/BDD (GMRES/BDD) are done only for the interfaces nodes while on the PCG/OBDD (GMRES/OBDD) they are done for all the nodes. We note however that in the proposed algorithms, after the first iteration of the OBDD, only on the extended boundary interfaces will have nonzero residuals and will remain so during the PCG iterations when RASHO coarse problems [2, 9] are considered (since the RASHO coarse basis functions are designed to have zero residual at non interface nodes). Hence a large saving in perform $A * v$ to compute residuals is possible. The BDD methods nowadays are well developed for several applications such as discontinuous coefficients, two and three dimensional elasticity, plates and shells, and are recently also extended to saddle point problems. For two and three dimensional elasticity and for discontinuous coefficients problems, we can apply some of the ideas in [8, 9] to design and analyze OBDD algorithms. The extension of OBDD algorithms to saddle point problems is not trivial and it is a very interesting subject for future research.

2 The Finite Element Formulation

Consider the Helmholtz problem

$$\begin{aligned} -\Delta u^* - (k(x))^2 u^* &= f \quad \text{in} \quad \Omega \\ u^* &= g_D \quad \text{on} \quad \partial\Omega_D \\ \frac{\partial u^*}{\partial n} &= g_N \quad \text{on} \quad \partial\Omega_N \\ \frac{\partial u^*}{\partial n} + iku^* &= g_S \quad \text{on} \quad \partial\Omega_S \end{aligned} \tag{9}$$

where Ω is a bounded polygonal region in $\Re^2$ with a diameter of size $O(1)$. The $\partial\Omega_D$, $\partial\Omega_N$, and $\partial\Omega_S$ are disjoint parts of $\partial\Omega$ where the Dirichlet, Neumann, and Sommerfeld boundary conditions are imposed. We note that the methods developed here also works for polyhedral regions in $\Re^3$. From a Green's formula and conjugation of the test functions, we can reduce (9) into the following variational form: find $u^* - u_D^* \in H_D^1(\Omega)$ such that,

$$\begin{aligned} a(u^*, v) &= \int_\Omega (\nabla u^* \cdot \nabla \bar{v} - k^2 u^* \bar{v})\, dx - ik \int_{\partial\Omega_S} u^* \bar{v}\, ds \\ &= \int_\Omega f \bar{v}\, dx + \int_{\partial\Omega_N} g \bar{v}\, ds = F(v), \ \forall v \in H_D^1(\Omega), \end{aligned} \tag{10}$$

where u_D^* is an extension of g_D to $H^1(\Omega)$, and $H_D^1(\Omega)$ is the subspace of $H^1(\Omega)$ of functions which vanishes on $\partial\Omega_D$. To treat the Poisson's problem, we let $k = 0$ and $\partial\Omega_S = \emptyset$.

Let $\mathcal{T}^h(\Omega)$ be a shape regular quasi-uniform triangulation of Ω and let $V \subset H^1_D(\Omega)$ be the finite element space consisting of continuous piecewise linear functions, associated with the triangulation, which vanish on $\partial\Omega_D$. Eliminating u_D we obtain the following discrete problem: Find $u \in V$ such that

$$a(u,v) = f(v), \quad \forall\, v \in V. \tag{11}$$

Using the standard basis functions, (11) can be rewritten as a linear system of equations of the form (1).

All the domains and subdomains are assumed to be open; i.e., boundaries are not included in their definitions. The superscript T means the adjoint of an operator.

3 Notation

Given the domain Ω and triangulation $\mathcal{T}^h(\Omega)$, we assume that a domain partition has been applied and resulted in N non-overlapping connected subdomains $\Omega_i, i = 1, \ldots N$ of size $O(H)$, such that

$$\overline{\Omega} = \cup_{i=1}^N \overline{\Omega}_i \quad \text{and} \quad \Omega_i \cap \Omega_j = \emptyset, \quad \text{for} \quad j \neq i.$$

We define the overlapping subdomains Ω_i^δ as follows. Let Ω_i^1 be the one-overlap element extension of Ω_i, where $\Omega_i^1 \supset \Omega_i$ is obtained by including all immediate neighboring elements $\tau_h \in \mathcal{T}^h(\Omega)$ of Ω_i such that $\overline{\tau}_h \cap \overline{\Omega}_i \neq \emptyset$. Using the idea recursively, define a δ-extension overlapping subdomains Ω_i^δ

$$\Omega_i = \Omega_i^0 \subset \Omega_i^1 \subset \cdots \subset \Omega_i^\delta.$$

Here the integer $\delta \geq 1$ indicates the level of element extension and δh is the approximate width of the extension. We note that this extension can be coded easily using the adjacency matrix associated to the mesh.

4 Local Problems: Definitions of D_i^δ, R_i^δ and T_i^δ

Consider a partition of unity on $\overline{\Omega}$ with the following usual properties: $\sum_{i=1}^N \theta_i^\delta(x) = 1$, $0 \leq \theta_i^\delta(x) \leq 1$, and $|\nabla\theta_i^\delta(x)| \leq C/(\delta h)$, when $x \in \overline{\Omega}$, and $\theta_i^\delta(x)$ vanish on $\overline{\Omega}\backslash\overline{\Omega}_i^\delta$; for details see [7, 10]. The diagonal weighting matrices D_i^δ are defined to have diagonal elements values equal to $\theta_i^\delta(x)$ at the nodes $x \in \overline{\Omega}$.

Let us denote by V_i^δ, $i = 1, \cdots, N$, the local space of functions in $H^1(\Omega_i^\delta)$ which are continuous and piecewise linear on the elements of $\mathcal{T}^h(\Omega_i^\delta)$ and which vanish on $\partial\Omega_D \cap \partial\Omega_i^\delta$. We remark that we do not assume that the functions in V_i^δ vanish on the whole of $\partial\Omega_i^\delta$. We then define the corresponding restriction operator R_i^δ

$$R_i^\delta : V \rightarrow V_i^\delta, \quad i = 1, \cdots, N,$$

and obtain (6) and the following subspace decomposition

$$D_i^\delta (R_i^\delta)^T V_i^\delta \subset V \quad \text{and} \quad V = \sum_{i=1}^{N} D_i^\delta (R_i^\delta)^T V_i^\delta .$$

To define the local solvers, we introduce the local bilinear forms on V_i^δ by

$$a_{\Omega_i^\delta}(u_i, v_i) = \int_{\Omega_i^\delta} (\nabla u_i \cdot \nabla \bar{v}_i - k^2 u_i \bar{v}_i)\, dx - ik \int_{\partial\Omega_i^\delta \setminus (\partial\Omega_D \cup \partial\Omega_N)} u_i \bar{v}_i \, ds. \qquad (12)$$

For the case $k = 0$, i.e. the Poisson problem, $a_{\Omega_i^\delta}$ reduces to the regular H^1-seminorm inner product. For the case $k \neq 0$, i.e. the Helmholtz case, the bilinear form $a_{\Omega_i^\delta}$ induces the Sommerfeld boundary condition on $\partial\Omega_i^\delta \setminus \partial\Omega_{N\cup D}$, Neumann on $\partial\Omega_i \cap \partial\Omega_N$ and Dirichlet on $\partial\Omega_i \cap \partial\Omega_D$; see also [1]. The associated local problems define $\tilde{T}_i^\delta : V \rightarrow V_i^\delta$ by: for any $u \in V$

$$a_{\Omega_i^\delta}(\tilde{T}_i^\delta u, v) = a(u, D_i^\delta (R_i^\delta)^T v), \quad \forall v \in V_i^\delta, \ i = 1, \cdots, N, \qquad (13)$$

and let $T_i^\delta = D_i^\delta (R_i^\delta)^T \tilde{T}_i^\delta$ to obtain (8). When $k = 0$ and Ω_i^δ is a floating subdomain, the matrix A_i^δ is singular. To obtain the compatibility condition (Poisson problem) or to accelerate the algorithm (Helmholtz problem) we next introduce coarse problems.

5 Coarse Problems: Definitions of R_0^δ and P_0^δ

We note that some of the functions $\vartheta_i^\delta = I_h \theta_i^\delta$ cannot be used as a coarse basis functions since some of them do not satisfy the zero Dirichlet boundary condition on $\partial\Omega_D$ and therefore, do not belong to V. Hence we modify them just in a δh layer near $\partial\Omega_D$. This is done by defining a smooth cut-off function ϕ_δ on a δh layer near $\partial\Omega_D$ and by defining the coarse basis functions by $\vartheta_i^\delta = I_h(\phi_\delta \theta_i^\delta)$. Here I_h is the regular pointwise interpolation operator to V.

For the Poisson's problem, we define the coarse space V_0^δ as the span of the coarse basis functions $\vartheta_i^\delta, i = 1, \cdots, N$.

For the Helmholtz's problem, we combine the ϑ_i^δ with N_p planar waves. The basis functions for the coarse space V_0^δ are given by $I_h(\vartheta_i^\delta Q_j), i = 1, \ldots, N$ and $j = 1, \cdots, N_p$, with $Q_j(x) = e^{ik\Theta_j^T x}$, and $\Theta_j^T = (\cos(\theta_j), \sin(\theta_j))$, with $\theta_j = (j-1) \times \dfrac{\pi}{N_p}, j = 1, \cdots, N_p$; see also [3] for the use of plane waves for FETI-H methods.

We define the restriction matrix $R_0^\delta : V \rightarrow V_0^\delta$ consisting of the columns ϑ_i^δ (Poisson) or $I_h(\vartheta_i^\delta Q_j)$ (Helmholtz). We define $P_0^\delta : V \rightarrow V_0^\delta$ by: for any $u \in V$

$$a(P_0^\delta u, v) = a(u, v), \quad \forall v \in V_0^\delta,$$

and in matrix notation, $P_0^\delta = (R_0^\delta)^T (A_0^\delta)^{-1} R_0^\delta$, where $A_0^\delta = R_0^\delta A (R_0^\delta)^T$.

For the Poisson case, we have [5]:

Theorem 1.

$$a(u, u) \preceq a(T_{OBDD} u, u) \preceq (1 + \frac{H}{\delta h}) a(u, u).$$

6 Numerical Experiments

Below we present numerical results for solving the Helmholtz's problem on the unit square with the following boundary condition: Dirichlet $g_D = 1$ on west side, homogeneous Neumann on north and south sides, and homogeneous Sommerfeld on east side; see [3]. For the Poisson's equation including a discussion on the parallel implementations see Kimn and Bourdin [4].

Table 1. Number of iterations (PGMREZ) to solve Helmholtz equation for a Guided Wave Problem, Wave coarse space $N_p = 4$, Tol=10^{-6}, $k = 20$.

ovlp =1, n =	33	65	129	257
sub = 4x4	18	22	43	82
sub = 8x8	9	11	14	21
sub = 16x16		8	10	13
sub = 32x32			8	10

Table 2. Number of iterations (PGMREZ) to solve Helmholtz equation for a Guided Wave Problem, Wave coarse space $N_s = 8$, Tol=10^{-6}, $k = 20$,

ovlp =1, n =	33	65	129	257
sub = 4x4	14	18	25	48
sub = 8x8	7	7	8	9
sub = 16x16		4	4	4
sub = 32x32			2	2

References

1. X.-C. Cai, M. A. Casarin, F. W. Elliott Jr., and O. B. Widlund, *Overlapping Schwarz algorithms for solving Helmholtz's equation*, in Domain decomposition methods, 10 (Boulder, CO, 1997), Amer. Math. Soc., Providence, RI, 1998, pp. 391–399.
2. X. C. Cai, M. Dryja, and M. Sarkis, *A restricted additive Schwarz preconditioner with harmonic overlap for symmetric positive definite linear systems*, SIAM J. Sci. Comput., (2002). Submitted.
3. 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 (2000), pp. 283–303.
4. J.-H. Kimn and B. Bourdin, *Numerical implementation of overlapping balancing domain decomposition methods on unstructured meshes*, in Proceedings of the 16th International Conference on Domain Decomposition Methods, O. B. Widlund and D. E. Keyes, eds., Springer, 2006. These proceedings.
5. J.-H. Kimn and M. Sarkis, *Analysis of overlapping balancing domain decomposition methods.* In preparation, 2006.
6. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.
7. M. Sarkis, *Partition of unity coarse space and Schwarz methods with harmonic overlap*, in Recent Developments in Domain Decomposition Method, L. F. Pavarino and A. Tosell, eds., Springer-Verlag, 2002, pp. 75–92.
8. ———, *A coarse space for elasticity: Partition of unity rigid body motions coarse space*, in Proceedings of the Applied Mathematics and Scientific Computing Dubrovnik, Croacia, June, 2001, Z. D. et. al., ed., Kluwer Academic Press, 2003, pp. 3–31.
9. ———, *Partition of unity coarse spaces: Enhanced versions, discontinuous coefficients and applications to elasticity*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003.
10. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

Developments in Overlapping Schwarz Preconditioning of High-Order Nodal Discontinuous Galerkin Discretizations

Luke N. Olson[1], Jan S. Hesthaven[1], and Lucas C. Wilcox[1]

Division of Applied Mathematics, Brown University, 182 George Street, Box F, Providence, RI 02912, USA. Luke.Olson@brown.edu, Jan.Hesthaven@brown.edu, lucasw@dam.brown.edu

Summary. Recent progress has been made to more robustly handle the increased complexity of high-order schemes by focusing on the local nature of the discretization. This locality is particularly true for many Discontinuous Galerkin formulations and is the focus of this paper. The contributions of this paper are twofold. First, novel observations regarding various flux representations in the discontinuous Galerkin formulation are highlighted in the context of overlapping Schwarz methods. Second, we conduct additional experiments using high-order elements for the indefinite Helmholtz equation to expose the impact of overlap.

1 Introduction

We consider the Helmholtz equation

$$-\nabla \cdot \nabla u(\mathbf{x}) - \omega^2 u(\mathbf{x}) = f(\mathbf{x}) \quad \text{in } \Omega, \tag{1a}$$

$$u(\mathbf{x}) = g(\mathbf{x}) \quad \text{on } \Gamma. \tag{1b}$$

Although the form presented in (1) is evidently straightforward, it does still expose a number of difficulties that we discuss in this paper. The problem turns cumbersome quickly as the wave number increases since the resulting system of equations becomes indefinite. Identifying the key components to efficiently solving this wave problem will likely carry over into more complicated situations, such as Maxwell's equations.

The approach taken in this paper is an overlapping Schwarz-type method. The method presented is motivated by efforts of a number of authors who have outlined several situations where Schwarz methods have proved to be effective: indefinite problems, discontinuous Galerkin discretizations, and high-order elements [4, 2, 3, 8, 9, 10]. Based on this previously detailed success, we study the performance of a additive Schwarz method that utilizes element overlap to maintain efficient performance as the order of the discontinuous spectral element method increases and as indefiniteness becomes more prominent.

2 DG

The LDG formulation which we adopt yields several advantageous properties in the resulting linear system of equations. The global mass matrix is block diagonal, allowing cheap inversion, while symmetry is preserved in the global discretization matrix.

We begin by considering an admissible, shape regular triangulation $\mathcal{K}$ of $\Omega \in \mathbb{R}^2$ and let $h_\kappa = 1/2 \cdot \operatorname{diam}(\kappa)$, for $\kappa \in \mathcal{K}$. The numerical approximation u_h on element $\kappa \in \mathcal{K}_h$ is composed of Lagrange interpolating polynomials $L_j(\mathbf{x})$ at selected degrees of freedom $\mathbf{x}_j$ within κ. In 1-D, we describe these locations as the Gauss-Lobatto-Legendre (GLL) quadrature points. Similarly, for our 2-D reference triangle, $\hat{\kappa}$, we choose a distribution of nodes governed by electrostatics [6]. $N_\kappa = \dfrac{(n+1)(n+2)}{2}$ points are needed to ensure an order n resolution in the local polynomial approximation on element κ. Figure 1 shows an example on the reference element. Finally, we define $\mathcal{P}_n(\kappa)$, the local spectral element space where we seek an approximation.

The standard LDG formulation [1] is described first by introducing a slack variable $\mathbf{q} = \nabla u$. The first-order system for (1) on an arbitrary element κ is

$$-\nabla \cdot \mathbf{q} - \omega^2 u = f \quad \text{in } \kappa, \tag{2a}$$

$$\mathbf{q} - \nabla u = 0 \quad \text{in } \kappa. \tag{2b}$$

Multiplying each equation by scalar and vector test functions $\phi(\mathbf{x})$ and $\boldsymbol{\psi}(\mathbf{x})$, respectively, and integrating by parts yields the weak formulation. The local traces of u and $\mathbf{q}$ are replaced by approximations u^* and $\mathbf{q}^*$, also referred to as *numerical fluxes*. With this substitution and integrating by parts again, the associated (and slightly stronger) weak discrete problem is: find $(u_{h,n}, \mathbf{q}_{h,n})$ such that

$$-\int_\kappa \nabla \cdot \mathbf{q}_{h,n} \phi_n \, d\mathbf{x} - \omega^2 \int_\kappa u_{h,n} \phi_n \, d\mathbf{x} = \int_\kappa f_{h,n} \phi_n \, d\mathbf{x} + \int_{\partial\kappa} \mathbf{n}_k \cdot (\mathbf{q}^* - \mathbf{q}_{h,n}) \phi_n \, d\mathbf{x}, \tag{3a}$$

$$\int_\kappa \mathbf{q}_{h,n} \cdot \boldsymbol{\psi}_n \, d\mathbf{x} - \int_\kappa \nabla u_{h,n} \cdot \boldsymbol{\psi}_n \, d\mathbf{x} = \int_{\partial\kappa} (u^* - u_{h,n}) \mathbf{n}_k \cdot \boldsymbol{\psi} \, d\mathbf{x}, \tag{3b}$$

for all $\kappa \in \mathcal{K}_h$ and $(\phi_n, \boldsymbol{\psi}_n)$. The function spaces are the local spectral element spaces defined using the Lagrange interpolation above.

Defining the numerical flux is what separates different discontinuous Galerkin approaches [1] and is the most distinguishing feature of a formulation since the interelement connectivity is solely defined by the representation of the numerical flux on each edge. This choice directly impacts the approximation properties as well as the stability of the method. Moreover, the resulting (global) linear system of equations will perhaps exhibit symmetry and varying sparsity patterns depending on how the trace is approximated along each edge of each element in the tessellation. For a given element κ, define u^- to be the value of u interior to the element and define u^+ to be the value of u in the adjacent, neighboring element. For a scalar function u and vector function $\mathbf{q}$, the *jump* and the *average* between neighboring elements are respectively defined as $[\![u]\!] = u^- \mathbf{n}^- + u^+ \mathbf{n}^+$, $\{\!\{u\}\!\} = \frac{1}{2}(u^- + u^+)$, $[\![\mathbf{q}]\!] = \mathbf{q}^- \cdot \mathbf{n}_{k^-} + \mathbf{q}^+ \cdot \mathbf{n}_{k^+}$, $\{\!\{\mathbf{q}\}\!\} = \frac{1}{2}(\mathbf{q}^- + \mathbf{q}^+)$. For $\kappa \in \mathcal{K}$ with $\partial\kappa \in \Gamma_{\text{bdy}}$, these values are adjusted by extending the solution to a ghost element.

By defining the numerical fluxes u^* and $\mathbf{q}^*$ independently of ∇u, we will be able to formulate the weak problem (3) independently of the slack variable $\mathbf{q}(\mathbf{x})$. In general, the numerical fluxes for the LDG method are defined as [1]

$$u^* = \{\!\{u_{n,h}\}\!\} + \beta \cdot [\![u_{n,h}]\!] \qquad \mathbf{q}^* = \{\!\{\mathbf{q}_{n,h}\}\!\} - \beta[\![\mathbf{q}_{n,h}]\!] - \eta_k[\![u_{n,h}]\!]. \tag{4}$$

The sign on β is specifically opposite to ensure symmetry of the associated stiffness matrix [1]. Adhering to this form of a numerical flux is beneficial since the method is consistent and locally conservative. Further, if $\eta_k > 0$ the method is considered stable [1]. Setting $\beta = 0$ yields a *central* flux for u^* and a stabilized central flux for $\mathbf{q}^*$, while using $\beta = 0.5\mathbf{n}^-$ results in an upwinding scheme. The impact computationally is addressed in Section 4.

The numerical flux u^* is independent of $\mathbf{q}_{h,n}$ allowing us to write the discrete system completely independent of the slack variable $\mathbf{q}$ (cf. lifting operators in [1]). As we sum the weak problem over all elements $\kappa \in \mathcal{K}$ we will need the following global matrices: S^x, S^y, and M, which are stiffness and mass matrices and $F_{u^*}^{x,y}$ and $F_{q^*}^{x,y}$, which couple nodes in adjacent elements. Introducing global data vectors $\tilde{\mathbf{q}}^x$, $\tilde{\mathbf{q}}^y$, and $\tilde{\mathbf{u}}$ and summing the weak problem (3) over all elements $\kappa \in \mathcal{K}$, we arrive at the following

$$-S^x\tilde{\mathbf{q}}^x - S^y\tilde{\mathbf{q}}^y - \omega^2 M\tilde{\mathbf{u}} = M\mathbf{f} + F_{q^*}^x\tilde{\mathbf{q}}^x + F_{q^*}^y\tilde{\mathbf{q}}^y - \tau F_{q^*}^\tau\tilde{\mathbf{u}}, \tag{5}$$

$$M\tilde{\mathbf{q}}^x - S^x\tilde{\mathbf{u}} = F_{u^*}^x\tilde{\mathbf{u}}, \tag{6}$$

$$M\tilde{\mathbf{q}}^y - S^y\tilde{\mathbf{u}} = F_{u^*}^y\tilde{\mathbf{u}}. \tag{7}$$

Solving for the slack variable $\tilde{\mathbf{q}}^{x,y}$ in equations (6) and (7), and substituting into (5) eliminates the dependence on $\tilde{\mathbf{q}}$. The system, written in compact form is then

$$\left(-S + F - \omega^2 M\right)\tilde{\mathbf{u}} = M\mathbf{f}, \tag{8}$$

where $S = S^x M^{-1} S^x + S^y M^{-1} S^y$ and $F = F_{q^*}^x M^{-1} S^x + F_{q^*}^x M^{-1} F_{u^*}^x + F_{q^*}^y M^{-1} S^y + F_{q^*}^y M^{-1} F_{u^*}^y - \tau F_{u^*}^\tau$ The operator S is clearly negative semi-definite, while for $\tau > 0$, the composite operator $S - F$ is strictly negative definite. A full eigenspectrum analysis is missing and the impact on the preconditioner is unknown. However, it suffices to say that for moderate ω, indefinite and near singular matrices should be expected.

3 Additive Schwarz

Extensive work by Cai et al. [4, 2, 3] and Elman [5] conclude that standard Krylov based iterative methods handle a moderate number of flipped eigenvalues quite well for this indefinite problem. We will also use this class of methods and, in particular, choose the Generalized Minimum Residual method (GMRES). GMRES can be applied to indefinite systems and, more importantly, the preconditioned implementation permits indefinite preconditioning matrices. This will be beneficial in the case of the additive Schwarz (AS) method. It is noteworthy that BiCGStab yielded slightly improved results in our tests, but the observed trends remained the same.

Our implementation is a culmination of approaches, which includes overlapping subdomains and a coarse grid solution phase with the ability to handle non-nested coarse grids. It is important to note that a global coarse solve does not improve the convergence process if the grid is not rich enough to fully resolve a wave. There are a couple notable features about our approach. First, given a coarse grid tessellation, Ω^H, and a subdomain $\Omega_s^h \subset \Omega^h$, we define the restriction operator based on a standard finite element interpolation as $R_{0_{ij}}^T = \phi_i(\mathbf{x}_j)$. Here, $\phi_i(\mathbf{x})$ is a coarse grid basis function (bilinear in our case) and $\mathbf{x}_j$ is a node in Ω_s on the fine grid. $R_{0_{ij}} = 0$ if $\mathbf{x}_j$ is not in the underlying footprint of ϕ_i and is thus still sparse, although not in comparison to the injection operators used in the subdomain solves. To efficiently implement this process, let V be the Vandermonde matrix built from our orthogonal set of polynomials: $V_{i,j} = p_j(\mathbf{x}_i)$. With this we can transfer between modal and nodal representations easily with $\mathbf{f} = V\hat{\mathbf{f}}$ and $\hat{f} = V^{-1}\mathbf{f}$ since V^{-1} can be built locally in preprocessing. The advantage is clear when we look at more general interpolation in this respect. Let V_{cc} be the coarse basis/coarse nodes Vandermonde matrix and V_{cf} be the coarse basis/fine nodes Vandermonde matrix. Then $P_0 = V_{cf}V_{cc}^{-1} \equiv R_0^T$ defines the equivalent interpolation operator at the expense of only a few operations. Second, in order to ensure proper interpolation of constant solutions, we incorporate a row equilibration technique, by rescaling each row of R_0 by the row sum:

$$R_{0_{ij}} \leftarrow \frac{1}{\sum_j R_{0_{ij}}} R_{0_{ij}}. \tag{9}$$

The composite preconditioning matrix is then defined to be $M^{-1} = R_0^T A_0^{-1} R_0 + \sum_{s=1}^{S} R_s^T A_s^{-1} R_s$.

Overlap is also introduced in our algorithm. This increases communication, but, as we show in the next section, overlap is an essential component particularly for high-order approximations and as the matrix increases in indefiniteness and size. We define $\delta = 0$ as the case with no geometric overlap, keeping in mind the nature of the discontinuous discretization, where degrees of freedom in neighboring elements may share a geometric location, resulting in some resemblance of overlap. By increasing δ, we simply mean that each subdomain is padded by δ layers of elements. At first glance, this may seem extreme, since Fischer and Lottes [9] extend only by strips of nodes into the adjacent elements. However, the class of problems we address is altogether different, requiring a large number of elements, and requiring only moderate polynomial degrees, making overlap overhead costs small as the mesh is further refined. Moreover, layers of nodes within an electrostatic distribution are not readily available either in the element itself or in the reference element, whereas they have a straightforward formation in the case of tensor-based element.

4 Numerics

Using the central flux in the DG method is more correctly termed the Brezzi method [1]. Due to the ease of implementation, this formulation has grown in popularity, also benefiting from slightly improved conditioning over a bona fide LDG method where $\beta = 0.5\mathbf{n}^-$. Unfortunately, if $\beta = 0$, the data from elements κ^+ is needed to describe equations (3) in element κ^- as well as data from the neighbors of κ^+, which we label κ^{++}. Thus the influence on one element extends two layers beyond a given element. The noncompact stencil is also prevalent for $\beta \neq 0$, unless $\beta = 0.5\mathbf{n}^-$, which corresponds to an upwind flux. This is considered the LDG method since fortuitous cancellation of the terms eliminates the extension to neighboring elements, resulting in a stencil width of only one layer. Figure 1 articulates this effect. A more detailed explanation of the effects on discretization error and the eigenspectrum can be found in [7], although convergence of the iterative solution process is not addressed.

Also shown in Figure 1 is the so-called Interior Penalty method (IP). Here, a local gradient is used in the definition of the flux, which also results in a compact stencil. The IP method offers a straightforward implementation, however the poor conditioning of this approach requires careful attention. Table 1 illustrates a typical situation. The results are presented for the definite case ($\omega = 0$) on a grid with $h \approx 1/8$. A single level additive Schwarz scheme is used to precondition the GMRES acceleration. The first column reiterates the fact that the Brezzi approach ($\beta = 0.0$) has slightly better conditioning than the LDG implementation ($\beta = 0.5\mathbf{n}^-$), while the IP system suffers from a very poor spectrum. Column 2 also provides insight, showing that while the LDG scheme is slightly more ill-conditioned, the local type preconditioning scheme is more effective due to the compact stencil. The Brezzi operator responds similarly under preconditioning, but due to the wide stencil, the relative improvement is not as drastic. The preconditioning also has significant influence on the IP method, but due to the poor conditioning, it is difficult to fully quantify the effect of AS. We will focus on the Brezzi method throughout the rest of the paper since it is a widely used formulation of DG and since we expect the preconditioning results to be on the pessimistic side. A more comprehensive study of the various DG methods and preconditioning, similar to Table 1, is an ongoing research effort.

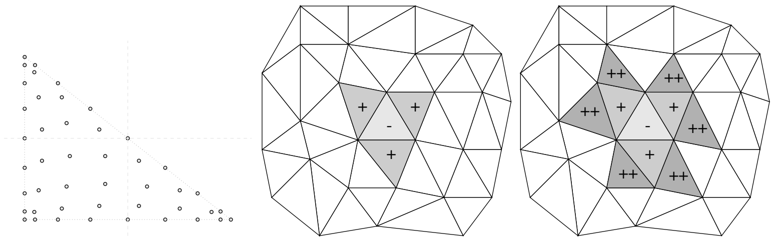
Table 1. GMRES iterations for Brezzi, LDG, and IP formulations with and without preconditioning.

	Brezzi		LDG		IP	
N	w/o AS	w/ AS	w/o AS	w/ AS	w/o AS	w/ AS
2	73	21	121	21	355	57
4	167	28	252	29	1291	151
6	316	30	456	32	> 2000	294
8	534	38	713	36	> 2000	568

Our test problem is basic, yet still exposes a principal difficulty: indefiniteness and high-order discretizations. We consider a smooth, solution $u(x, y) = \sin(2\pi\omega x)\sin(2\pi\omega y)$.

Comparing the iterations in Table 2 indicates that a coarse grid is beneficial for high-order discretizations. The number of GMRES iterations are reduced for each polynomial order when using a richer coarse grid. It is interesting to further note that the relative improvement is consistent as the order is increased. Overlap, however, has a much larger impact on the convergence of the preconditioned iterative method as indicated in Table 2.

As the frequency ω increases, more degrees of freedom are needed to fully resolve the solution. When the problem is viewed on a coarser grid, the discretization lacks resolution and the solution found on the coarse grid no longer resembles an accurate approximation to the fine grid solution. Thus the two-level error correction



(a) Reference Element (b) LDG and IP (c) Brezzi

Fig. 1. Stencil width relative to element κ^-.

Table 2. GMRES iterations with $h_f \approx 1/8$, $\omega = 1.0$: adding overlap.

	$\delta = 0$												$\delta = 1$										
	order n																						
h_c	1	2	3	4	5	6	7	8	9	10	11												
0	26	38	49	60	71	82	93	105	116	128	140												
1/4	22	32	39	50	58	67	72	81	88	100	108	→	22	22	23	24	24	25	25	26	26	27	28
1/8	14	25	30	36	43	47	55	60	66	73	79												

becomes ineffective and possibly pollutes the fine grid solution. Figure 2 shows that the iteration counts remain bounded as the polynomial order is increased for each selected ω. The iterations increase as the frequency is increased, but this is to be expected as more low eigenvalues are shifted to the positive half-plane. As expected, coarse solves do not improve solution for large wave numbers, however there is significant improvement as we introduce overlap, particularly for the case of the highly indefinite problem, $\omega = 50$.

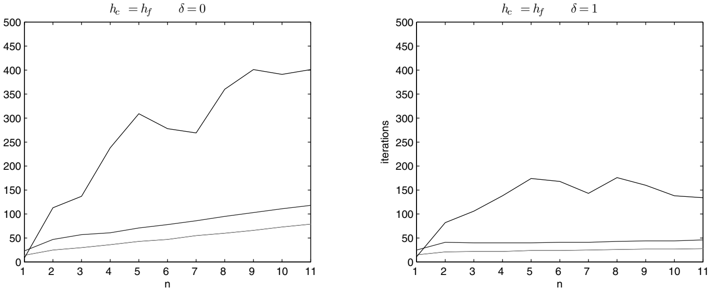


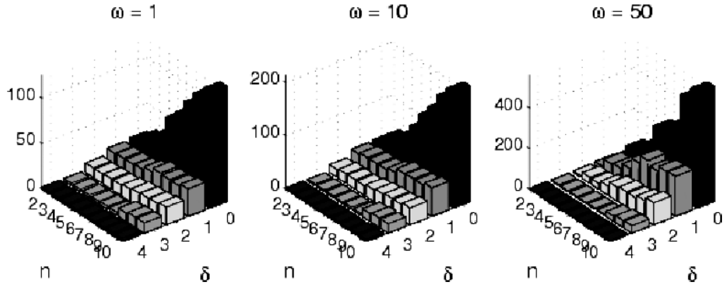
Fig. 2. GMRES iterations versus polynomial order: Comparing overlap impact for $\omega = 1.0$, 10.0, 50.0.

A more definitive test is to investigate problems where the discretization is neither under nor over resolved. Referring to dispersion analysis, using around several degrees of freedom per wavelength (in 1-D) is generally considered well resolved. Table 3 confirms the importance of overlap. Its relative improvement as n increases is attributed to the fact that larger subdomain solves are being used. The trend in overlap continues only so far. Figure 3 illustrates that performance is improved as the overlap is increased, however the relative impact becomes less.

Table 3. GMRES iterations: $h_f \approx 1/4$, no coarse grid.

		No AS	$\delta = 0$	$\delta = 1$
n	ω	avg. iterations		
1	$0 \dots 7$	48	23	18
2	$6 \dots 10$	106	43	27
3	$9 \dots 13$	170	57	30
4	$12 \dots 16$	271	72	36
5	$15 \dots 20$	392	106	48
6	$19 \dots 23$	534	151	67
7	$22 \dots 26$	705	193	72

Fig. 3. GMRES iterations versus polynomial (n) order and overlap (δ).



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 (2002), pp. 1749–1779.
2. X.-C. Cai, *A family of overlapping Schwarz algorithms for nonsymmetric and indefinite elliptic problems*, in Domain-based parallelism and problem decomposition methods in computational science and engineering, D. E. Keyes, Y. Saad, and D. G. Truhlar, eds., SIAM, Philadelphia, PA, 1995, pp. 1–19.
3. X.-C. Cai, M. A. Casarin, F. W. Elliott Jr., and O. B. Widlund, *Overlapping Schwarz algorithms for solving Helmholtz's equation*, in Domain decomposition methods, 10 (Boulder, CO, 1997), vol. 218 of Contemp. Math., AMS, Providence, RI, 1998, pp. 391–399.
4. X.-C. Cai and O. B. Widlund, *Domain decomposition algorithms for indefinite elliptic problems*, SIAM J. Sci. Statist. Comput., 13 (1992), pp. 243–258.

5. H. C. Elman, O. G. Ernst, and D. P. O'Leary, *A multigrid method enhanced by Krylov subspace iteration for discrete Helmhotz equations*, SIAM J. Sci. Comput., 23 (2001), pp. 1291–1315.
6. J. S. Hesthaven, *From electrostatics to almost optimal nodal sets for polynomial interpolation in a simplex*, SIAM J. Numer. Anal., 35 (1998), pp. 655–676.
7. R. M. Kirby, *Toward dynamic spectral/hp refinement: algorithms and applications to flow-structure interactions*, PhD thesis, Brown University, May 2003.
8. C. Lasser and A. Toselli, *Overlapping preconditioners for discontinuous Galerkin approximations of second order problems*, in Thirteenth international conference on domain decomposition, N. Debit, M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, eds., ddm.org, 2001, pp. 78–84.
9. J. W. Lottes and P. F. Fischer, *Hybrid multigrid/Schwarz algorithms for the spectral element method*, Tech. Rep. ANL/MCS-P1052-0403, Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL, May 2003.
10. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

Domain-decomposed Fully Coupled Implicit Methods for a Magnetohydrodynamics Problem*

Serguei Ovtchinnikov[1], Florin Dobrian[2], Xiao-Chuan Cai[3], David Keyes[4]

[1] University of Colorado at Boulder, Department of Computer Science, 430 UCB, Boulder, CO 80309, USA. serguei.ovtchinnikov@colorado.edu
[2] Old Dominion University, Department of Computer Science, Norfolk, VA 23529, USA. dobrian@cs.odu.edu
[3] University of Colorado at Boulder, Department of Computer Science, 430 UCB, Boulder, CO 80309, USA. cai@cs.colorado.edu
[4] Columbia University, Department of Applied Physics & Applied Mathematics, 500 W. 120th St., New York, NY 10027, USA. david.keyes@columbia.edu

Summary. We present a parallel fully coupled implicit Newton-Krylov-Schwarz algorithm for the numerical solution of the unsteady magnetic reconnection problem described by a system of reduced magnetohydrodynamics equations in two dimensions. In particular, we discuss the linear and nonlinear convergence, the parallel performance of a third-order implicit algorithm and compare to solutions obtained with an explicit method.

1 Introduction

In the magnetohydrodynamics (MHD) formalism plasma is treated as a conducting fluid satisfying the Navier-Stokes equations coupled with Maxwell's equations [5]. The behavior of an MHD system is complex since it admits phenomena such as Alfvén waves and their instabilities. One of the intrinsic features of MHD is the formation of a singular current density sheet, which is linked to the reconnection of magnetic field lines [2, 8, 9, 11], which in turn leads to the release of energy stored in the magnetic field. Numerical simulation of the reconnection plays an important role in our understanding of physical systems ranging from the solar corona to laboratory fusion devices. Capturing the change of the magnetic field topology requires a more general model than ideal MHD. A resistive Hall MHD system is considered in this paper. To simulate this multi-scale, multi-physics phenomenon, a robust solver has

*The work was partially supported by DOE DE-FC02-01ER25479, DEFC02-04ER25595, NSF ACI-0305666 and ACI-0352334.

to be applied in order to deal with the high degree of nonlinearity and the nonsmooth blowup behavior in the system. One of the successful approaches to the numerical solution of the MHD system is based on the splitting of the system into two parts, where equations for the current and the vorticity are advanced in time, and the corresponding potentials are obtained by solving Poisson-like equations in a separate step. In such an explicit approach, to satisfy the CFL condition, the time step may become very small, especially in the case of fine meshes, and the Poisson solves must therefore be performed frequently. On the other hand, implicit time stepping presents an alternative approach that may allow the use of larger time steps. However, the non-smooth nature of the solution often results in convergence difficulties. In this work we take a fully coupled approach such that no operator splitting is applied to the system of MHD equations. More precisely, we first apply a third-order implicit time integration scheme, and then, to guarantee nonlinear consistency, we use a one-level Newton-Krylov-Schwarz algorithm to solve the large sparse nonlinear system of algebraic equations containing all physical variables at every time step. The focus of this paper is on the convergence and parallel performance studies of the proposed implicit algorithm.

2 Model MHD Problem

We consider a model MHD problem described as follows [1, 6]:

$$\begin{cases} \nabla^2 \phi = U \\ \nabla^2 \psi = \dfrac{1}{d_e^2}(\psi - F) \\ \dfrac{\partial U}{\partial t} + [\phi, U] = \dfrac{1}{d_e^2}[F, \psi] + \nu \nabla^2 U \\ \dfrac{\partial F}{\partial t} + [\phi, F] = \rho_s^2 [U, \psi] + \eta \nabla^2 (\psi - \psi^0), \end{cases} \tag{1}$$

where U is the vorticity, F is the canonical momentum, ϕ and ψ are the stream functions for the vorticity and current density, respectively, ν is the plasma viscosity, η is the normalized resistivity, $d_e = c/\omega_{pe}$ is the inertial skin depth, and $\rho_s = \sqrt{T_e/T_i \rho_i}$ is the ion sound Larmor radius. The current density is obtained by $J = (F - \psi)/d_e^2$. The Poisson bracket is defined as: $[A, B] \equiv (\partial A/\partial x)(\partial B/\partial y) - (\partial A/\partial y)(\partial B/\partial x)$. Every variable in the system is assumed to be the sum of an equilibrium and a perturbation component; i.e. $\phi = \phi^0 + \phi^1$, $\psi = \psi^0 + \psi^1$, $U = U^0 + U^1$, and $F = F^0 + F^1$, where $\phi^0 = U^0 = 0$, $\psi^0 = \cos(x)$, and $F^0 = (1 + d_e^2)\cos(x)$ are the equilibrium components. After substitutions, we arrive at the following system for the perturbed variables:

$$\begin{cases} \nabla^2 \phi^1 = U^1 \\ \nabla^2 \psi^1 = \dfrac{1}{d_e^2}(\psi^1 - F^1) \\ \dfrac{\partial U^1}{\partial t} + [\phi^1, U^1] = \dfrac{1}{d_e^2}[F^1, \psi^1] + \nu \nabla^2 U^1 + \dfrac{1}{d_e^2}\left(\dfrac{\partial \psi^1}{\partial y} F_{eqx} + \dfrac{\partial F^1}{\partial y} B_{eqy}\right) \\ \dfrac{\partial F^1}{\partial t} + [\phi^1, F^1] = \rho_s^2 [U^1, \psi^1] + \eta \nabla^2 \psi^1 + \left(\dfrac{\partial \phi^1}{\partial y} F_{eqx} + \rho_s^2 \dfrac{\partial U^1}{\partial y} B_{eqy}\right), \end{cases} \tag{2}$$

where $F_{eqx} = -(1+d_e^2)\sin(x)$ and $B_{eqy} = \sin(x)$. The system is defined on a rectangular domain $\Omega \equiv [l_x, l_y] \equiv [2\pi, 4\pi]$, and doubly periodic boundary conditions are assumed. For initial conditions, we use a nonzero initial perturbation in ϕ^1 and a zero initial perturbation in ψ^1. The exact form of the perturbation follows after some useful definitions. The aspect ratio is $\varepsilon = l_x/l_y$. The perturbation's magnitude is scaled by $\delta = 10^{-4}$. We define $\tilde{d_e} = \max\{d_e, \rho_s\}$ and $\gamma = \varepsilon\tilde{d_e}$. For the initial value of the ϕ perturbation we use

$$\phi^1(x,y,0) = \begin{cases} \delta\,\dfrac{\gamma}{\varepsilon}\,\mathrm{erf}\left(\dfrac{x}{\sqrt{2}\tilde{d_e}}\right)\sin(\varepsilon y) & \text{if } 0 \le x < \dfrac{\pi}{2} \\ -\delta\,\dfrac{\gamma}{\varepsilon}\,\mathrm{erf}\left(\dfrac{x-\pi}{\sqrt{2}\tilde{d_e}}\right)\sin(\varepsilon y) & \text{if } \dfrac{\pi}{2} \le x < \dfrac{3\pi}{2} \\ \delta\,\dfrac{\gamma}{\varepsilon}\,\mathrm{erf}\left(\dfrac{x-2\pi}{\sqrt{2}\tilde{d_e}}\right)\sin(\varepsilon y) & \text{if } \dfrac{3\pi}{2} \le x \le 2\pi. \end{cases} \tag{3}$$

Other quantities are set as: $U^1(x,y,0) = \nabla^2\phi^1(x,y,0)$ and $F^1(x,y,0) = \psi^1(x,y,0) - d_e\nabla^2\psi^1(x,y,0)$. From now on, we drop the superscript and assume that the four fields ϕ, ψ, U and F represent the perturbed components only. In order to connect the stream functions to physical quantities the following definitions are used: $\mathbf{v} = e_z \times \nabla\phi$ and $\mathbf{B} = B_0 e_z + \nabla\psi \times e_z$. Here $\mathbf{B}$ stands for the total magnetic field, B_0 is the guiding field in the z direction, and $\mathbf{v}$ is the velocity in the plane perpendicular to the guiding field.

We discretize the system of PDEs with finite differences on a uniform mesh of sizes h_x and h_y in x and y directions, respectively. At time level t^k, we denote the grid values of the unknown functions $\phi(x,y,t)$, $\psi(x,y,t)$, $U(x,y,t)$, and $F(x,y,t)$, as $\phi^k_{i,j}$, $\psi^k_{i,j}$, $U^k_{i,j}$, and $F^k_{i,j}$. The time independent components of the system (2) are discretized with the standard second-order central difference method. For the time discretization, we use some multistep formulas, known as backward differentiation formulas (BDF) [7]. In this paper, we focus on a third-order temporal and second-order spatial discretizations as shown in (4), where $R^{k+1}_\phi(i,j)$, $R^{k+1}_\psi(i,j)$, $R^{k+1}_U(i,j)$, and $R^{k+1}_F(i,j)$ are the second-order accurate spatial discretizations of the time-independent components. We need to know solutions at time steps $k-2$, $k-1$ and k in order to compute a solution at time step $k+1$ in (4). Lower order schemes are employed at the beginning of the time integration for these start-up values.

$$\begin{cases} R^{k+1}_\phi(i,j) = 0 \\ R^{k+1}_\psi(i,j) = 0 \\ \dfrac{h_x h_y}{6\Delta t}\left(11U^{k+1}_{i,j} - 18U^k_{i,j} + 9U^{k-1}_{i,j} - 2U^{k-2}_{i,j}\right) - R^{k+1}_U(i,j) = 0 \\ \dfrac{h_x h_y}{6\Delta t}\left(11F^{k+1}_{i,j} - 18F^k_{i,j} + 9F^{k-1}_{i,j} - 2F^{k-2}_{i,j}\right) - R^{k+1}_F(i,j) = 0 \end{cases} \tag{4}$$

3 One-level Newton-Krylov-Schwarz Method

At each time step, the discretized fully coupled system of equations (4) can be represented by $G(E) = 0$, where $E = \{\phi, \psi, U, F\}$. The unknowns are ordered mesh point by mesh point, and at each mesh point they are in the order ϕ, ψ, U, and F. The mesh points are ordered subdomain by subdomain for the purpose of parallel processing. The system is solved with a one-level Newton-Krylov-Schwarz (NKS), which is a general purpose parallel algorithm for solving systems of nonlinear algebraic equations. The Newton iteration is given as: $E_{k+1} = E_k - \lambda_k J(E_k)^{-1} G(E_k)$, $\ k = 0, 1, ...$, where E_0 is a solution obtained at the previous time step, $J(E_k) = G'(E_k)$ is the Jacobian at E_k, and λ_k is the steplength determined by a linesearch procedure [3]. Due to doubly periodic boundary conditions, the Jacobian has a one-dimensional null-space that is removed by projecting out a constant. The accuracy of the Jacobian solve is determined by some $\eta_k \in [0, 1)$ and the condition $\|G(E_k) + J(E_k)s_k\| \leq \eta_k \|G(E_k)\|$. The overall algorithm can be described as follows:

(a) Inexactly solve the linear system $J(E_k)s_k = -G(E_k)$ for s_k using a preconditioned GMRES(30) [10].
(b) Perform a full Newton step with $\lambda_0 = 1$ in the direction s_k.
(c) If the full Newton step is unacceptable, backtrack λ_0 using a backtracking procedure until a new λ is obtained that makes $E_+ = E_k + \lambda s_k$ an acceptable step.
(d) Set $E_{k+1} = E_+$, go to step 1 unless a stopping condition has been met.

In step 1 above we use a right-preconditioned GMRES to solve the linear system; i.e., the vector s_k is obtained by approximately solving the linear system $J(E_k)M_k^{-1}(M_k s_k) = -G(E_k)$, where M_k^{-1} is a one-level additive Schwarz 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 grid cells all around the perimeter, i.e., $ovlp = (H_x' - H_x)/2 = (H_y' - H_y)/2$ for every subdomain. The boundary subdomains are also extended all around their perimeters because of the doubly periodic physical boundary. On each extended subdomain Ω_l', we construct a subdomain preconditioner B_l, whose elements are $B_l^{i,j} = \{J_{ij}\}$, where the node indexed by (i, j) belongs to Ω_l'. The entry J_{ij} is calculated with finite differences $J_{ij} = 1/(2\delta)(G_i(E_j + \delta) - G_i(E_j - \delta))$, where $0 < \delta \ll 1$ is a constant. Homogeneous Dirichlet boundary conditions are used on the subdomain boundary $\partial\Omega_l'$. The additive Schwarz preconditioner can be written as

$$M_k^{-1} = (R_1)^T B_1^{-1} R_1 + \cdots + (R_N)^T B_N^{-1} R_N. \tag{5}$$

Let n be the total number of mesh points and n_l' the total number of mesh points in Ω_l'. Then, R_l is an $n_l' \times n$ block matrix that is defined as: its 4×4 block element $(R_l)_{i,j}$ is an identity block if the integer indices $1 \leq i \leq n_l'$ and $1 \leq j \leq n$ belong to a mesh point in Ω_l', or a block of zeros otherwise. The R_l serves as a restriction matrix because its multiplication by a block $n \times 1$ vector results in a smaller $n_l' \times 1$ block vector by dropping the components corresponding to mesh points outside Ω_l'. Various inexact additive Schwarz preconditioners can be constructed by replacing the matrices B_l in (5) with convenient and inexpensive to compute matrices, such as those obtained with incomplete and complete factorizations. In this paper we employ the LU factorization.

4 Numerical Results

To illustrate model behavior, we choose nominal values of the inertial skin depth $d_e = 0.08$ and the ion sound Larmor radius $\rho_s = 0.24$. The normalized resistivity and viscosity are chosen in the range $\eta, \nu \in [10^{-4}, 10^{-2}]$. Time in the system is normalized to the Alfvén time $\tau_A = \sqrt{4\pi n m_i} l_x / B_{y0}$, where B_{y0} is the characteristic magnitude of the equilibrium magnetic field and l_x is the macroscopic scale length [6]. Ω is uniformly partitioned into rectangular meshes up to 600×600 in size. The stopping conditions for the iterative processes are given as follows: relative reduction in nonlinear function norm $\|G(E_k)\| \leq 10^{-7}\|G(E_0)\|$, absolute tolerance in nonlinear function norm $\|G(E_k)\| \leq 10^{-7}$, relative reduction in linear residual norm $\|r_k\| \leq 10^{-10}\|r_0\|$, and absolute tolerance in linear residual norm $\|r_k\| \leq 10^{-7}$.

A typical solution is shown in Fig. 1. The initial perturbation in ϕ produces a feature-rich behavior in ψ, U, and F. The four variables in the system evolve at different rates: ϕ and ψ evolve at a slower rate than F and U. For $\eta = 10^{-3}$ and $\nu = 10^{-3}$ we observe an initial slow evolution of current density profiles up to time $100\tau_A$ and the solution blows up at time near $290\tau_A$. In the middle of the domain the notorious "X" structure is developed, as can be seen in the F contours, where the magnetic flux is reconnected. Similar reconnection areas are developed on the boundaries of the domain due to the periodicity of boundary conditions and the shape of the initial ϕ perturbation. In the reconnection regions sharp current density peaks (Fig. 2 (a)) are formed. We compare solutions obtained by our implicit method with these obtained with an explicit method [4]. Fig. 2 (b) shows that the third-order implicit method allows for much larger time steps and produces a solution that is very close to the solution obtained with the explicit algorithm, where the size of the time step is determined by the CFL constraint.

Next, we look at some of the machine dependent properties of the algorithm. Our main focus is on the scalability, which is an important quality in evaluating parallel algorithms. First, we look at the total computing time as a function of the number of subdomains and calculate $t(16)/t(np)$ which gives a ratio of time needed to solve the problem with sixteen processors to the time needed to solve the problem with np processors. Fig. 3 shows the results for a 600×600 mesh, and an overlap of 6 is used in all cases. We can see that the one-level algorithm scales reasonably well in terms of the compute time. Table 1 illustrates results obtained on a 600×600 mesh. The compute time scalability is attained despite the fact that the total number of linear iterations increases with the number of subdomains.

5 Conclusions and Future Work

The proposed fully coupled implicit scheme with a third-order temporal discretization allows much larger time steps than the explicit method, while still preserving the solution accuracy. One-level NKS converges well with the problem parameters in the specified range, given the right stopping conditions. Without a coarse space, the algorithm scales reasonably well for a large number of processors with a medium subdomain overlap. Future continuation of this work may include solutions of the MHD problem on finer meshes with a larger number of processors. Longer time integration with various η and ν values, as well as higher ρ_s to d_e ratios, may be helpful

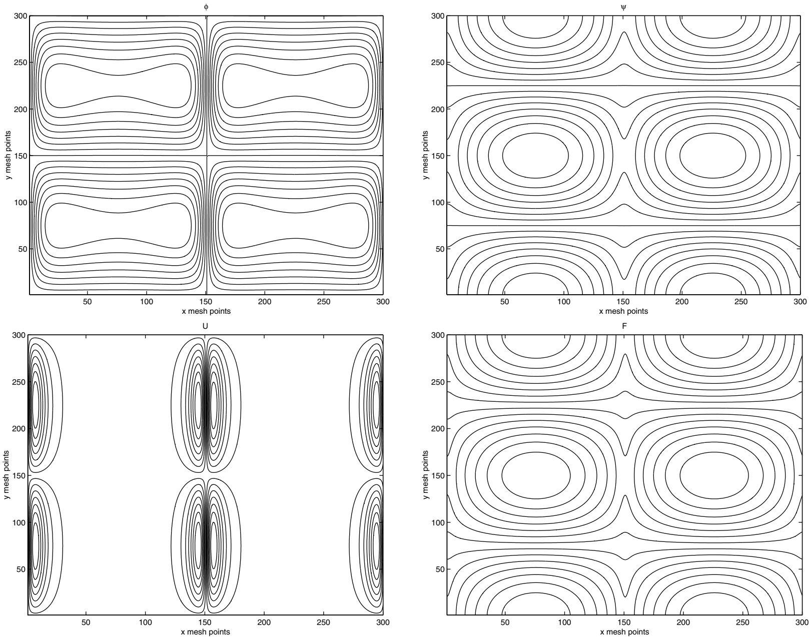


Fig. 1. Contour plots of ϕ (top left), ψ (top right), U (bottom left), and F (bottom right). The results are obtained on 300×300 mesh, $\Delta t = 1.0\tau_A$, $t = 100\tau_A$, $\eta = 10^{-3}$, $\nu = 10^{-3}$, implicit time stepping.

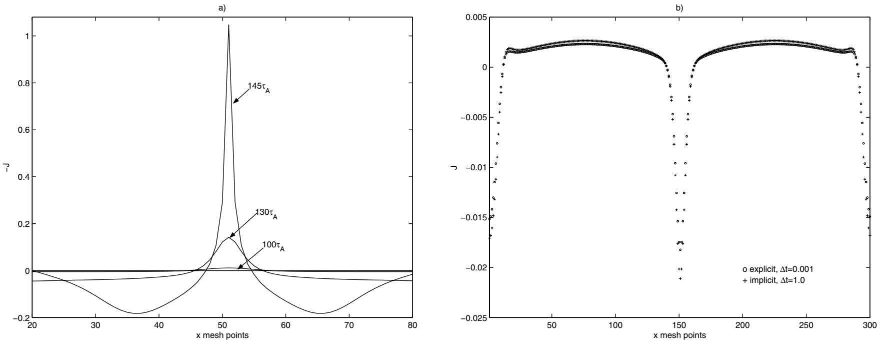


Fig. 2. a) Formation of current density peaks in the reconnection region, J, 100×100 mesh, $\eta = 10^{-2}$, $\nu = 10^{-2}$, $\Delta t = 1.0\tau_A$. b) Comparison plots of J obtained with the explicit method ($\Delta t = 0.001\tau_A$) and the implicit with $\Delta t = 1.0\tau_A$ at $t = 200\tau_A$ on 300×300 mesh with $\eta = 10^{-3}$ and $\nu = 10^{-3}$.

Table 1. Scalability with respect to the number of processors, 600 × 600 mesh. *LU* factorization for all subproblems, *ovlp* = 6. Time step $\Delta t = 1.0\tau_A$, 10 time steps, $t = 280\tau_A$. The problem is solved with 16 – 400 processors.

np	t[sec]	Total Nonlinear	Total Linear	Linear/Nonlinear
16	2894.8	30	1802	60.1
36	1038.1	30	2154	71.8
64	542.8	30	2348	78.3
100	340.5	30	2637	87.9
144	239.5	30	2941	98.0
225	167.8	30	3622	120.7
400	120.4	30	4792	159.7

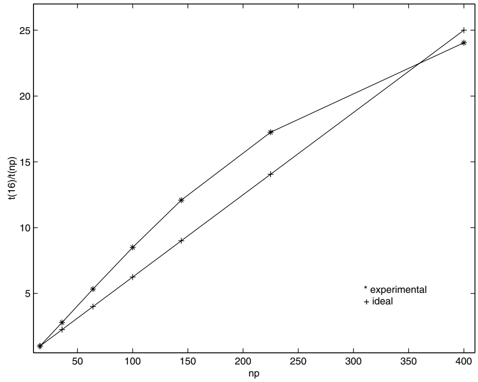


Fig. 3. Computing time scalability $t(16)/t(np)$, 600×600 mesh, $\eta = 10^{-3}$, $\nu = 10^{-3}$, $\Delta t = 1.0\tau_A$ with 16 – 400 processors, $t = 280\tau_A$. The data are collected over 10 time steps. The "∗" shows experimental speedup values and "+" depicts the ideal speedup.

in the further understanding of the algorithm for the numerical solutions of MHD problems.

References

1. E. Cafaro, D. Grasso, F. Pegoraro, F. Porcelli, and A. Saluzzi, *Invariants and geometric structures in nonlinear Hamiltonian magnetic reconnection*, Phys. Rev. Lett., 80 (1998), pp. 4430–4433.
2. L. Chacón, D. A. Knoll, and J. M. Finn, *An implicit, nonlinear reduced resistive MHD solver*, J. Comput. Phys., 178 (2002), pp. 15–36.
3. J. E. Dennis, Jr. and R. B. Schnabel, *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*, SIAM, Philadelphia, PA, 1996.
4. K. Germaschewski. Personal communications.
5. R. J. Goldston and P. H. Rutherford, *Introduction to Plasma Physics*, Institute of Physics (IOP) Publishing, Philadelphia, PA, 1995.
6. D. Grasso, F. Pegoraro, F. Porcelli, and F. Califano, *Hamiltonian magnetic reconnection*, Plasma Phys. Control Fusion, 41 (1999), pp. 1497–1515.

7. E. Hairer, S. P. Norsett, and G. Wanner, *Solving Ordinary Differential Equations I: Nonstiff Problems*, Springer-Verlag, 1993.
8. M. Ottaviani and F. Porcelli, *Nonlinear collisionless magnetic reconnection*, Phys. Rev. Lett., 71 (1993), pp. 3802–3805.
9. ———, *Fast nonlinear magnetic reconnection*, Phys. Plasmas, 2 (1995), pp. 4104–4117.
10. Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, second ed., 2003.
11. H. R. Strauss and D. W. Longcope, *An adaptive finite element method for magnetohydrodynamics*, J. Comput. Phys., 147 (1998), pp. 318–336.

A Proposal for a Dynamically Adapted Inexact Additive Schwarz Preconditioner

Marcus Sarkis[1] and Daniel B. Szyld[2]

[1] Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, Brazil, and Worcester Polytechnic Institute, Worcester, MA 01609, USA.
msarkis@fluid.impa.br

[2] Department of Mathematics, Temple University, Philadelphia, PA 19122, USA.
szyld@math.temple.edu

1 Introduction

Additive Schwarz is a powerful preconditioner used in conjuction with Krylov subspace methods (e.g., GMRES [7]) for the solution of linear systems of equations of the form $Au = f$, especially those arising from discretizations of differential equations on a domain divided into p (overlapping) subdomains [5], [9], [10]. In this paper we consider right preconditioning, i.e., the equivalent linear system is $AM^{-1}w = f$, with $Mu = w$. The additive Schwarz preconditioner is

$$M^{-1} = \sum_{i=1}^{p} R_i^T A_i^{-1} R_i, \tag{1}$$

where R_i is a restriction operator and $A_i = R_i A R_i^T$ is a restriction of A to a subdomain. The strength of this preconditioner stems in part from having overlap between the subdomains, and in part from the efficiency of local solvers, i.e., solutions of the "local" problems

$$A_i x = R_i v. \tag{2}$$

We also consider a weighted additive Schwarz preconditioner with harmonic extension (WASH), a preconditioner in the family of restricted additive Schwarz (RAS) preconditioners [3] of the form

$$M^{-1} = \sum_{i=1}^{p} R_i^T A_i^{-1} R_i^{\omega}, \tag{3}$$

in which the restriction operator R_i^{ω} is such that all variables corresponding to a point in the overlap are weighted with weights that add up to one, i.e., $\sum_{i=1}^{p} R_i^T R_i^{\omega} = I$ [4].

In this paper we consider the case when the local problems are either too large or too expensive to be solved exactly. Therefore, the systems (2) are solved using an iterative method. Usually, one takes a fixed number of (inner) iterations. We are interested instead in prescribing a certain (inner) tolerance so that the iterative method for the solution of (2) stops when the local residual

$$s_{i,k} = A_i x_j - R_i v_k$$

has norm below the inner tolerance ($j = j(i,k)$ being the index of the inner iteration, and we write $x_j = \tilde{A}_{i,k}^{-1} R_i v_k$, where the subscript in $\tilde{A}_{i,k}$ indicates that the inexact local solvers changes also with k). Inexact local solvers have been used extensively (see, e.g., [9]); what is new here is that the inexactness changes as the (outer) iterations proceed. In this case, the (global) preconditioner changes from step to step, i.e.,

$$M_k^{-1} = \sum_{i=1}^{p} R_i^T \tilde{A}_{i,k}^{-1} R_i, \tag{4}$$

and one needs to use a flexible Krylov subspace method, such as FGMRES [6].

Recent results have shown that it is possible to vary how inexact a preconditioner is without degradation of the overall performance of a Krylov method; see [1], [8] and references therein, and in particular we mention [2] where Schur complement methods were studied. More precisely, the preconditioned system has to be solved more exactly at first, while the exactness can be relaxed as the (outer) iterative method progresses. In this paper we propose to apply these new ideas to additive Schwarz preconditioning and its restricted variants, thus providing a way of dynamically choosing the inner tolerance for the local solvers in each step k of the (outer) iterative method. Our proposed strategy is illustrated with numerical experiments, which show that there is a great potential in savings while maintaining the performance of the overall process.

2 A Dynamic Stopping Criterion for the Local Solvers

The algorithmic setup is as follows, in each step k of the (outer) Krylov subspace method for the solution of $Au = f$ (we use FGMRES here), we apply a preconditioner of the form (4), where the symbol $\tilde{A}_{i,k}$ indicates that the solution of local problem (2) is approximated by a Krylov subspace method (we use GMRES) iterated until $\|s_{i,k}\| \le \varepsilon_{i,k}$.

In this setup, at the kth iteration instead of the usual matrix-vector product $AM^{-1}v_k$ we have

$$\begin{aligned} AM_k^{-1} v_k &= A \sum_{i=1}^{p} R_i^T \tilde{A}_{i,k}^{-1} R_i v_k \\ &= A \sum_{i=1}^{p} R_i^T A_i^{-1} R_i v_k + A \sum_{i=1}^{p} R_i^T (\tilde{A}_{i,k}^{-1} - A_i^{-1}) R_i v_k \\ &= AM^{-1} v_k + A \sum_{i=1}^{p} R_i^T A_i^{-1} s_{i,k}. \end{aligned}$$

Thus, we can write $AM_k^{-1} v_k = (AM^{-1} + E_k) v_k$, where E_k is the inexactness of the preconditioned matrix at the kth step, and $f_k = E_k v_k = A \sum_{i=1}^{p} R_i^T A_i^{-1} s_{i,k}$, so that

$$\|f_k\| = \|E_k v_k\| \leq \sum_{i=1}^{p} \|AR_i^T A_i^{-1}\| \|s_{i,k}\|. \tag{5}$$

In the situation we are describing, namely of inexact preconditioner, the inexact Arnoldi relation that holds is

$$AV_m + [f_1, f_2, \ldots, f_m] = V_{m+1} H_{m+1,m},$$

where the $V_m = [v_1, v_2, \ldots, v_m]$ has orthonormal columns, and $H_{m+1,m}$ is upper Hessenberg. Let $W_m = V_{m+1} H_{m+1,m}$, and r_k be the GMRES (outer) residual at the kth step. It follows from [8, sections 4 and 5] that

$$\|W_m^T r_m\| \leq \kappa(H_{m+1,m}) \sum_{k=1}^{m} \|f_k\| \|r_{k-1}\|, \tag{6}$$

$$\|r_m - \tilde{r}_m\| \leq \frac{1}{\sigma_{min}(H_{m+1,m})} \sum_{k=1}^{m} \|f_k\| \|r_{k-1}\|, \tag{7}$$

where $\kappa(H_{m+1,m}) = \sigma_{max}(H_{m+1,m})/\sigma_{min}(H_{m+1,m})$ is the condition number of $H_{m+1,m}$, and $\tilde{r}_m = r_0 - V_{m+1} H_{m+1,m} y_m$ is the computed residual. In the exact case, i.e., when $\varepsilon_{i,k} = 0$, $i = 1, \ldots, p$, $k = 1, 2, \ldots$, then $W_m^T r_m = 0$. Equation (6) indicates how far from that optimal situation we may be. The residual gap (7) is the norm of the difference between the "true" residual $r_m = f - AV_m y_m$ and the computed one. As $\tilde{r}_m \to 0$, we have that if the right hand side of (7) is of order ε, then $\|r_m\| \to \mathcal{O}(\varepsilon)$; cf. [8, Figure 9.1].

Using (5) we obtain the following result.

Proposition 1. *If the local residuals satisfy $\|s_{i,k}\| \leq \varepsilon_k$, $i = 1, \ldots, p$, then the kth GMRES (outer) residual satisfies the following two relations:*

$$\|W_m^T r_m\| \leq \kappa(H_{m+1,m}) \sum_{i=1}^{p} \|AR_i^T A_i^{-1}\| \sum_{k=1}^{m} \varepsilon_k \|r_{k-1}\|, \tag{8}$$

$$\|r_m - \tilde{r}_m\| \leq \frac{1}{\sigma_{min}(H_{m+1,m})} \sum_{i=1}^{p} \|AR_i^T A_i^{-1}\| \sum_{k=1}^{m} \varepsilon_k \|r_{k-1}\|. \tag{9}$$

We can then conclude that an *a posteriori* result holds.

Proposition 2. *If ε_k, the bound of the local residual norms, satisfy*

$$\varepsilon_k \leq K_m \frac{1}{\|r_{k-1}\|}\ \varepsilon, \tag{10}$$

with

$$K_m = 1/m\kappa(H_{m+1,m}) \sum_{i=1}^{p} \|AR_i^T A_i^{-1}\|, \tag{11}$$

then $\|W_m^T r_m\| \leq \varepsilon$, and if (10) *holds with*

$$K_m = \sigma_{min}(H_{m+1,m})/m \sum_{i=1}^{p} \|AR_i^T A_i^{-1}\|, \tag{12}$$

then $\|r_m - \tilde{r}_m\| \leq \varepsilon$.

We mention that these results apply to the case of inexact WASH preconditioning as well, where the restriction R_i on the right of each term in (4) is replaced with R_i^{ω}.

3 Implementation Considerations

The power of Proposition 2 is to point out that one can relax the local residual norms in a way inversely proportional to the norm of the (outer or global) residual from the previous step; cf. [1], [8]. The constants K_m as stated in (11) and (12), which do not depend on k, depend in part on A, i.e., on the problem to be solved, the preconditioner, through the local problems represented by A_i, as well as on how the inexact strategy is implemented, through $H_{m+1,m}$. Observe that since $m\kappa(H_{m+1,m}) \gg 1$ it is natural from (11) to expect $K_m \leq 1$.

Depending on the problem, we could obtain an *a priori* bound for K_m which would not depend on the specifics of the inexact strategy, for example by setting $\kappa(H_{m+1,m}) \approx \gamma\kappa(AM^{-1})$, for some fixed number γ, or similarly $\sigma_{min}(H_{m+1,m}) \approx \gamma\sigma_{min}(AM^{-1})$. While this may appear as an oversimplification, we are justified in part because the bounds (8) and (9) are very far from being tight.

In many problems though, the value of K_m may not be known in advance, or it may be hard to estimate, and we can just try some number, say 1, and decrease it until a good convergence behavior is achieved. One could also use the information from a first run, to estimate a value of K_m. In our preliminary experiments, reported in the next section, we have used the value of $K_m = 1$.

4 Numerical Experiments

We present numerical experiments on finite difference discretizations of two partial differential equations with Dirichlet boundary conditions on the two-dimensional unit square: the Laplacian $-\Delta u = f$, and a convection diffusion equation $-\Delta u + b.\nabla u = f$, with $b^T = [10, 20]$, where upwind differences are used, and the components of f are random, uniformly distributed between 0 and 1. We use an uniform discretization in each direction of 128 points, so the matrices are of order 16129, i.e., 16129 nodes in the grid. We partition the grid into 8×8 subdomains. In Table 1 we report experiments with varying degree of overlap: no overlap (0), one or two lines of overlap (1,2). Our (global) tolerance is $\varepsilon = 10^{-6}$. We compare the performance of using a fixed inner tolerance in each local solve, $\varepsilon_k = 10^{-4}$ for $k = 1, \ldots$, with the dynamic choice (10) using $K = K_m = 1$. We remark that both of these strategies correspond to varying the degree of inexactness and are expressed by the preconditioner (4). We run our experiments with the Additive Schwarz preconditioner (4) (ASM) and with weighted additive Schwarz preconditioner with harmonic extension (WASH). We have used a minimum of five (inner) iterations in each of the local solvers. We report the average number of inner iterations, which in this case well reflects the total work in each case, and in parenthesis the number of outer FGMRES iterations needed for convergence.

It can be appreciated from Table 1 that the proposed dynamic strategy for the inexact local solvers can reach the same (outer) tolerance using up to 20% less work. We point out that we have used the same value of $K = 1$ for all overlaps, although the preconditioners certainly change. A better estimate of K as a function of the overlap is expected to produce better results. We also mention that both the fixed inner tolerance and the dynamically chosen one usually require less storage than the exact local solvers (1) and (3).

Table 1. Average number of inner iterations (and number of outer iterations). Fixed or dynamic inner tolerance ($K = 1$).

	problem	Laplacian			Conv. Diff.		
	overlap	0	1	2	0	1	2
ASM	Fixed 10^{-4}	1923(64)	1536(46)	1388(38)	1825(60)	1458(43)	1295(35)
	Dynamic	1557(73)	1316(60)	1201(53)	1762(66)	1434(51)	1288(44)
WASH	Fixed 10^{-4}	1692(56)	1317(40)	1100(31)	1601(53)	1220(37)	1020(29)
	Dynamic	1387(61)	1089(45)	948(38)	1570(56)	1216(40)	1060(35)

Acknowledgments

The first author was supported in part by CNPQ (Brazil) under grant 305539/2003-8 and by the U.S. National Science Foundation under grant CGR 9984404. The second author was supported in part by the U.S. National Science Foundation under grant DMS-0207525.

References

1. A. Bouras and V. Frayseé, *Inexact matrix-vector products in Krylov methods for solving linear systems: a relaxation strategy*, SIAM J. Matrix Anal. Appl., 26 (2005), pp. 660–678.
2. A. Bouras, V. Frayseé, and L. Giraud, *A relaxation strategy for inner–outer linear solvers in domain decomposition methods*, Tech. Rep. TR/PA/00/17, CERFACS, Toulouse, France, 2000.
3. X.-C. Cai and M. Sarkis, *A restricted additive Schwarz preconditioner for general sparse linear systems*, SIAM J. Sci. Comput., 21 (1999), pp. 792–797.
4. A. Frommer and D. B. Szyld, *Weighted max norms, splittings, and overlapping additive Schwarz iterations*, Numerische Mathematik, 83 (1999), pp. 259–278.
5. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.
6. Y. Saad, *A flexible inner-outer preconditioned GMRES algorithm*, SIAM J. Scientific Comput., 14 (1993), pp. 461–469.
7. Y. Saad and M. H. Schultz, *GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems*, SIAM J. Sci. Stat. Comp., 7 (1986), pp. 856–869.
8. V. Simoncini and D. B. Szyld, *Theory of inexact Krylov subspace methods and applications to scientific computing*, SIAM J. Sci. Comput., 25 (2003), pp. 454–477.
9. B. F. Smith, P. E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
10. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

MINISYMPOSIUM 7: FETI and Neumann-Neumann Methods with Primal Constraints

Organizers: Axel Klawonn[1] and Kendall Pierson[2]

[1] University of Duisburg-Essen axel.klawonn@uni-essen.de
[2] Sandia National Laboratories khpiers@sandia.gov

FETI and Neumann-Neumann iterative substructuring algorithms are among the best known and most severely tested domain decomposition methods. Most of the recent developments are on methods with primal constraints, namely dual-primal FETI and balancing domain decomposition with constraints (BDDC) algorithms. In this minisymposium, we bring together active researchers in the field of dual-primal FETI and BDDC algorithms coming from the fields of numerical analysis, scientific computing and computational mechanics. The talks will be on new algorithmic developments and new theoretical results as well as on large-scale computational applications.

Parallel Scalability of a FETI–DP Mortar Method for Problems with Discontinuous Coefficients

Nina Dokeva and Wlodek Proskurowski

Department of Mathematics, University of Southern California, Los Angeles, CA 90089–1113, USA. dokeva,proskuro@usc.edu

Summary. We consider elliptic problems with discontinuous coefficients discretized by finite elements on non-matching triangulations across the interface using the mortar technique. The resulting discrete problem is solved by a FETI–DP method using a preconditioner with a special scaling described in a forthcoming paper by Dokeva, Dryja and Proskurowski. Experiments performed on up to a thousand processors show that this FETI–DP mortar method exhibits good parallel scalability.

1 Introduction

Parallelization of finite element algorithms enables one to solve problems with a large number of degrees of freedom in a reasonable time, which becomes possible if the method is scalable.

We adopt here the definition of scalability of [3] and [4]: solving n-times larger problem using an n-times larger number of processors in nearly constant cpu time. Domain decomposition algorithms using FETI-DP solvers ([7], [8], [9], [10]) have been demonstrated to provide scalable performance on massively parallel processors, see [4] and the references therein.

The aim of this paper is to experimentally demonstrate that a scalable performance on hundreds of processors can be achieved for a mortar discretization using FETI-DP solvers described in [5] and [6].

In view of the page limitation, Section 2 describing the FETI-DP method and preconditioner is abbreviated to a minimum. For a complete presentation refer to [5]. Section 3 contains the main results.

2 FETI-DP equation and preconditioner

We consider the following differential problem.

Find $u^* \in H_0^1(\Omega)$ such that

$$a(u^*, v) = f(v), \quad v \in H_0^1(\Omega), \tag{1}$$

where

$$a(u,v) = (\rho(x)\nabla u, \nabla u)_{L^2(\Omega)}, \quad f(v) = (f, v)_{L^2(\Omega)}.$$

We assume that Ω is a polygonal region and $\overline{\Omega} = \bigcup_{i=1}^{N} \overline{\Omega}_i$, Ω_i are disjoint polygonal subregions, $\rho(x) = \rho_i$ is a positive constant on Ω_i and $f \in L^2(\Omega)$. We solve (1) by the finite element method on geometrically conforming non–matching triangulation across $\partial\Omega_i$. To describe a discrete problem the mortar technique is used.

We impose on Ω_i a triangulation with triangular elements and a parameter h_i. The resulting triangulation of Ω is non-matching across $\partial\Omega_i$. Let $X_i(\Omega_i)$ be a finite element space of piecewise linear continuous functions defined on the triangulation introduced. We assume that the functions of $X_i(\Omega_i)$ vanish on $\partial\Omega_i \cap \partial\Omega$.

Let $X^h(\Omega) = X_1(\Omega_1) \times \ldots \times X_N(\Omega_N)$ and let $V^h(\Omega)$ be a subspace of $X^h(\Omega)$ of functions which satisfy the mortar condition

$$\int_{\delta_m} (u_i - u_j)\psi ds = 0, \quad \psi \in M(\delta_m). \tag{2}$$

Here, $u_i \in X_i(\Omega_i)$ and $u_j \in X_j(\Omega_j)$ on Γ_{ij}, an edge common to Ω_i and Ω_j and $M(\delta_m)$ is a space of test (mortar) functions.

Let $\Gamma_{ij} = \partial\Omega_i \cap \partial\Omega_j$ be a common edge of two substructures Ω_i and Ω_j. Let Γ_{ij} as an edge of Ω_i be denoted by $\gamma_{m(i)}$ and called *mortar* (master), and let Γ_{ij} as an edge of Ω_j be denoted by $\delta_{m(j)}$ and called *non-mortar* (slave). Denote by $W_j\left(\delta_{m(j)}\right)$ the restriction of $X_j(\Omega_j)$ to $\delta_{m(j)}$.

Using the nodal basis functions $\varphi^{(l)}_{\delta_{m(i)}} \in W_i\left(\delta_{m(i)}\right)$, $\varphi^{(k)}_{\gamma_{m(j)}} \in W_j\left(\gamma_{m(j)}\right)$ and $\psi^{(p)}_{\delta_{m(i)}} \in M\left(\delta_{m(i)}\right)$, the matrix formulation of (2) is

$$B_{\delta_{m(i)}} u_{i\delta_{m(i)}} - B_{\gamma_{m(j)}} u_{j\gamma_{m(j)}} = 0, \tag{3}$$

where $u_{i\delta_{m(i)}}$ and $u_{j\gamma_{m(j)}}$ are vectors which represent $u_i\big|_{\delta_{m(i)}} \in W_i\left(\delta_{m(i)}\right)$ and $u_j\big|_{\gamma_{m(j)}} \in W_j\left(\gamma_{m(j)}\right)$, and

$$B_{\delta_{m(i)}} = \left\{ (\psi^{(p)}_{\delta_{m(i)}}, \varphi^{(k)}_{\delta_{m(i)}})_{L^2(\delta_{m(i)})} \right\}, \; p = 1, \ldots, n_{\delta(i)}, \; k = 0, \ldots, n_{\delta(i)} + 1,$$

$$B_{\gamma_{m(j)}} = \left\{ (\psi^{(p)}_{\delta_{m(i)}}, \varphi^{(l)}_{\gamma_{m(j)}})_{L^2(\gamma_{m(j)})} \right\}, \; p = 1, \ldots, n_{\delta(i)}, \; l = 0, \ldots, n_{\gamma_{(j)}} + 1.$$

We rewrite the discrete problem for (1) in V^h as a saddle-point problem using Lagrange multipliers, λ. Its solution is $(u_h^*, \lambda_h^*) \in \widetilde{X}^h(\Omega) \times M(\Gamma)$, where $\widetilde{X}^h(\Omega)$ denotes a subspace of $X^h(\Omega)$ of functions which are continuous at vertices common to the substructures. We partition $u_h^* = \left(u^{(i)}, u^{(c)}, u^{(r)}\right)$ into vectors containing the interior nodal points of Ω_l, the vertices of Ω_l, and the remaining nodal points of $\partial\Omega_l \backslash \partial\Omega$, respectively.

Let $K^{(l)}$ be the stiffness matrix of $a_l(\,\cdot\,,\cdot\,)$. It is represented as

$$K^{(l)} = \begin{pmatrix} K_{ii}^{(l)} & K_{ic}^{(l)} & K_{ir}^{(l)} \\ K_{ci}^{(l)} & K_{cc}^{(l)} & K_{cr}^{(l)} \\ K_{ri}^{(l)} & K_{rc}^{(l)} & K_{rr}^{(l)} \end{pmatrix}, \tag{4}$$

where the rows correspond to the interior unknowns, its vertices and its edges.

Using this notation and the assumption of continuity of u_h^* at the vertices of $\partial\Omega_l$, the saddle point problem can be written as

$$\begin{pmatrix} K_{ii} & K_{ic} & K_{ir} & 0 \\ K_{ci} & \widetilde{K}_{cc} & K_{cr} & B_c^T \\ K_{ri} & K_{rc} & K_{rr} & B_r^T \\ 0 & B_c & B_r & 0 \end{pmatrix} \begin{pmatrix} u^{(i)} \\ u^{(c)} \\ u^{(r)} \\ \widetilde{\lambda}^* \end{pmatrix} = \begin{pmatrix} f^{(i)} \\ f^{(c)} \\ f^{(r)} \\ 0 \end{pmatrix}. \tag{5}$$

Here, the matrices K_{ii} and K_{rr} are diagonal block-matrices of $K_{ii}^{(l)}$ and $K_{rr}^{(l)}$, while $\widetilde{K}_{cc}$ is a diagonal block built by matrices $K_{cc}^{(l)}$ using the fact that $u^{(c)}$ are the same at the common vertices of the substructures. The mortar condition is represented by the global matrix $B = (B_c, B_r)$.

In the system (5) we eliminate the unknowns $u^{(i)}$ and $u^{(c)}$ to obtain

$$\begin{pmatrix} \widetilde{S} & \widetilde{B}^T \\ \widetilde{B} & \widetilde{S}_{cc} \end{pmatrix} \begin{pmatrix} u^{(r)} \\ \widetilde{\lambda}^* \end{pmatrix} = \begin{pmatrix} \widetilde{f}_r \\ \widetilde{f}_c \end{pmatrix}, \tag{6}$$

where (since $K_{ic} = 0 = K_{ci}$ in the case of triangular elements and a piecewise linear continuous finite element space used in the implementation):

$$\widetilde{S} = K_{rr} - K_{ri}K_{ii}^{-1}K_{ir} - K_{rc}\widetilde{K}_{cc}^{-1}K_{cr}, \quad \widetilde{f}_r = f^{(r)} - K_{ri}K_{ii}^{-1}f^{(i)} - K_{rc}\widetilde{K}_{cc}^{-1}f^{(c)}$$

$$\widetilde{B} = B_r - B_c\widetilde{K}_{cc}^{-1}K_{cr}, \quad \widetilde{S}_{cc} = -B_c\widetilde{K}_{cc}^{-1}B_c^T, \quad \text{and} \quad \widetilde{f}_c = -B_c\widetilde{K}_{cc}^{-1}f_c.$$

We next eliminate the unknown $u^{(r)}$ to get for $\widetilde{\lambda}^* \in M(\Gamma)$

$$F\widetilde{\lambda}^* = d, \tag{7}$$

where

$$F = \widetilde{B}\widetilde{S}^{-1}\widetilde{B}^T - \widetilde{S}_{cc}, \quad \text{and} \quad d = \widetilde{B}\widetilde{S}^{-1}\widetilde{f}_r - \widetilde{f}_c. \tag{8}$$

This is the FETI-DP equation for the Lagrange multipliers. Since F is positive definite, the problem has a unique solution. This problem can be solved by conjugate gradient iterations with a preconditioner discussed below.

Let $S^{(l)}$ denote the Schur complement of $K^{(l)}$, see (4), with respect to unknowns at the nodal points of $\partial\Omega_l$. This matrix is represented as

$$S^{(l)} = \begin{pmatrix} S_{rr}^{(l)} & S_{rc}^{(l)} \\ S_{cr}^{(l)} & S_{cc}^{(l)} \end{pmatrix}, \tag{9}$$

where the second row corresponds to unknowns at the vertices of $\partial\Omega_l$ while the first one corresponds to the remaining unknowns of $\partial\Omega_l$. Note that B_r is a matrix obtained from B defined on functions with zero values at the vertices of Ω_l and let

$$S_{rr} = \operatorname{diag}\left\{S_{rr}^{(l)}\right\}_{l=1}^{N}, \; S_{cc} = \operatorname{diag}\left\{S_{cc}^{(l)}\right\}_{l=1}^{N}, \; S_{cr} = \left(S_{cr}^{(1)}, \ldots, S_{cr}^{(N)}\right). \qquad (10)$$

We employ a special scaling appropriate for problems with discontinuous coefficients. The preconditioner M for (7) is defined as, see [5]

$$M^{-1} = \widehat{B}_r \widehat{S}_{rr} \widehat{B}_r^T, \qquad (11)$$

where $\widehat{S}_{rr} = \operatorname{diag}\left\{\widehat{S}_{rr}^{(i)}\right\}_{i=1}^{N}$, $\widehat{S}_{rr}^{(i)} = S_{rr}^{(i)}$ for $\rho_i = 1$ and we define

$\widehat{B}|_{\delta_{m(i)}} = \left(\rho_i^{1/2} I_{\delta_{m(i)}}, -\dfrac{h_{\delta_{m(i)}}}{h_{\gamma_{m(j)}}} \dfrac{\rho_i}{\rho_j} \rho_i^{1/2} B_{\delta_{m(i)}}^{-1} B_{\gamma_{m(j)}}\right)$, for $\delta_{m(i)} \subset \partial\Omega_i$, $i = 1, \ldots, N$; $h_{\delta_{m(i)}}$ and $h_{\gamma_{m(j)}}$ are the mesh parameters on $\delta_{m(i)}$ and $\gamma_{m(j)}$, respectively.

We have, following [5]

Theorem 1. *Let the mortar side be chosen where the coefficient ρ_i is larger. Then for $\lambda \in M(\Gamma)$ the following holds*

$$c_0 \left(1 + \log \frac{H}{h}\right)^{-2} \langle M\lambda, \lambda\rangle \leq \langle F\lambda, \lambda\rangle \leq c_1 \left(1 + \log \frac{H}{h}\right)^{2} \langle M\lambda, \lambda\rangle, \qquad (12)$$

where c_0 and c_1 are positive constants independent of h_i, H_i, and the jumps of ρ_i; $h = \min_i h_i$, $H = \max_i H_i$.

This estimate allows us to achieve numerical scalability, an essential ingredient in a successful parallel implementation.

3 Parallel implementation and results

Our parallel implementation problem is divided into three types of tasks: solvers on the subdomains (with different meshes of discretization) which run individually and in parallel, a problem on the interfaces between the subdomains which can be solved in parallel with only a modest amount of global communication, and a "coarse" problem on the vertices between the subdomains which is a global task. A proper implementation of the coarse problem is crucial when the number of processors/subdomains is large.

We discuss some details of the implementation and present experimental results demonstrating that this method is well scalable. The numerical experiments were performed on up to 1024 processors provided by the University of Southern California Center for High Performance Computing and Communications (`http://www.usc.edu/hpcc`). All jobs were run on identically configured nodes equipped with dual Intel Pentium 4 Xeon 3.06 GHz processors, 2 GB of RAM and low latency Myrinet networking. Our code was written in C and MPI, using the PETSc toolkit (see [2]) which interfaces many different solvers.

The test example for our experiments is the weak formulation of

$$-\operatorname{div}(\rho(x)\nabla u) = f(x) \text{ in } \Omega, \qquad (13)$$

with Dirichlet boundary conditions on $\partial\Omega$, where $\Omega = (0,1) \times (0,1)$ is a union of $N = n^2$ disjoint square subregions Ω_i, $i = 1, \ldots, N$ and $\rho(x) = \rho_i$ is a positive constant in each Ω_i. The coefficients $\rho(x)$ are chosen larger on the mortar sides of the interfaces, see Theorem 1.

The distribution of the coefficients ρ_i and grids h_i in Ω_i, $i = 1, \ldots, 4$ with a maximum mesh ratio 8 : 1 used in our tests (for larger number of subregions, this pattern of coefficients is repeated) is here with $h = \dfrac{1}{32n}$:

$$\begin{pmatrix} 1e6 & 1e4 \\ 1e2 & 1 \end{pmatrix}, \quad \begin{pmatrix} h/8 & h/4 \\ h/2 & h \end{pmatrix}. \tag{14}$$

Each of the N processors works on a given subdomain and communicates mostly with the processors working on the neighboring subdomains.

For the subdomain solvers, we employ a symmetric block sparse Cholesky solver provided by the SPOOLES library (see [1]). The matrices are factored during the first solve and afterwards only a forward and backward substitutions are needed.

In each preconditioned conjugate gradient (PCG) iteration to solve the FETI-DP equation (7) for the Lagrange multipliers, there are two main operations:

1. multiplication by the preconditioner $M^{-1} = \widehat{B}_r \widehat{S}_{rr} \widehat{B}_r^T$ which involves solving N Dirichlet problems that are uncoupled, and some operations on the interfaces between the neighboring subdomains.

2. multiplication by $F = \widetilde{B}\widetilde{S}^{-1}\widetilde{B}^T - \widetilde{S}_{cc}$ which involves solving N coupled Neumann problems connected through the vertices.

The latter task involves solving a system with the global stiffness matrix K, see (5), of the form:

$$\begin{pmatrix} K_{ii} & 0 & K_{ir} \\ 0 & \widetilde{K}_{cc} & K_{cr} \\ K_{ri} & K_{rc} & K_{rr} \end{pmatrix} \begin{pmatrix} v_i \\ v_c \\ v_r \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ p \end{pmatrix}. \tag{15}$$

Its Schur complement matrix C with respect to the vertices is

$$C = \widetilde{K}_{cc} - (0,\ K_{cr}) \begin{pmatrix} K_{ii} & K_{ir} \\ K_{ri} & K_{rr} \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ K_{rc} \end{pmatrix}. \tag{16}$$

C is a sparse, block tridiagonal $(n-1)^2 \times (n-1)^2$ matrix which has 9 nonzero diagonals. Solving a "coarse" problem with C is a global task while the subdomain solvers are local and run in parallel.

Proper implementation of the coarse system solving is important for the scalability especially when the number of processors/subdomains, N is large. Without assembling C, the coarse system could be solved iteratively (for example, with PCG using symmetric Gauss-Seidel preconditioner). Since the cpu cost then depends on N, it is preferable to assemble C.

We implemented two approaches discussed in [4]. In the case of relatively small C studied here one can invert C in parallel by duplicating it across a group of processors so that each computes a column of C^{-1} by a direct solver, for which we employed SPOOLES.

When C is larger the above approach may not be efficient or even possible; in that case one can use distributed storage for C and then a parallel direct solver. In a second implementation, we employed the block sparse Cholesky solver from the MUMPS package (see [11] and [12]) interfaced through PETSc. For simplicity, the matrix C was stored on $n-1$ or $(n-1)^2$ processors, with the first choice yielding better performance.

In the tests run on up to (the maximum available to us) $N = 1024$ processors the two implementations performed almost identically. In Table 1 and Fig. 1 and 2 we present results from our first implementation when the coarse problem is solved by computing columns of C^{-1}.

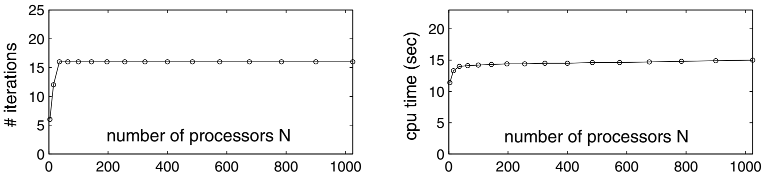


Fig. 1. Iterations and execution time vs number of processors.

Fig. 1 shows that the number of PCG iterations remains constant after $N = 36$ when the number of subdomains/processors is increased. The graph of the execution time (on the right) has a similar pattern. Although the number of degrees of freedom is increasing, the cpu time remains almost constant, see Table 1.

N	# it	d.o.f.	cpu time
4	6	87 037	11.4
16	13	350 057	13.3
36	16	789 061	14.0
64	16	1 404 049	14.1
100	16	2 195 021	14.2
144	16	3 161 977	14.3
196	16	4 304 917	14.4
256	16	5 623 841	14.4
324	16	7 118 749	14.5
400	16	8 789 641	14.5
484	16	10 636 517	14.6
576	16	12 659 377	14.6
676	16	14 858 221	14.7
784	16	17 233 049	14.8
900	16	19 783 861	14.9
1024	16	22 510 657	15.0

Table 1. Number of iterations, number of degrees of freedom and execution time in seconds.

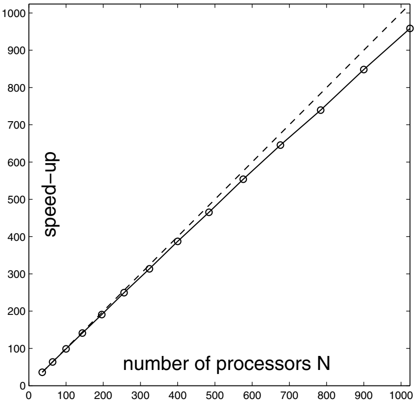


Fig. 2. Speed-up.

Fig. 2 shows the speed-up of the algorithm, where the dashed line represents the ideal (linear) and the solid line the actual speed-up, respectively.

We adopt the definition of the speed-up of [3]. Here, it is adjusted to $N_0 = 36$ as a reference point, after which the number of iterations remains constant, see Table 1:

$$Sp = \frac{36 \times T_{36}}{T_{N_p}} \times \frac{N_{dof_{N_s}}}{N_{dof_{36}}},$$

where T_{36} and T_{N_p} denote the CPU time corresponding to 36 and N_p processors, respectively, and $N_{dof_{36}}$ and $N_{dof_{N_s}}$ denote the number of d.o.f. of the global problems corresponding to 36 and N_s subdomains, respectively.

This definition accounts both for the numerical and parallel scalability.

4 Conclusions

In this paper we study the parallel performance of the FETI–DP mortar preconditioner developed in [5] for elliptic 2D problems with discontinuous coefficients. The computational evidence presented illustrates good scalability of the method (an almost linear speed-up).

We would like to thank Max Dryja for his collaboration throughout. The first author would like to thank Panayot Vassilevski for the guidance during her summer internship at the Lawrence Livermore National Laboratory. The USC Center for High Performance Computing and Communications (HPCC) is acknowledged for generously providing us with the use of its Linux cluster.

References

1. C. Ashcraft and R. G. Grimes, *SPOOLES: An object-oriented sparse matrix library*, in Proceedings of the Ninth SIAM Conference on Parallel Processing for Scientific Computing, 1999.
2. S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, L. C. McInnes, and B. F. Smith, *PETSc home page*. http://www.mcs.anl.gov/petsc, 2001.
3. M. Bhardwaj, D. Day, C. Farhat, M. Lesoinne, K. Pierson, and D. Rixen, *Application of the FETI method to ASCI problems - scalability results on one thousand processors and discussion of highly heterogeneous problems*, Int. J. Numer. Meth. Engrg., 47 (2000), pp. 513–535.
4. M. Bhardwaj, K. Pierson, G. Reese, T. Walsh, D. Day, K. Alvin, J. Peery, C. Farhat, and M. Lesoinne, *Salinas: A scalable software for high-performance structural and solid mechanics simulations*, in Proceedings of 2002 ACM/IEEE Conference on Supercomputing, 2002, pp. 1–19. Gordon Bell Award.
5. 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 (2006), pp. 283–299.
6. M. Dryja and O. B. Widlund, *A FETI-DP method for a mortar discretization of elliptic problems*, vol. 23 of Lecture Notes in Computational Science and Engineering, Springer, 2002, pp. 41–52.
7. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.

8. C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numer. Lin. Alg. Appl., 7 (2000), pp. 687–714.
9. 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 (2002), pp. 159–179.
10. J. Mandel and R. Tezaur, *On the convergence of a dual-primal substructuring method*, Numer. Math., 88 (2001), pp. 543–558.
11. I. S. D. Patrick R. Amestoy and J.-Y. L'Excellent, *Multifrontal parallel distributed symmetric and unsymmetric solvers*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 501–520.
12. I. S. D. Patrick R. Amestoy, J.-Y. L'Excellent, and J. Koster, *A fully asynchronous multifrontal solver using distributed dynamic scheduling*, SIAM J. Matrix Anal. Appl., 23 (2001), pp. 15–41.

Neumann-Neumann Algorithms (Two and Three Levels) for Finite Element Elliptic Problems with Discontinuous Coefficients on Fine Triangulation

Maksymilian Dryja[1]* and Olof Widlund[2]

[1] Department of Mathematics, Informatics and Mechanics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland. dryja@mimuw.edu.pl
[2] Courant Institute of Mathematical Sciences, New York University, 252 Mercer Street, New York, NY 10012, USA. widlund@courant.nyu.edu

1 Introduction

We design and analyze Neumann-Neumann (N-N) algorithms for elliptic problems imposed in the 2-D polygonal region Ω with discontinuous coefficients on fine triangulation. We first discuss the two-level N-N algorithm and then we extend it to three levels. The coefficients ϱ_i given on the coarse Ω_i triangulation are discontinuous functions with respect to a fine triangulation in Ω_i. We assume, for simplicity of representation, that $\varrho_i = \varrho_i^k$ is constant on τ_i^k triangles of the fine triangulation in Ω_i. The resulting fine triangulation on $\bar{\Omega} = \cup_i \bar{\Omega}_i$ is matching. We assume that $\bar{\varrho}_i \sim \varrho_i^k$ for each $\tau_i^k \subset \Omega_i$ where $\bar{\varrho}_i = \dfrac{1}{|\Omega_i|} \sum\limits_{\tau_i^k \subset \Omega_i} |\tau_i^k| \varrho_i^k$. It means that ϱ_i^k are moderated in Ω_i, i.e. $\min\limits_k \varrho_i^k$ and $\max\limits_k \varrho_i^k$ are the same order. Under this assumption we prove that the two-level N-N algorithm is almost optimal and its rate of convergence is independent of the parameters of coarse and fine triangulation and the jumps of ϱ_i across $\partial\Omega_i$.

This result is extended to the three-level N-N algorithm defined by three triangulation of Ω: super-coarse $\{\Omega_i\}$, coarse $\{\Omega_i^j\}$ with $\Omega_i^j \subset \Omega_i$ and fine with $\tau_{ij}^k \subset \Omega_i^j$ on which the problem is discritized. The discontinuities of ϱ_i are given on the coarse triangulation. The three-level N-N algorithm in each iteration reduces to solving a global problem on $\{\Omega_i\}$, coarse local problems on $\{\Omega_i^j\}$ in each Ω_i and local problems on the fine triangulation in each Ω_i^j. The global and coarse local problems are defined on the coarse triangulation. The rate of convergence of the three-level

*This work was partially supported by Polish Scientific Grant 2/P03A/005/24. The work was supported in part by U.S. Department of Energy under contract DE-FG02-92ER25127

N-N algorithm is proved under the assumption as above in Ω_i and Ω_i^j but it is independent of the jumps of ϱ_i across $\partial\Omega_i$.

The methods discussed in this paper can be generalized to elliptic problems with discontinuous coefficients in three dimensions.

The N-N algorithms (two-level) are well understood for conforming finite element discretization of elliptic problems with coefficients ϱ which are constants on each Ω_i, the coarse triangulation of Ω, see [2], [1] and the books [4] and [3], and literature theirin. The first goal of this paper is to generalize the method to solving the problem with coefficients ϱ_i which are also discontinuous in each Ω_i. The second goal is to design and analyze the three-level method for solving the problem. To our knowledge the N-N algorithms designed and analyzed in this paper for solving FE discretization of elliptic problems with discontinuous coefficients also in Ω_i have not previously been discussed in the literature.

The paper is organized as follows. In Section 2, the differential problem and its FE discretization are described. Section 3 is devoted to designing and analyzing the two-level N-N algorithm. In Section 4, the three-level N-N algorithm is designed and analyzed.

2 Differential and discrete problems

Find $u^* \in H_0^1(\Omega)$ such that

$$a_\varrho(u^*, v) = f(v), \quad v \in H_0^1(\Omega) \tag{1}$$

where

$$a_\varrho(u, v) = \sum_{i=1}^{N} \int_{\Omega_i} \varrho_i \nabla u \nabla v dx, \quad f(v) = \int_\Omega f v dx \tag{2}$$

and Ω is a polygonal region in R^2, Ω_i are polygons, $\bar{\Omega} = \bigcup \bar{\Omega}_i; \varrho_i(x) \geq \varrho_i^0 > 0, f \in L^2(\Omega)$.

We assume that $\{\Omega_i\}$ forms a coarse triangulation with a parameter H. We introduce a fine triangulation in Ω_i with triangles τ_i^k. The resulting triangulation on Ω with parameter h have to be matching. The coarse and fine triangulation by the assumption are shape regular in the common sense of FE theory. We assume, for simplicity of presentation, that $\varrho_i(x) = \varrho_i^k > 0$ on $\tau_i^k \subset \bar{\Omega}_i$ where ϱ_i^k are constants.

Let $V_h(\Omega)$ be a finite element space of piecewise linear continuous functions on the fine triangulation with zero values on $\partial\Omega$. The discrete problem is of the form:

Find $u_h^* \in V_h(\Omega)$ such that

$$a_\varrho(u_h^*, v_h) = f(v_h), \quad v_h \in V_h(\Omega). \tag{3}$$

3 Two level Neumann-Neumann algorithm

The problem (3) is reduced to the Schur complement problem of the form:

$$S^\varrho u_B = b_B \tag{4}$$

where

$$(S^{\varrho}u_B, v_B) = \sum_{i=1}^{N}(S_i^{\varrho}u_B^{(i)}, v_B^{(i)}).$$

Here $u \in V_h(\Omega)$ on $\bar{\Omega}_i$ is denoted by $u^{(i)}$ and decomposed into $u_I^{(i)}$ and $u_B^{(i)}$ which correspond to interior and boundary values of $u^{(i)}$, respectively;

$$(S_i^{\varrho}u_B^{(i)}, u_B^{(i)}) = a_i^{\varrho}(\mathcal{H}_i u_B^{(i)}, \mathcal{H}_i u_B^{(i)})$$

where the discrete harmonic extension $\mathcal{H}_i$ is understood in the sense of

$$a_i^{\varrho}(u,v) = \sum_{\tau_i^k \subset \Omega_i} \varrho_i^k (\nabla u, \nabla v)_{L^2(\tau_i^k)},$$

the original form restricted to Ω_i.

We will use also the discrete harmonic functions $\hat{\mathcal{H}}_i u$ in the sense of

$$\hat{a}_i(u,v) \equiv \int_{\Omega_i} \nabla u \nabla v dx + \frac{1}{H_i^2}\int_{\Omega_i} uv dx$$

where H_i is a diameter of Ω_i.

Let

$$V_h(\Omega) = V_h^{\mathcal{H}}(\Omega) \oplus V_h^{P}(\Omega)$$

where $\mathcal{H} = \{\mathcal{H}_i\},\ P = \{P_i\}$ and P_i is the projection in the sense of $a_i^{\varrho}(.,.)$, i.e.

$$u_{|\Omega_i} = \mathcal{H}_i u + P_i u.$$

The problem (4) is considered in the space $V_h^{\mathcal{H}}(\Omega)$ which below is denoted by $V_h(\Gamma)$ where $\Gamma = (\bigcup_i \partial\Omega_i)\backslash\partial\Omega$.

For (4) we design a Neumann-Neumann (N-N) algorithm (two-levels) as ASM. For that the general theory of ASMs is used, see [1] or the books [4] and [3].

3.1 Decomposition of $\mathbf{V_h(\Gamma)}$

This is of the form:

$$V_h(\Gamma) = V_0(\Gamma) + V_1(\Gamma) + \cdots + V_N(\Gamma).$$

The spaces $V_i, i = 1, \cdots, N$, are defined as

$$V_i(\Gamma) = \{u \in V_h(\Gamma) : u(x) = 0, \quad x \in \Gamma_h \backslash \partial\Omega_{ih}\}$$

where Γ_h and $\partial\Omega_{ih}$ are the sets of nodal points of Γ and $\partial\Omega_i$, respectively. We point out that here the discrete harmonic functions are in the sense of $a_i^{\varrho}(.,.)$. The space V_0 is defined as

$$V_0(\Gamma) = span\{I_h(\bar{\varrho}_i^{1/2}\bar{\mu}_i^{\dagger})\}_{i \in \mathcal{N}_I}, \qquad \bar{\varrho}_i = \frac{1}{|\Omega_i|}\sum_{\tau_i^k \subset \bar{\Omega}_i} |\tau_i^k| \varrho_i^k,$$

where

$$\bar{\mu}_i(x) = \sum_j \bar{\varrho}_j^{1/2}, \quad x \in \partial\Omega_{h,i}; \quad \bar{\mu}_i = 0, \quad \Gamma_h \backslash \partial\Omega_{ih}.$$

Here I_h is the linear interpolant on the fine triangulation and the sum of j is taken over the values of j for which $x \in \partial\Omega_j$, and $\mathcal{N}_I$ is the set of Ω_i which do not touch $\partial\Omega$. Note that the harmonic extension of $I_h(\bar{\varrho}_i^{1/2}\bar{\mu}_i^\dagger)$ is in the sense of $a_i^{(\varrho)}(.,.)$. For simplicity of presentation, we assume that any Ω_i for $i \in \mathcal{N}_B$ touches $\partial\Omega$ by an edge where $\mathcal{N}_B$ is the set of Ω_i which touch $\partial\Omega$. This guarantees that $V_0(\Gamma) \subset V_h(\Gamma)$.

Remark 1. The space V_0 can be extended by adding basis functions corresponding to Ω_i for $i \in \mathcal{N}_B$ with modified $\mu_i^\dagger$. It can be done in the same way as for the standard case, see [1] for details.

3.2 Inexact solver

Let for $i = 1, \cdots, N$ and $u, v \in V_i(\Gamma)$

$$b_i(u, v) \equiv \hat{a}_i(\hat{\mathcal{H}}_i I_h(\bar{\mu}_i u), \hat{\mathcal{H}}_i I_h(\bar{\mu}_i v)).$$

For $i = 0$ and $u, v \in V_0(\Gamma)$

$$b_0(u, v) \equiv (1 + log\frac{H}{h})^{-1} a_\varrho(u, v).$$

3.3 The equation

Let $T_i : V_h(\Gamma) \to V_i(\Gamma), \;\; i = 0, \cdots, N$, be defined as

$$b_i(T_i u, v) = a_\varrho(u, v), \qquad v \in V_i(\Gamma)$$

and

$$T \equiv T_0 + T_1 + \cdots + T_N.$$

The problem (4) is replaced by

$$Tu_h^* = g_h$$

where $g_h = \sum_{i=0}^{N} g_i, \;\; g_i = T_i u_h^*$. Note that to find g_i we do not need to know u_h^*, the solution of (4).

Theorem 1. *Let for $i = 1, \cdots, N$,*

$$\bar{\varrho}_i \sim \varrho_i^k \; for \, \tau_i^k \subset \Omega_i.$$

Then for $u \in V_h(\Gamma)$

$$C_0 S^\varrho(u, u) \le S^\varrho(Tu, u) \le C_1(1 + log\frac{H}{h})^2 S^\varrho(u, u)$$

where C_0 and C_1 are positive constants independent of h and H, and the jumps of coefficients across $\partial\Omega_i$.

4 Three level Neumann-Neumann algorithm

In this section we design the three-level N-N algorithm for solving the problem (3) defined by three-level triangulation of Ω: supercoarse $\{\Omega_i\}$ with h_{sc} parameter, coarse $\{\Omega_i^j\}$ with $\Omega_i^j \subset \Omega_i$ and parameter h_c and fine $\{\tau_{ij}^k\}$ with $\tau_{ij}^k \subset \Omega_i^j$ and h parameter. Thus $\bar{\Omega} = \bigcup_{i=1}^{N} \bar{\Omega}_i$, $\bar{\Omega}_i = \bigcup_{j=1}^{N_i} \bar{\Omega}_i^j$, $\bar{\Omega}_i^j = \bigcup_k \bar{\tau}_{ij}^k$ where Ω_i are polygons while Ω_i^j and τ_{ij}^k are triangles. We assume that these three triangulation are shape regular in the common sense of FE theory.

The problem (3) is discretized on the fine triangulation with elements τ_{ij}^k and the coefficients ϱ_{ij}^k on these elements. We assume, that $\varrho_{ij}^k = \varrho_i^j$ for all $\tau_{ij}^k \subset \Omega_i^j$ and they are positive constants. If ϱ_{ij}^k are piecewise constants on the fine triangulation in Ω_i^j then ϱ_i^j is defined as the integral average of ϱ_{ij}^k over Ω_i^j. The Schur complement problem (4) is now defined on the $\{\partial\Omega_i^j\}$ triangulation and $V_h(\Gamma)$ is a space of discrete harmonic functions in each Ω_i^j, in the sense of $a_{ij}^{\varrho}(.,.)$, the restriction $a_{\varrho}(.,.)$ to Ω_i^j, with data on $\partial\Omega_i^j$; $\Gamma = (\bigcup_i \bigcup_j \partial\Omega_i^j)\backslash\partial\Omega$, $\Gamma_0 = (\bigcup_i \partial\Omega_i)\backslash\partial\Omega$.

The three-level N-N algorithm for solving (4) is designed and analyzed using the general theory of ASMs, see [1] or the books [4] and [3].

4.1 Decomposition of $\mathbf{V_h(\Gamma)}$

Let

$$V_h(\Gamma) = V_{00}(\Gamma_0) + \sum_{i=1}^{N}(V_{0i}^{\mathcal{H}}(\Gamma_0) + V_{0i}^{P}(\Gamma_0)) + \sum_{i=1}^{N}\sum_{j=1}^{N_i} V_{ij}(\Gamma). \qquad (5)$$

The spaces V_{ij}, $i = 1, \cdots, N$, $j = 1, \cdots, N_i$, are of the form:

$$V_{ij}(\Gamma) := \{v \in V_h(\Gamma) : v(x) = 0 \text{ at } x \in \Gamma_h\backslash\partial\Omega_{ih}^j\}$$

where Γ_h and $\partial\Omega_{ih}^j$ are the sets of nodal points of Γ and $\partial\Omega_i^j$, respectively.

To define $V_{00}, V_{0i}^{\mathcal{H}}$ and V_{0i}^{P} we introduce first two auxiliary spaces $V_0(\Gamma)$ and $V_0^{(c)}(\Gamma_0)$. Let

$$\mu_i^j(x) = \sum_{l,k}(\varrho_l^k)^{1/2}, \quad x \in \partial\Omega_{i,h}^j; \quad \mu_i^j = 0, \quad x \in \Gamma_h\backslash\partial\Omega_{ih}^j$$

where the sum is taken over substructures Ω_l^k, for which $x \in \partial\Omega_{l,h}^k$. Let us introduce

$$V_0(\Gamma) = span\{I_h((\varrho_i^j)^{1/2}(\mu_i^j)^{\dagger})\}, \;\; i \in \mathcal{N}_I^{(c)}, \;\; j \in \mathcal{N}_{I,i}^{(c)}.$$

Here $\mathcal{N}_I^{(c)}$ are the set of Ω_i which to not touch $\partial\Omega$ while $\mathcal{N}_{I,i}^{(c)}$ is the set of Ω_i^j in Ω_i. We assume here and below, for simplicity of presentation, that if Ω_i^j touches $\partial\Omega$ it touches its by an edge. We should point out that the function $I_h((\varrho_i^j)^{1/2}(\mu_i^j)^{\dagger})$ given on $\{\partial\Omega_l^k\}$ is extended to $\{\Omega_l^k\}$ as discrete harmonic in the sense of $a_{kl}^{\varrho}(.,.)$, the restriction $a_{\varrho}(.,.)$ to Ω_i^k. Note that $V_0(\Gamma) \subset V_h(\Gamma)$ is the coarse space in the case of the two-level N-N algorithm based on $\{\tau_{ij}^k\}$ and $\{\Omega_i^j\}$ triangulation.

Let I_c be the linear interpolant on the coarse triangulation with the parameter h_c. Let $V_0^{(c)}(\Gamma) = I_c V_0(\Gamma)$. Functions $V_0^{(c)}(\Gamma)$ are piecewise linear continuous on $\{\Omega_i^j\}$ and defined by values given at vertices of Ω_i^j. Thus the two-level decomposition of $V_h(\Gamma)$ is of the form

$$V_h(\Gamma) = V_0^{(c)}(\Gamma) + \sum_{i=1}^{N} \sum_{j=1}^{N_i} V_{ij}(\Gamma). \tag{6}$$

We now further decompose $V_0^{(c)}(\Gamma)$ to get the three-level decomposition of $V_h(\Gamma)$. Let $u_0 \in V_0^{(c)}(\Gamma)$ on Ω_i be

$$u_0|_{\Omega_i} = \mathcal{H}_i^{(c)} u_0 + P_i^{(c)} u_0, \quad i = 1, \cdots, N \tag{7}$$

where $\mathcal{H}_i^{(c)} u_0$ is discrete harmonic in Ω_i on the coarse triangulation $\{\Omega_i^j\}$ in the sense of $a_i^\varrho(.,.)$, the restriction $a_\varrho(.,.)$ to Ω_i, with data u_0 on $\partial\Omega_i$. Let $V_{0i}^{\mathcal{H}}(\Gamma_0)$ and $V_{0i}^P(\Gamma_0)$ denote subspaces of $V_0^{(c)}(\Gamma)$ defined as follows: $V_{0i}^{\mathcal{H}}(\Gamma_0)$ is a space of discrete harmonic functions in $\{\Omega_j\}$ in the sense of $\mathcal{H}_i^{(c)}$ with data u_0 on $\partial\Omega_i$ and zero on $\Gamma_{0h} \backslash \partial\Omega_{ih}$. $V_{0i}^P(\Gamma_0)$ is $P_c V_0(\Gamma)$ with zero outside Ω_i where $P_c = \{P_i^{(c)}\}$. The decomposition of $V_0^{(c)}$ is of the form

$$V_0^{(c)}(\Gamma) = V_{00}(\Gamma_0) + \sum_{i=1}^{N} (V_{0i}^{\mathcal{H}}(\Gamma_0) + V_{0i}^P(\Gamma_0)). \tag{8}$$

The space $V_{00}(\Gamma_0)$ is defined as ($\mathcal{H}_c = \{\mathcal{H}_i^{(c)}\}$)

$$V_{00}(\Gamma_0) = span\{\mathcal{H}_c I_c(\bar{\varrho}_i^{1/2}(\bar{\mu}_i^\dagger))\}, \;\; i \in \mathcal{N}_I, \;\; \bar{\varrho}_i = \frac{1}{|\Omega_i|} \sum_{\Omega_i^j \subset \Omega_i} \varrho_i^j |\Omega_i^j|$$

and

$$\bar{\mu}_i = \sum_j \bar{\varrho}_j^{1/2}, \;\; x \in \partial\Omega_i; \;\; \bar{\mu}_i = 0, \quad x \in \Gamma_{0h} \backslash \partial\Omega_{ih}.$$

We point out that $I_c(\bar{\varrho}_i^{1/2}(\bar{\mu}_i)^\dagger)$ given on $\{\partial\Omega_j\}$ is extended to Ω_j on the coarse triangulation $\{\Omega_j^k\}$ as discrete harmonic function in the sense of $a_j^\varrho(.,.)$, the restriction $a_\varrho(.,.)$ to Ω_j.We note that $V_{00}(\Gamma_0) \subset V_0^{(c)}(\Gamma) \subset V_h(\Gamma)$ since the discrete harmonic function in the sense of a_i^ϱ (.,.), with data on $\cup_i \partial\Omega_i$, is also the discrete harmonic function on $\{\Omega_i^j\}$ in the sense of $a_{ij}^\varrho(u, v)$. Using (8) in (6) we get the three-level decomposition (5) of $V_h(\Gamma)$.

4.2 Inexact solver

Let for $i = 1, \cdots, N$, $b_i^j(., .)$ be defined as in Section 3 with respect to the coarse triangulation $\{\Omega_i^j\}$, i.e. for $u, v \in V_{ij}$

$$b_i^j(u, v) = \hat{a}_{ij}(\hat{\mathcal{H}}_i^j I_h(\mu_i^j u), \hat{\mathcal{H}}_i^j I_h(\mu_i^j v))(1 + log\frac{h_c}{h})^{-1},$$

$i = 1, \cdots, N$, $j = 1, \cdots, N_i$ where $\hat{a}_{ij}(\cdot, \cdot)$ is defined as in Section 3 with Ω_i^j and H_{ij}^2 instead of Ω_i and H_i^2, where H_{ij} is a diameter of Ω_i^j.

In the space $V_{00}(\Gamma_0)$ we set

$$b_{00}(u,v) = a_\varrho(u,v)(1+log\frac{h_{sc}}{h_c})^{-1}(1+log\frac{h_c}{h})^{-1}, \quad u,v \in V_{00}(\Gamma_0),$$

where $h_{sc} = \max_i H_i$, $h_c = \max_{ij} H_{ij}$ and H_i and H_{ij} are diameters of Ω_i and Ω_i^j, respectively.

In the space $V_{0i}^{\mathcal{H}}(\Gamma_0)$ and $V_{0i}^P(\Gamma_0)$, $i = 1, \cdots, N$, we set

$$b_{0i}^{\mathcal{H}}(u,v) = \hat{a}_i(\hat{\mathcal{H}}_i^{(c)} I_c(\bar{\mu}_i u), \hat{\mathcal{H}}_i^{(c)} I_c(\bar{\mu}_i u))(1+log\frac{h_c}{h})^{-1}, \ u,v \in V_{0i}^{\mathcal{H}}(\Gamma_0),$$

and

$$b_{0i}^P(u,v) = a_i^\varrho(u,v)(1+log\frac{h_c}{h})^{-1}, \ u,v \in V_{0i}^P(\Gamma_0).$$

Here $\hat{\mathcal{H}}_i^{(c)}$ is defined as in (7) on the coarse triangulation in Ω_i with H_i, a diameter of Ω_i.

4.3 The equation

Let $T_i^j : V_h(\Gamma) \to V_{ij}(\Gamma)$ for $i = 1, \cdots, N$, $j = 1, \cdots, N_i$, be defined by

$$b_i^j(T_i^j u, v) = a_\varrho(u,v), \ v \in V_i^j(\Gamma).$$

Let $T_{0i}^{\mathcal{H}} : V_h(\Gamma) \to V_{0i}^{\mathcal{H}}(\Gamma_0)$ and $T_{0i}^P;\ V_h(\Gamma) \to V_{i0}^P(\Gamma_0)$, $i = 1, \cdots, N$, be defined by

$$b_{0i}^{\mathcal{H}}(T_{0i}^{\mathcal{H}} u, v) = a_\varrho(u,v), \ v \in V_{0i}(\Gamma)$$

and

$$b_{0i}^P(T_{0i}^P u, v) = a_\varrho(u,v), \ v \in V_{0i}^P(V_0).$$

Let finally $T_{00} : V_h(\Gamma) \to V_{00}(\Gamma_0)$ be defined by

$$b_{00}(T_{00}u, v) = a_\varrho(u,v), \quad v \in V_{00}(\Gamma_0).$$

Let

$$T = T_{00} + \sum_{i=1}^N (T_{0i}^{\mathcal{H}} + T_{0i}^P) + \sum_{i=1}^N \sum_{j=1}^{N_i} T_i^j.$$

The problem (4), defined on $\{\partial\Omega_i^j\}$ with the coefficients ϱ_i^j on Ω_i^j, is replaced by

$$Tu_h^* = g_h$$

where $g_h = g_{00} + \sum_{i=1}^N (g_{0i}^{\mathcal{H}} + g_{0i}^P) + \sum_{i=1}^N \sum_{j=1}^{N_i} g_{ij}$, $\ g_{00} = T_{00}u_h^*$, $\ g_{0i}^{\mathcal{H}} = T_{0i}^{\mathcal{H}} u_h^*$, $\ g_{0i}^P = T_{0i}^P$, $\ g_{ij} = T_i^j u_h^*$.

Theorem 2. *Let for* $i = 1, \cdots, N$

$$\bar{\varrho}_i \sim \varrho_i^j \ \ for \ \ \Omega_i^j \subset \Omega_i.$$

Then for $u \in V_h(\Gamma)$

$$C_0 S^\varrho(u,u) \le S^\varrho(Tu,u) \le \alpha C_1 S^\varrho(u,u)$$

where

$$\alpha = \max\{(1+log\frac{h_{sc}}{h_c})^2(1+log\frac{h_c}{h}), (1+log\frac{h_c}{h})^3\}$$

and C_0 *and* C_1, *are positive constants independent of* h, h_c *and* h_{sc}, *and the jumps of coefficients across* $\partial\Omega_i$.

References

1. M. Dryja and O. B. Widlund, *Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems*, Comm. Pure Appl. Math., 48 (1995), pp. 121–155.
2. J. Mandel and M. Brezina, *Balancing domain decomposition for problems with large jumps in coefficients*, Math. Comp., 65 (1996), pp. 1387–1401.
3. B. F. Smith, P. E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
4. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

The Primal Alternatives of the FETI Methods Equipped with the Lumped Preconditioner

Yannis Fragakis[1,2] and Manolis Papadrakakis[2]

[1] International Center for Numerical Methods in Engineering (CIMNE), Technical University of Catalonia, Edificio C1, Campus Norte, Gran Capitan s/n, Barcelona, 08034, Spain. fragayan@cimne.upc.es

[2] Institute of Structural Analysis and Seismic Research, National Technical University Athens 9, Iroon Polytechniou, Zografou Campus, GR-15780 Athens, Greece. fragayan@central.ntua.gr, mpapadra@central.ntua.gr

Summary. In the past few years, Domain Decomposition Methods (DDM) have emerged as advanced solvers in several areas of computational mechanics. In particular, during the last decade, in the area of solid and structural mechanics, they reached a considerable level of advancement and have been shown to be more efficient than popular solvers, like advanced sparse direct solvers. The present contribution follows the lines of a series of recent publications by the authors on DDM. In these papers, the authors developed a unified theory of primal and dual methods and presented a family of DDM that were shown to be more efficient than previous methods. The present paper extends this work, presenting a new family of related DDM, thus enriching the theory of the relations between primal and dual methods.

1 Introduction

In the last decade Domain Decomposition Methods (DDM) have progressed significantly leading to a large number of methods and techniques, capable of giving solution to various problems of computational mechanics. In the field of solid and structural mechanics, in particular, this fruitful period has led to the extensive parallel development of two large families of methods: (a) the Finite Element Tearing and Interconnecting (FETI) methods and (b) the Balancing Domain Decomposition (BDD) methods. Both categories of methods were introduced at the beginning of the 90s [1, 6] and today include a large number of variants. However, their distinct theories have led to a lack of extensive studies to interconnect them in the past. Thus, in the present decade two studies [5, 2] have attempted to determine the relations between the two methods.

In particular, studies [2, 3] set the basis of a unified theory of primal and dual DDM. This effort also led to the introduction of a new family of methods, under the name “Primal class of FETI methods”, or in abbreviation “P-FETI methods”. These methods are derived from the Dirichlet preconditioned FETI methods. They, thus, inherit the high computational efficiency properties of these methods, while their primal flavour gives them increased efficiency and robustness in ill-conditioned problems. However, so far a primal alternative for the lumped preconditioned FETI methods has not been presented. Filling this hole is the object of the present study and even though the new formulations do not appear to share the same advantages as the P-FETI formulations, they serve the purpose of diversifying our knowledge of the relations of primal and dual methods.

Thus, this paper presents the primal alternatives of the lumped preconditioned FETI methods and is organised as follows: Section 2 presents the base formulation of the introduced methods and section 3 transforms the algorithms into a more economical form. Section 4 presents numerical results for comparing the new formulation with previous ones and section 5 gives some concluding statements.

2 Basic formulation of the primal alternatives of the FETI methods equipped with the lumped preconditioner

The P-FETI methods were built on the concept of preconditioning the Schur complement method with the first estimate of displacements obtained during the FETI methods. Accordingly, the primal counterparts of the lumped preconditioned methods will be obtained by similarly preconditioning the intact global problem. Thus, the following equation

$$Ku = f \Leftrightarrow L^T K^s L u = L^T f^s \tag{1}$$

will be preconditioned with the first displacement estimate of a FETI method. In eq. (1), K, u, and f represent the global stiffness matrix, displacement and force vectors, respectively, while

$$K^s = \begin{bmatrix} K^{(1)} & & \\ & \ddots & \\ & & K^{(n_s)} \end{bmatrix} \quad , \quad u^s = \begin{bmatrix} u^{(1)} \\ \vdots \\ u^{(n_s)} \end{bmatrix} \quad , \quad f^s = \begin{bmatrix} f^{(1)} \\ \vdots \\ f^{(n_s)} \end{bmatrix} \tag{2}$$

are the matrix block-diagonal assemblage of the correponding quantities of the subdomains $s = 1, ..., n_s$ and L is a Boolean restriction matrix, such that $u^s = Lu$. Using the original FETI formulation, usually refered to as “one-level FETI” or “FETI-1”, the following preconditioner for (1) is derived (this equation is obtained following an analysis almost identical to [2], section 6):

$$\tilde{A}^{-1} = L_p^T \tilde{A}^{s^{-1}} L_p \tag{3}$$

where:

$$\tilde{A}^{s^{-1}} = H^T K^{s^+} H \quad , \quad H = I - B^T Q G (G^T Q G)^{-1} R^{s^T} \quad , \quad G = B R^s \tag{4}$$

Here, R^s and K^{s^+} are the block-diagonal assemblage of subdomain zero energy modes and generalized inverses of subdomain stiffness matrices, respectively. B is a mapping matrix such that $null(B) = range(L)$, Q is a symmetric positive definite matrix used in the FETI-1 coarse projector (see for instance [1]), while L_p and B_p are scaled variants of L and B (see the expressions gathered from various DDM papers in [2]). Similar ideas lead to the corresponding preconditioners that are derived from other FETI variants. Comparing the lumped preconditioned FETI-1 method with the method of this section, it is noted that the present method has a significantly higher computational cost, because it operates on the full displacement vector u of the structure and also needs multiplications with the full stiffness matrices of the subdomains. In order to diminish its cost, this algorithm will be transformed into a more economical version, by respresenting its primal variables with dual variables.

3 Change of variables

The primal variables of the algorithm of the previous section will be represented with dual variables, based on the theorem: If the initial solution vector of the PCG algorithm applied for the solution of eq. (1) with the preconditioner of eq. (3), is set equal to:

$$u^0 = \tilde{A}^{-1} f \tag{5}$$

then there exist suitable vectors (denoted below with the subscript "1"), such that the following variables of the PCG can be written in the forms ($k = 0, 1, ...$):

$$z^k = -L_p^T {\tilde{A}^{s}}^{-1} B^T z_1^k \quad , \quad p^k = -L_p^T {\tilde{A}^{s}}^{-1} B^T p_1^k \tag{6}$$

$$r^k = L^T K^s B_p^T r_1^k \quad , \quad q^k = L^T K^s B_p^T q_1^k \tag{7}$$

In eqs. (5) - (7) and what follows, we use the notation and steps of Algorithm 1. Eqs. (6) - (7) allow for expressing the PCG vectors, which have the size of the total number of degrees of freedom (d.o.f.), with respect to vectors whose size is equal to the row size of matrix B (which in turn is equal to the number of Lagrange multipliers used in dual DDM). They thus allow a reduction of the cost of the algorithm. The relatively small length of the present paper does not allow a full proof for the above theorem. This proof is obtained by following the steps of the PCG and thus proving recursively the eqs. (6) - (7) (The full proof can be found in a larger version of this paper [4]). Using eqs. (6) - (7) and the definitions:

- Initialize

$$r^0 = f - Ku^0 \quad , \quad z^0 = \tilde{A}^{-1} r^0 \quad , \quad p^0 = z^0 \quad , \quad q^0 = Kp^0 \quad , \quad \eta^0 = \frac{{p^0}^T r^0}{{p^0}^T q^0}$$

- Iterate $k = 1, 2, ...$ until convergence

$$u^k = u^{k-1} + \eta^{k-1} p^{k-1} \quad , \quad r^k = r^{k-1} - \eta^{k-1} q^{k-1} \quad , \quad z^k = \tilde{A}^{-1} r^k$$

$$p^k = z^k - \sum_{i=0}^{k-1} \frac{{z^k}^T q^i}{{p^i}^T q^i} p^i \quad , \quad q^k = Kp^k \quad , \quad \eta^k = \frac{{p^k}^T r^k}{{p^k}^T q^k}$$

Algorithm 1. The PCG algorithm for solving system $Ku = f$ preconditioned with $\tilde{A}^{-1}$ (full reorthogonalization)

$$z_2^k = B\tilde{A}^{s^{-1}} B^T z_1^k \quad , \quad z_3^k = B_p K^s B_p^T z_2^k \tag{8}$$

$$p_2^k = B\tilde{A}^{s^{-1}} B^T p_1^k \quad , \quad p_3^k = B_p K^s B_p^T p_2^k \tag{9}$$

$$r_2^k = B_p K^s B_p^T r_1^k \quad , \quad r_3^k = B\tilde{A}^{s^{-1}} B^T r_2^k \tag{10}$$

$$q_2^k = B_p K^s B_p^T q_1^k \quad , \quad q_3^k = B\tilde{A}^{s^{-1}} B^T q_2^k \tag{11}$$

it is thus shown following the proof of the above theorem that the PCG algorithm for solving eq. (1) with preconditioner of eq. (3) is transformed into Algorithm 2 (in the case of full reorthogonalization). In Algorithm 2, it is worth noting that even though the formulation is primal, the final algrorithm is very similar to the algorithm of the FETI-1 method with the lumped preconditioner. In particular:

- The matrices $B\tilde{A}^{s^{-1}} B^T$ and $B_{p_b}^T K_{bb}^s B_{p_b}^T$ that are used during the iterations are equal to the FETI-1 matrix operator and lumped preconditioner, respectively.
- The algorithm iterates on vectors of the size of the Lagrange multipliers.
- From the equations that compute vectors r^k and q^k $(k = 0, 1, ...)$ in Algorithm 2, it follows that the residuals r^k vanish in internal d.o.f. of the subdomains, when these d.o.f. are not adjacent to the interface, again as in FETI-1 with the lumped preconditioner.

On the other hand, each iteration of the present algorithm requires more linear combinations of vectors than a dual algorithm. These operations become important in the case of reorthogonalization. In this case, the required dot products $z_1^{k^T}(q_3^i - q_1^i)$, $i = 0, ..., k-1$ imply the same computational cost as in FETI-1, because at each iteration $q_3^k - q_1^k$ is computed and stored. However, compared to FETI-1, this algorithm requires twice as many linear combinations for computing the vectors p_1^k and p_2^k, that represent the direction vectors p^k. In total, in this algorithm reorthogonalization requires 50% more floating point operations than in FETI-1. In addition, while FETI-1 reorthogonalization requires storing two vectors per iteration, here it is required to store the three vectors p_1^k, p_2^k and $q_3^k - q_1^k$, which implies 50% higher memory requirements for reorthogonalization in Algorithm 2.

- Initialize

$$u^0 = L_p^T \tilde{A}^{s^{-1}} L_p f \quad , \quad \tilde{u}^0 = 0 \quad , \quad r_1^0 = B\tilde{A}^{s^{-1}} L_p f$$
$$r^0 = \begin{bmatrix} L_b^T K_{bb}^s \\ K_{ib}^s \end{bmatrix} B_{p_b}^T r_1^0 \quad , \quad p_1^0 = z_1^0 = B_{p_b}^T K_{bb}^s B_{p_b}^T r_1^0$$
$$q_1^0 = p_2^0 = r_3^0 = z_2^0 = B\tilde{A}^{s^{-1}} B^T z_1^0 \quad , \quad q^0 = \begin{bmatrix} L_b^T K_{bb}^s \\ K_{ib}^s \end{bmatrix} B_{p_b}^T q_1^0$$
$$p_3^0 = q_2^0 = B_{p_b}^T K_{bb}^s B_{p_b}^T q_1^0 \quad , \quad \eta^0 = \frac{(p_3^{0^T} - p_1^{0^T}) r_1^0}{(p_3^{0^T} - p_1^{0^T}) q_1^0}$$

- Iterate $k = 1, 2, ...$ until convergence $(\left\| r^k \right\| < \varepsilon)$

$$\tilde{u}_1^k = \tilde{u}_1^k + \eta^{k-1} p_1^{k-1} \quad , \quad r^k = r^{k-1} - \eta^{k-1} q^{k-1} \quad , \quad r_1^k = r_1^{k-1} - \eta^{k-1} q_1^{k-1}$$
$$z_1^k = r_2^k = r_2^{k-1} - \eta^{k-1} q_2^{k-1} \quad , \quad r_3^k = z_2^k = B\tilde{A}^{s^{-1}} B^T z_1^k$$
$$q_3^{k-1} = \left(1 \middle/ \eta^{k-1}\right)\left(r_3^{k-1} - r_3^k\right) \quad , \quad p_1^k = z_1^k - \sum_{i=0}^{k-1} \frac{z_1^{k^T}(q_3^i - q_1^i)}{p_1^{i^T}(q_3^i - q_1^i)} p_1^i$$

$$q_1^k = p_2^k = z_2^k - \sum_{i=0}^{k-1} \frac{{z_1^k}^T (q_3^i - q_1^i)}{{p_1^i}^T (q_3^i - q_1^i)} p_2^i \quad , \quad q^k = \begin{bmatrix} L_b^T K_{bb}^s \\ K_{ib}^s \end{bmatrix} B_{p_b}^T p_2^k$$

$$p_3^k = q_2^k = B_{p_b}^T K_{bb}^s B_{p_b}^T p_2^k \quad , \quad \eta^k = \frac{({p_3^k}^T - {p_1^k}^T) r_1^k}{({p_3^k}^T - {p_1^k}^T) q_1^k}$$

- After convergence

$$u^k = u^0 - L_p^T {\tilde{A}^s}^{-1} B^T \tilde{u}_1^k$$

Algorithm 2: The primal alternative of the FETI-1 method with the lumped preconditioner (full reorthogonalization)

4 Numerical results

We have implemented the FETI-1 and FETI-DP methods with the lumped preconditioner and their primal alternatives in our Matlab code and we consider the 3-D elasticity problem of Fig. 1. This cubic structure is composed of five layers of two different materials and is discretized with 28 × 28 × 28 8-node brick elements. Additionally, it is pinned at the four corners of its left surface. Various ratios E_A/E_B of the Young modulus and ρ_A/ρ_B of the density of the two materials are considered, while their Poisson ratio is set equal to $\nu_A = \nu_B = 0.30$. Two decompositions P1 and P2 of this heterogeneous model of 73, 155 d.o.f. in 100 subdomains, are considered (see [2] for details).

Table 1 presents the iterations required by primal and dual formulations of the lumped preconditioned FETI-1 method. The results show that like in the case of comparing dual and primal formulations of the Dirichlet preconditioned FETI methods, the iterations of the two formulations of the lumped preconditioned FETI-1 methods are comparable. More precisely, it is noted that in the more ill-conditioned cases the primal method requires slightly fewer iterations (up to 11%) than the dual one. In fact, judging also from many other tests that we have performed comparing the two formulations of FETI-1 and FETI-DP with the lumped preconditioner, it appears that the difference between the number of iterations of primal and dual formulations in ill-conditioned problems is more pronounced in the case of the lumped preconditioner than in the case of the Dirichlet preconditioner. A probable explanation is that the lumped preconditioned methods lead by themselves to more ill-conditioned systems than the Dirichlet ones.

On the other hand, bearing in mind that the primal formulation implies a 50% higher reorthogonalization cost, we conclude that statistically the primal formulation will be probably slower than the dual one for well-conditioned problems and probably faster for ill-conditioned problems with relatively low reorthogonalization cost. In addition, in the case of the lumped preconditioner, our results do not show the increased robustness (measured in terms of the maximum achievable solution accuracy in ill-conditioned problems) of the primal formulation that has been seen in the case of the P-FETI formulations. A probable explanation of this observation is given by the increased operations required in each iteration of the primal algorithm as oposed to the dual one and also by the fact that due to setting the initial solution vector equal to eq. (20), the initial residual of the primal methods is equal to the initial residual of the dual methods (see the expression of the residual r^0 in Algorithm 2, which is equal to the initial residual of the FETI-1 method). Thus, contrary to the P-FETI formulations, the residuals of the primal formulations of the lumped preconditioned FETI methods begin from relatively high values, as in the dual formulations.

5 Conclusions

The roots of the work presented in this paper can be traced back to the paper [2]. That paper introduced the P-FETI methods, as the primal alternatives of the Dirichlet preconditioned FETI methods. Compared to the original FETI formulations, the P-FETI methods present the advantage of being more robust and faster in the solution of ill-conditioned problems. [2] also introduced an open question of the existence of a primal alternative for the lumped preconditioned FETI methods. In the last few years it has become clear that the lumped preconditioner leads to faster solutions, in the cases where a problem needs to be decomposed in a relatively small number of subdomains. These cases and also the cases where the lumped preconditioner leads to implementations that require less memory (in large problems where this can be the main issue), appear to be the cases where the lumped preconditioner is used in modern DDM practice.

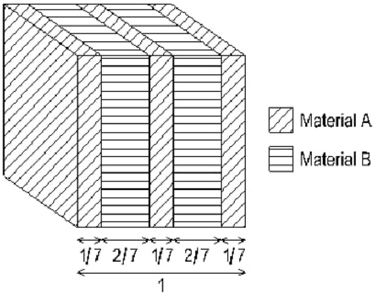


Fig. 1. A cubic structure composed of two materials.

The present work introduces primal alternatives of the lumped preconditioned FETI methods. These new formulations do not appear to present the advantages of the P-FETI formulations, since they are slightly slower or faster than their dual counterparts depending on the problem and do not exhibit higher robustness than the dual methods. Their principal value lies in the fact that they add a new level of completion to the theory of the relations of primal and dual methods. The fact that a primal algorithm can be turned to an algorithm which uses dual operators and vectors appears to be new. It is also worth noting that the same transformations

Table 1. Number of iterations (Tolerance:10^{-3}) of the lumped preconditioned FETI-1 method and its primal alternative for the solution of the example of Fig. 1.

Ratio of Young moduli	Type of decomposition	Dual formulation	Primal formulation
10^0	P1	25	24
10^3	P1	44	41
10^3	P2	25	24
10^6	P1	30	26
10^6	P2	53	47

used in this paper can be used for the P-FETI and the BDD methods in order to transform them into algorithms that operate on dual quantities. This and many other recent studies [5, 7] show more and more that primal and dual formulations are closely connected.

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 (2000), pp. 513–536.
2. Y. Fragakis and M. Papadrakakis, *The mosaic of high performance domain decomposition methods for structural mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods*, Comput. Methods Appl. Mech. Engrg, 192 (2003), pp. 3799–3830.
3. ———, *The mosaic of high performance domain decomposition methods for structural mechanics – part II: Formulation enhancements, multiple right-hand sides and implicit dynamics*, Comput. Methods Appl. Mech. Engrg., 193 (2004), pp. 4611–4662.
4. ———, *Derivation of the primal alternatives of the lumped preconditioned FETI methods*, tech. rep., Institute of Structural Analysis and Seismic Research, National Technical University of Athens, Athens, Greece, 2005. Available from http://users.ntua.gr/fragayan/publications.htm.
5. A. Klawonn and O. B. Widlund, *FETI and Neumann–Neumann iterative substructuring methods: Connections and new results*, Comm. Pure Appl. Math., 54 (2001), pp. 57–90.
6. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.
7. J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Appl. Numer. Math., 54 (2005), pp. 167–193.

Balancing Domain Decomposition Methods for Mortar Coupling Stokes-Darcy Systems

Juan Galvis[1] and Marcus Sarkis[2]

[1] Instituto Nacional de Matemática Pura e Aplicada, Estrada Dona Castorina 110, CEP 22460320, Rio de Janeiro, Brazil. jugal@fluid.impa.br

[2] Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, Brazil, and Worcester Polytechnic Institute, Worcester, MA 01609, USA. msarkis@fluid.impa.br

1 Introduction and Problem Setting

We consider Stokes equations in the fluid region Ω_f and Darcy equations for the filtration velocity in the porous medium Ω_p, and coupled at the interface Γ with adequate transmission conditions. Such problem appears in several applications like well-reservoir coupling in petroleum engineering, transport of substances across groundwater and surface water, and (bio)fluid-organ interactions. There are some works that address numerical analysis issues such as inf-sup and approximation results associated to the continuous and discrete formulations Stokes-Darcy systems [8, 7, 6] and Stokes-Laplacian systems [2, 3], mortar discretizations analysis [12, 6], preconditioning analysis for Stokes-Laplacian systems [4, 1]. Here we are interested on preconditionings for *Stokes-Mortar-Darcy* with *flux boundary conditions*, therefore the global system as well as the local systems require flux compatibilities. Here we propose two preconditioners based on balancing domain decomposition methods [9, 11, 5]; in the first one the energy of the preconditioner is controlled by the Stokes system while in the second one it is controlled by the Darcy system. The second is more interesting because it is scalable for the parameters faced in practice.

Let $\Omega_f,\ \Omega_p \subset \Re^n$ be polyhedral subdomains, $\Omega = \text{int}(\overline{\Omega}_f \cup \overline{\Omega}_p)$ and $\Gamma = \text{int}(\partial\Omega_f \cup \partial\Omega_p)$, with outward unit normal vectors on $\partial\Omega_j$ denoted by $\boldsymbol{\eta}_j$, $j = f, p$. The tangent vectors of Γ are denoted by $\boldsymbol{\tau}_1$ $(n = 2)$, or $\boldsymbol{\tau}_l$, $l = 1, 2$ $(n = 3)$. Define $\Gamma_j := \partial\Omega_j \setminus \Gamma$, $j = f, p$. Fluid velocities are denoted by $\boldsymbol{u}_j : \Omega_j \rightarrow \Re^n$, $j = f, p$. Pressures are $p_j : \Omega_j \rightarrow \Re$, $j = f, p$. We have:

$$\begin{array}{cc} \text{Stokes equations} & \text{Darcy equations} \\ \left\{ \begin{aligned} -\nabla\cdot T(\boldsymbol{u}_f, p_f) &= \boldsymbol{f}_f \text{ in } \Omega_f \\ \nabla\cdot\boldsymbol{u}_f &= g_f \text{ in } \Omega_f \\ \boldsymbol{u}_f &= \boldsymbol{h}_f \text{ on } \Gamma_f \end{aligned} \right. & \left\{ \begin{aligned} \boldsymbol{u}_p &= -\frac{\kappa}{\mu}\nabla p_p \text{ in } \Omega_p \\ \nabla\cdot\boldsymbol{u}_p &= g_p \qquad \text{in } \Omega_p \\ \boldsymbol{u}_p\cdot\boldsymbol{\eta}_p &= h_p \qquad \text{on } \Gamma_p \end{aligned} \right. \end{array} \tag{1}$$

Here $T(\boldsymbol{v}, p) := -pI + 2\mu\boldsymbol{D}\boldsymbol{v}$ where μ is the viscosity and $\boldsymbol{D}\boldsymbol{v} := \dfrac{1}{2}(\nabla\boldsymbol{v} + \nabla\boldsymbol{v}^T)$ is the linearized strain tensor. κ represents the rock permeability and μ the fluid viscosity. For simplicity in the analysis we assume that κ is a real positive constant. We also impose the compatibility condition (see [6])

$$\langle g_f, 1\rangle_{\Omega_f} + \langle g_p, 1\rangle_{\Omega_p} - \langle \boldsymbol{h}_f \cdot \boldsymbol{\eta}_f, 1\rangle_{\Gamma_f} - \langle h_p, 1\rangle_{\Gamma_p} = 0,$$

and the following interface matching conditions across Γ (see [8, 3, 2, 4] and references therein):

1. **Conservation of mass across Γ:** $\boldsymbol{u}_f \cdot \boldsymbol{\eta}_f + \boldsymbol{u}_p \cdot \boldsymbol{\eta}_p = 0$ on Γ.

2. **Balance of normal forces across Γ:** $p_f - 2\mu \boldsymbol{\eta}_f^T \boldsymbol{D}(\boldsymbol{u}_f)\boldsymbol{\eta}_f = p_p$ on Γ.

3. **Beavers-Joseph-Saffman condition:** This condition is an empirical law that gives an expression for the component of Σ in the tangential direction of $\boldsymbol{\tau}$. It is expressed by:

$$\boldsymbol{u}_f \cdot \boldsymbol{\tau}_j = -\frac{\sqrt{\kappa}}{\alpha_f} 2\boldsymbol{\eta}_f^T \boldsymbol{D}(\boldsymbol{u}_f)\boldsymbol{\tau}_j \quad j = 1, d-1; \text{ on } \Gamma. \tag{2}$$

2 Weak Formulations and Discretization.

Without loss of generality we consider the case where $\boldsymbol{h}_f = 0$, $h_p = 0$, and $\alpha_f = \infty$. Here we use the energy of α_f-harmonic Stokes and harmonic Laplacian extensions are equivalents independent of α_f; see [6].

The problem is formulated as: *Find* $(\boldsymbol{u}, p, \lambda) \in \boldsymbol{X} \times M \times \Lambda$ *satisfying, for all* $(\boldsymbol{v}, q, \mu) \in \boldsymbol{X} \times M \times \Lambda$:

$$\begin{cases} a(\boldsymbol{u}, \boldsymbol{v}) + b(\boldsymbol{v}, p) + b_\Gamma(\boldsymbol{v}, \lambda) = \ell(\boldsymbol{v}) \\ b(\boldsymbol{u}, q) \qquad\qquad\qquad\qquad\quad = g(q) \\ b_\Gamma(\boldsymbol{u}, \mu) \qquad\qquad\qquad\qquad = 0, \end{cases} \tag{3}$$

where $\boldsymbol{X} = \boldsymbol{X}_f \times \boldsymbol{X}_f := H_0^1(\Omega_f, \Gamma_f)^2 \times \boldsymbol{H}_0(\text{div}, \Omega_p, \Gamma_p)$; $M := L_0^2(\Omega) \subset L^2(\Omega_f) \times L^2(\Omega_p)$. Here $H_0^1(\Omega_f, \Gamma_f)$ denotes the subspace of $H^1(\Omega_f)$ of functions that vanish on Γ_f. Analogously, $\boldsymbol{H}_0(\text{div}, \Omega_p, \Gamma_p)$ denotes the subspace of $\boldsymbol{H}(\text{div}, \Omega_p)$ of functions with its normal trace restricted to Γ_p zero. The Lagrange multiplier space is $\Lambda := H^{1/2}(\Gamma)$. Also

$$a(\boldsymbol{u}, \boldsymbol{v}) := a_f(\boldsymbol{u}_f, \boldsymbol{v}_f) + a_p(\boldsymbol{u}_p, \boldsymbol{v}_p), \qquad b(\boldsymbol{v}, p) := b_f(\boldsymbol{v}_f, p_f) + b_p(\boldsymbol{v}_p, p_p),$$

and $b_\Gamma(\boldsymbol{v}, \mu) := \langle \boldsymbol{v}_f \cdot \boldsymbol{\eta}_f, \mu\rangle_\Gamma + \langle \boldsymbol{v}_p \cdot \boldsymbol{\eta}_p, \mu\rangle_\Gamma$, $\boldsymbol{v} = (\boldsymbol{v}_f, \boldsymbol{v}_p) \in \boldsymbol{X}, \mu \in \Lambda$, where $\langle \boldsymbol{v}_p \cdot \boldsymbol{\eta}_p, \mu\rangle_\Gamma := \langle \boldsymbol{v}_p \cdot \boldsymbol{\eta}_p, E\boldsymbol{\eta}_p(\mu)\rangle_{\partial\Omega_p}$. Here $E\boldsymbol{\eta}_p$ is any continuous lifting. The bilinear forms a_j, b_j are associated to Stokes equations, $j = f$, and Darcy law, $j = p$. The bilinear for a_f incorporates conditions 2 and 3 above. The bilinear form b_Γ is the weak version of condition 1 above. For the analysis of this weak formulation and the well-posedness of the problem see [6].

From now on we assume that Ω_i, $i = f, p$, are *two dimensional* polygonal subdomains. Let $\mathcal{T}_i^{h_i}$ be a triangulation of Ω_i, $i = f, p$. We do not assume that they match at the interface Γ. For the fluid region, let $\boldsymbol{X}_f^{h_f}$ and $M_f^{h_f}$ be the $P2/P1$

triangular Taylor-Hood finite elements and denote $\mathring{M}_f^{h_f} = M_f^{h_f} \cap L_0^2(\Omega_f)$. For the porous region, let $\boldsymbol{X}_p^{h_p}$ and $M_p^{h_p}$ be the lowest order Raviart-Thomas finite elements based on triangles and denote $\mathring{M}_p^{h_p} = M_p^{h_p} \cap L_0^2(\Omega_p)$. We assume in the definition of the discrete velocities that the boundary conditions are included, i.e., for $\boldsymbol{v}_f^{h_f} \in \boldsymbol{X}_f^{h_f}$ we have $\boldsymbol{v}_f^{h_f} = \boldsymbol{0}$ on Γ_f and for $\boldsymbol{v}_p^{h_p} \in \boldsymbol{X}_p^{h_p}$, $\boldsymbol{v}_p^h \cdot \boldsymbol{\eta}_p = 0$ holds on Γ_p.

We choose piecewise constant Lagrange multiplier space:

$$\Lambda^{h_p} := \left\{ \lambda \ : \ \lambda|_{e_j^p} = \lambda_{e_j^p} \ \text{ is constant in each edge } e_j^p \text{ of } \mathcal{T}_p^{h_p}(\Gamma) \right\},$$

i.e., the mortar is on the fluid region side and the slave on the porous region side, and leads to a nonconforming approximation on Λ^{h_p} since piecewise constant functions do not belong to $H^{1/2}(\Gamma)$. Define $\boldsymbol{X}^h := \boldsymbol{X}_f^{h_f} \times \boldsymbol{X}_p^{h_p}$, and

$$\boldsymbol{Z}_\Gamma^h := \left\{ \boldsymbol{v}^h \in \boldsymbol{X}^h \ : \ (\boldsymbol{v}_f^{h_f} \cdot \boldsymbol{\eta}_f + \boldsymbol{v}_p^{h_p} \cdot \boldsymbol{\eta}_p, \mu)_\Gamma = 0 \ \forall \mu \in \Lambda^{h_p} \right\}. \tag{4}$$

3 Matrix and Vector Representations

To simplify notation, we drop the subscript h associated to the discrete variables. We consider the following partition of the degrees of freedom:

$$\begin{bmatrix} \boldsymbol{u}_I^i \\ p_I^i \\ u_\Gamma^i \\ \bar{p}^i \end{bmatrix} \begin{array}{l} \text{Interior displacements + tangential velocities at } \Gamma, \\ \text{Interior pressures with zero average in } \Omega_i, \\ \text{Interface normal displacements on } \Gamma, \\ \text{Constant pressure in } \Omega_i, \end{array} \qquad i = f, p.$$

Then, we have the following matrix representation of the coupled problem:

$$\left[\begin{array}{cccc|cccc|c} A_{II}^f & B_{II}^{fT} & A_{\Gamma I}^{fT} & 0 & 0 & 0 & 0 & 0 & 0 \\ B_{II}^f & 0 & B_{I\Gamma}^f & 0 & 0 & 0 & 0 & 0 & 0 \\ A_{\Gamma I}^f & B_{I\Gamma}^{fT} & A_{\Gamma\Gamma}^f & \bar{B}^{fT} & 0 & 0 & 0 & 0 & B_f^T \\ 0 & 0 & \bar{B}^f & 0 & 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & A_{II}^p & B_{II}^{pT} & A_{\Gamma I}^{pT} & 0 & 0 \\ 0 & 0 & 0 & 0 & B_{II}^p & 0 & B_{I\Gamma}^p & 0 & 0 \\ 0 & 0 & 0 & 0 & A_{\Gamma I}^p & B_{I\Gamma}^{pT} & A_{\Gamma\Gamma}^p & \bar{B}^{pT} & B_p^T \\ 0 & 0 & 0 & 0 & 0 & 0 & \bar{B}^p & 0 & 0 \\ \hline 0 & 0 & B_f & 0 & 0 & 0 & -B_p & 0 & 0 \end{array} \right] \begin{bmatrix} \boldsymbol{u}_I^f \\ p_{\mathrm{i}}^f \\ u_\Gamma^f \\ \bar{p}^f \\ \boldsymbol{u}_I^p \\ p_{\mathrm{i}}^p \\ u_\Gamma^p \\ \bar{p}^p \\ \lambda \end{bmatrix}$$

and in each subdomain (see [11, 5]) given by:

$$\left[\begin{array}{cc|cc} A_{II}^i & B_{II}^{iT} & A_{\Gamma I}^{iT} & 0 \\ B_{II}^i & 0 & B_{I\Gamma}^i & 0 \\ \hline A_{\Gamma I}^i & B_{I\Gamma}^{iT} & A_{\Gamma\Gamma}^i & \bar{B}^{iT} \\ 0 & 0 & \bar{B}^i & 0 \end{array} \right] = \begin{bmatrix} K_{II}^i & K_{\Gamma I}^{iT} \\ K_{\Gamma I}^i & K_{\Gamma\Gamma}^i \end{bmatrix}. \tag{5}$$

The mortar condition 4 on Γ (Darcy side as the slave side) is imposed as $u^p_\Gamma = -B_p^{-1} B_f u^f_\Gamma = \Pi u^f_\Gamma$, where $-\Pi$ is the $L^2(\Gamma)$ projection onto the space of piecewise constant functions on each e^p_i. We note that that B_p is a diagonal matrix for the lowest order Raviart-Thomas elements.

We now eliminate $\boldsymbol{u}^i_I$, p^i_I, $i = f, p.$, and λ, to obtain the following (saddle point) Schur complement equations

$$S \begin{bmatrix} u^f_\Gamma \\ \bar{p}^f \\ \bar{p}^p \end{bmatrix} = \begin{bmatrix} b \\ \bar{b}^f \\ \bar{b}^p \end{bmatrix},$$

which is solvable when $\bar{b}^f + \bar{b}^p = 0$. Here S is given by

$$S := S^f + \tilde{\Pi}^T S^p \tilde{\Pi} = \left[\begin{array}{c|cc} S^f_\Gamma + \Pi^T S^p_\Gamma \Pi & \bar{B}^{fT} & \Pi^T \bar{B}^{pT} \\ \hline \bar{B}^f & 0 & 0 \\ \bar{B}^p \Pi & 0 & 0 \end{array}\right] = \begin{bmatrix} S_\Gamma & \bar{B}^T \\ \bar{B} & 0 \end{bmatrix},$$

where $\tilde{\Pi} := \begin{bmatrix} \Pi & 0 \\ 0 & I_{2\times 2} \end{bmatrix}$ and $S^i := K^i_{\Gamma\Gamma} - K^i_{\Gamma I} \left(K^i_{\Gamma\Gamma} \right)^{-1} K^{iT}_{\Gamma I} = \begin{bmatrix} S^i_\Gamma & \bar{B}^{iT} \\ \bar{B}^i & 0 \end{bmatrix}$.

Define $\boldsymbol{V}_\Gamma := \left\{ \boldsymbol{v} \in \boldsymbol{Z}^h : \boldsymbol{v}_f = \mathcal{SH}(\boldsymbol{v}_f \cdot \boldsymbol{\eta}_f|_\Gamma) \text{ and } \boldsymbol{v}_p = \mathcal{DH}(\boldsymbol{v}_p \cdot \boldsymbol{\eta}_p|_\Gamma)|_\Gamma) \right\}$ and

$$\boldsymbol{M}_0 := \left\{ q \in M^h : q_i = \text{const. in } \Omega_i, i = f, p; \text{ and } \int_{\Omega_f} q_f + \int_{\Omega_p} q_p = 0 \right\}.$$

Here $\mathcal{SH}$ ($\mathcal{DH}$) is the velocity component of the discrete Stokes (Darcy) harmonic extension operator that maps discrete interface normal velocity $\hat{u}^f_\Gamma \in H^{1/2}_{00}(\Gamma)$ ($\hat{u}^p_\Gamma \in (H^{1/2}(\Gamma))'$) to the solution of the problem: find $\boldsymbol{u}_i \in \boldsymbol{X}^{h_i}_f$ and $p_i \in \mathring{M}^{h_i}_i$ such that in Ω_i and $\forall \boldsymbol{v}_i \in \boldsymbol{X}^{h_i}_i$ and $\forall q_i \in \mathring{M}^{h_i}_i$ we have:

$$\begin{cases} a_f(\boldsymbol{u}_f, \boldsymbol{v}_f) + b_f(\boldsymbol{v}_f, p_f) = 0 \\ b_f(\boldsymbol{u}_f, q_f) \qquad\qquad\quad = 0 \\ \boldsymbol{u}_f \cdot \boldsymbol{\eta} = \hat{u}^f_\Gamma \text{ on } \Gamma \\ \boldsymbol{u}_f \cdot \boldsymbol{\eta} = 0 \text{ on } \Gamma_f \\ \boldsymbol{u}_f \cdot \boldsymbol{\tau} = 0 \text{ on } \partial\Omega_f \end{cases} \qquad \begin{cases} a_p(\boldsymbol{u}_p, \boldsymbol{v}_p) + b_p(\boldsymbol{v}_p, p_p) = 0 \\ b_p(\boldsymbol{u}_p, q_p) \qquad\qquad\quad = 0 \\ \boldsymbol{u}_p \cdot \boldsymbol{\eta} = \hat{u}^p_\Gamma \text{ on } \Gamma \\ \boldsymbol{u}_p \cdot \boldsymbol{\eta} = 0 \text{ on } \Gamma_f. \end{cases} \tag{6}$$

Associated with the coupled problem we introduce the *balanced subspace*:

$$\boldsymbol{V}_{\Gamma,\bar{B}} := \mathrm{Ker}\bar{B} = \left\{ \boldsymbol{v} \in \boldsymbol{V}_\Gamma : \int_\Gamma \boldsymbol{v}^i \cdot \boldsymbol{\eta}_i = 0, i = f, p \text{ and } \boldsymbol{u}^p_\Gamma = \Pi v^f_\Gamma \right\}. \tag{7}$$

4 Balancing Domain Decomposition Preconditioner I

For the sake of simplicity in the analysis we assume that $\Gamma = \{0\} \times [0,1]$, $\Omega_f = (-1,0) \times (0,1)$ and $\Omega_p = (0,1) \times (0,1)$. We introduce the velocity coarse space on Γ as the span of the $\phi^0_f = y(y-1)$ and let v_0 be its vector representation. Define:

$$R_0 = \begin{bmatrix} v_0^T & 0 \\ 0 & I_{2\times 2} \end{bmatrix}, \quad S_0 = R_0 S R_0^T \quad \text{and} \quad Q_0 = R_0^T S_0^\dagger R_0.$$

Because v_0 is not balanced, S_0 is invertible when pressures restricted to M_0. The low dimensionality of the coarse space and the shape of ϕ_f^0 are kept fixed with respect to mesh parameters, imply stable discrete inf-sup condition for the coarse problem. Denote $\tilde{S}_0 := v_0^T S_\Gamma v_0$ and $\tilde{S} := \bar{B} v_0 \tilde{S}_0^{-1} v_0^T \bar{B}^T$. A simple calculation gives $I - Q_0 S = \begin{bmatrix} I - \mathcal{P} & 0 \\ \mathcal{G} & 0 \end{bmatrix}$, where

$$\begin{aligned}
\mathcal{P} &:= \left(v_0 \tilde{S}_0^{-1} v_0^T S_\Gamma - v_0 \tilde{S}_0^{-1} v_0^T \bar{B}^T \tilde{S}^{-1} \bar{B} v_0 \tilde{S}_0^{-1} v_0^T S_\Gamma \right) + v_0 \tilde{S}_0^{-1} v_0^T \bar{B}^T \tilde{S}^{-1} \bar{B} \\
\mathcal{G} &:= \tilde{S}^{-1} \bar{B} - \tilde{S}^{-1} \bar{B} v_0 \tilde{S}_0^{-1} v_0^T S_\Gamma.
\end{aligned}$$

Note that $\mathcal{P}$ is a projection and that $\bar{B}(I - \mathcal{P}) = 0$, i.e. the image of $I - \mathcal{P}$ is contained in the balanced subspace defined in (7); see also [11]. Given a residual r, the coarse problem $Q_0 r$ is the solution of a coupled problem with one velocity degree of freedom (v_0) and a constant pressure per subdomain Ω_i, $i = f, p$ with mean zero on Ω. Hence, when v_Γ and u_Γ are balanced functions, the S_Γ-inner product is defined by (see (3)):

$$\langle u_\Gamma, v_\Gamma \rangle_{S_\Gamma} := \langle S_\Gamma u_\Gamma, v_\Gamma \rangle = u_\Gamma^T S_\Gamma v_\Gamma$$

coincides with the S-inner product defined by

$$\left\langle \begin{bmatrix} v_\Gamma \\ \bar{q} \end{bmatrix}, \begin{bmatrix} u_\Gamma \\ \bar{p} \end{bmatrix} \right\rangle_S := \begin{bmatrix} v_\Gamma \\ \bar{q} \end{bmatrix}^T S \begin{bmatrix} u_\Gamma \\ \bar{p} \end{bmatrix}.$$

Consider the following BDD preconditioner operator (See [5]):

$$S_N^{-1} = Q_0 + (I - Q_0 S) \left(S^f \right)^{-1} (I - S Q_0). \tag{8}$$

Also observe that $S_N^{-1} S = Q_0 S + (I - Q_0 S) \left(S^f \right)^{-1} S (I - Q_0 S)$, and when u_Γ, v_Γ are balanced functions we have:

$$\langle S_N^{-1} S \begin{bmatrix} u_\Gamma \\ \bar{p} \end{bmatrix}, \begin{bmatrix} v_\Gamma \\ \bar{q} \end{bmatrix} \rangle_S = \langle \left(S_\Gamma^f \right)^{-1} S_\Gamma u_\Gamma, v_\Gamma \rangle_{S_\Gamma},$$

and

$$c \langle u_\Gamma^f, u_\Gamma^f \rangle_{S_\Gamma} \le \langle \left(S^f \right)^{-1} S_\Gamma u_\Gamma^f, u_\Gamma^f \rangle_{S_\Gamma} \le C \langle u_\Gamma^f, u_\Gamma^f \rangle_{S_\Gamma}$$

is equivalent to

$$c \langle S_f u_\Gamma^f, u_\Gamma^f \rangle \le \langle S_\Gamma u_\Gamma^f, u_\Gamma^f \rangle \le C \langle S_f u_\Gamma^f, u_\Gamma^f \rangle. \tag{9}$$

Proposition 1 *If u_Γ^f is a balanced function then*

$$\langle S_\Gamma^f u_\Gamma^f, u_\Gamma^f \rangle \le \langle S_\Gamma u_\Gamma^f, u_\Gamma^f \rangle \preceq (1 + \frac{1}{\kappa}) \langle S_f u_\Gamma^f, u_\Gamma^f \rangle.$$

Proof. The lower bound follows trivially from S_Γ^f and S_Γ^p being positive on the subspace of balanced functions. We next concentrate on the upper bound.

Let v_Γ^f a balanced function and $v_\Gamma^p = \Pi v_\Gamma^f$. Define $\boldsymbol{v}_p = \mathcal{DH} v_\Gamma^p$. Using properties ([10]) of the discrete operator $\mathcal{DH}$ we obtain

$$\langle S_\Gamma^p v_\Gamma^p, v_\Gamma^p \rangle = a_p(\boldsymbol{v}_p, \boldsymbol{v}_p) \asymp \frac{\mu}{\kappa} \|v_\Gamma^p\|^2_{(H^{1/2})'(\Gamma)}.$$

Using the L_2-stability property of mortar projection Π we have

$$\|v_\Gamma^p\|^2_{(H^{1/2})'(\Gamma)} \preceq \|v_\Gamma^p\|^2_{L^2(\Gamma)} = \|v_\Gamma^f\|^2_{L^2(\Gamma)} \preceq \|v_\Gamma^f\|^2_{H_{00}^{1/2}(\Gamma)}.$$

Defining $\boldsymbol{v}_f = \mathcal{SH} v_\Gamma^f$ and using properties of $\mathcal{SH}$ ([11],GS05) we have

$$\mu \|v_\Gamma^f\|^2_{H_{00}^{1/2}(\Gamma)} \asymp a_f(\boldsymbol{v}_f, \boldsymbol{v}_f).$$

5 Balancing Domain Decomposition Preconditioner II

We note that the previous preconditioner is scalable with respect to mesh parameters, however it deteriorates when the permeability κ gets small. In real life applications, permeabilities are in general very small, hence the previous preconditioner becomes irrelevant in practice. In addition, to capture the boundary layer behavior of Navier-Stokes flows near the interface Γ, the size of the fluid mesh h_f needs to be small while the Darcy mesh does not. With those two issues in mind, we were motivated to propose the second preconditioner. In contrast to the former preconditioner, we now control the Stokes energy by the Darcy energy.

We assume that the fluid side discretization on Γ is a *refinement* of the corresponding porous side discretization. For $j = 1, \dots, M^p$, and on Γ, we introduce normal velocity Stokes functions ϕ_f^j (a bubble $P2$ function) with support in the interval $e_p^j = 0 \times [(j-1)h_p], jh_p]$. Under the assumption of nested refinement and $P2/P1$ Tatlor-Hood discretization, $\phi_f^j \in \boldsymbol{X}^f|_\Gamma$. Denote by $\boldsymbol{X}_f^b$ as the subspace spanned by all ϕ_f^j and by $\boldsymbol{X}_n^f$ as the subspace spanned by the functions of v_Γ^f which has zero average on all edges e_p^j. Note that $\boldsymbol{X}_f^b$ and $\boldsymbol{X}_n^f$ form a direct sum for $\boldsymbol{X}^f|_\Gamma$ and the image $\Pi \boldsymbol{X}_n^f$ is the zero vector. Using this space decomposition we can write

$$S_\Gamma^f = \begin{bmatrix} S_{bb}^f & S_{nb}^{fT} \\ S_{nb}^f & S_{nn}^f \end{bmatrix}$$

and by eliminating the variables associated with the spaces $\boldsymbol{X}_n^f$ we obtain

$$\hat{S}_\Gamma^f = S_{bb}^f - S_{nb}^{fT} (S_{nn}^f)^{-1} S_{nb}^f,$$

and end up again with a Schur complement of the form

$$\hat{S} := \hat{S}^f + \begin{bmatrix} -B_p^{-1}\hat{B}_f & 0 \\ 0 & I_{2\times 2} \end{bmatrix}^T S^p \begin{bmatrix} -B_p^{-1}\hat{B}_f & 0 \\ 0 & I_{2\times 2} \end{bmatrix} = \hat{S}^f + \hat{S}^p,$$

where the matrix $\hat{S}$ is applied to vectors of the form $\begin{bmatrix} u_\Gamma^b & p_0^f & p_0^p \end{bmatrix}^T$. Note that $\hat{B}_f$ and B_p are diagonal matrices of the same dimension and are spectrally equivalent. We introduce the following preconditioner operator

$$\hat{S}_N^{-1} = \hat{Q}_0 + (I - \hat{Q}_0\hat{S})(\hat{S}^p)^{-1}(I - \hat{S}\hat{Q}_0). \tag{10}$$

Using the same arguments as before we prove:

Proposition 2 *If u_Γ^b is a balanced function then*

$$\langle \hat{S}_\Gamma^p u_\Gamma^b, u_\Gamma^b\rangle \leq \langle \hat{S}_\Gamma u_\Gamma^b, u_\Gamma^b\rangle \preceq (1 + \frac{\kappa}{h_p^2})\langle \hat{S}_\Gamma^p u_\Gamma^b, u_\Gamma^b\rangle.$$

Proof. Let $v_\Gamma^b = \sum_{j=1}^{M_p} \beta_j \phi_f^j$. And notice that the support of the basis functions ϕ_f^j do not overlap each other on Γ. We have:

$$\|v_\Gamma^b\|_{L^2(\Gamma)}^2 = \sum_{j=1}^{M_p} \beta_j^2 \|\phi_f^j\|_{L^2(\Gamma)}^2 \asymp h_p \sum_{j=1}^{M_p} \beta_j^2,$$

and using $H_{00}^{1/2}$ arguments on the intervals e_p^j we have

$$\|v_\Gamma^b\|_{H_{00}^{1/2}(\Gamma)}^2 \preceq \sum_{j=1}^{M_p} \beta_j^2 \|\phi_f^j\|_{H_{00}^{1/2}(e_p^j)}^2 \asymp \sum_{j=1}^{M_p} \beta_j^2.$$

Note that, by considering $\boldsymbol{v}_\Gamma^f = v_\Gamma^b$, we have

$$\langle \hat{S}^f v^b, v^b\rangle \leq a_f(\mathcal{SH}\boldsymbol{v}_\Gamma^f, \mathcal{SH}\boldsymbol{v}_\Gamma^f) \asymp \mu \|\boldsymbol{v}_f r_\Gamma\|_{H_{00}^{1/2}(\Gamma)}^2,$$

since the space for discrete Stokes harmonic extension now is richer (includes also $\boldsymbol{X}_n^f$) than in $\mathcal{SH}$, and we also use the equivalence results between discrete Stokes and Laplacian harmonic extensions. We obtain

$$\langle \hat{S}_\Gamma^f v^b, v^b\rangle \preceq \frac{\mu}{h_p}\|v_\Gamma^b\|_{L^2(\Gamma)}^2 \preceq \frac{\mu}{h_p^2}\mu\|\Pi v_\Gamma^b\|_{(H^{1/2})'(\Gamma)}^2 \asymp \frac{\kappa}{h_p^2}\langle \hat{S}_\Gamma^p v^b, v^b\rangle,$$

where we have used an inverse inequality for piecewise constant functions.

References

1. M. Discacciati, *Iterative methods for Stokes/Darcy coupling*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., vol. 40 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 563–570.
2. M. Discacciati, E. Miglio, and A. Quarteroni, *Mathematical and numerical modeling for coupling surface and groundwater flows*, Appl. Numer. Math., 43 (2002), pp. 57–74.
3. M. Discacciati and A. Quarteroni, *Analysis of a domain decomp. method for the coupling for the Stokes and Darcy equations*, in Numerical analysis and advanced applications – Proceedings of ENUMATH 2001, F. Brezzi, A. Buffa, S. Corsaro, and A. Murli, eds., Springer-Verlag Italia, 2003.

4. ———, *Convergence analysis of a subdomain iterative method for the finite element approximation of the coupling of Stokes and Darcy equations*, Comput. Vis. Sci., 6 (2004), pp. 93–103.
5. M. Dryja and W. Proskurowski, *On preconditioners for mortar discretization of elliptic problems*, Numer. Linear Algebra Appl., 10 (2003), pp. 65–82.
6. J. Galvis and M. Sarkis, *Inf-sup conditions and discrete error analysis for a non-matching mortar discretization for coupling Stokes-Darcy equations.* Submitted, 2006.
7. J. C. Galvis and M. Sarkis, *Inf-sup for coupling Stokes-Darcy*, in Proceedings of the XXV Iberian Latin American Congress in Computational Methods in Engineering, A. L. et al., ed., Universidade Federal de Pernambuco, 2004.
8. W. J. Layton, F. Schieweck, and I. Yotov, *Coupling fluid flow with porous media flow*, SIAM J. Num. Anal., 40 (2003), pp. 2195–2218.
9. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.
10. T. P. Mathew, *Domain Decomposition and Iterative Refinement Methods for Mixed Finite Element Discretizations of Elliptic Problems*, PhD thesis, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, September 1989.
11. L. F. Pavarino and O. B. Widlund, *Balancing Neumann-Neumann methods for incompressible Stokes equations*, Comm. Pure Appl. Math., 55 (2002), pp. 302–335.
12. B. M. Rivière and I. Yotov, *Locally conservative coupling of Stokes and Darcy flows*, SIAM J. Numer. Anal., 42 (2005), pp. 1959–1977.

A FETI-DP Formulation for Compressible Elasticity with Mortar Constraints

Hyea Hyun Kim

Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY10012, USA. hhk2@cims.nyu.edu

Summary. A FETI-DP formulation for three-dimensional elasticity problems on non-matching grids is considered. To resolve the nonconformity of the finite elements, a mortar matching condition is imposed on subdomain interfaces. The mortar matching condition are considered as weak continuity constraints in the FETI-DP formulation. A relatively large set of primal constraints, which include average and moment constraints over interfaces (faces) as well as vertex constraints, is further introduced to achieve a scalable FETI-DP method. A condition number bound, $C(1+\log(H/h))^2$, for the FETI-DP formulation with a Neumann-Dirichlet preconditioner is then proved for elasticity problems with discontinuous material parameters when the primal constraints are enforced on only some of the faces instead of all of them. These faces are called primal faces. An algorithm for selecting a quite small number of primal faces is described in [6].

1 A model problem

Let Ω be a polyhedral domain in $\mathbf{R}^3$. The space $H^1(\Omega)$ is the set of functions in $L^2(\Omega)$ that are square integrable up to first weak derivatives and equipped with the standard Sobolev norm: $\|v\|_{1,\Omega}^2 := |v|_{1,\Omega}^2 + \|v\|_{0,\Omega}^2$, where $|v|_{1,\Omega}^2 = \displaystyle\int_\Omega \nabla v \cdot \nabla v \, dx$ and $\|v\|_{0,\Omega} = \displaystyle\int_\Omega v^2 \, dx$. We assume that $\partial\Omega$ is divided into two parts $\partial\Omega_D$ and $\partial\Omega_N$ on which a Dirichlet boundary condition and a natural boundary condition are specified, respectively. The subspace $H_D^1(\Omega) \subset H^1(\Omega)$ is a set of functions having zero trace on $\partial\Omega_D$. For the elasticity problem, we introduce the vector-valued Sobolev spaces

[‡] This work was supported in part by the Applied Mathematical Sciences Program of the U.S. Department of Energy under contract DE-FG02-00ER25053 and in part by the Post-doctoral Fellowship Program of Korea Science and Engineering Foundation (KOSEF)

$$\mathbf{H}^1_D(\Omega) = \prod_{i=1}^{3} H^1_D(\Omega), \quad \mathbf{H}^1(\Omega) = \prod_{i=1}^{3} H^1(\Omega)$$

equipped with the product norm.

We consider the following variational form of the compressible elasticity problem: find $\mathbf{u} \in \mathbf{H}^1_D(\Omega)$ such that

$$\int_\Omega G(\mathbf{x})\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v})\, d\mathbf{x} + \int_\Omega G(\mathbf{x})\beta(\mathbf{x})\nabla \cdot \mathbf{u}\, \nabla \cdot \mathbf{v}\, d\mathbf{x} = \langle \mathbf{F}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathbf{H}^1_D(\Omega), \quad (1)$$

where $G = E/(1+\nu)$ and $\beta = \nu/(1-2\nu)$ are material parameters depending on the Young's modulus $E > 0$ and the Poisson ratio $\nu \in (0, 1/2]$ bounded away from $1/2$. The linearized strain tensor is defined by

$$\varepsilon(\mathbf{u})_{ij} := \frac{1}{2}\left(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\right) \quad i, j = 1, 2, 3,$$

and the tensor product and the force term are given by

$$\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) = \sum_{i,j=1}^{3} \varepsilon_{ij}(\mathbf{u})\varepsilon_{ij}(\mathbf{v}), \quad \langle \mathbf{F}, \mathbf{v} \rangle = \int_\Omega \mathbf{f} \cdot \mathbf{v}\, d\mathbf{x} + \int_{\partial\Omega_N} \mathbf{g} \cdot \mathbf{v} d\sigma.$$

Here $\mathbf{f}$ is the body force and $\mathbf{g}$ is the surface force on the natural boundary part $\partial\Omega_N$.

The space $\mathbf{ker}(\varepsilon)$ has the following six rigid body motions as its basis, which are three translations

$$\mathbf{r}_1 = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix}, \ \mathbf{r}_2 = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix}, \ \mathbf{r}_3 = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}, \quad (2)$$

and three rotations

$$\mathbf{r}_4 = \frac{1}{H}\begin{pmatrix} x_2 - \widehat{x}_2 \\ -x_1 + \widehat{x}_1 \\ 0 \end{pmatrix}, \ \mathbf{r}_5 = \frac{1}{H}\begin{pmatrix} -x_3 + \widehat{x}_3 \\ 0 \\ x_1 - \widehat{x}_1 \end{pmatrix}, \ \mathbf{r}_6 = \frac{1}{H}\begin{pmatrix} 0 \\ x_3 - \widehat{x}_3 \\ -x_2 + \widehat{x}_2 \end{pmatrix}. \quad (3)$$

Here $\widehat{\mathbf{x}} = (\widehat{x}_1, \widehat{x}_2, \widehat{x}_3) \in \Omega$ and H is the diameter of Ω. This shift and the scaling make the L_2-norm of the six vectors scale in the same way with H.

2 FETI-DP formulation

2.1 Finite elements and mortar matching condition

We divide the domain Ω into a geometrically conforming partition $\{\Omega_i\}_{i=1}^N$ and we assume that the coefficients $G(\mathbf{x})$ and $\beta(\mathbf{x})$ are positive constants in each subdomain

$$G(\mathbf{x})|_{\Omega_i} = G_i, \quad \beta(\mathbf{x})|_{\Omega_i} = \beta_i.$$

Since we confine our study to the compressible elasticity problem, we can associate the conforming P_1-finite element space $\mathbf{X}_i$ to a quasi-uniform triangulation τ_i of each subdomain Ω_i. In addition, functions in the space $\mathbf{X}_i$ satisfy the Dirichlet boundary condition on $\partial\Omega_i\cap\partial\Omega_D$. The triangulations $\{\tau_i\}_{i=1}^N$ may not match across subdomain interfaces. We associate the finite element space $\mathbf{W}_i$ to the boundary of subdomain Ω_i; it is the trace space of $\mathbf{X}_i$ on $\partial\Omega_i$. Throughout this paper, we will use H_i and h_i to denote the diameter of Ω_i and the typical mesh size of τ_i, respectively.

For each interface (face) $F^{ij} = \partial\Omega_i \cap \partial\Omega_j$, we will choose the one with larger $G(\mathbf{x})$ as the mortar side and the other as the nonmortar side. We then introduce the finite element space on the interface F^{ij}

$$\mathbf{W}_{ij} = \left\{\mathbf{w} \in \mathbf{H}_0^1(F^{ij}) \;:\; \mathbf{w} = \mathbf{v}|_{F^{ij}} \text{ for } \mathbf{v} \in \mathbf{X}_{n(ij)}\right\},$$

where $n(ij)$ denotes the nonmortar side. A Lagrange multiplier space $\mathbf{M}_{ij}$, which depends on the space $\mathbf{W}_{ij}$ is given. We refer to [4] for the detailed construction of the dual Lagrange multiplier space and to [1] for the standard Lagrange multiplier space. The mortar matching condition is written as

$$\int_{F_{ij}} (\mathbf{v}_i - \mathbf{v}_j)\cdot\boldsymbol{\lambda}\, ds = 0 \quad \forall\boldsymbol{\lambda}\in\mathbf{M}_{ij},\ \forall F_{ij}. \tag{4}$$

For each subdomain Ω_i, we define the set m_i containing the subdomain indices j that are mortar sides of interfaces $F\subset\partial\Omega_i$:

$$m_i := \{j \;:\; \Omega_i \text{ is the nonmortar side of } F(:=\partial\Omega_i\cap\partial\Omega_j)\ \forall F\subset\partial\Omega_i\}.$$

We then introduce the finite element spaces on the interfaces

$$\mathbf{W} = \prod_{i=1}^N \mathbf{W}_i, \quad \mathbf{W}_n = \prod_{i=1}^N\prod_{j\in m_i}\mathbf{W}_{ij}, \quad \mathbf{M} = \prod_{i=1}^N\prod_{j\in m_i}\mathbf{M}_{ij}.$$

2.2 Primal constraints

Selection of primal constraints is important in achieving scalability of FETI-DP algorithms as well as making each subdomain problem invertible. FETI-DP algorithms have been developed for elasticity problems with conforming discretization [2] and numerical results in [3] further show that primal constraints with faces average and vertex constraints provide a scalable algorithm for three dimensional problems. Klawonn and Widlund [8] considered various types of primal constraints for elasticity problems with discontinuous coefficients. Their primal constraints are edge average and edge moment constraints, and vertex constraints. Furthermore, they introduced the concepts of an acceptable face path and an acceptable vertex path in an attempt to reduce the number of primal constraints. For the case of mortar constraints, we are able to construct primal constraints based on faces. Thus, in [5], we introduce face average constraints for three-dimensional elliptic problems with mortar discretizations and show that the condition number is bounded by a polylogarithmic function of the subdomain problem size independently of the mesh parameters and the coefficients.

We will now select primal constraints on each face for the elasticity problems with mortar discretization. For an interface F^{ij}, we consider the rigid body motions $\{\mathbf{r}_i\}_{i=1}^6$ as in (2) and (3), where H is the diameter of the interface F^{ij} and $\widehat{\mathbf{x}}$ is a point in F^{ij}. We define a projection $\mathbf{Q}:\mathbf{H}^{1/2}(F^{ij})\to\mathbf{M}_{ij}$ by

$$\int_{F^{ij}} (\mathbf{Q}(\mathbf{w}) - \mathbf{w}) \cdot \boldsymbol{\phi} \, ds = 0 \quad \forall \boldsymbol{\phi} \in \mathbf{W}_{ij}.$$

We then construct the projected rigid body motions $\{\mathbf{Q}(\mathbf{r}_i)\}_{i=1}^{6}$. Since the space $\mathbf{M}_{ij}$ contains the translational rigid body motions, $\mathbf{Q}(\mathbf{r}_i) = \mathbf{r}_i$ for $i = 1, 2, 3$. We now consider the following constraints on the face F^{ij}

$$\int_{F^{ij}} (\mathbf{v}_i - \mathbf{v}_j) \cdot \mathbf{Q}(\mathbf{r}_l) \, ds = 0 \quad \forall l = 1, \cdots, 6.$$

For $\{\mathbf{Q}(\mathbf{r}_l)\}_{l=1}^{3}$, these constraints are nothing but the average matching conditions across the interface (face). The remaining constraints with $\{\mathbf{Q}(\mathbf{r}_l)\}_{l=4}^{6}$ are similar to the moment matching constraints which were introduced for fully primal edges in [7] except that our constraints use the projected rotations and are imposed on faces. We call $\{\mathbf{Q}(\mathbf{r}_l)\}_{l=4}^{6}$ the moment constraints.

To reduce the size of the coarse problem, we select only some faces as primal among all the faces and we impose the primal constraints over only them. For the remaining (non-primal faces), we assume that they satisfy an acceptable face path condition. This assumption makes it possible for the FETI-DP method to have a condition number bound comparable to when all faces are chosen to be primal.

Definition 1. (Acceptable face path) *For a pair of subdomains (Ω_i, Ω_j) having the common face F^{ij} with $G_i \leq G_j$, an acceptable face path is a path $\{\Omega_i, \Omega_{k_1}, \cdots, \Omega_{k_n}, \Omega_j\}$ from Ω_i to Ω_j such that the coefficient G_{k_l} of Ω_{k_l} satisfy the conditions*

$$TOL * (1 + log(H_i/h_i))^{-1} \, (1 + log(H_{k_l}/h_{k_l}))^2 * G_{k_l} \geq G_i \tag{5}$$

and the path from one subdomain to another is always through a primal face.

Furthermore, we choose some of the vertices as primal vertices at which we impose a pointwise matching condition. We assume that enough primal vertices are taken so as to make each local problem invertible. Based on these primal constraints, we introduce the following subspaces

$$\begin{aligned} \widetilde{\mathbf{W}} : = \{\mathbf{w} \in \mathbf{W} \ : \ & \mathbf{w} \text{ satisfies vertex constraints at the primal vertices} \\ & \text{and the face constraints across the primal faces}\}, \\ \widetilde{\mathbf{W}}_n : = \{\mathbf{w}_n \in \mathbf{W}_n \ : \ & \mathbf{w}_n \text{ satisfies zero average and zero moment} \\ & \text{constraints for each primal faces}\}. \end{aligned}$$

For $\mathbf{w}_n \in \widetilde{\mathbf{W}}_n$, let $E(\mathbf{w}_n) \in \mathbf{W}$ be the zero extension of $\mathbf{w}_n$ to the whole interface, i.e., mortar and nonmortar interfaces. We can easily see that $E(\mathbf{w}_n)$ belongs to $\widetilde{\mathbf{W}}$.

2.3 The FETI-DP equation

Let $A^{(i)}$ denote the stiffness matrix of the bilinear form

$$a_i(\mathbf{u}_i, \mathbf{v}_i) := G_i \int_{\Omega_i} \varepsilon(\mathbf{u}_i) : \varepsilon(\mathbf{v}_i) \, dx + G_i \beta_i \int_{\Omega_i} \nabla \cdot \mathbf{u}_i \nabla \cdot \mathbf{v}_i \, dx,$$

and let $S^{(i)}$ be the Schur complement of the matrix $A^{(i)}$. The matrix $B^{(i)}$ denotes the mortar matching matrix for the unknowns of $\partial\Omega_i$ and the mortar matching condition for $\mathbf{w} = (\mathbf{w}_1, \cdots, \mathbf{w}_N) \in \mathbf{W}$ can then be written as

$$\sum_{i=1}^{N} B^{(i)} \mathbf{w}_i = 0.$$

Let V_c be the set of unknowns at the primal vertices, let $V_c^{(i)}$ be the restriction of V_c on the subdomain Ω_i, and let the mapping $R_c^{(i)} : V_c \to V_c^{(i)}$ denote a restriction. The matrix $B^{(i)}$ and the vector $\mathbf{w}_i \in \mathbf{W}_i$ are ordered as

$$B^{(i)} = \begin{pmatrix} B_r^{(i)} & B_c^{(i)} \end{pmatrix}, \quad \mathbf{w}_i = \begin{pmatrix} \mathbf{w}_r^{(i)} \\ \mathbf{w}_c^{(i)} \end{pmatrix},$$

where c stands for the unknowns at the primal vertices in $V_c^{(i)}$ and r stands for the remaining unknowns. We then assemble vectors and matrices of each subdomains

$$\mathbf{w}_r = \begin{pmatrix} \mathbf{w}_r^{(1)} \\ \vdots \\ \mathbf{w}_r^{(N)} \end{pmatrix}, \quad B_r = \begin{pmatrix} B_r^{(1)} & \dots & B_r^{(N)} \end{pmatrix}, \quad B_c = \sum_{i=1}^{N} B_c^{(i)} R_c^{(i)}.$$

Since the primal face constraints are the mortar constraints, we express them by using an appropriate matrix R

$$R^t (B_r \mathbf{w}_r + B_c \mathbf{w}_c) = 0,$$

where $\mathbf{w}_c$ represents the unknowns at the global primal vertices.

By introducing Lagrange multipliers $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ for the primal face constraints and for the mortar matching constraints, respectively, we get the following mixed formulation of (1)

$$\begin{pmatrix} S_{rr} & S_{rc} & B_r^t R & B_r^t \\ S_{cr} & S_{cc} & B_c^t R & B_c^t \\ R^t B_r & R^t B_c & 0 & 0 \\ B_r & B_c & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbf{w}_r \\ \mathbf{w}_c \\ \boldsymbol{\mu} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{g}_r \\ \mathbf{g}_c \\ 0 \\ 0 \end{pmatrix}.$$

We now eliminate all the unknowns except $\boldsymbol{\lambda}$ and obtain

$$F_{DP} \boldsymbol{\lambda} = \mathbf{d}.$$

This matrix F_{DP} satisfies the well-known relation

$$\langle F_{DP} \boldsymbol{\lambda}, \boldsymbol{\lambda} \rangle = \max_{\mathbf{w} \in \widetilde{\mathbf{W}}} \frac{\langle B\mathbf{w}, \boldsymbol{\lambda} \rangle^2}{\langle S\mathbf{w}, \mathbf{w} \rangle},$$

where

$$S = \operatorname{diag}(S^{(i)}), \quad B = \begin{pmatrix} B^{(1)} & \dots & B^{(N)} \end{pmatrix}.$$

We now introduce the Neumann-Dirichlet preconditioner M^{-1} given by

$$\langle M \boldsymbol{\lambda}, \boldsymbol{\lambda} \rangle = \max_{\mathbf{w}_n \in \widetilde{\mathbf{W}}_n} \frac{\langle BE(\mathbf{w}_n), \boldsymbol{\lambda} \rangle^2}{\langle SE(\mathbf{w}_n), E(\mathbf{w}_n) \rangle},$$

where $E(\mathbf{w}_n)$ is the zero extension of $\mathbf{w}_n$ into the space $\mathbf{W}$. From the fact that $E(\mathbf{w}_n)$ belongs to $\widetilde{\mathbf{W}}$ for $\mathbf{w}_n \in \widetilde{\mathbf{W}}_n$, we obtain

$$\langle M \boldsymbol{\lambda}, \boldsymbol{\lambda} \rangle = \max_{\mathbf{w}_n \in \widetilde{\mathbf{W}}_n} \frac{\langle BE(\mathbf{w}_n), \boldsymbol{\lambda} \rangle^2}{\langle SE(\mathbf{w}_n), E(\mathbf{w}_n) \rangle} \le \max_{\mathbf{w} \in \widetilde{\mathbf{W}}} \frac{\langle B\mathbf{w}, \boldsymbol{\lambda} \rangle^2}{\langle S\mathbf{w}, \mathbf{w} \rangle} = \langle F_{DP} \boldsymbol{\lambda}, \boldsymbol{\lambda} \rangle. \quad (6)$$

Therefore the lower bound of the FETI-DP operator is bounded from below by 1.

3 Condition number analysis

In the following, we will provide several lemmas that will be used to obtain the upper bound of the FETI-DP operator. For a face $F \subset \partial\Omega_i$, the space $H^{1/2}_{00}(F)$ consists of the functions whose zero extension onto the whole boundary $\partial\Omega_i$ belongs to the space $H^{1/2}(\partial\Omega_i)$ and it is equipped with the norm

$$\|v\|_{H^{1/2}_{00}(F)} := \left(|v|^2_{H^{1/2}(F)} + \int_F \frac{v(x)^2}{\text{dist}(x, \partial F)}\, ds \right)^{1/2}.$$

We note that we can extend this norm to the product space $\mathbf{H}^{1/2}_{00}(F) := [H^{1/2}_{00}(F)]^3$ by using the usual product norm. We now provide several inequalities for the mortar projection of functions.

Definition 2. *(***Mortar projection***) The mortar projection $\pi_{ij} : \mathbf{L}^2(F^{ij}) \to \mathbf{W}_{ij}$ is given by*

$$\int_{F^{ij}} (\pi_{ij}(\mathbf{v}) - \mathbf{v}) \cdot \boldsymbol{\psi}\, ds = 0 \quad \forall \boldsymbol{\psi} \in \mathbf{M}_{ij}.$$

Lemma 1. *For $F^{ij} (= \partial\Omega_i \cap \partial\Omega_j)$, a primal face with $G_i \le G_j$, and for $\mathbf{w} \in \widetilde{\mathbf{W}}$, we have*

$$G_i \|\pi_{ij}(\mathbf{w}_i - \mathbf{w}_j)\|^2_{H^{1/2}_{00}(F^{ij})} \le C \left\{ \left(1 + log\frac{H_i}{h_i}\right)^2 |\mathbf{w}_i|^2_{S_i} \right. \\ \left. + \frac{G_i}{G_j} \left(1 + log\frac{H_j}{h_j}\right) \left(1 + log\frac{H_j}{h_j} + \frac{h_j}{h_i}\right) |\mathbf{w}_j|^2_{S_j} \right\},$$

where $|\mathbf{w}_l|^2_{S_l} = \langle S_l \mathbf{w}_l, \mathbf{w}_l \rangle$ for $l = i, j$.

Lemma 2. *For a non-primal face $F = \partial\Omega_i \cap \partial\Omega_j$ with $G_i \le G_j$, assume that there is an acceptable face path $\{\Omega_i, \Omega_{k_1}, \cdots, \Omega_{k_n}, \Omega_j\}$. Then, for $\mathbf{w} \in \widetilde{\mathbf{W}}$, we have*

$$G_i \|\pi_{ij}(\mathbf{w}_i - \mathbf{w}_j)\|^2_{H^{1/2}_{00}(F)} \le C \left\{ \left(1 + log\frac{H_i}{h_i}\right)^2 |\mathbf{w}_i|^2_{S_i} \right. \\ + L * \sum_{l=1}^{n} \left(1 + log\frac{H_i}{h_i}\right) \frac{G_i}{G_{k_l}} |\mathbf{w}_{k_l}|^2_{S_{k_l}} \\ \left. + \frac{G_i}{G_j} \left(1 + log\frac{H_j}{h_j}\right) \left(1 + log\frac{H_j}{h_j} + \frac{h_j}{h_i}\right) |\mathbf{w}_j|^2_{S_j} \right\},$$

where $\mathbf{w}_i = \mathbf{w}|_{\partial\Omega_i}$, $\mathbf{w}_j = \mathbf{w}|_{\partial\Omega_j}$, and the constant L is the number of subdomains on the acceptable face path.

To bound the term $(G_i/G_j)(h_j/h_i)$ by a constant independent of mesh parameters, we need to impose an assumption on mesh sizes.

Assumption on mesh sizes. For subdomains Ω_i and Ω_j that have a common face F with $G_i \le G_j$, the mesh sizes h_i and h_j satisfy

$$\frac{h_j}{h_i} \le C \left(\frac{G_j}{G_i}\right)^{\gamma} \text{ for some } 0 \le \gamma \le 1. \tag{7}$$

By combining Lemmas 1 and 2 with the assumption on the mesh sizes and the acceptable face path condition (5), we have the following upper bound for the FETI-DP operator.

Lemma 3. *Assume that the mesh sizes satisfy the assumption* (7) *and that every non-primal face satisfies the acceptable face path condition with given TOL and L. We then have*

$$\langle F_{DP}\boldsymbol{\lambda},\boldsymbol{\lambda}\rangle^2 = \max_{\mathbf{w}\in\widetilde{\mathbf{W}}} \frac{\langle B\mathbf{w},\boldsymbol{\lambda}\rangle^2}{\langle S\mathbf{w},\mathbf{w}\rangle} \le C(TOL,L) \max_{i=1,\cdots,N} \left\{ \left(1 + log\frac{H_i}{h_i}\right)^2 \right\} \langle M\boldsymbol{\lambda},\boldsymbol{\lambda}\rangle,$$

where the constant C depends on the TOL and L but not on the mesh parameters and the coefficients G_i.

The lower bound in (6) and the upper bound from Lemma 3 lead to the following condition number bound.

Theorem 1. *Under the assumptions in Lemma 3, we obtain the condition number bound*

$$\kappa(M^{-1}F_{DP}) \le C(TOL,L) \max_{i=1,\cdots,N} \left\{ \left(1 + log\frac{H_i}{h_i}\right)^2 \right\}.$$

Here the constant C is independent of the mesh parameters and the coefficients G_i, but depends on TOL and L, the maximum face path length.

Acknowledgement. The author is deeply grateful to Professor Olof B. Widlund for his strong support and valuable discussions.

References

1. F. B. Belgacem and Y. Maday, *The mortar element method for three dimensional finite elements*, Math. Model. Numer. Anal., 31 (1997), pp. 289–302.
2. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.
3. C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numer. Lin. Alg. Appl., 7 (2000), pp. 687–714.
4. C. Kim, R. Lazarov, J. Pasciak, and P. Vassilevski, *Multiplier spaces for the mortar finite element method in three dimensions*, SIAM J. Numer. Anal., 39 (2001), pp. 519–538.
5. H. H. Kim, *A preconditioner for the FETI-DP formulation with mortar methods in three dimensions*, Tech. Rep. 04-19, Division of Applied Mathematics, Korea Advanced Instititue of Science and Technology, 2004.
6. ———, *A FETI-DP formulation of three dimensional elasticity problems with mortar discretization*, Tech. Rep. 863, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, April 2005.

7. A. Klawonn and O. Widlund, *Dual-Primal FETI methods for linear elasticity*, Tech. Rep. 855, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, September 2004.
8. A. Klawonn and O. B. Widlund, *FETI and Neumann–Neumann iterative substructuring methods: Connections and new results*, Comm. Pure Appl. Math., 54 (2001), pp. 57–90.

Some Computational Results for Robust FETI-DP Methods Applied to Heterogeneous Elasticity Problems in 3D

Axel Klawonn and Oliver Rheinbach

Fachbereich Mathematik, Universität Duisburg-Essen, Universitätsstr. 3, D-45117 Essen, Germany. axel.klawonn@uni-essen.de,oliver.rheinbach@uni-essen.de

1 Introduction

Robust FETI-DP methods for heterogeneous, linear elasticity problems in three dimensions were developed and analyzed in [7]. For homogeneous problems or materials with only small jumps in the Young moduli, the primal constraints can be chosen as edge averages of the displacement components over well selected edges; see [7] and for numerical experimental work, [5]. In the case of large jumps in the material coefficients, first order moments were introduced as additional primal constraints in [7], in order to obtain a robust condition number bound. In the present article, we provide some first numerical results which confirm the theoretical findings in [7] and show that in some cases, first order moments are necessary to obtain a good convergence rate.

2 Linear elasticity and finite elements

The equations of linear elasticity model the displacement of a linear elastic material under the action of external and internal forces. The elastic body occupies a domain $\Omega \subset \mathbb{R}^3$, which is assumed to be polyhedral and of diameter one. We denote its boundary by $\partial\Omega$ and assume that one part of it, $\partial\Omega_D$, is clamped, i.e., with homogeneous Dirichlet boundary conditions, and that the rest, $\partial\Omega_N := \partial\Omega \setminus \partial\Omega_D$, is subject to a surface force $\mathbf{g}$, i.e., a natural boundary condition. We can also introduce a body force $\mathbf{f}$, e.g., gravity. With $\mathbf{H^1}(\Omega) := (H^1(\Omega))^3$, the appropriate space for a variational formulation is the Sobolev space $\mathbf{H_0^1}(\Omega, \partial\Omega_D) := \{\mathbf{v} \in \mathbf{H^1}(\Omega) : \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}$. The linear elasticity problem consists in finding the displacement $\mathbf{u} \in \mathbf{H_0^1}(\Omega, \partial\Omega_D)$ of the elastic body Ω, such that

$$\int_\Omega G(\mathbf{x})\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) d\mathbf{x} + \int_\Omega G(\mathbf{x})\, \beta(\mathbf{x})\, \mathrm{div}\mathbf{u}\ \mathrm{div}\mathbf{v}\, d\mathbf{x} = \langle \mathbf{F}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathbf{H_0^1}(\Omega, \partial\Omega_D). \quad (1)$$

Here G and β are material parameters which depend on the Young modulus $E > 0$ and the Poisson ratio $\nu \in (0, 1/2]$; we have $G = E/(1+\nu)$ and $\beta = \nu/(1-2\nu)$. In this article, we only consider the case of compressible elasticity, which means that the

Poisson ratio ν is bounded away from 1/2. Furthermore, $\varepsilon_{ij}(\mathbf{u}) := \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$ is the linearized strain tensor, and

$$\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) = \sum_{i,j=1}^{3} \varepsilon_{ij}(\mathbf{u})\varepsilon_{ij}(\mathbf{v}), \quad \langle \mathbf{F}, \mathbf{v} \rangle := \int_{\Omega} \mathbf{f}^T \mathbf{v}\, d\mathbf{x} \; + \; \int_{\partial\Omega_N} \mathbf{g}^T \mathbf{v}\, d\sigma.$$

For convenience, we also introduce the notation

$$(\varepsilon(\mathbf{u}), \varepsilon(\mathbf{v}))_{L_2(\Omega)} := \int_{\Omega} \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) d\mathbf{x}.$$

The bilinear form associated with linear elasticity is then

$$a(\mathbf{u}, \mathbf{v}) = (G\, \varepsilon(\mathbf{u}), \varepsilon(\mathbf{v}))_{L_2(\Omega)} \; + \; (G\, \beta\, \mathrm{div}\mathbf{u}, \mathrm{div}\mathbf{v})_{L_2(\Omega)}.$$

The wellposedness of the linear system (1) follows immediately from the continuity and ellipticity of the bilinear form $a(\cdot, \cdot)$, where the first follows from elementary inequalities and the latter from Korn's first inequality; see, e.g., [2]. The null space $\ker(\varepsilon)$ of ε is the space of the six rigid body motions. which is spanned by the three translations $\mathbf{r}_i := \mathbf{e}_i, i = 1, 2, 3$, where the $\mathbf{e}_i$ are the three standard unit vectors, and the three rotations

$$\mathbf{r}_4 := \begin{bmatrix} x_2 - \hat{x}_2 \\ -x_1 + \hat{x}_1 \\ 0 \end{bmatrix}, \mathbf{r}_5 := \begin{bmatrix} -x_3 + \hat{x}_3 \\ 0 \\ x_1 - \hat{x}_1 \end{bmatrix}, \mathbf{r}_6 := \begin{bmatrix} 0 \\ x_3 - \hat{x}_3 \\ -x_2 + \hat{x}_2 \end{bmatrix}. \tag{2}$$

Here $\hat{\mathbf{x}} \in \Omega$ to shift the origin to a point in Ω.

We will only consider compressible elastic materials. It is therefore sufficient to discretize our elliptic problem of linear elasticity (1) by low order, conforming finite elements, e.g., linear or trilinear elements.

Let us assume that a triangulation τ^h of Ω is given which is shape regular and has a typical diameter of h. We denote by $\mathbf{W}^h := \mathbf{W}^h(\Omega)$ the corresponding conforming finite element space of finite element functions. The associated discrete problem is then

$$a(\mathbf{u}_h, \mathbf{v}_h) = \langle \mathbf{F}, \mathbf{v}_h \rangle \quad \forall \mathbf{v}_h \in \mathbf{W}^h. \tag{3}$$

When there is no risk of confusion, we will drop the subscript h.

Let the domain $\Omega \subset \mathbb{R}^3$ be decomposed into nonoverlapping subdomains $\Omega_i, i = 1, \ldots, N$, each of which is the union of finite elements with matching finite element nodes on the boundaries of neighboring subdomains across the interface Γ. The interface Γ is the union of three different types of open sets, namely, subdomain faces, edges, and vertices; see [7] or [5] for a detailed definition. In the case of a decomposition into regular substructures, e.g., cubes or tetrahedra, our definition of faces, edges, and vertices is conform with our basic geometric intuition. In the definition of dual-primal FETI methods, we need the notion of edge averages, and in the case of heterogeneous materials, also of edge first order moments. We note that the rigid body modes $\mathbf{r}_1, \ldots, \mathbf{r}_6$, restricted to a straight edge provide only five linearly independent vectors, since one rotation is always linearly dependent on other rigid body modes. For the following definition, we assume that we have used an appropriate change of coordinates such that the edge under consideration coincides with the x_1-axis and the special rotation is then $\mathbf{r}_6$. The edge averages and first order moments over this specific edge $\mathcal{E}$ are of the form

$$\frac{\int_{\mathcal{E}} \mathbf{r}_k^T \mathbf{u} dx}{\int_{\mathcal{E}} \mathbf{r}^T \mathbf{r} dx}, \quad k \in \{1, \ldots, 5\}, \mathbf{u} = (u_1^T, u_2^T, u_3^T)^T \in \mathbf{W}^h. \tag{4}$$

3 The FETI-DP algorithm

For each subdomain $\Omega_i, i = 1, \ldots, N$, we assemble local stiffness matrices $K^{(i)}$ and local load vectors $\mathbf{f}^{(i)}$. By $\mathbf{u}^{(i)}$ we denote the local solution vectors of nodal values.

In the dual-primal FETI methods, we distinguish between dual and primal displacement variables by the way the continuity of the solution in those variables is established. Dual displacement variables are those, for which the continuity is enforced by a continuity constraint and Lagrange multipliers $\boldsymbol{\lambda}$ and thus, continuity is not established until convergence of the iterative method is reached, as in the classical one-level FETI methods; see, e.g., [8]. On the other hand, continuity of the primal displacement variables is enforced explicitly in each iteration step by subassembly of the local stiffness matrices $K^{(i)}$ at the primal displacement variables. This subassembly yields a symmetric, positive definite stiffness matrix $\widetilde{K}$ which is coupled at the primal displacement variables but block diagonal otherwise. Let us note that this coupling yields a global problem which is necessary to obtain a numerically scalable algorithm.

We will use subscripts I, Δ, and Π, to denote the interior, dual, and primal displacement variables, respectively, and obtain for the local stiffness matrices, load vectors, and solution vectors of nodal values

$$K^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Delta I}^{(i)T} & K_{\Pi I}^{(i)T} \\ K_{\Delta I}^{(i)} & K_{\Delta\Delta}^{(i)} & K_{\Pi\Delta}^{(i)T} \\ K_{\Pi I}^{(i)} & K_{\Pi\Delta}^{(i)} & K_{\Pi\Pi}^{(i)} \end{bmatrix}, \mathbf{u}^{(i)} = \begin{bmatrix} \mathbf{u}_I^{(i)} \\ \mathbf{u}_\Delta^{(i)} \\ \mathbf{u}_\Pi^{(i)} \end{bmatrix}, \mathbf{f}^{(i)} = \begin{bmatrix} \mathbf{f}_I^{(i)} \\ \mathbf{f}_\Delta^{(i)} \\ \mathbf{f}_\Pi^{(i)} \end{bmatrix}.$$

We also introduce the notation

$$\mathbf{u}_B = [\mathbf{u}_I \ \mathbf{u}_\Delta]^T, \mathbf{f}_B = [\mathbf{f}_I \ \mathbf{f}_\Delta]^T, \mathbf{u}_B^{(i)} = [\mathbf{u}_I^{(i)} \ \mathbf{u}_\Delta^{(i)}]^T, \text{ and } \mathbf{f}_B^{(i)} = [\mathbf{f}_I^{(i)} \ \mathbf{f}_\Delta^{(i)}]^T.$$

Accordingly, we define

$$K_{BB} = \operatorname{diag}_{i=1}^N (K_{BB}^{(i)}), \quad K_{BB}^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Delta I}^{(i)T} \\ K_{\Delta I}^{(i)} & K_{\Delta\Delta}^{(i)} \end{bmatrix}, \quad K_{\Pi B} = [K_{\Pi B}^{(1)} \ldots K_{\Pi B}^{(N)}].$$

We note that K_{BB} is a block diagonal matrix. By subassembly in the primal displacement variables, we obtain

$$\widetilde{K} = \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix},$$

where a tilde indicates the subassembled matrices and where

$$\widetilde{K}_{\Pi B} = [\widetilde{K}_{\Pi B}^{(1)} \cdots \widetilde{K}_{\Pi B}^{(N)}].$$

Introducing local assembly operators $R_\Pi^{(i)}$ which map from the local primal displacement variables $\mathbf{u}_\Pi^{(i)}$ to the global, assembled $\widetilde{\mathbf{u}}_\Pi$, we have

$$\widetilde{K}_{\Pi B}^{(i)} = R_{\Pi}^{(i)} K_{\Pi B}^{(i)}, \qquad \mathbf{u}_{\Pi}^{(i)} = R_{\Pi}^{(i)T} \widetilde{\mathbf{u}}_{\Pi}, \qquad i = 1, \ldots, N,$$
$$\widetilde{K}_{\Pi\Pi} \;=\; \sum_{i=1}^{N} R_{\Pi}^{(i)} K_{\Pi\Pi}^{(i)} R_{\Pi}^{(i)T}.$$

Due to the subassembly of the primal displacement variables, Lagrange multipliers have to be used only for the dual displacement variables $\mathbf{u}_{\Delta}$ to enforce continuity. We introduce a discrete jump operator $B = [O\, B_{\Delta}]$ such that the solution $\mathbf{u}_{\Delta}$, associated with more than one subdomain, coincides when $B\mathbf{u}_B = B_{\Delta}\mathbf{u}_{\Delta} = 0$ with $\mathbf{u}_B = [\mathbf{u}_I^T, \mathbf{u}_{\Delta}^T]^T$. Since we assume pointwise matching grids across the interface Γ, the entries of the matrix B are $0, 1$, and -1. However, we will otherwise use all possible constraints and thus work with a fully redundant set of Lagrange multipliers as in [8, Section 5]; cf. also [9]. Thus, for an edge node common to four subdomains, we will use six constraints rather than choosing as few as three.

We can now reformulate the finite element discretization of (3) as

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^T & B^T \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & O \\ B & O & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_B \\ \widetilde{\mathbf{u}}_{\Pi} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_B \\ \widetilde{\mathbf{f}}_{\Pi} \\ \mathbf{0} \end{bmatrix}. \tag{5}$$

Elimination of the primal variables $\widetilde{\mathbf{u}}_{\Pi}$ and of the interior and dual displacement variables $\mathbf{u}_B$ leads to a a reduced linear system of the form

$$F\boldsymbol{\lambda} = \mathbf{d},$$

where the matrix F and the right hand side $\mathbf{d}$ are formally obtained by block Gauss elimination. Let us note that the matrix F is never built explicitly but that in every iteration appropriate linear systems are solved; see [4], [7] or [5] for further details.

To define the FETI-DP Dirichlet preconditioner M^{-1}, we introduce a scaled jump operator B_D; this is done by scaling the contributions of B associated with the dual displacement variables from individual subdomains. We define

$$B_D = [B_D^{(1)}, \ldots, B_D^{(N)}],$$

where the $B_D^{(i)}$ are defined as follows: each row of $B^{(i)}$ with a nonzero entry corresponds to a Lagrange multiplier connecting the subdomain Ω_i with a neighboring subdomain Ω_j at a point $x \in \partial\Omega_{i,h} \cap \partial\Omega_{j,h}$. We obtain $B_D^{(i)}$ by multiplying each such row of $B^{(i)}$ with $1/|\mathcal{N}_x|$, where $|\mathcal{N}_x|$ denotes the multiplicity of the interface point $x \in \Gamma$. This scaling is called multiplicity scaling and is suitable for homogeneous problems; see [7] or [5] for a scaling suitable for heterogeneous materials. Our preconditioner is then given in matrix form by

$$M^{-1} = B_D R_{\Gamma}^T S R_{\Gamma} B_D^T = \sum_{i=1}^{N} B_D^{(i)} R_{\Gamma}^{(i)T} S^{(i)} R_{\Gamma}^{(i)} B_D^{(i)T}. \tag{6}$$

Here, $R_{\Gamma}^{(i)}$ are restriction matrices that restrict the degrees of freedom of a subdomain to its interface and $R_{\Gamma} = \operatorname{diag}_i(R_{\Gamma}^{(i)})$.

We have to decide how to choose the primal displacement variables. The simplest choice is to select them as certain primal vertices of the subdomains; see [3], where this approach was first considered; this version has been denoted by Algorithm A. Unfortunately, this choice does not always lead to good convergence results in three dimensions. To obtain better convergence for three dimensional problems, a different coarse problem was suggested by introducing additional constraints. These constraints are averages or first order moments over selected edges or faces, which are enforced to have the same values across the interface. For further details, see [4], [7], or [5]. To obtain robust condition number bounds for highly heterogeneous materials, additional first order moments over selected edges have to be used; cf. [7]. There are different ways of implementing these additional primal constraints. One is to use additional, optional Lagrange multipliers, see [4] or [7], another one is to apply a transformation of basis, see [7] and [5]. In this article, we will use the approach with a transformation of basis. Let us note that this approach leads again to a mixed linear system of the form (5) and that the same algorithmic form as for Algorithm A can be used; see [7], [5], and [6] for further details. For our FETI-DP algorithm, using a well selected set of primal constraints of edge averages or first order moments and in some very difficult cases also of primal vertices, we have the estimate, cf. [7],

Theorem 1. *The condition number satisfies*

$$\kappa(M^{-1}F) \leq C\,(1+\log(H/h))^2.$$

Here, $C > 0$ is independent of h, H, and the values of the coefficients G_i.

A more general result can be shown if the concept of acceptable paths is introduced; cf. [7] for more details.

4 Numerical results

We first consider a model problem, where two subdomains are surrounded by subdomains with much smaller stiffnesses, i.e., Young moduli. Furthermore, we assume that these two special subdomains share only an edge; cf. Figure 1. In [7] it was shown that a well selected set of primal constraints, which has five linearly independent primal constraints related to that special edge shared by the two stiffer subdomains and otherwise six linearly independent edge constraints for each face, is sufficient to prove a condition number bound as in Theorem 1. In that article, the five linearly independent constraints are chosen as three edge averages and two properly chosen first order moments; cf. also (4). Here, the six linearly independent constraints for each face can be chosen as edge averages (and moments) over appropriately chosen edges of the considered face. In a set of experiments, we have tested different combinations of edge constraints on the specific edge shared by the two stiffer subdomains; cf. Table 1. In the case of three constraints only edge averages are used, in the case of five, additionally two first order moments are applied. On all other edges, an edge average over each displacement component is used to define the primal constraints. We see that using no constraints or only edge average constraints on the specific edge leads to a large condition number. Applying all five constraints leads to a good condition number which is bounded independently of the jump in the Young moduli. Since we only have one difficult edge in this example, the iteration count is not increased accordingly; the eigenvalues are still well clustered except for two outliers in the case of three edge averages, see [6]. Next, we analyze a more involved example, where we will see that additional first order moments not only improve the condition number but are absolutely necessary to obtain convergence. We consider a linear elasticity model problem with a material

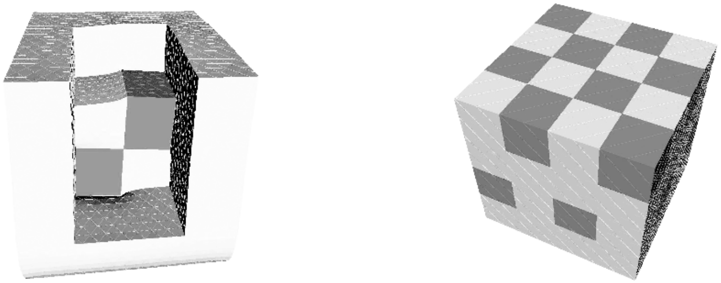


Fig. 1. Left: Two stiff cubic subdomains sharing an edge surrounded by softer material. Cubic domain Ω cut open in front and on top.
Right: Alternating layers of a heterogeneous material distributed in a checkerboard pattern and a homogeneous material.

Table 1. Comparison of different number of edge constraints on the edge shared by the two stiffer subdomains; $3 \times 4 \times 4 = 48$ brick-shaped subdomains of 1 536 d.o.f. each, 55 506 total d.o.f. Stopping criterion: Relative residual reduction of 10^{-10}.

# edge constraints	0		3		5	
E_1/E_2	Iter.	Cond.	Iter.	Cond.	Iter.	Cond.
10^0	29	9.21	28	9.10	**28**	**9.09**
10^3	47	4.36×10^2	37	7.51×10^1	**30**	**9.03**
10^6	70	4.24×10^5	47	7.16×10^4	**30**	**9.03**

consisting of different layers as shown on the right side in Figure 1. The ratio of the different Young moduli is $E_2/E_1 = 10^6$ with $E_2 = 210$ and a Poisson ratio of $\nu = 0.29$ for both materials. Here, in addition to three edge averages on each edge, we have also used two first order moments as primal constraints; see [7] and [6] for more details. The results clearly show that the additional first order moments help to improve the convergence significantly; see [7] for theoretical results. In Table 3 the parallel scalability is shown for a cube of eight layers; cf. Figure 1 (right). All computations were carried out using PETSc; see [1]. The numerical results given in Tables 2 and 3 were obtained on a 16 processor (2.2 Ghz Opteron 248; Gigabit Ethernet) computing cluster in Essen. A more detailed numerical study is current work in progress; cf. [6].

Table 2. Heterogeneous linear elasticity: Comparison of FETI-DP algorithm using edge averages vs. edge averages and first order moments; 1 728 cubic subdomains of 5 184 d.o.f. each, 7 057 911 total d.o.f. Stopping criterion: Relative residual reduction of 10^{-10}.

edge averages			edge averages + first order moments		
Cond.	Iter.	Time	Cond.	Iter.	Time
2.14×10^5	> **1 000**	> 6 686s	5.19	**24**	629s

Table 3. FETI-DP: Parallel scalability using edge averages and first order moments. Stopping criterion: Relative residual reduction of 10^{-7}.

Proc.	$\frac{\text{Subdom.}}{\text{Proc.}}$	$\frac{\text{d.o.f.}}{\text{Subdom.}}$	Total d.o.f.	Iter.	Cond.	Time
1	512	5 184	2 114 907	17	5.18	**1 828s**
2	256	5 184	2 114 907	17	5.18	**842s**
4	128	5 184	2 114 907	17	5.18	**428s**
8	64	5 184	2 114 907	17	5.18	**215s**
16	32	5 184	2 114 907	17	5.18	**122s**

References

1. S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, L. C. McInnes, and B. F. Smith, *PETSc home page*. http://www.mcs.anl.gov/petsc, 2001.

2. P. G. Ciarlet, *Mathematical Elasticity Volume I: Three–Dimensional Elasticity*, North-Holland, 1988.
3. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.
4. C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numer. Lin. Alg. Appl., 7 (2000), pp. 687–714.
5. A. Klawonn and O. Rheinbach, *A parallel implementation of Dual-Primal FETI methods for three dimensional linear elasticity using a transformation of basis*, Tech. Rep. SM-E-601, Department of Mathematics, University of Duisburg–Essen, Germany, February 2005.
6. ———, *Robust FETI-DP methods for heterogeneous elasticity problems*, Tech. Rep. SM-E-607, Department of Mathematics, University of Duisburg–Essen, Germany, July 2005.
7. A. Klawonn and O. Widlund, *Dual-Primal FETI methods for linear elasticity*, Tech. Rep. 855, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, September 2004.
8. A. Klawonn and O. B. Widlund, *FETI and Neumann–Neumann iterative substructuring methods: Connections and new results*, Comm. Pure Appl. Math., 54 (2001), pp. 57–90.
9. D. Rixen and C. Farhat, *A simple and efficient extension of a class of substructure based preconditioners to heterogeneous structural mechanics problems*, Int. J. Numer. Meth. Engrg., 44 (1999), pp. 489–516.

Dual-primal Iterative Substructuring for Almost Incompressible Elasticity

Axel Klawonn[1], Oliver Rheinbach[1], and Barbara Wohlmuth[2]

[1] Fachbereich Mathematik, Universität Duisburg-Essen, Campus Essen, Universitätsstraße 3, 45117 Essen, Germany.
axel.klawonn@uni-essen.de,oliver.rheinbach@uni-essen.de

[2] Institut für Angewandte Analysis und Numerische Simulation, Universität Stuttgart, Pfaffenwaldring 57, 70569 Stuttgart, Germany.
wohlmuth@ians.uni-stuttgart.de

1 Introduction

There exist a large number of publications devoted to the construction and analysis of finite element approximations for problems in solid mechanics, in which it is necessary to circumvent volumetric locking. Of special interest are nearly incompressible materials where standard low order finite element discretizations do not ensure uniform convergence in the incompressible limit. Methods associated with the enrichment or enhancement of the strain or stress field by the addition of carefully chosen basis functions have proved to be highly effective and popular. The key work dealing with enhanced assumed strain formulations is that of [14]. Of exclusive interest in our paper are situations corresponding to a pure displacement based formulation which is obtained by a local static condensation of a mixed problem satisfying a uniform inf-sup condition. We work with conforming bilinear approximations for the displacement and a pressure space of piecewise constants. Unfortunately, the standard $Q1 - P0$ pairing does not satisfy a uniform inf-sup condition. To obtain a stable scheme, we have to extract from the pressure space the so-called checkerboard modes. For some earlier references on the construction of uniformly bounded domain decomposition and multigrid methods in the incompressible limit, see [5] for Neumann-Neumann methods and [15] and [13] for multigrid solvers. Let us note that there are also recent results on FETI-DP and BDDC domain decomposition methods for mixed finite element discretizations of Stokes' equations, see [12] and [11], and almost incompressible elasticity, see [1]. In this work, we propose a dual-primal iterative substructuring method for almost incompressible elasticity. Numerical results illustrate the performance and the scalability of our method in the incompressible limit.

2 Almost incompressible elasticity and finite elements

The equations of linear elasticity model the displacement of a homogeneous linear elastic material under the action of external and internal forces. The elastic body occupies a domain $\Omega \subset \mathbb{R}^2$, which is assumed to be polyhedral and of diameter one. We denote its boundary by $\partial\Omega$ and assume that one part of it, $\partial\Omega_D$, is clamped, i.e., with homogeneous Dirichlet boundary conditions, and that the rest, $\partial\Omega_N := \partial\Omega \setminus \partial\Omega_D$, is subject to a surface force $\mathbf{g}$, i.e., a natural boundary condition. We can also introduce a body force $\mathbf{f}$, e.g., gravity. With $\mathbf{H^1}(\Omega) := (H^1(\Omega))^2$, the appropriate space for a variational formulation is the Sobolev space $\mathbf{H_0^1}(\Omega, \partial\Omega_D) := \{\mathbf{v} \in \mathbf{H^1}(\Omega) : \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega_D\}$. The linear elasticity problem consists of finding the displacement $\mathbf{u} \in \mathbf{H_0^1}(\Omega, \partial\Omega_D)$ of the elastic body Ω, such that

$$\int_\Omega 2\mu\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) d\mathbf{x} + \int_\Omega \lambda \operatorname{div}\mathbf{u} \operatorname{div}\mathbf{v}\, d\mathbf{x} = \langle \mathbf{F}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathbf{H_0^1}(\Omega, \partial\Omega_D). \qquad (1)$$

Here μ and λ are the Lamé parameters, which are constant in view of the assumption of a homogeneous body, and which are assumed positive. Of particular interest is the incompressible limit, which corresponds to $\lambda \to \infty$. The Lamé parameters are related to the pair (E, ν), where E is Young's modulus and ν is Poisson's ratio by

$$E = \frac{\mu(2\mu + 3\lambda)}{\mu + \lambda}, \qquad \nu = \frac{\lambda}{2(\mu + \lambda)} .$$

Furthermore, $\varepsilon_{ij}(\mathbf{u}) := \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})$ is the linearized strain tensor, and

$$\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) = \sum_{i,j=1}^{2} \varepsilon_{ij}(\mathbf{u})\varepsilon_{ij}(\mathbf{v}), \quad \langle \mathbf{F}, \mathbf{v} \rangle := \int_\Omega \mathbf{f}^T \mathbf{v}\, d\mathbf{x} \; + \int_{\partial\Omega_N} \mathbf{g}^T \mathbf{v}\, d\sigma.$$

Our finite element discretization is based on the conforming space $\mathbf{V}_h$ of continuous piecewise bilinear approximations on quadrilaterals. The quasi-uniform mesh is denoted by $\mathcal{T}_h$, and we assume that it has a macro-element structure, i.e., $\mathcal{T}_h$ is obtained from a coarser mesh $\mathcal{T}_h^m$ by decomposing each element into four subelements. We first consider the abstract pair $(\mathbf{V}_h, M_h)$

$$\begin{aligned} 2\mu(\varepsilon(\mathbf{u}_h), \varepsilon(\mathbf{v}_h))_0 + (\operatorname{div}\mathbf{v}_h, p_h)_0 &= \langle \mathbf{F}, \mathbf{v}_h \rangle \quad && \forall \mathbf{v}_h \in \mathbf{V}_h \;, \\ (\operatorname{div}\mathbf{u}_h, q_h)_0 \qquad - \frac{1}{\lambda}(p_h, q_h)_0 &= 0 && \forall q_h \in M_h \,. \end{aligned}$$

In terms of static condensation, we can eliminate the pressure and obtain a displacement-based formulation

$$\int_\Omega 2\mu\varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v}) d\mathbf{x} + \int_\Omega \lambda\, \Pi_{M_h} \operatorname{div}\mathbf{u}\; \Pi_{M_h} \operatorname{div}\mathbf{v}\, d\mathbf{x} = \langle \mathbf{F}, \mathbf{v} \rangle \quad \forall \mathbf{v} \in \mathbf{V}_h, \qquad (2)$$

where Π_{M_h} denotes the L^2-projection onto M_h. It is well known that the choice $M_h = M_h^u$

$$M_h^u = \{q \in L_0^2(\Omega) \mid \; q|_K \in P_0(K), \; K \in \mathcal{T}_h\},$$

does not yield a uniform inf-sup condition and checkerboard modes in the pressure might be observed, see, e.g., [4]. Thus it is necessary to make M_h a proper subset of M_h^u. There exist different possibilities to overcome this difficulty. One option is to work with macro-elements and to extract from M_h^u the checkerboard mode on each macro-element, as in [4]. The restrictions of functions in M_h^u to a macro-element are spanned by the four functions depicted in Figure 1.

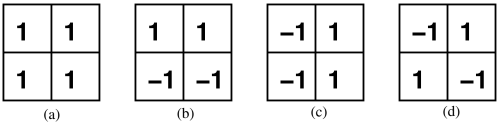


Fig. 1. Basis functions for the pressure space related to a single macro element.

The function indicated in Figure 1 (d) is the local checkerboard modes p^c. To obtain a stable pairing, we have to work with $M_h = M_h^s$

$$M_h^s = \{q \in M_h^u \mid (q, p^c)_{0;K} = 0, \; K \in \mathcal{T}_h^m\}.$$

From now on, we call the choice $M_h = M_h^u$ the unstable or the not stabilized $Q1 - P0$ formulation and the choice $M_h = M_h^s$ the stabilized $Q1 - P0$ formulation. The analysis and the implementation will be based on the reduced problem (2). We note that in both cases the L^2-projection Π_{M_h} can be carried out locally.

3 The FETI-DP algorithm

Let the domain Ω be decomposed into nonoverlapping subdomains $\Omega_i, i = 1, \dots, N$, each of which is the union of finite elements with matching finite element nodes across the interface Γ. The interface Γ is the union of the interior subdomain edges and vertices. For each subdomain Ω_i, we assemble local stiffness matrices $K^{(i)}$ and local load vectors $\mathbf{f}^{(i)}$. By $\mathbf{u}^{(i)}$ we denote the local solution vectors of nodal values.

In the dual-primal FETI methods, we distinguish between dual and primal displacement variables by the way the continuity of the solution in those variables is established. Dual displacement variables are those, for which the continuity is enforced by a continuity constraint and Lagrange multipliers $\boldsymbol{\lambda}$ and thus, continuity is not established until convergence of the iterative method is reached, as in the classical one-level FETI methods; see, e.g., [8]. On the other hand, continuity of the primal displacement variables is enforced explicitly in each iteration step by subassembly of the local stiffness matrices $K^{(i)}$ at the primal displacement variables. This subassembly yields a symmetric, positive definite stiffness matrix $\widetilde{K}$ which is not block diagonal anymore but is coupled at the primal displacement variables. Let us note that this coupling yields a global problem which is necessary to obtain a numerically scalable algorithm.

We will use subscripts I, Δ, and Π, to denote the interior, dual, and primal displacement variables, respectively, and obtain for the local stiffness matrices, load vectors, and solution vectors of nodal values

$$K^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Delta I}^{(i)T} & K_{\Pi I}^{(i)T} \\ K_{\Delta I}^{(i)} & K_{\Delta\Delta}^{(i)} & K_{\Pi\Delta}^{(i)T} \\ K_{\Pi I}^{(i)} & K_{\Pi\Delta}^{(i)} & K_{\Pi\Pi}^{(i)} \end{bmatrix}, \mathbf{u}^{(i)} = \begin{bmatrix} \mathbf{u}_I^{(i)} \\ \mathbf{u}_\Delta^{(i)} \\ \mathbf{u}_\Pi^{(i)} \end{bmatrix}, \mathbf{f}^{(i)} = \begin{bmatrix} \mathbf{f}_I^{(i)} \\ \mathbf{f}_\Delta^{(i)} \\ \mathbf{f}_\Pi^{(i)} \end{bmatrix}.$$

We also introduce the notation

$$\mathbf{u}_B = [\mathbf{u}_I \; \mathbf{u}_\Delta]^T, \mathbf{f}_B = [\mathbf{f}_I \; \mathbf{f}_\Delta]^T, \mathbf{u}_B^{(i)} = [\mathbf{u}_I^{(i)} \; \mathbf{u}_\Delta^{(i)}]^T, \text{ and } \mathbf{f}_B^{(i)} = [\mathbf{f}_I^{(i)} \; \mathbf{f}_\Delta^{(i)}]^T.$$

Accordingly, we define

$$K_{BB} = \operatorname{diag}_{i=1}^{N}(K_{BB}^{(i)}), \;\; K_{BB}^{(i)} = \begin{bmatrix} K_{II}^{(i)} & K_{\Delta I}^{(i)T} \\ K_{\Delta I}^{(i)} & K_{\Delta\Delta}^{(i)} \end{bmatrix}, \;\; K_{\Pi B} = [K_{\Pi B}^{(1)} \ldots K_{\Pi B}^{(N)}].$$

We note that K_{BB} is a block diagonal matrix. By subassembly in the primal displacement variables, we obtain

$$\widetilde{K} = \begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^{T} \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} \end{bmatrix},$$

where a tilde indicates the subassembled matrices and where

$$\widetilde{K}_{\Pi B} = [\widetilde{K}_{\Pi B}^{(1)} \cdots \widetilde{K}_{\Pi B}^{(N)}].$$

Introducing local assembly operators $R_{\Pi}^{(i)}$ which map from the local primal displacement variables $\mathbf{u}_{\Pi}^{(i)}$ to the global, assembled $\widetilde{\mathbf{u}}_{\Pi}$, we have

$$\widetilde{K}_{\Pi B}^{(i)} = R_{\Pi}^{(i)} K_{\Pi B}^{(i)}, \qquad \widetilde{\mathbf{u}}_{\Pi} = \sum_{i=1}^{N} R_{\Pi}^{(i)} \mathbf{u}_{\Pi}^{(i)}, \qquad \widetilde{K}_{\Pi\Pi} = \sum_{i=1}^{N} R_{\Pi}^{(i)} K_{\Pi\Pi}^{(i)} R_{\Pi}^{(i)T},$$

for $i = 1, \ldots, N$. Due to the subassembly of the primal displacement variables, Lagrange multipliers have to be used only for the dual displacement variables $\mathbf{u}_{\Delta}$ to enforce continuity. We introduce a discrete jump operator B such that the solution $\mathbf{u}_{\Delta}$, associated with more than one subdomain, coincides when $B\mathbf{u}_B = 0$; the interior variables $\mathbf{u}_I$ remain unchanged and thus the corresponding entries in B remain zero. Since we assume pointwise matching grids across the interface Γ, the entries of the matrix B are $0, 1$, and -1.

We can now reformulate the finite element discretization of (2) as

$$\begin{bmatrix} K_{BB} & \widetilde{K}_{\Pi B}^{T} & B^{T} \\ \widetilde{K}_{\Pi B} & \widetilde{K}_{\Pi\Pi} & O \\ B & O & O \end{bmatrix} \begin{bmatrix} \mathbf{u}_B \\ \widetilde{\mathbf{u}}_{\Pi} \\ \boldsymbol{\lambda} \end{bmatrix} = \begin{bmatrix} \mathbf{f}_B \\ \widetilde{\mathbf{f}}_{\Pi} \\ \mathbf{0} \end{bmatrix}. \tag{3}$$

Elimination of the primal variables $\widetilde{\mathbf{u}}_{\Pi}$ and the interior and dual displacement variables $\mathbf{u}_B$ leads to a a reduced linear system of the form

$$F\boldsymbol{\lambda} = \mathbf{d},$$

where the matrix F and the right hand side $\mathbf{d}$ are formally obtained by block Gauss elimination. Let us note that the matrix F is never built explicitly but that in every iteration appropriate linear systems are solved; see [3], [7] or [6] for further details.

To define the FETI-DP Dirichlet preconditioner M^{-1}, we introduce a scaled jump operator B_D; this is done by scaling the contributions of B associated with the dual displacement variables from individual subdomains. We define $B_D = [B_D^{(1)}, \ldots, B_D^{(N)}]$, where the $B_D^{(i)}$ are defined as follows: each row of $B^{(i)}$ with a nonzero entry corresponds to a Lagrange multiplier connecting the subdomain Ω_i with a neighboring subdomain Ω_j at a point $x \in \partial\Omega_{i,h} \cap \partial\Omega_{j,h}$. We obtain $B_D^{(i)}$ by multiplying each such row of $B^{(i)}$ with $1/|\mathcal{N}_x|$, where $|\mathcal{N}_x|$ denotes the multiplicity of the interface point $x \in \Gamma$. This scaling is called the multiplicity scaling and is suitable for homogeneous problems; see [7]. Our preconditioner is then given in matrix form by

$$M^{-1} = B_D R_\Gamma^T S R_\Gamma B_D^T = \sum_{i=1}^{N} B_D^{(i)} R_\Gamma^{(i)T} S^{(i)} R_\Gamma^{(i)} B_D^{(i)T}. \tag{4}$$

Here, $R_\Gamma^{(i)}$ are restriction matrices that restrict the degrees of freedom of a subdomain to its interface and $R_\Gamma = \mathrm{diag}_i(R_\Gamma^{(i)})$.

We have to decide how to choose the primal displacement variables. The simplest choice is to choose them as certain selected vertices of the subdomains, see [2], where this approach was first considered. Following the notation introduced in [9], we will denote the FETI-DP algorithm which uses exclusively selected vertices as primal displacement constraints as Algorithm A. Unfortunately, Algorithm A does not yield uniform bounds in the incompressible limit. To obtain better convergence properties, we have to introduce additional constraints. These constraints are averages over the edges, which are enforced to have the same values across the interface. This variant has been introduced in [9] for scalar problems and is denoted by Algorithm B.

For our FETI-DP algorithm B, we have the following condition number estimate, cf. [10],

Theorem 1. *The condition number for the choice $M_h = M_h^s$ satisfies*

$$\kappa(M^{-1}F) \leq C\,(1 + \log(H/h))^2.$$

Here, $C > 0$ is independent of h, H, and the values of the Poisson ratio ν.

4 Numerical results

We apply Algorithms A and B to (2), where $\Omega = (0,1)^2$ and the Young modulus is defined as $E = 1$. We will present results for different Poisson ratios ν. Algorithm A uses all subdomain vertices as primal constraints and Algorithm B, additionally, uses all edge averages as primal constraints. For the experiments in Table 1, we use a structured grid with 240×240 macro elements ($= 480 \times 480$ elements). In small portions of the boundary in all four corners of the unit square homogeneous Dirichlet boundary conditions were applied (see Figure 2) and the domain was subjected to a volume force directed towards $(1,1)^T$. The domain was decomposed into 64 square subdomains with 7 442 d.o.f. each; this results in an overall problem of 462 722 d.o.f. The stopping criterion is a relative residual reduction of 10^{-10}. The experiments were carried out on two Opteron 248 (2.2 Ghz) 64-bit processors. The differences in computing time between the unstable and the stabilized $Q1-P0$ element, e.g., for $\nu = 0.4$, are due to the different sparsity patterns of the stiffness matrices. The stabilized $Q1-P0$ element leads up to 50% more nonzero entries in the corresponding stiffness matrix.

For the experiments in Table 2, the unit square is decomposed into 4 to 1 024 subdomains with 1 250 d.o.f. each. Homogeneous Dirichlet boundary conditions are applied on the bottom and the left side. Again, a volume force directed towards $(1,1)^T$ is applied. The calculations were carried out on a single Opteron 144 (1.8 Ghz) 64-bit processor. We used as a stopping criterion the relative residual reduction of 10^{-14}.

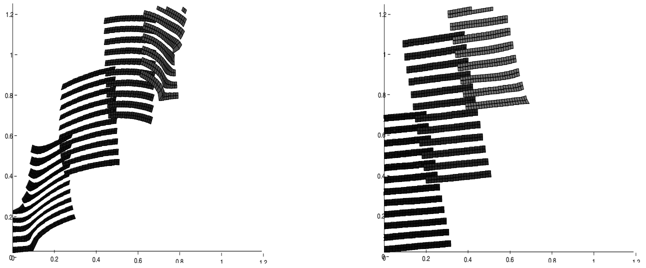


Fig. 2. Deformed configuration for the experiments in Table 1 (left) and for the experiments in Table 2 (right). In both cases a coarser grid than used in the calculations is depicted.

ν	It.	$\lambda_{\max}$	$\lambda_{\min}$	Time	It.	$\lambda_{\max}$	$\lambda_{\min}$	Time
Alg. B		**(stabilized)**				**(not stabilized)**		
0.4	23	**6.98**	1.0075	55s	23	**6.98**	1.0075	47s
0.49	23	**6.81**	1.0079	55s	23	**6.86**	1.0086	47s
0.499	24	**6.79**	1.0078	56s	23	**6.79**	1.0090	47s
0.4999	24	**6.79**	1.0078	56s	29	**6.48**	1.0087	53s
0.49999	24	**6.79**	1.0080	56s	55	**39.98**	1.0088	80s
0.499999	25	**6.79**	1.0076	57s	97	**366**	1.0086	124s
0.4999999	25	**6.79**	1.0078	57s	131	**3 632**	1.0096	159s
Alg. A		**(stabilized)**				**(not stabilized)**		
0.4	53	**42.52**	1.012	82s	53	**42.52**	1.012	81s
0.49	103	**316**	1.017	139s	67	**85.93**	1.015	78s
0.499	192	**3 037**	1.018	241s	137	**723**	1.017	143s
0.4999	270	$\mathbf{3.02} \times 10^4$	1.020	332s	220	**7 069**	1.020	221s
0.49999	368	$\mathbf{3.02} \times 10^5$	1.020	445s	315	$\mathbf{7.05} \times 10^4$	1.021	310s
0.499999	465	$\mathbf{3.02} \times 10^6$	1.022	558s	$>$ 500	$\mathbf{7.05} \times 10^5$	1.037	$>$ 486s
0.4999999	$>$ 500	$\mathbf{3.02} \times 10^7$	1.032	$>$ 599s	$>$ 500	$\mathbf{7.05} \times 10^6$	1.159	$>$ 484s

Table 1. Algorithms B and A, 462 722 d.o.f. and 64 subdomains.

Acknowledgement. The first and third author gratefully acknowledge the support of the “Research in Pairs” (RiP) program while being at the Mathematisches Forschungsinstitut Oberwolfach.

Algorithm B			$\nu = \mathbf{0.4999999}$			$\nu = \mathbf{0.4}$		
N	Mesh	d.o.f.	It.	$\lambda_{\max}$	$\lambda_{\min}$	It.	$\lambda_{\max}$	$\lambda_{\min}$
4	48×48	4 802	17	2.51	1.0011	13	2.19	1.0015
9	72×72	10 658	21	3.38	1.0020	19	3.47	1.0024
16	96×96	18 818	24	4.03	1.0023	22	4.13	1.0025
36	144×144	42 050	26	4.53	1.0024	24	4.64	1.0025
64	192×192	74 498	27	4.69	1.0024	25	4.80	1.0026
100	240×240	116 162	29	4.75	1.0022	26	4.86	1.0025
144	288×288	167 042	29	4.78	1.0023	27	4.88	1.0026
256	384×384	296 450	30	4.79	1.0022	30	4.91	1.0024
576	576×576	665 858	32	4.80	1.0021	32	4.77	1.0024
1 024	768×768	1 182 722	32	4.80	1.0021	33	4.81	1.0024

Table 2. Numerical scalability of Algorithm B, $Q_1 - P_0$ (stabilized).

References

1. C. R. Dohrmann, *A substructuring preconditioner for nearly incompressible elasticity problems*, Tech. Rep. SAND 2004-5393, Sandia National Laboratories, October 2004.
2. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.
3. C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numer. Lin. Alg. Appl., 7 (2000), pp. 687–714.
4. V. Girault and P.-A. Raviart, *Finite Element Methods for Navier-Stokes Equations*, Springer-Verlag, New York, 1986.
5. P. Goldfield, *Balancing Neumann-Neumann preconditioners for the mixed formulation of almost-incompressible linear elasticity*, PhD thesis, New York University, Department of Mathematics, 2003.
6. A. Klawonn and O. Rheinbach, *A parallel implementation of Dual-Primal FETI methods for three dimensional linear elasticity using a transformation of basis*, Tech. Rep. SM-E-601, Univ. Duisburg-Essen, Department of Mathematics, Germany, February 2005.
7. A. Klawonn and O. Widlund, *Dual-Primal FETI methods for linear elasticity*, Tech. Rep. 855, Department of Computer Science, Courant Institute of Mathematical Sciences, New York, September 2004.
8. A. Klawonn and O. B. Widlund, *FETI and Neumann–Neumann iterative substructuring methods: Connections and new results*, Comm. Pure Appl. Math., 54 (2001), pp. 57–90.
9. 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 (2002).
10. A. Klawonn and B. I. Wohlmuth, *FETI-DP for almost incompressible elasticity in the displacement formulation.* in preparation, 2006.
11. J. Li, *Dual-Primal FETI methods for stationary Stokes and Navier-Stokes equations*, PhD thesis, Courant Institute of Mathematical Sciences, New York University, 2002.

12. J. Li and O. B. Widlund, *BDDC algorithms for incompressible Stokes equations*, Tech. Rep. TR-861, New York University, Department of Computer Science, 2005.
13. J. Schöberl, *Multigrid methods for a parameter-dependent problem in primal variables*, Numer. Math., 84 (1999), pp. 97–119.
14. J. C. Simo and M. S. Rifai, *A class of mixed assumed strain methods and the method of incompatible modes*, Internat. J. Numer. Methods Engrg., 29 (1990), pp. 1595–1638.
15. C. Wieners, *Robust multigrid methods for nearly incompressible elasticity*, Computing, 64 (2000), pp. 289–306.

Inexact Fast Multipole Boundary Element Tearing and Interconnecting Methods

Ulrich Langer[1][2], Günther Of[3], Olaf Steinbach[4], and Walter Zulehner[1]

[1] Johannes Kepler University Linz, Institute of Computational Mathematics, Linz, Austria.

[2] Austrian Academy of Sciences, Johann Radon Institute for Computational and Applied Mathematics, Linz, Austria.

[3] University of Stuttgart, Institute for Applied Analysis and Numerical Simulation, Stuttgart, Germany.

[4] Graz University of Technology, Institute of Mathematics, Graz, Austria.

Summary. The Boundary Element Tearing and Interconnecting (BETI) methods have recently been introduced as boundary element counterparts of the well–established Finite Element Tearing and Interconnecting (FETI) methods. In this paper we present inexact data–sparse versions of the BETI methods which avoid the elimination of the primal unknowns and dense matrices. The data–sparse approximation of the matrices and the preconditioners involved is fully based on Fast Multipole Methods (FMM). This leads to robust solvers which are almost optimal with respect to the asymptotic complexity estimates.

1 Introduction

Langer and Steinbach [8] have recently introduced the BETI methods as boundary element counterparts of the well–established FETI methods which were proposed by Farhat and Roux [3]. We refer the reader to the monograph by Toselli and Widlund [12] for more information and references to FETI and FETI–DP methods. In particular, we mention the paper by Klawonn and Widlund [5] who introduced and investigated the inexact FETI technique that avoids the elimination of the primal unknowns (displacements).

In this paper we introduce inexact BETI methods for solving the inhomogeneous Dirichlet boundary value problem (BVP) for the homogeneous potential equation in 3D bounded domains, where all matrices and preconditioners involved in the BETI solver are data-sparse via FMM representations. However, instead of symmetric and positive definite systems, we finally have to solve two–fold saddle point problems. The proposed iterative solver and preconditioner result in an almost optimal solver the complexity of which is proportional to the numbers of unknowns on the skeleton up to some polylogarithmical factor. More precisely, the solver requires $\mathcal{O}((H/h)^{(d-1)}(1+\log(H/h))^4 \log \varepsilon^{-1})$ arithmetical operations in a parallel regime and $\mathcal{O}((H/h)^{(d-1)}(1+\log(H/h))^2)$ storage units per processor, where $d = 3$ in the 3D case considered here, and $\varepsilon \in (0,1)$ is the relative accuracy of the iteration error in a suitable norm. H and h denote the usual scalings of the subdomains and the boundary elements, respectively. Moreover, the solvers are robust with respect to large coefficient jumps. For the sake of simplicity, we present here only the case where all subdomains are non-floating. All results remain valid for the general case that is discussed together with some other issues including other preconditioners in the forthcoming paper by Langer, Of, Steinbach and Zulehner [6] where the reader can also find the proofs in detail.

The rest of the paper is organized as follows. In Section 2, we introduce the fast multipole boundary element domain decomposition (DD) method. Section 3 is devoted to the inexact BETI method. In Section 4, we describe the ingredients from which the preconditioner and the solver for the two–fold saddle point problem that we finally have to solve is built. In Section 5, we present and discuss the results of our numerical experiments. Finally, we draw some conclusions.

2 Fast Multipole Boundary Element DD Methods

Let us consider the Dirichlet BVP for the potential equation

$$-\mathrm{div}[a(x)\nabla \hat{u}(x)] = 0 \text{ for } x \in \Omega \subset \mathbf{R^3}, \quad \hat{\mathbf{u}}(\mathbf{x}) = \mathbf{g}(\mathbf{x}) \text{ for } \mathbf{x} \in \mathbf{\Gamma} = \partial\mathbf{\Omega}, \tag{1}$$

with given Dirichlet data $g \in H^{1/2}(\Gamma)$ as a typical model problem, where Ω is a bounded Lipschitz domain that is assumed to be decomposed into p non–overlapping subdomains Ω_i with Lipschitz boundaries $\Gamma_i = \partial\Omega_i$. We further assume that the coefficient function $a(\cdot)$ in the potential equation (1) is piecewise constant such that $a(x) = a_i > 0$ for $x \in \Omega_i,\ i = 1,\dots,p$.

The solution $\hat{u}$ of (1) is obviously harmonic in all subdomains Ω_i. Using the representation formula and its normal derivative on Γ_i, we can reformulate the BVP (1) as a DD boundary integral variational problem living on the skeleton $\Gamma_S = \cup_{i=1}^p \Gamma_i$ of the DD, see [2] and [4]. After homogenization of the Dirichlet boundary condition via the ansatz $\hat{u} = \hat{g} + u$ with $\hat{g}_{|\Gamma} = g$ and $u_{|\Gamma} = 0$, this DD boundary integral variational problem can be written as a mixed variational problem of the form: find $t = (t_1, t_2, \dots, t_p) \in T = T_1 \times T_2 \times \dots \times T_p = H^{-1/2}(\Gamma_1) \times H^{-1/2}(\Gamma_2) \times \dots \times H^{-1/2}(\Gamma_p)$ and $u \in U = \{v_{|\Gamma_S} : v \in H_0^1(\Omega)\}$ such that

$$a_i \left[\langle \tau_i, V_i t_i \rangle_{\Gamma_i} - \langle \tau_i, (\frac{1}{2}I + K_i) u_{|\Gamma_i} \rangle_{\Gamma_i} \right] = a_i \langle \tau_i, (\frac{1}{2}I + K_i)\hat{g}_{|\Gamma_i} \rangle_{\Gamma_i} \tag{2}$$

for all $\tau_i \in T_i$, $i = 1, 2, \dots, p$, and

$$\sum_{i=1}^p a_i \left[-\langle (\frac{1}{2}I + K_i')t_i, v_{|\Gamma_i} \rangle_{\Gamma_i} - \langle D_i u_{|\Gamma_i}, v_{|\Gamma_i} \rangle_{\Gamma_i} \right] = \sum_{i=1}^p a_i \langle D_i \hat{g}_{|\Gamma_i}, v_{|\Gamma_i} \rangle_{\Gamma_i} \tag{3}$$

for all $v \in U$, where V_i, K_i, K_i', and D_i denote the local single layer potential operator, the local double layer potential operator, its adjoint, and the local hypersingular boundary integral operator, respectively.

Let us now introduce the boundary element trial spaces $U_h = S_h^1(\mathit{\Gamma}_S) = \operatorname{span}\{\varphi_m\}_{m=1}^{M} \subset U$ and $T_{i,h} = S_h^0(\mathit{\Gamma}_i) = \operatorname{span}\{\psi_k^i\}_{k=1}^{N_i} \subset T_i$ spanned by continuous piecewise linear basis functions φ_m and by piecewise constant basis functions ψ_k^i with respect to a regular globally quasi–uniform boundary element mesh with the average mesh size h on $\mathit{\Gamma}_S$ and $\mathit{\Gamma}_i$, respectively. The Galerkin discretization finally leads to a large–scale symmetric and indefinite system of form

$$\begin{pmatrix} a_1\widetilde{V}_{1,h} & & & -a_1\widetilde{K}_{1,h}R_{1,h} \\ & \ddots & & \vdots \\ & & a_p\widetilde{V}_{p,h} & -a_p\widetilde{K}_{p,h}R_{p,h} \\ -a_1R_{1,h}^{\top}\widetilde{K}_{1,h}^{\top} & \dots & -a_pR_{p,h}^{\top}\widetilde{K}_{p,h}^{\top} & -\widetilde{D}_h \end{pmatrix} \begin{pmatrix} \underline{\widetilde{t}}_1 \\ \vdots \\ \underline{\widetilde{t}}_p \\ \underline{\widetilde{u}} \end{pmatrix} = \begin{pmatrix} a_1\underline{\widetilde{g}}_1 \\ \vdots \\ a_p\underline{\widetilde{g}}_p \\ \underline{\widetilde{f}} \end{pmatrix} \tag{4}$$

for defining the coefficient vectors $\underline{\widetilde{t}}_i \in \mathbf{R}^{\mathbf{N_i}}$ and $\underline{\widetilde{u}} \in \mathbf{R}^{\mathbf{M}}$. The matrices $\widetilde{V}_{i,h}$, $\widetilde{K}_{i,h}$ and $\widetilde{D}_h$ are data–sparse FMM approximations to the originally dense Galerkin matrices $V_{i,h}$, $K_{i,h}$ and $D_h = \sum_{i=1}^{p} a_i R_{i,h}^{\top} D_{i,h} R_{i,h}$, respectively. The use of the FMM is indicated by the "tilde" on the matrices and vectors. The FMM approximation of these matrices reduces the quadratic complexity with respect to the number of unknowns to an almost linear one, but without disturbing the accuracy. The restriction operator $R_{i,h}$ maps some global coefficient vector $\underline{v} \in \mathbf{R}^{\mathbf{M}}$ to the local vector $\underline{v}_i \in \mathbf{R}^{\mathbf{M_i}}$ containing those components of $\underline{v}$ which correspond to $\mathit{\Gamma}_i$ only, $i = 1, 2, \dots, p$. The matrices $R_{i,h}$ are Boolean matrices which are sometimes also called subdomain connectivity matrices.

3 Inexact BETI Methods

Introducing the local unknowns $\underline{\widetilde{u}}_i = R_{i,h}\underline{\widetilde{u}}$ as individual variables and enforcing again the global continuity of the potentials by the constraints

$$\sum_{i=1}^{p} B_i \underline{\widetilde{u}}_i = \underline{0}, \tag{5}$$

we immediately arrive at the two–fold saddle point problem

$$\begin{pmatrix} V & K & 0 \\ K^{\top} & -D & B^{\top} \\ 0 & B & 0 \end{pmatrix} \begin{pmatrix} \underline{t} \\ \underline{u} \\ \underline{\lambda} \end{pmatrix} = \begin{pmatrix} \underline{g} \\ \underline{f} \\ \underline{0} \end{pmatrix} \tag{6}$$

that is obviously equivalent to (4), where $\underline{t} = (\underline{\widetilde{t}}_1, \dots, \underline{\widetilde{t}}_p)^{\top}$, $\underline{u} = (\underline{\widetilde{u}}_1, \dots, \underline{\widetilde{u}}_p)^{\top}$, and $\underline{\lambda} \in \mathbf{R}^{\mathbf{L}}$ is the vector of the Lagrange multipliers. The matrices $V = \operatorname{diag}(a_i\widetilde{V}_{i,h})$, $K = \operatorname{diag}(-a_i\widetilde{K}_{i,h})$ and $D = \operatorname{diag}(a_i\widetilde{D}_{i,h})$ are block–diagonal whereas $B = (B_1, \dots, B_p)$. As in the FETI method each row of the matrix B is connected with a pair of matching nodes across the subdomain boundaries. The entries of such a row are 1 and -1 for the indices corresponding to the matching nodes on the interface (coupling boundaries) $\mathit{\Gamma}_C = \mathit{\Gamma}_S \setminus \mathit{\Gamma}$ and 0 otherwise. We assume here that the number of constraints at some matching node is equal to the number of matching subdomains minus one. This method of a minimal number of constraints respectively multipliers is called non–redundant (see, e.g., [12]). The matrices $\widetilde{V}_{i,h}$ are symmetric and positive definite (SPD). For non–floating subdomains assumed in this paper the matrices $\widetilde{D}_{i,h}$ are SPD as well. In the more complicated case of floating subdomains, the matrices $\widetilde{D}_{i,h}$ must be modified due to the non-trivial kernel $\ker(\widetilde{D}_{i,h})$ $=\operatorname{span}\{\underline{1}_i\}$, where $\{\underline{1}_i\} = (1,\dots,1)^\top$, see [8] or [6].

4 Solvers and Preconditioners

Following [13], who extended the special conjugate gradient (CG) method proposed by [1] for solving one–fold saddle point problems, to n–fold saddle point problems, we are able to construct a very efficient saddle point conjugate gradient (SPCG) solver for our two–fold saddle point problem (6) provided that appropriate preconditioners for the single layer potential matrices $\widetilde{V}_{i,h}$, the local boundary element Schur complements $\widetilde{S}_{i,h} = \widetilde{D}_{i,h} + \widetilde{K}_{i,h}^\top \widetilde{V}_{i,h}^{-1} \widetilde{K}_{i,h}$ and the BETI Schur complement $\widetilde{F} = \sum\limits_{i=q+1}^{p} a_i^{-1} B_i \widetilde{S}_{i,h}^{-1} B_i^\top$ are available. We propose the following data–sparse preconditioners which are also used in our numerical experiments:

(a) *Data–sparse algebraic or geometric multigrid preconditoners $\widetilde{\mathcal{V}}_{i,h}$ for the matrices $\widetilde{V}_{i,h}$:* For the geometric multigrid method, [7] proved the spectral equivalence inequalities

$$\underline{c}_V \widetilde{\mathcal{V}}_{i,h} \le \widetilde{V}_{i,h} \le \overline{c}_V \widetilde{\mathcal{V}}_{i,h} \tag{7}$$

where the spectral equivalence constants $\underline{c}_V$ and $\overline{c}_V$ are positive and independent of h and H.

(b) *Data–sparse opposite order preconditioners $\widetilde{\mathcal{S}}_{i,h}$ for the local boundary element Schur complements $\widetilde{S}_{i,h}$*: In order to construct efficient preconditioners $\widetilde{\mathcal{S}}_{i,h}$, we apply the concept of boundary integral operators of the opposite order proposed by [11]. Based on the local trial space $U_{i,h} = S_h^1(\mathit{\Gamma}_i)$ of piecewise linear basis functions φ_m^i, as used for the Galerkin discretization of the local hypersingular boundary integral operators D_i, we define the Galerkin matrices $\bar{V}_{i,h}$ and $\bar{M}_{i,h}$ by

$$\bar{V}_{i,h}[n,m] \;=\; \langle \varphi_n^i, V\varphi_m^i\rangle_{\mathit{\Gamma}_i}, \quad \bar{M}_{i,h}[n,m] \;=\; \langle \varphi_n^i, \varphi_m^i\rangle_{\mathit{\Gamma}_i}$$

for $m,n = 1,\dots,M_i$. The inverse preconditioners are now defined by

$$\widetilde{\mathcal{S}}_{i,h}^{-1} \;=\; \bar{M}_{i,h}^{-1} \widetilde{\bar{V}}_{i,h} \bar{M}_{i,h}^{-1} \quad \text{for } i = 1,\dots,p, \tag{8}$$

where the tilde on the top of $\widetilde{\bar{V}}_{i,h}$ again indicates that the application of the discrete single layer potential $\bar{V}_{i,h}$ is realized by using the FMM. In [6] we prove the spectral equivalence inequalities

$$\underline{c}_S (1+\log(H/h))^{-2} \widetilde{\mathcal{S}}_{i,h} \le \widetilde{S}_{i,h} \le \overline{c}_S \widetilde{\mathcal{S}}_{i,h}, \tag{9}$$

where the spectral equivalence constants $\underline{c}_S$ and $\overline{c}_S$ are positive and independent of h and H. The log–term disappears in the case of floating subdomains.

(c) *Data–sparse BETI preconditioner $\widetilde{\mathcal{F}}$ for the BETI Schur complements $\widetilde{F}$*: Following [8], we define the inverse BETI preconditioner

$$\widetilde{\mathcal{F}}_{i,h}^{-1} = (BC_a^{-1}B^T)^{-1}\sum_{i=1}^{p} B_i C_\alpha^{-1}\widetilde{D}_{i,h}C_{a,i}^{-1}B_i^\top (BC_a^{-1}B^\top)^{-1}, \qquad (10)$$

with the help of the local data–sparse discrete hypersingular operators $\widetilde{D}_{i,h}$ and the scaling matrix $C_a = \operatorname{diag}(C_{a,i})$. The definition of the diagonal matrices $C_{a,i}$ can be found in [12]. In [6], the spectral equivalence inequalities

$$\underline{c}_F\widetilde{\mathcal{F}} \le \widetilde{F} \le \overline{c}_F(1+\log(H/h))^2\widetilde{\mathcal{F}} \qquad (11)$$

were proved, where the spectral equivalence constants $\underline{c}_S$ and $\overline{c}_S$ are positive and independent of h, H and the a_i's (coefficients jumps). In the general case where non–floating as well as floating subdomains are present in the DD, the spectral equivalence inequalities (11) remain valid on an appropriate subspace.

Combining these spectral equivalence estimates with the results obtained by [13] and taking into account the complexity estimate for the FMM, we can easily prove the following theorem.

Theorem 1. *If the two–fold saddle point problem (6) is solved by the SPCG method where the preconditioner is build from the block preconditioners $\widetilde{\mathcal{V}}_{i,h}$, $\widetilde{\mathcal{S}}_{i,h}$, and $\widetilde{\mathcal{F}}$, then not more than $I(\varepsilon) = \mathcal{O}((1+\log(H/h))^2\log\varepsilon^{-1})$ iterations and ops$(\varepsilon) = \mathcal{O}((H/h)^2(1+\log(H/h))^4\log\varepsilon^{-1})$ arithmetical operations are required in order to reduce the initial error by the factor $\varepsilon \in (0,1)$ in a parallel regime. The number of iterations $I(\varepsilon)$ is robust with respect to the jumps in the coefficients. Moreover, not more than $\mathcal{O}((H/h)^2(1+\log(H/h))^2)$ storage units are needed per processor.*

The results of the theorem remain valid also in the general case where also floating subdomains are present in the domain decomposition (see [6]). The proposed SPCG solver is asymptotically almost optimal with respect to the complexity in arithmetic and storage as well as very efficient on a parallel computer with distributed memory.

Remark 1. If we used optimal preconditioners $\widetilde{\mathcal{S}}_{i,h}$ for the local boundary element Schur complements $\widetilde{S}_{i,h}$, then the number of iteration $I(\varepsilon)$ of our SPCG solver would behave like $\mathcal{O}((1+\log(H/h))\log\varepsilon^{-1})$, whereas the arithmetical complexity would decrease from $\mathcal{O}((H/h)^2(1+\log(H/h))^4\log\varepsilon^{-1})$ to $\mathcal{O}((H/h)^2(1+\log(H/h))^3\log\varepsilon^{-1})$. Such preconditioners are available. If we convert the non–floating subdomains having a Dirichlet boundary part to floating subdomains by including the Dirichlet boundary condition into the constraints, then the data–sparse opposite order preconditioners $\widetilde{\mathcal{S}}_{i,h}$ given above is optimal.

5 Numerical Results

Let us consider the unit cube which is subdivided into eight similar subdomains. In order to check the behavior of the discretization error, we take the Dirichlet data $g = \hat{u}_{|\Gamma}$ as the trace of a regular solution $\hat{u}$ of the boundary value problem (1) on the boundary Γ. We perform numerical experiments for the Laplace equation ($a_i = 1$ for all $i = 1,\dots,8$) and for the potential equation with large jumps in the coefficients ($a_i \in \{1, 10^5\}$).

Starting from the coarsest grid level $L = 0$ with 192 triangles on $\cup\partial\Omega_i$, we successively refine the mesh by subdividing each triangle into four smaller similar triangles. N and M denote the total numbers of triangles and nodes, respectively. M_c is the total number of coupling nodes. The numbers of local triangles and nodes on $\partial\Omega_i$ are given by N_i and M_i, respectively. If the boundary mesh of one subdomain Ω_i on level $L = 6$ with 98304 triangles was uniformly extended to the interior of the subdomain, then the corresponding finite element mesh would consist of 6291456 tetrahedrals resulting in more than 50 millions tetrahedrals for the whole computational domain. In Table 1, together with the mesh features L, N, M, M_c, N_i and M_i, the time t_1 [sec] for generating the system (6) and for setting up the preconditioner, the time t_2 [sec] spent by the SPCG solver, the number of iterations $I(\varepsilon)$ and the absolute $L_2(\Gamma_i)$ discretization error $\|\hat{u} - \hat{u}_h\|_{L_2(\Gamma_i)}$ are displayed. The relative accuracy ε of the iteration error is chosen to be 10^{-8}. The first line in each row for the columns t_1, t_2, $I(\varepsilon)$ and $L_2(\Gamma_i)$–error corresponds to the Laplace case whereas the second line corresponds to the case of jumping coefficients. Table 1 shows that the growth in the number of iterations and in the CPU times is in good agreement with the complexity estimates given in Theorem 1. The efficiency of our SPCG solver is not affected by large jumps in the coefficients of the potential equations (1). Moreover, the number of iterations are less than in the Laplace case. In addition, the CPU time for the finest level $L = 6$ is half of the time needed for a primal preconditioned Schur complement solver in the case of jumping coefficients. All numerical experiments were performed on standard PCs with 3.06 Ghz Intel processors and 1 GB of RAM.

L	N	M	M_c	N_i	M_i	t_1	t_2	$I(\varepsilon)$	L_2-error
0	192	63	13	24	14	0	0	6	2,8527E–03
						1	0	6	2,8527E–08
1	768	261	67	96	50	1	1	33	7,1318E–04
						1	1	29	7,1318E–09
2	3072	1089	319	384	194	5	6	36	1,7830E–04
						5	6	34	1,7830E–09
3	12288	4473	1399	1536	770	16	34	38	4,4574E–05
						15	30	36	4,4577E–10
4	49152	18153	5863	6144	3074	81	186	41	1,1143E–05
						79	172	38	1,1144E–10
5	196608	73161	24007	24576	12290	316	1469	46	2,7859E–06
						310	1346	44	2,7859E–11
6	786432	293769	97159	98304	49154	1314	7250	55	6,9647E–07
						1319	7034	49	6,9651E–12

Table 1. Numerical features for the SPCG solver.

6 Conclusions

Inexact data–sparse BETI methods introduced in this paper show an almost optimal behavior with respect to the number of iterations, the arithmetical costs and the memory consumption. Moreover, the methods are robust with respect to large jumps in the coefficients of the potential equation (1). These results have been rigorosly proved and have also been confirmed by our numerical experiments. The treatment of the outer Dirichlet problem as well as other boundary conditions is straightforward. Inexact data–sparse BETI methods can naturally be generalized to linear elasticity BVP including elasticity problems for almost incompressible materials (cf. [10]). Combining the results of this paper with the results on inexact FETI methods obtained by Klawonn and Widlund [5], we can develop inexact data-sparse BETI–FETI solvers for coupled boundary and finite element equations (cf. [9] for the exact version).

Acknowledgement. This work has been supported by the Austrian Science Fund 'Fonds zur Förderung der wissenschaftlichen Forschung (FWF)' under the grants P14953 and SFB F013 'Numerical and Symbolic Scientific Computing', and by the German Research Foundation 'Deutsche Forschungsgemeinschaft (DFG)' under the grant SFB 404 'Multifield Problems in Continuum Mechanics'.

References

1. J. H. Bramble and J. E. Pasciak, *A preconditioning technique for indefinite systems resulting from mixed approximations of elliptic problems*, Mathematics of Computation, 50 (1988), pp. 1–17.
2. M. Costabel, *Symmetric methods for the coupling of finite elements and boundary elements*, in Boundary Elements IX, C. A. Brebbia, W. L. Wendland, and G. Kuhn, eds., vol. 1, Springer-Verlang, 1987, pp. 411–420.
3. C. Farhat and F.-X. Roux, *A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm*, Int. J. Numer. Meth. Engrg., 32 (1991), pp. 1205–1227.
4. G. Hsiao and W. Wendland, *Domain decomposition in boundary element methods*, in Proceedings of the Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, Y. Kuznetsov, G. Meurant, J. Périaux, and O. B. Widlund, eds., Philadelphia, 1991, SIAM, pp. 41–49.
5. A. Klawonn and O. B. Widlund, *A domain decomposition method with Lagrange multipliers and inexact solvers for linear elasticity*, SIAM J. Sci. Comput., 22 (2000), pp. 1199–1219.

6. U. Langer, G. Of, O. Steinbach, and W. Zulehner, *Inexact data–sparse boundary element tearing and interconnecting methods*, Tech. Rep. 2005-7, RICAM, Johann Radon Institute for Computational and Applied Mathematics, Autrian Academy of Sciences, Linz, Austria, 2005.
7. U. Langer and D. Pusch, *Convergence analysis of geometrical multigrid methods for solving data–sparse boundary element equations*, Tech. Rep. 2005-16, RICAM, Johann Radon Institute for Computational and Applied Mathematics, Autrian Academy of Sciences, Linz, Austria, 2005.
8. U. Langer and O. Stinbach, *Boundary element tearing and interconnecting method*, Computing, (2003), pp. 205–228.
9. ———, *Coupled boundary and finite element tearing and interconnecting methods*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., vol. 40 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 83–97.
10. O. Steinbach, *A robust boundary element method for nearly incompressible linear elasticity*, Numer. Math., 95 (2003), pp. 553–562.
11. O. Steinbach and W. L. Wendland, *The construction of some efficient preconditioners in the boundary element method*, Adv. Comput. Math., 9 (1998), pp. 191–216.
12. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.
13. W. Zulehner, *Uzawa–type methods for block–structured indefinite linear systems*, Tech. Rep. 2005–5, Johannes Kepler University, Linz, Austria, 2005. SFB F013.

A BDDC Preconditioner for Saddle Point Problems

Jing Li[1] and Olof Widlund[2]

[1] Department of Mathematical Sciences, Kent State University, Kent, OH 44242, USA. li@math.kent.edu

[2] Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA. widlund@cs.nyu.edu

Summary. The purpose of this paper is to extend the BDDC (balancing domain decomposition by constraints) algorithm to saddle point problems that arise when mixed finite element methods are used to approximate the system of incompressible Stokes equations. The BDDC algorithms are defined in terms of a set of primal continuity constraints, which are enforced across the interface between the subdomains, and which provide a coarse space component of the preconditioner. Sets of such constraints are identified for which bounds on the rate of convergence can be established that are just as strong as previously known bounds for the elliptic case. The preconditioned operator is positive definite and a conjugate gradient method can be used. A close connection is also established between the BDDC and FETI-DP algorithms for the Stokes case.

1 Introduction

The BDDC algorithms are domain decomposition methods based on nonoverlapping subdomains into which the domain of a given partial differential equation is divided. Introduced by Dohrmann [1] and analyzed in the elliptic case by him, Mandel, and Tezaur [9], these methods represent an important advance over the balancing Neumann-Neumann methods that have been used extensively in the past to solve large finite element problems; cf. [10, Section 6.2] where references to earlier work can also be found. It has also been established that the preconditioned operators of a pair of BDDC and FETI-DP algorithms, with the same primal constraints, have the same nonzero eigenvalues for positive definite elliptic problems; see [9, 3, 7].

In this paper, a BDDC preconditioner is developed for mixed finite element approximations of the incompressible Stokes equations in a very similar way; see also [8] for many more details. If the set of primal constraints on the velocity across the interface satisfies a certain assumption, we are then able to show that the preconditioned operator is positive definite and has the same nonzero eigenvalues as the

FETI-DP operator developed in [6]. With an additional assumption, a bound on the convergence rate as strong as for the standard elliptic case can be proved.

2 Discretization of a Saddle Point Problem

Let us consider the incompressible Stokes problem on a bounded, polyhedral domain Ω, in two or three dimensions. We denote the boundary of the domain by $\partial\Omega$; for simplicity a homogeneous Dirichlet boundary condition is enforced. The weak solution has the following saddle point formulation: find $\mathbf{u} \in \left(H_0^1(\Omega)\right)^d = \{\mathbf{v} \in (H^1(\Omega))^d \mid \mathbf{v} = \mathbf{0} \text{ on } \partial\Omega\}$, $d = 2$ or 3, and $p \in L_0^2(\Omega) = \{q \in L^2(\Omega) \mid \int_\Omega q = 0\}$, such that,

$$\begin{cases} a(\mathbf{u},\mathbf{v}) + b(\mathbf{v},p) = (\mathbf{f},\mathbf{v}), \ \forall \mathbf{v} \in \left(H_0^1(\Omega)\right)^d, \\ b(\mathbf{u},q) \qquad\qquad\quad = 0, \qquad \forall q \in L_0^2(\Omega) \ , \end{cases} \tag{1}$$

where $a(\mathbf{u},\mathbf{v}) = \int_\Omega \nabla\mathbf{u} : \nabla\mathbf{v}$, or $a(\mathbf{u},\mathbf{v}) = 2\int_\Omega \varepsilon(\mathbf{u}) : \varepsilon(\mathbf{v})$ and $b(\mathbf{u},q) = -\int_\Omega (\nabla \cdot \mathbf{u})q$. Here the strain tensor $\varepsilon(\mathbf{u})$ is defined by $\varepsilon_{ij}(\mathbf{u}) = (\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i})/2$. The operator form of the Stokes problem with Dirichlet boundary conditions is the same for either choice of the bilinear form $a(\cdot,\cdot)$, but we adopt the second which gives rise to a natural boundary condition which is consistent with physics.

In our mixed finite element methods for solving the saddle point problem (1), the velocity solution space will be denoted by $\widehat{\mathbf{W}}$. It consists of vector-valued, low order piecewise polynomial functions which are continuous across element boundaries. The pressure space $Q \subset L_0^2(\Omega)$ consists of scalar, discontinuous functions. A characteristic diameter of the elements of the underlying triangulation is denoted by h. The finite element approximation $(\mathbf{u},p)$ of the variational problem (1) can be written as

$$\begin{bmatrix} A & B^T \\ B & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u} \\ p \end{bmatrix} = \begin{bmatrix} \mathbf{f} \\ 0 \end{bmatrix}. \tag{2}$$

We will always assume that the chosen mixed finite element space $\widehat{\mathbf{W}} \times Q$ is inf-sup stable, i.e., that there exists a positive constant β, independent of h, such that

$$\sup_{\mathbf{w}\in\widehat{\mathbf{W}}} \frac{b(\mathbf{w},q)}{\|\mathbf{w}\|_{H^1}} \geq \beta\|q\|_{L^2}, \qquad \forall q \in Q. \tag{3}$$

The domain Ω is decomposed into N nonoverlapping polyhedral subdomains Ω_i, $i = 1,2,...,N$, of characteristic diameter H. The subdomain interface is defined by $\Gamma = (\cup\partial\Omega_i)\backslash\partial\Omega$, and the interface of an individual subdomain Ω_i is $\Gamma_i = \partial\Omega_i \cap \Gamma$. We decompose the discrete velocity and pressure spaces $\widehat{\mathbf{W}}$ and Q into $\widehat{\mathbf{W}} = \mathbf{W}_I \bigoplus \widehat{\mathbf{W}}_\Gamma$ and $Q = Q_I \bigoplus Q_0$, where $\mathbf{W}_I$ and Q_I are direct sums of subdomain interior velocity spaces $\mathbf{W}_I^{(i)}$, and subdomain interior pressure spaces $Q_I^{(i)}$, respectively. The elements of $\mathbf{W}_I^{(i)}$ are supported in the subdomain Ω_i and vanish on its interface Γ_i, while the elements of $Q_I^{(i)}$ are restrictions of elements in Q to Ω_i which satisfy $\int_{\Omega_i} q_I^{(i)} = 0$. $\widehat{\mathbf{W}}_\Gamma$ is the space of traces on Γ of functions in $\widehat{\mathbf{W}}$

and Q_0 is the subspace of Q with constant values $q_0^{(i)}$ in the subdomain Ω_i that satisfy $\int_{\Omega} q_0 dx = \sum_{i=1}^{N} q_0^{(i)}\ m(\Omega_i) = 0$, where $m(\Omega_i)$ is the measure of the subdomain Ω_i.

We denote the space of interface velocity variables on the subdomain Ω_i by $\mathbf{W}_\Gamma^{(i)}$, and the associated product space by $\mathbf{W}_\Gamma = \prod_{i=1}^{N} \mathbf{W}_\Gamma^{(i)}$; generally functions in $\mathbf{W}_\Gamma$ are discontinuous across the interface. Eliminating the independent subdomain interior variables $(\mathbf{u}_I, p_I)$ from the global problem (2), we have the global interface problem

$$\begin{bmatrix} \widehat{S}_\Gamma & \widehat{B}_{0\Gamma}^T \\ \widehat{B}_{0\Gamma} & 0 \end{bmatrix} \begin{bmatrix} \mathbf{u}_\Gamma \\ p_0 \end{bmatrix} = \begin{bmatrix} \mathbf{g}_\Gamma \\ 0 \end{bmatrix}. \tag{4}$$

Here, $\mathbf{g}_\Gamma$ is a reduced load vector obtained when the interior variables are eliminated. $\widehat{S}_\Gamma$ is assembled from subdomain Stokes Schur complements

$$S_\Gamma^{(i)} = A_{\Gamma\Gamma}^{(i)} - \begin{bmatrix} A_{\Gamma I}^{(i)} & B_{I\Gamma}^{(i)^T} \end{bmatrix} \begin{bmatrix} A_{II}^{(i)} & B_{II}^{(i)^T} \\ B_{II}^{(i)} & 0 \end{bmatrix}^{-1} \begin{bmatrix} A_{\Gamma I}^{(i)^T} \\ B_{I\Gamma}^{(i)} \end{bmatrix},$$

i.e., $\widehat{S}_\Gamma = \sum_{i=1}^{N} R_\Gamma^{(i)^T} S_\Gamma^{(i)} R_\Gamma^{(i)}$, where $R_\Gamma^{(i)}$ is the operator which maps functions in the continuous interface velocity space $\widehat{\mathbf{W}}_\Gamma$ to their subdomain components in the space $\mathbf{W}_\Gamma^{(i)}$. Denote by S_Γ and R_Γ the direct sums of $S_\Gamma^{(i)}$ and $R_\Gamma^{(i)}$, respectively. $\widehat{S}_\Gamma$ can then be written as $\widehat{S}_\Gamma = R_\Gamma^T S_\Gamma R_\Gamma$.

We denote the operator of the interface problem (4) by $\widehat{S}$. Since $\widehat{S}$ is symmetric and indefinite, we could use the minimal residual method, possibly with a positive definite block preconditioner, as in [10, Section 9.2], to solve problem (4). We will instead propose a different type of preconditioner and show that the preconditioned operator is positive definite, provided that a suitable set of primal constraints are chosen; cf. Assumption 1.

3 A BDDC Preconditioner for Stokes Equations

We introduce a partially assembled interface velocity space $\widetilde{\mathbf{W}}_\Gamma$ by

$$\widetilde{\mathbf{W}}_\Gamma = \widehat{\mathbf{W}}_\Pi \bigoplus \mathbf{W}_\Delta = \widehat{\mathbf{W}}_\Pi \bigoplus \left(\prod_{i=1}^{N} \mathbf{W}_\Delta^{(i)} \right).$$

Here, $\widehat{\mathbf{W}}_\Pi$ is the continuous coarse level, primal interface velocity space that is typically spanned by subdomain vertex nodal basis functions, and/or by interface edge and/or face basis functions with constant values, or with values of weight functions, on these edges or faces. These basis functions correspond to the primal interface velocity continuity constraints. We will always assume that the basis has been changed so that each primal basis function corresponds to an explicit degree of freedom which is shared by the neighboring subdomains; see [7], [5, Section 6], and

[4] for more details of the change of basis. The complimentary space $\mathbf{W}_{\Delta}$ is the direct sum of the subdomain dual interface velocity spaces $\mathbf{W}_{\Delta}^{(i)}$, which correspond to the remaining interface velocity degrees of freedom and are spanned by basis functions which vanish at the primal degrees of freedom. Thus, an element in the space $\widetilde{\mathbf{W}}_{\Gamma}$ has a continuous primal velocity and typically a discontinuous dual velocity component.

We define $R_{\Delta}^{(i)}$ as the operator which maps a function in the space $\widehat{\mathbf{W}}_{\Gamma}$ to its dual component in the space $\mathbf{W}_{\Delta}^{(i)}$. $R_{\Gamma\Pi}$ is the restriction operator from the space $\widehat{\mathbf{W}}_{\Gamma}$ to its subspace $\widehat{\mathbf{W}}_{\Pi}$ and $R_{\Pi}^{(i)}$ is the operator which maps $\widehat{\mathbf{W}}_{\Pi}$ into its Γ_i-component. $\widetilde{R}_{\Gamma}$ is the direct sum of $R_{\Gamma\Pi}$ and the $R_{\Delta}^{(i)}$, and it is a map from $\widehat{\mathbf{W}}_{\Gamma}$ into $\widetilde{\mathbf{W}}_{\Gamma}$.

The interface velocity Schur complement $\widetilde{S}_{\Gamma}$ is defined on the partially assembled interface velocity space $\widetilde{\mathbf{W}}_{\Gamma}$ by $\widetilde{S}_{\Gamma} = \overline{R}_{\Gamma}^{T} S_{\Gamma} \overline{R}_{\Gamma}$, where $\overline{R}_{\Gamma}$ maps $\widetilde{\mathbf{W}}_{\Gamma}$ into the product space $\mathbf{W}_{\Gamma}$ associated with the set of subdomains. We recall that the global interface Schur operator $\widehat{S}_{\Gamma}$ is obtained by fully assembling the $S_{\Gamma}^{(i)}$ across the subdomain interface. $\widehat{S}_{\Gamma}$ can therefore also be obtained from $\widetilde{S}_{\Gamma}$ by further assembling the dual interface velocity part, i.e., $\widehat{S}_{\Gamma} = \widetilde{R}_{\Gamma}^{T} \widetilde{S}_{\Gamma} \widetilde{R}_{\Gamma}$. Correspondingly, we define $\widetilde{B}_{0\Gamma}$, which is obtained from the subdomain operators $B_{0\Gamma}^{(i)}$ by assembling the primal interface velocity part only. The operator $\widehat{B}_{0\Gamma}$ can then be obtained from $\widetilde{B}_{0\Gamma}$ by assembling the dual interface velocity part on the subdomain interfaces, i.e., $\widehat{B}_{0\Gamma} = \widetilde{B}_{0\Gamma} \widetilde{R}_{\Gamma}$. We can therefore write $\widehat{S}$, the operator of the global interface problem (4), as $\widehat{S} = \widetilde{R}^{T} \widetilde{S} \widetilde{R}$, where

$$\widetilde{R} = \begin{bmatrix} \widetilde{R}_{\Gamma} & \\ & I \end{bmatrix}, \quad \widetilde{S} = \begin{bmatrix} \widetilde{S}_{\Gamma} & \widetilde{B}_{0\Gamma}^{T} \\ \widetilde{B}_{0\Gamma} & 0 \end{bmatrix}. \tag{5}$$

To define the BDDC preconditioner, we need certain scaling functions. For each interface node $x \in \Gamma_i$, we set $\delta_i^{\dagger}(x) = 1/\mathbf{N}_x, x \in \Gamma_i$, where $\mathbf{N}_x$ is the number of subdomain to which x belongs. Given the scaling factors at the subdomain interface nodes, we can define scaled restriction operators $R_{D,\Delta}^{(i)}$. Each row of $R_{\Delta}^{(i)}$ has only one nonzero entry which corresponds to a node $x \in \Gamma_i$, and multiplying each such element with the scaling factor $\delta_i^{\dagger}(x)$ gives us $R_{D,\Delta}^{(i)}$. The scaled operator $\widetilde{R}_{D,\Gamma}$ is the direct sum of $R_{\Gamma\Pi}$ and the $R_{D,\Delta}^{(i)}$. For elasticity problems, these scaling factors should depend on the first Lamé constant μ, which can be allowed to change across the interface between neighboring subdomains; see [10, Section 8.5.1] and [5].

The BDDC preconditioner for solving the interface saddle point problem (4) is $M^{-1} = \widetilde{R}_{D}^{T} \widetilde{S}^{-1} \widetilde{R}_{D}$, where $\widetilde{R}_{D}$ is of the same form as $\widetilde{R}$ in (5), except that $\widetilde{R}_{\Gamma}$ is replaced by $\widetilde{R}_{D,\Gamma}$. To compute the product of $\widetilde{S}^{-1}$ and a vector, a coarse level saddle point problem, for the primal variables, and subdomain Neumann problems, each with a few primal constraints, need to be solved; cf. [7, 8].

4 Condition Number Bounds

We define an average operator

$$E_D = \widetilde{R}\widetilde{R}_{D}^{T} = \begin{bmatrix} \widetilde{R}_{\Gamma} & \\ & I \end{bmatrix} \begin{bmatrix} \widetilde{R}_{D,\Gamma}^{T} & \\ & I \end{bmatrix} = \begin{bmatrix} E_{D,\Gamma} & \\ & I \end{bmatrix}, \tag{6}$$

which maps $\widetilde{\mathbf{W}}_\Gamma \times Q_0$, with generally discontinuous interface velocities, to elements with continuous interface velocities in the same space. $E_{D,\Gamma} = \widetilde{R}_\Gamma \widetilde{R}_{D,\Gamma}^T$, provides the average of the interface velocities across the interface Γ. Denoting the primal and dual parts of $\mathbf{w}_\Gamma$ by $\mathbf{w}_\Pi$ and $\mathbf{w}_\Delta$, we can write $E_{D,\Gamma}\mathbf{w}_\Gamma$ as the direct sum of $\mathbf{w}_\Pi$ and $E_{D,\Delta}\mathbf{w}_\Delta$, where $E_{D,\Delta}\mathbf{w}_\Delta$ is the dual part of the averaged vector.

The following two assumptions will be needed in the condition number bound of the preconditioned operator.

Assumption 1 *For any* $\mathbf{w}_\Delta \in \mathbf{W}_\Delta$, $\int_{\partial\Omega_i} \mathbf{w}_\Delta^{(i)} \cdot \mathbf{n} = 0$ *and* $\int_{\partial\Omega_i} (E_{D,\Delta}\mathbf{w}_\Delta)^{(i)} \cdot \mathbf{n} = 0$, *where* $\mathbf{n}$ *is the unit outward normal of* $\partial\Omega_i$.

Assumption 2 *There exists a positive constant* C*, which is independent of* H*,* h*, and the number of subdomains, such that*

$$|\overline{R}_\Gamma \left(E_{D,\Gamma}\mathbf{w}_\Gamma\right)|_{\mathbf{E}(\Gamma)} \le C\left(1+\log\frac{H}{h}\right)|\overline{R}_\Gamma \mathbf{w}_\Gamma|_{\mathbf{E}(\Gamma)}, \quad \forall \mathbf{w}_\Gamma \in \widetilde{\mathbf{W}}_\Gamma,$$

where $|\cdot|_{\mathbf{E}(\Gamma)}$ *is defined on the space* $\mathbf{W}_\Gamma$ *by* $|\mathbf{w}_\Gamma|^2_{\mathbf{E}(\Gamma)} = \sum_{i=1}^N |\mathbf{w}_\Gamma^{(i)}|^2_{\mathbf{E}(\Gamma_i)}$ *with*

$$|\mathbf{w}_\Gamma^{(i)}|_{\mathbf{E}(\Gamma_i)} = \inf_{\substack{\mathbf{v}^{(i)} \in (H^1(\Omega_i))^d \\ \mathbf{v}^{(i)}|_{\Gamma_i} = \mathbf{w}_\Gamma^{(i)}}} \|\varepsilon(\mathbf{v}^{(i)})\|_{L^2(\Omega_i)}.$$

These two assumptions can be satisfied with an appropriate choice of the primal continuity constraints on the interface velocity variables; for two-dimensional problems, Assumptions 1 and 2 are satisfied if all subdomain vertices are primal, i.e, both components of the velocity are continuous at those nodes, and $\int_{\Gamma^{ij}} \mathbf{w}_\Gamma^{(i)} \cdot \mathbf{n}_{ij} = \int_{\Gamma^{ij}} \mathbf{w}_\Gamma^{(j)} \cdot \mathbf{n}_{ij}$, is enforced on all the subdomain interface edges. Here $\mathbf{n}_{ij}$ is a normal of Γ_{ij}. For the more complicated three-dimensional case, see [2, 8, 5].

The interface velocity subspaces $\widehat{\mathbf{W}}_{\Gamma,B}$ and $\widetilde{\mathbf{W}}_{\Gamma,B}$ are defined by $\widehat{\mathbf{W}}_{\Gamma,B} = \{\mathbf{w}_\Gamma \in \widehat{\mathbf{W}}_\Gamma \mid \widehat{B}_{0\Gamma}\mathbf{w}_\Gamma = 0\}$, and $\widetilde{\mathbf{W}}_{\Gamma,B} = \{\mathbf{w}_\Gamma \in \widetilde{\mathbf{W}}_\Gamma \mid \widetilde{B}_{0\Gamma}\mathbf{w}_\Gamma = 0\}$. We will call $\widehat{\mathbf{W}}_{\Gamma,B} \times Q_0$ and $\widetilde{\mathbf{W}}_{\Gamma,B} \times Q_0$ the *benign subspaces* of $\widehat{\mathbf{W}}_\Gamma \times Q_0$ and $\widetilde{\mathbf{W}}_\Gamma \times Q_0$, respectively.

The preconditioned operator $\widetilde{R}_D^T \widetilde{S}^{-1} \widetilde{R}_D \widehat{S}$ is indefinite on the space $\widehat{\mathbf{W}}_\Gamma \times Q_0$, since both $\widehat{S}$ and $\widetilde{S}$ are indefinite. However, both $\widehat{S}$ and $\widetilde{S}$ are positive semi-definite, when restricted to the benign subspaces $\widehat{\mathbf{W}}_{\Gamma,B} \times Q_0$ and $\widetilde{\mathbf{W}}_{\Gamma,B} \times Q_0$, respectively. We will also know, from Lemma 1, that $M^{-1}\widehat{S}$ maps $\widehat{\mathbf{W}}_{\Gamma,B} \times Q_0$ into itself and that $M^{-1}\widehat{S}$ is symmetric with respect to the bilinear form $\langle\cdot,\cdot\rangle_{\widehat{S}}$. Theorem 1 will show that $M^{-1}\widehat{S}$ is positive definite, when restricted to the benign subspace $\widehat{\mathbf{W}}_{\Gamma,B} \times Q_0$. Therefore a preconditioned conjugate gradient method can be used. The following lemmas will be used in the proof of Theorem 1.

Lemma 1 *Let Assumption 1 hold. Then,* $\widetilde{R}_D^T \mathbf{w} \in \widehat{\mathbf{W}}_{\Gamma,B} \times Q_0$*, for any* $\mathbf{w} \in \widetilde{\mathbf{W}}_{\Gamma,B} \times Q_0$.

Lemma 2 *Let Assumptions 1 and 2 hold. There then exists a positive constant C, which is independent of H, h, and the number of subdomains, such that,*

$$< E_D\mathbf{w}, E_D\mathbf{w} >_{\widetilde{S}} \leq C\frac{1}{\beta^2}\left(1+\log\frac{H}{h}\right)^2 < \mathbf{w}, \mathbf{w} >_{\widetilde{S}}, \quad \forall \mathbf{w} \in \widetilde{\mathbf{W}}_{\Gamma,B} \times Q_0.$$

Here, β is the inf-sup stability constant of Equation (3).

Theorem 1 *Let Assumptions 1 and 2 hold. The preconditioned operator $M^{-1}\widehat{S}$ is then symmetric, positive definite with respect to the bilinear form $\langle\cdot,\cdot\rangle_{\widehat{S}}$ on the benign space $\widehat{\mathbf{W}}_{\Gamma,B} \times Q_0$. Its minimum eigenvalue is 1 and its maximum eigenvalue is bounded by*

$$C\frac{1}{\beta^2}\left(1+\log\frac{H}{h}\right)^2.$$

Here, C is a constant which is independent of H, h, and the number of subdomains and β is the inf-sup stability constant defined in Equation (3).

Just as in the positive definite elliptic case, we can also establish that the preconditioned BDDC operator and the preconditioned FETI-DP operator in [6] have the same nonzero eigenvalues; cf. [7, 8]. We have,

Theorem 2 *Let Assumption 1 hold. The preconditioned FETI–DP and BDDC operators have the same nonzero eigenvalues, when the same set of primal constraints are applied.*

5 Numerical Experiments

We solve an incompressible Stokes problem on the domain $\Omega = [0,1] \times [0,1]$ with Dirichlet boundary condition, where the velocity is $(1,0)$ on the upper side, and vanishes on the other three sides. We use a uniform mesh, as in Figure 1. The mixed finite element method is also indicated in Figure 1; the velocity is continuous and linear in each element and the pressure is constant on macro elements which are unions of four triangles.

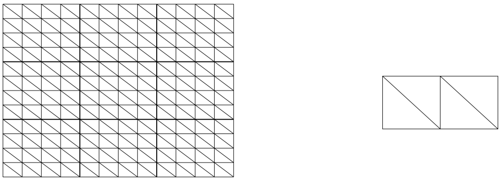


Fig. 1. The mesh and the mixed finite elements.

Both the BDDC and FETI–DP algorithms have been tested. The preconditioned conjugate gradient method is used and the iteration is halted when the L_2-norm of the residual has been reduced by a factor 10^{-6}. The primal velocity space is spanned by the subdomain vertex nodal basis functions for both components and by a constant vector in the direction normal to the edge for each interface edge. Both Assumptions 1 and 2 are then satisfied. From Tables 1 and 2, we see that the preconditioned BDDC and FETI–DP operators are both positive definite and quite well-conditioned as established in Theorems 1 and 2. We observe that the extreme eigenvalues and the iteration counts of the BDDC and FETI–DP algorithms match very well, and that the condition numbers of both algorithms are independent of the number of subdomains, and increases only slowly with the number of elements across each subdomain, all as predicted by the theory.

Table 1. Spectral bounds and iteration counts for a pair of BDDC and FETI–DP algorithms, with different number of subdomains, for $H/h = 8$ and a primal space spanned by both corner and normal edge basis functions.

Num. of subs	BDDC			FETI–DP		
$n_x \times n_y$	λ_{min}	λ_{max}	$iter.$	λ_{min}	λ_{max}	$iter.$
4×4	1.00	3.14	11	1.00	3.14	11
8×8	1.00	3.88	12	1.00	3.88	12
12×12	1.00	4.02	12	1.00	4.02	13
16×16	1.00	4.06	12	1.00	4.07	13
20×20	1.00	4.08	12	1.00	4.08	13

Table 2. Spectral bounds and iteration counts for a pair of BDDC and FETI–DP algorithms, with different H/h, for 4×4 subdomains and a primal space spanned by both corner and normal edge basis functions.

	BDDC			FETI–DP		
H/h	λ_{min}	λ_{max}	$iter.$	λ_{min}	λ_{max}	$iter.$
4	1.00	2.17	8	1.00	2.17	9
8	1.00	3.14	11	1.00	3.14	11
16	1.00	4.22	13	1.00	4.22	12
32	1.00	5.42	14	1.00	5.42	14

When Assumption 1 is not satisfied, e.g., when only vertex velocity variables are primal, the preconditioned BDDC operator is no longer positive definite, and the iteration counts will depend on both the number of subdomains as well as on the number of elements across each subdomain; cf. [8].

References

1. C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258.
2. ———, *A substructuring preconditioner for nearly incompressible elasticity problems*, Tech. Rep. SAND 2004-5393, Sandia National Laboratories, October 2004.
3. Y. Fragakis and M. Papadrakakis, *The mosaic of high performance domain decomposition methods for structural mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods*, Comput. Methods Appl. Mech. Engrg, 192 (2003), pp. 3799–3830.
4. A. Klawonn and O. Rheinbach, *A parallel implementation of Dual-Primal FETI methods for three dimensional linear elasticity using a transformation of basis*, Tech. Rep. SM-E-601, Univ. Duisburg-Essen, Department of Mathematics, Germany, February 2005.
5. A. Klawonn and O. Widlund, *Dual-Primal FETI methods for linear elasticity*, Tech. Rep. 855, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, September 2004.
6. J. Li, *Dual-Primal FETI methods for stationary Stokes and Navier-Stokes equations*, Tech. Rep. 816, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, 2001.
7. J. Li and O. B. Widlund, *FETI-DP, BDDC, and block Cholesky methods*, Tech. Rep. 857, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 2004.
8. ———, *BDDC algorithms for incompressible Stokes equations*, Tech. Rep. TR-861, New York University, Department of Computer Science, 2005.
9. J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Appl. Numer. Math., 54 (2005), pp. 167–193.
10. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Springer Series in Computational Mathematics, Springer, 2005.

Adaptive Coarse Space Selection in the BDDC and the FETI-DP Iterative Substructuring Methods: Optimal Face Degrees of Freedom

Jan Mandel[1*] and Bedřich Sousedík[2†]

[1] Department of Mathematics, University of Colorado at Denver, P.O. Box 173364, Campus Box 170, Denver, CO 80217, USA. jmandel@math.cudenver.edu

[2] Department of Mathematics, Faculty of Civil Engineering, Czech Technical University in Prague, Thákurova 7, 166 36 Prague 6, Czech Republic. bedrich.sousedik@fsv.cvut.cz

Summary. We propose an adaptive selection of the coarse space of the BDDC and FETI-DP iterative substructuring methods by adding coarse degrees of freedom (dofs) on faces between substructures constructed using eigenvectors associated with the faces. Provably the minimal number of coarse dofs on the faces is added to decrease a heuristic indicator of the condition number under a target value specified a priori. It is assumed that the corner dofs are already sufficient to prevent relative rigid body motions of any two substructures with a common face. It is shown numerically on a 2D elasticity problem that the indicator is reasonably close to the actual condition number and that the method can find automatically the hard part of the problem and concentrate the computational work there to achieve the target value for the condition number and good convergence of the iterations, at a modest cost.

1 Introduction

The BDDC and FETI-DP methods are iterative substructuring methods that use coarse degrees of freedom associated with corners and edges (in 2D) or faces (in 3D, further on just faces) between substructures, and they are currently the most advanced versions of the BDD and FETI families of methods. The BDDC method [2] is a Neumann-Neumann method of Schwarz type [3]. The BDDC method iterates on the system of primal variables reduced to the interfaces between the substructures

[*] Supported by NSF grant 0325314 and by Sandia National Laboratories.

[†] Partially supported by the grant agency of the Czech Republic under grant 106/05/2731 and Program of the Information Society 1ET400300415.

and it can be understood as a further development of the BDD method [10]. The FETI-DP method [5, 4] is a dual method that iterates on a system for Lagrange multipliers that enforce continuity on the interfaces. Algebraic relations between FETI and BDD methods were pointed out in [6, 7, 12]. A common bound on the condition number of both the FETI and the BDD method in terms of a single inequality was given in [7]. In the case of corner constraints only, methods identical to BDDC were derived as primal versions of FETI-DP in [1, 6]. In [11], it was proved that the eigenvalues of BDDC and FETI-DP are identical and a bound on the condition number was obtained in terms of matrix data only.

In this contribution, we show how to use the algebraic estimate of the condition number from [11] to develop an adaptive fast method. First we estimate the condition number as the solution of an eigenvalue problem, then obtain a reliable heuristic indicator from the eigenvalues for two substructures with a common face faces. Finally, we show how to use the eigenvectors to obtain coarse degrees of freedom that result in an optimal decrease of the indicator. We demonstrate on numerical examples that the indicator is quite close that such an adaptive approach results in the concentration of computational work in a small part of the problem, leading to good convergence behavior at a small added cost.

Related work on adaptive coarse space selection has focused on the global problem of selecting the smallest number of corners to prevent coarse mechanisms [9] and the smallest number of coarse degrees of freedom to assure asymptotically optimal convergence estimates [8]. In contrast, our indicator of condition number is local in nature and we assume that corner degrees of freedom are already sufficient to prevent relative rigid body motions of any two substructures with a common face.

2 Formulation of BDDC and FETI-DP

We need to briefly recall the formulation of the methods and the condition number bound. Let K_s be the stiffness matrix and v_s the vector of degrees of freedom (dofs) for substructure s. We want to solve the problem in decomposed form

$$\frac{1}{2}v^T Kv - v^T f \to \min, \quad v = \begin{bmatrix} v_1 \\ \vdots \\ v_N \end{bmatrix} \quad K = \begin{bmatrix} K_1 & & \\ & \ddots & \\ & & K_N \end{bmatrix}$$

subject to continuity dofs between substructures. Partitioning the dofs in each subdomain s into internal and interface (boundary)

$$K_s = \begin{bmatrix} K_s^{ii} & K_s^{ib} \\ K_s^{ib\,T} & K_s^{bb} \end{bmatrix}, \quad v_s = \begin{bmatrix} v_s^i \\ v_s^b \end{bmatrix}, \quad f_s = \begin{bmatrix} f_s^i \\ f_s^b \end{bmatrix},$$

and eliminating the interior dofs we obtain the problem reduced to interfaces

$$\frac{1}{2}w^T Sw - w^T g \to \min, \quad S = \mathrm{diag}(S_s), \quad S_s = K_s^{bb} - K_s^{ib^T} K_s^{ii^{-1}} K_s^{ib},$$

again subject to continuity of dofs between substructures.

In BDD type methods, the continuity of dofs between substructures is enforced by imposing common values on substructures interfaces: $w = Ru$ for some u, where

$$R = \begin{bmatrix} R_1 \\ \vdots \\ R_N \end{bmatrix}$$

and R_s is the operator of restriction of global dofs on the interfaces to substructure s. In FETI type methods, continuity of dofs between substructures is enforced by the constraint $Bw = 0$, where the entries of B are typically $0, \pm 1$. By construction, we have $R_s R_s^T = I$ and $\operatorname{range} R = \operatorname{null} B$.

Node is the set of all dofs associated with the same location in space. Nodes such that no other node is adjacent to the same set of substructures are called corners. Face is the set of all dofs shared by two substructures that contains more than one node.

A BDDC or FETI-DP method is specified by the choice of coarse dofs and the choice of weights for intersubdomain averaging. To define the coarse problem for BDDC, choose a matrix Q_P^T that selects coarse dofs u_c from global interface dofs u, e.g. as values at corners or averages on faces:

$$u_c = Q_P^T u.$$

The space $\widetilde{W}$ will consist of all vectors of substructure interface dofs such that the coarse dofs are continuous between substructures,

$$\widetilde{W} = \{ w \in W : \exists u_c \forall s : C_s w_s = R_{cs} u_c \}$$

where $C_s = R_{cs} Q_P^T R_s^T$ maps a collection of substructure dofs to a collection of coarse dofs on substructure s, and R_{cs} restricts a vector of all coarse dof values into a vector of coarse dof values that can be nonzero on substructure s. The dual approach in FETI-DP is to construct Q_D such that $\widetilde{W} = \operatorname{null} Q_D^T B$.

In BDDC, the intersubdomain averaging is defined by the matrices $D_P = \operatorname{diag}(D_{Ps})$ that form a decomposition of unity, $R^T D_P R = I$. The corresponding dual matrices in FETI-DP are $B_D = [D_{D1} B_1, \ldots D_{DN} B_N]$, where the dual weights D_{Ds} are defined so that $B_D^T B + R R^T D_P = I$.

The BDDC method is then the method of conjugate gradients for the assembled system $Au = R^T g$ with the system matrix $A = R^T S R$ and the preconditioner P defined by $Pr = R^T D_P (\Psi u_c + z)$, where u_c is the solution of the coarse problem $\Psi^T S \Psi u_c = \Psi^T D_P^T R r$ and z is the solution of

$$\begin{array}{lcl} Sz + C^T \mu & = & D_P^T R r \\ Cz & = & 0 \end{array},$$

which is a collection of independent substructure problems. The coarse basis functions Ψ are defined by energy minimization,

$$\begin{bmatrix} S & C^T \\ C & 0 \end{bmatrix} \begin{bmatrix} \Psi \\ \Lambda \end{bmatrix} = \begin{bmatrix} 0 \\ R_c \end{bmatrix}.$$

The FETI-DP method solves the saddle point problem

$$\min_{w \in \widetilde{W}} \max_{\lambda} \mathcal{L}(w, \lambda) = \max_{\lambda} \min_{w \in \widetilde{W}} \mathcal{L}(w, \lambda),$$

where $\mathcal{L}(w, \lambda) = \frac{1}{2} w^T S w - w^T f + w^T B^T \lambda$ by iterating on the dual problem $\frac{\partial \mathcal{F}(\lambda)}{\partial \lambda} = F\lambda - h = 0$, where

$$\mathcal{F}(\lambda) = \min_{w \in \widetilde{W}} \mathcal{L}(w, \lambda),$$

by conjugate gradients with the preconditioner $M = B_D S B_D^T$. See [11] for more details.

3 Indicator of the Condition Number

Theorem 1 ([11]). *The eigenvalues of the preconditioned operators PA of BDDC and MF of FETI-DP are same except for eigenvalues of zero and one, and the condition numbers satisfy*

$$\kappa_{\text{BDDC}} = \kappa_{\text{FETI-DP}} \leq \omega = \sup_{w \in \widetilde{W}} J(\omega), \quad J(\omega) = \frac{\left\| B_D^T B w \right\|_S^2}{\| w \|_S^2}.$$

Here, the condition number is the ratio of the largest and the smallest nonzero eigenvalue. Zero eigenvalues in FETI-DP are caused by redundant constraints, common in practice.

As an indicator of the condition number, we propose the maximum of the bounds from Theorem 1 computed by considering only one pair of adjacent substructures s, t with a common face at a time:

$$\omega \approx \widetilde{\omega} = \max_{st} \omega_{st}, \quad \omega_{st} = \sup_{w_{st} \in \widetilde{W}_{st}} J_{st}(w_{st}). \tag{1}$$

All quantities with the subscript $_{st}$ are the same as without the subscript but defined using the domain consisting of the substructures s and t only.

Theorem 2. *Let $a > 0$, Π_{st} be the orthogonal projection onto $\widetilde{W}_{st}$, and $I - \overline{\Pi}_{st}$ be the orthogonal projection onto*

$$\operatorname{null}(\Pi_{st} S_{st} \Pi_{st} + a(I - \Pi_{st})).$$

Then the stationary values $\omega_{st,1} \geq \omega_{st,2} \geq \ldots$ and the corresponding stationary vectors $w_{st,k}$ of the Rayleigh quotient J_{st} on $\widetilde{W}_{st}$ satisfy

$$X_{st} w_{st,k} = \omega_{st,k} Y_{st} w_{st,k} \tag{2}$$

with Y_{st} positive definite, where

$$\begin{aligned} X_{st} &= \Pi_{st} B_{st}^T B_{Dst} S_{st} B_{Dst}^T B_{st} \Pi_{st}, \\ Y_{st} &= \left(\overline{\Pi}_{st} \left(\Pi_{st} S_{st} \Pi_{st} + a(I - \Pi_{st}) \right) \overline{\Pi}_{st} + a\left(I - \overline{\Pi}_{st}\right) \right) \end{aligned}$$

The eigenvalue problem (2) is obtained by projecting the gradient of the Rayleigh quotient $J_{st}(w_{st})$ onto the complement in $\widetilde{W}_{st}$ of the subspace where its denominator $\|w_{st}\|_{S_{st}}^2 = 0$, in two steps. Both projections Π_{st} and $\overline{\Pi}_{st}$ are computed by matrix algebra, which is straightforward to implement numerically. The computation of Π_{st} involves minimization with Lagrange multipliers for the condition that the values of the coarse dofs on the the substructures s and t coincide. The projection $I - \overline{\Pi}_{st}$ is onto a subspace of null S_{st}, and it is easily constructed computationally if a matrix Z_{st} is given such that null $S_{st} \subset$ range Z_{st}. The matrix Z_{st} with columns consisting of the coarse basis functions can be used because the span of the coarse basis functions contains the rigid body modes. However, often these modes are available directly, which leads to a smaller matrix Z_{st} and thus cheaper computation. Since Y_{st} is positive definite, its Choleski decomposition exists, and we reduce (2) a symmetric eigenvalue problem, which is easier and more efficient to solve numerically.

4 Optimal Coarse Degrees of Freedom on Faces

Writing $\widetilde{W}_{st}$ in the dual form $\widetilde{W}_{st}=\operatorname{null} Q_{Dst}^T B_{st}$ suggests how to add coarse dofs in an optimal way to decrease the value of indicator $\widetilde{\omega}$. The following theorem follows immediately from the standard characterization of eigenvalues as minima and maxima of the Rayleigh quotient on subspaces spanned by eigvectors, applied to (2).

Theorem 3. *Suppose $\ell_{st} \geq 0$ and the dual coarse dof selection matrix Q_{Dst}^T is augmented to become $\left[Q_{Dst}^T, q_{Dst,1}^T, \ldots, q_{Dst,\ell_{st}}^T\right]$ with $q_{Dst,k}^T = w_{st,k}^T B_{st}^T B_{Dst} S_{st} B_{Dst}^T$, where $w_{st,k}^T$ are the eigenvectors from (2). Then $\omega_{st} = \omega_{st,\ell_{st}+1}$, and $\omega_{st} \geq \omega_{st,\ell_{st}+1}$ for any other augmentation of Q_{Dst}^T by at most ℓ_{st} columns.*

In particular, if $\omega_{st,\ell_{st}+1} \leq \tau$ for all pairs of substructures s, t with a common face, then $\widetilde{\omega} \leq \tau$.

Theorem 3 allows us to guarantee that the condition number indicator $\widetilde{\omega} \leq \tau$ for a given target value τ, by adding the smallest possible number of face coarse dofs.

The primal coarse space selection mechanism that corresponds to this augmentation can be seen easily in the case when the entries of B_{st} are $+1$ for substructure s and -1 for substructrure t. Then $w_{st} \in \widetilde{W}_{st}$ can be written as

$$Q_{Dst}^T (I_{st} w_s - I_{ts} w_t) = 0$$

where I_{st} is the $0-1$ matrix that selects from w_s the degrees of freedom on the intersection of the substructures s and t. Each column of q_D of Q_{Dst} defines a coarse degree of freedom associated with the interface of substructures s and t. The corresponding column q_P of Q_P is such that

$$q_P^T R_s^T = q_D^T I_{st} \tag{3}$$

Because R_s is also a $0-1$ matrix, this means that the vector q_P is formed by a scattering of the entries of the vector q_D.

5 Numerical Results

Consider plane elasticity discretized by bilinear elements on a rectangular mesh decomposed into 16 substructure, with one single edge between substructures being jagged (Fig. 1). We have computed the eigenvalues and eigenvectors of (2) by setting up the matrices and using standard methods for the symmetric eigenvalue problem in Matlab. The eigenvalues $\omega_{st,k}$ associated with edges between substructures (Table 1) clearly distinguish between the single problematic edge and the others. Adding

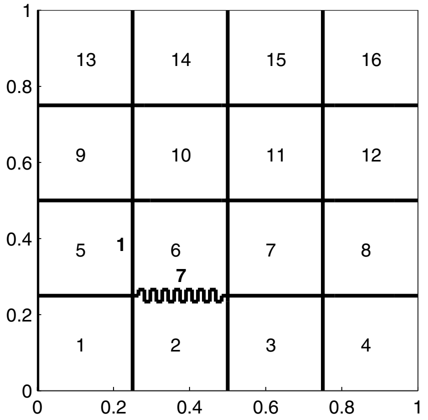


Fig. 1. Mesh with $H/h = 16$, 4×4 substructures, and one jagged edge between substructures 2 and 6. Zero displacement is imposed on the left edge. For compressible elasticity (Tables 1 and 2(a)) and tolerance $\tau = 10$, 7 coarse dofs at the jagged edge and 1 coarse dof at an adjacent edge are added automatically.

s	t	$\omega_{st,1}$	$\omega_{st,2}$	$\omega_{st,3}$	$\omega_{st,4}$	$\omega_{st,5}$	$\omega_{st,6}$	$\omega_{st,7}$	$\omega_{st,8}$
1	2	3.7	2.3	1.4	1.3	1.1	1.1	1.1	1.1
1	5	5.8	3.2	2.3	1.4	1.2	1.1	1.1	1.1
2	3	6.0	2.5	1.7	1.3	1.2	1.1	1.	1.1
2	6	21.7	19.5	17.8	14.9	14.5	11.7	11.2	9.7
3	4	3.3	2.3	1.4	1.3	1.1	1.1	1.1	1.1
3	7	7.1	5.1	3.2	1.8	1.4	1.3	1.2	1.1
4	8	5.9	3.4	2.6	1.4	1.2	1.1	1.1	1.1
5	6	12.0	4.9	4.4	1.8	1.6	1.3	1.3	1.2
5	9	5.9	3.4	2.6	1.4	1.3	1.3	1.1	1.1
6	7	8.7	4.9	3.9	1.8	1.5	1.3	1.2	1.1
6	10	7.3	4.8	3.4	1.8	1.4	1.3	1.2	1.1

Table 1. Several largest eigenvalues $\omega_{st,k}$ for several edges for the elasticity problem from Fig. 1 with $H/h = 16$. $(s,t) = (2,6)$ is the jagged edge.

H/h	$Ndof$	τ	Nc	$\widetilde{\omega}$	κ	it
4	578		42	10.3	5.6	19
		10	43	5.2	4.0	18
		3	44	3.0	4.0	18
		2	58	2.0	2.8	15
16	8450		42	22	20	37
		10	50	8.7	9.9	29
		3	77	3.0	4.6	22
		2	112	2.0	2.6	15
64	132098		42	87	40	55
		10	89	9.8	9.9	36
		3	151	3.0	4.7	22
		2	174	2.0	2.9	17

(a) compressible elasticity

H/h	$Ndof$	τ	Nc	$\widetilde{\omega}$	κ	it
4	578		42	285	208	64
		10	68	8.0	8.6	28
		3	89	2.9	4.6	22
		2	114	2.0	2.6	16
16	8450		42	1012	1010	161
		10	87	9.8	9.9	29
		3	77	3.0	4.6	22
		2	126	2.0	2.9	19
64	132098		42	6910	NA	∞
		10	183	9.8	9.7	37
		3	213	3.0	4.9	26
		2	274	2.0	3.0	20

(b) almost incompressible

Table 2. BDDC for plane elasticity on a square with one jagged edge. The Lamé coefficients are $\lambda = 1$ and $\mu = 2$ for (a), and $\lambda = 1000$ and $\mu = 2$ for (b). H/h is the number of elements per substructure in one direction, $Ndof$ the number of dofs in the problem, τ the condition number tolerance as in Theorem 3, Nc the number of coarse dofs, $\widetilde{\omega}$ the value of the condition number indicator from (1), κ the approximate condition number from the Lanczos sequence in conjugate gradients, and it the number of BDDC iterations for relative residual tolerance 10^{-8}.

the coarse dofs created from the associated eigenvectors according to Theorem 3 decreases the value of the condition number indicator $\widetilde{\omega}$ and improves convergence at the cost of increasing the number of coarse dofs. This effect is even more pronounced for almost incompressible elasticity where the iterations converge poorly or not at all without the additional coarse dofs. This incompressible elasticity problem is particularly hard for an iterative method because standard bilinear elements were used instead of stable elements or reduced integration. In all cases, values of the condition number indicator $\widetilde{\omega}$ are quite close to the actually observed condition numbers κ (Table 2).

References

1. J.-M. Cros, *A preconditioner for the Schur complement domain decomposition method*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003, pp. 373–380.
2. C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258.
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 (1995), pp. 121–155.
4. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.

5. C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numer. Lin. Alg. Appl., 7 (2000), pp. 687–714.
6. Y. Fragakis and M. Papadrakakis, *The mosaic of high performance domain decomposition methods for structural mechanics: Formulation, interrelation and numerical efficiency of primal and dual methods*, Comput. Methods Appl. Mech. Engrg, 192 (2003), pp. 3799–3830.
7. A. Klawonn and O. B. Widlund, *FETI and Neumann–Neumann iterative substructuring methods: Connections and new results*, Comm. Pure Appl. Math., 54 (2001), pp. 57–90.
8. ———, *Selecting constraints in dual-primal FETI methods for elasticity in three dimensions*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., vol. 40 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 67–81.
9. M. Lesoinne, *A FETI-DP corner selection algorithm for three-dimensional problems*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003, pp. 217–223.
10. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.
11. J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Appl. Numer. Math., 54 (2005), pp. 167–193.
12. 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*, Int. J. Numer. Meth. Engrg., 46 (1999), pp. 501–533.

Applications of the FETI-DP-RBS-LNA Algorithm on Large Scale Problems with Localized Nonlinearities

Jun Sun[1], Pan Michaleris[2], Anshul Gupta[3], and Padma Raghavan[4]

[1] Department of Mechanical and Nuclear Engineering, 307 Reber Building, Pennsylvania State University, University Park, PA 16802, USA. junsun@psu.edu

[2] Department of Mechanical and Nuclear Engineering, 232 Reber Building, Pennsylvania State University, University Park, PA 16802, USA. pxm32@psu.edu

[3] IBM T. J. Watson Research Center, P. O. Box 218, Yorktown Heights, NY 10598, USA. anshul@watson.ibm.com

[4] Department of Computer Science and Engineering, 343K IST Building, Pennsylvania State University, University Park, PA 16802, USA. raghavan@cse.psu.edu

Summary. Large scale computing is a well-known research area since it is heavily desired by many science and engineering disciplines to simulate complex and sophisticated problems. However, due to the unprecedented amount of data and computations involved, it also poses challenges for current available numerical algorithms and computer hardware. In this paper, the Dual-Primal Finite Element Tearing and Interconnecting method (FETI-DP) is carefully investigated, and a reduced back-substitution (RBS) algorithm is proposed to accelerate the time consuming preconditioned conjugate gradient (PCG) iterations involved in the interface problems. Linear and nonlinear identification analysis (LNA) is also proposed for large scale problems with localized nonlinearities. This combined approach is named the FETI-DP-RBS-LNA algorithm. Serial CPU time of this approach is measured and compared with a direct sparse solver and the standard FETI-DP method on a welding problem. Parallelism of the FETI-DP-RBS-LNA algorithm is also implemented by using MPI and the performance is reported. The empirical results demonstrate the effectiveness of the proposed computational approach for welding applications, which is representative of a large class of three dimensional linear-nonlinear large scale problems with localized nonlinearities.

1 Introduction

Large scale finite element analysis is an important research area due to its wide applicability in modeling and simulating complicated scientific and engineering applications, such as structural mechanics, heat transfer, and biomechanics. For realistic and sophisticated models, high density meshes are required to capture the underlying physics in areas that are of particular interest or with complex geometry or loading. Accordingly, the total degrees of freedom in systems discritized by finite element method may easily exceed millions, and it poses many computational challenges for current available numerical algorithms as well as computer hardware.

Extensive research has been conducted to develop efficient and reliable numerical methods which have the capabilities to solve large scale systems arising from various disciplines. Two well-known approaches in this field are direct and iterative methods. Direct sparse solvers are recognized as robust and efficient choices for most of the applications, and they are widely employed in many commercial finite element softwares. However, the high memory demands and the not-so-well parallel scalability of direct sparse solvers restrict its applications to large scale problems [4]. Traditional iterative solvers are excellent from the memory point of view. However, they are problem dependent and the convergence is not guaranteed. For complex ill-conditioned engineering problems, they are not as reliable as direct sparse solvers.

Several novel approaches, such as Domain Decomposition (DD) methods and adaptive meshing methods [5, 6], have also been studied extensively for their possible applications to solve large scale systems. DD methods are based on the native divide and conquer concept, they partition the physical domain into subdomains with either overlapping or non-overlapping interfaces. Coarse-grain parallel processing can then be applied to the computations of these subdomains to reduce overall analysis time. Adaptive meshing refines or coarsens the meshes in different regions of the model during the analysis based on their corresponding resolution requirements. Therefore, this approach is capable of reducing the computational costs while still maintain the quality of the solution.

The objective of this paper is to present the FETI-DP-RBS-LNA algorithm [7] and to investigate its serial and parallel performance for large scale problems with localized nonlinearity. The FETI-DP-RBS-LNA algorithm is based on one type of DD methods, the Dual-Primal Finite Element Tearing and Interconnecting method (FETI-DP) [2, 1]. Reduced Back-Substitution (RBS) algorithm is proposed to accelerate costly local back-substitutions, and Linear and Nonlinear Analysis (LNA) is introduced to reduce unnecessary re-factorizations of linear subdomains in the analysis. The distributed version of this algorithm is implemented with Message Passing Interface (MPI) and the performance is measured on a distributed PC cluster for a welding mechanical analysis problem with one million degrees of freedom.

2 Review of The FETI-DP-RBS-LNA Algorithm

2.1 The FETI-DP Algorithm

FETI-DP can be viewed as a combination of direct and iterative methods. Based on the underlining divide and conquer concept, the physical domain is divided into subdomains with non-overlapping interfaces. The related nodes after finite element discretization can be classified into three groups based on their locations, and they are marked as corner nodes, non-corner interface nodes and internal nodes in Figure 1, respectively. More details of FETI-DP can be found in Ref [7, 2, 1].

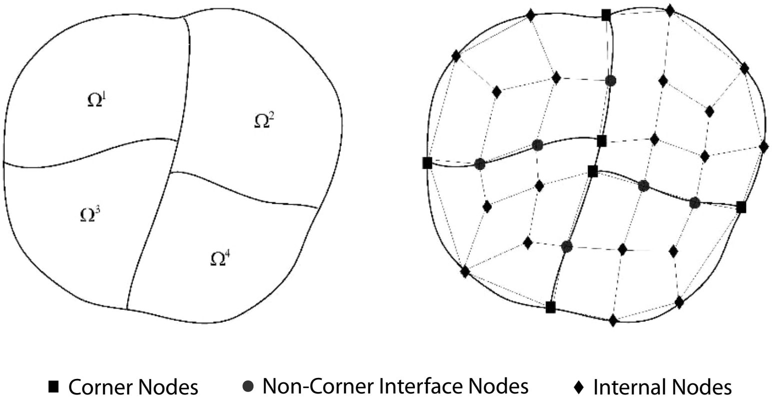


Fig. 1. Subdomains with non-overlapping interfaces, their meshes and nodes classification.

Through the similar concepts of super elements and substructures, the high level interface problem can first be formulated and solved by an iterative Preconditioned Conjugate Gradient (PCG) method. Once the interface solution is available, corner information can be further solved. After that, all the low level subdomains are independent and can be solved by direct sparse solvers in a parallel fashion. These procedures are illustrated in Figure 2.

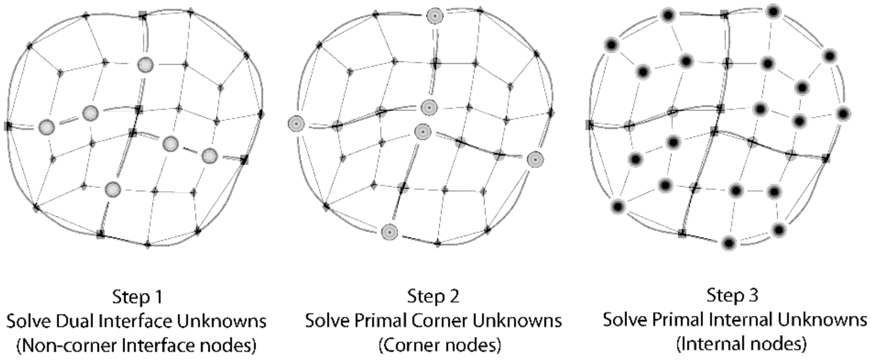

Step 1
Solve Dual Interface Unknowns
(Non-corner Interface nodes)
Step 2
Solve Primal Corner Unknowns
(Corner nodes)
Step 3
Solve Primal Internal Unknowns
(Internal nodes)

Fig. 2. Solution scheme of FETI-DP.

2.2 Reduced Back-Substitution Algorithm

Based on the CPU statistics in Ref [3] and the welding simulation problem in this paper, the PCG iterations for large interface problems are found to be the time consuming part in the FETI family algorithms. Within the PCG costs, a high percentage (around 64.3% for the mechanical analysis of the welding problem in this paper) of the CPU time is actually consumed by the local back-substitutions inside the PCG iterations. Therefore, a reduction of the computations in the local back-substitutions will greatly improve the overall performance of the FETI-DP algorithm.

During each PCG iteration, the most time consuming steps are calculating the following two matrix-vector multiplications listed in Equation (1). Each multiplication has several back-substitutions involved.

$$(F_{I_{rr}} + F_{I_{rc}} {K_{cc}^{*}}^{-1} {F_{I_{rc}}}^{T}) \cdot \lambda \qquad \text{and} \qquad {F_{I_{rr}}^{D}}^{-1} \cdot \lambda \tag{1}$$

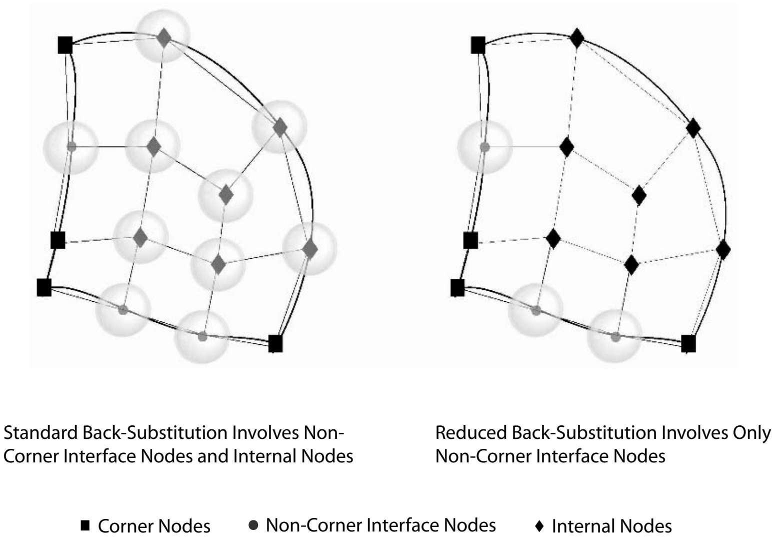
Taking one sub-step from the first multiplication $F_{I_{rr}} \cdot \lambda$ as an example, after substituting the detailed expression of $F_{I_{rr}}$ [2, 1], it yields the following equation:

$$F_{I_{rr}} \cdot \lambda = \sum_{s=1}^{n_s} B_r^s {K_{rr}^s}^{-1} {B_r^s}^T \lambda \tag{2}$$

In the FETI-DP algorithm, ${B_r^s}^T$ is first applied to λ through scatter operations to get ${B_r^s}^T \lambda$, then ${K_{rr}^s}^{-1}({B_r^s}^T \lambda)$ is solved as a whole through the back-substitution on the subdomain level, where ${K_{rr}^s}^{-1}$ is the inverse of subdomain matrix which has already been factorized with its factors stored. Finally, B_r^s is applied to the solution vector ${K_{rr}^s}^{-1}({B_r^s}^T \lambda)$ through gather operations to form $B_r^s({K_{rr}^s}^{-1}({B_r^s}^T \lambda))$ and summed over all the subdomains. The reason this process requires much computational time lies in the relatively large number of equations in each subdomain, as the back-substitution is actually performed for each subdomain internal and non-corner interface degrees of freedom (equations). The left part graph of Figure 3 shows the nodes involved in this standard back-substitution.

${B_r^s}^T$ and B_r^s connect subdomain level information to global domain information through scatter and gather operations. If written in matrix format, their representations are sparse matrices. Based on the analysis in Ref [7], assuming the number of equations corresponding to non-corner interface degrees of freedom is m, and these equations are numbered last. Only the last m components from λ are required as the input for the back-substitutions in Equation (2) since B_r^s zeros the remaining components, and only the last m components from the back-substitution result ${K_{rr}^s}^{-1} {B_r^s}^T \lambda$ are required as the output for the same reason. Thus the back-substitution is actually performed on the last m equations. m is a much smaller number compared to the sum of subdomain internal degrees of freedom and non-corner interface degrees of freedom. Therefore, much time can be saved with the reduced back-substitution (RBS). The nodes involved in this RBS algorithm are shown in the right part of Figure 3. Compared to standard back-substitution, many internal nodes need not be included anymore.

It must be mentioned that the proposed reduced back-substitutions will affect the ordering scheme since it imposes a restriction on the ordering of the related equations. This re-numbering adds to the cost of the numerical factorizations compared to a good ordering, such as the nested dissection scheme. More detailed discussion on this issue is in Ref [7].



Standard Back-Substitution Involves Non-Corner Interface Nodes and Internal Nodes

Reduced Back-Substitution Involves Only Non-Corner Interface Nodes

Fig. 3. Standard back-substitution and reduced back-substitution for subdomain Ω^2 in Figure 1.

2.3 Linear-Nonlinear Analysis

Linear-nonlinear analysis (LNA) is a well-known and efficient strategy to solve problems with localized nonlinearity. It exploits information about which subdomain remains linear during a nonlinear analysis. Therefore, repeated factorizations of linear subdomains can be avoided and computation costs can be saved. More implementation details on LNA can also be found in Ref [7].

3 Serial and Distributed Performance Results

3.1 Software and Hardware

The software and hardware implementation for the serial performance measurement is described in Ref [7]. The standard MPICH libarary has been implemented in the in-house code for distributed computing. The distributed computing simulations are performed on the Penn State LION-XM cluster, which consists 168 computing nodes where each node has 2 Intel Xeon (3.2 GHz) Processors and 4 GB memory.

3.2 16-Subdomain Hollow Beam Model and Simulation Information

A 16-Subdomain hollow beam model is chosen to be the large scale welding problem for performance measurements in this paper. The model and welding information can be found in [7]. The total number of Hex20 element in this model is 65664, and the total number of equations is 1007634. The number of interface equations is 8460 and the number of corner equations is 174.

3.3 Serial Performance Results

CPU Time (s)	Serial Direct	FETI-DP	FETI-DP	FETI-DP	FETI-DP
	Sparse Solver		RBS	LNA	RBS & LNA
IO & SF	42.11	81.45	103.58	80.99	103.58
NF	47262.12	26525.69	40601.13	1849.01	2582.91
BS	1273.22	—	—	—	—
PCG (LBS)	—	58759.03	8879.37	58335.77	8900.07
	—	(54880.92)	(5083.20)	(54497.29)	(5110.52)
TOTAL	48577.45	85366.17	49584.08	60265.77	11586.56

Table 1. Mechanical analysis serial performance, first 50 time increments.

The serial CPU costs of the IBM Watson direct sparse solver, FETI-DP, FETI-DP-RBS, FETI-DP-LNA and FETI-DP-RBS-LNA in the mechanical analysis are measured and compared in Table 1, where IO stands for solver initialization and ordering, SF is symbolic factorization, NF is numeric factorization, BS is back-substitution, PCG is Preconditioned Conjugate Gradient iterations, LBS is local back-substitution in PCG, LNA is Linear-Nonlinear Analysis, and RBS is Reduced Back-Substitution. Detailed analysis of serial CPU time is given in Ref [7].

3.4 Distributed Performance Results

Wallclock Time (s)	UNISYS, 1 Processor	LION-XM, 16 Nodes	SpeedUp
	(16 Subdomains)	(1 Subdomain Per Node)	
NF	288.53	20.07-30.72	9.4
PCG	54.64	6.51	8.4

Table 2. Mechanical analysis distributed performance and speedUp, first iteration.

Distributed computing performance results are measured for the numeric factorization and PCG iterations during the first iteration, as shown in Table 2. 16 computing nodes of the LION-XM cluster are used in the simulation and each computing node contains one subdomain.

The subdomain level computations, such as, forming the subdomain stiffness matrices, local numeric factorizations, local back-substitutions and residual computations are all performed on each individual computing node in a parallel fashion. MPI is mainly implemented to gather and broadcast the intermediate results during the procedure of solving the interface problem by the PCG method.

The speedup gained during numeric factorization is 9.4. Perfect scalability is not achieved due to the fact that the numbers of interface DOFs of the subdomain differ. Therefore, the computational cost of each subdomain is also not the same.

Some subdomains have large interfaces and require more time to be factorized. The MPI wallclock time is measured based on the longest factorization time.

The speedup gained during PCG iterations is 8.4. In the total 6.51s wallclock time, around 2.6s is spent on inter-processor communications to gather and broadcast the intermediate solution results during the interface solves. Therefore, from the computational point of view, the numerical scalability is very good and higher speedup can be expected when high-speed network interconnect is implemented.

4 Conclusion and Future Work

In this paper, a fast implementation of the FETI-DP algorithm: the FETI-DP-RBS-LNA algorithm is proposed for solving large scale problems with localized nonlinearity. Serial performance of the FETI-DP-RBS-LNA algorithm is tested to give a correct estimation of floating point performance. Distributed performance is also evaluated for the first iteration to measure the speedup gained from distributed computing. Future work will continue the study of the distributed performance of the FETI-DP-RBS-LNA algorithm when linear nonlinear analysis is applied.

5 Acknowledgments

The authors would like to acknowledge the funding from Office of Naval Research, award number N00014-00-1-0645 and ONR program manager Julie Christodoulou. The authors would also like to acknowledge Unisys, Inc. and Pennsylvania State University for providing the Unisys ES7000 systems and Jeff Nucciarone from Academic Services and Emerging Technologies, Pennsylvania State University, for his help and suggestions during the execution of the simulation on the Unisys system.

References

1. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.
2. C. Farhat, M. Lesoinne, and K. Pierson, *A scalable dual-primal domain decomposition method*, Numer. Lin. Alg. Appl., 7 (2000), pp. 687–714.
3. C. Farhat, K. Pierson, and M. Lesoinne, *The second generation of FETI methods and their application to the parallel solution of large-scale linear and geometrically nonlinear structural analysis problems*, Comput. Methods Appl. Mech. Engrg., 184 (2000), pp. 333–374.
4. R. J. Lipton, D. J. Rose, and R. E. Tarjan, *Generalized nested dissection*, SIAM J. Numer. Anal., 16 (1979), pp. 346–358.
5. N. Prasad and T. K. Sankaranrayanan, *Estimation of residual stresses in weldments using adaptive grids*, Computers and Structures, 60 (1996), pp. 1037–1045.

6. H. Runnemalm and S.-J. Hyun, *Three-dimensional welding analysis using an adaptive mesh scheme*, Comput. Methods Appl. Mech. Engrg., 189 (2000), pp. 515–523.
7. J. Sun, P. Michaleris, A. Gupta, and P. Raghavan, *A fast implementation of the FETI-DP method: FETI-DP-RBS-LNA and applications on large scale problems with localized non-linearities*, Internat. J. Numer. Methods Engrg., 63 (2005), pp. 833–858.

Three-level BDDC

Xuemin Tu

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

Summary. BDDC (Balancing Domain Decomposition by Constraints) methods, so far developed for two levels [3, 7, 8], are similar to the balancing Neumann-Neumann algorithms. However, the BDDC coarse problem is given in terms of a set of primal constraints and the matrix of the coarse problem is generated and factored by direct solvers at the beginning of the computation. The coarse component of the preconditioner can ultimately become a bottleneck if the number of subdomains is very large. In this paper, two three-level BDDC methods are introduced for solving the coarse problem approximately in two and three dimensions, while still maintaining a good convergence rate. Estimates of the condition numbers are provided for the two three-level BDDC methods and numerical experiments are also discussed.

1 The two-level BDDC method

We consider a second order scalar elliptic problem in a two or three dimensional region Ω: find $u \in H_0^1(\Omega)$, such that

$$\int_{\Omega} \rho \nabla u \cdot \nabla v = \int_{\Omega} f v \quad \forall v \in H_0^1(\Omega), \tag{1}$$

where $\rho(x) > 0$ for all $x \in \Omega$. We introduce a mesh, subdomains Ω_i, and an interface Γ on the domain Ω with notation as in [10, Section 4.2].

Let $\mathbf{W}^{(i)}$ be the standard conforming first-order finite elements on Ω_i. We assume that these functions vanish on $\partial\Omega$. Each $\mathbf{W}^{(i)}$ can be decomposed into a subdomain interior part $\mathbf{W}_I^{(i)}$ and a subdomain interface part $\mathbf{W}_\Gamma^{(i)}$. The subdomain interface part $\mathbf{W}_\Gamma^{(i)}$ can be further decomposed into a primal subspace $\mathbf{W}_\Pi^{(i)}$ and a dual subspace $\mathbf{W}_\Delta^{(i)}$, i.e., $\mathbf{W}^{(i)} = \mathbf{W}_I^{(i)} \bigoplus \mathbf{W}_\Gamma^{(i)} = \mathbf{W}_I^{(i)} \bigoplus \mathbf{W}_\Pi^{(i)} \bigoplus \mathbf{W}_\Delta^{(i)}$.

We denote the associated product spaces by $\mathbf{W} := \prod_{i=1}^{N} \mathbf{W}^{(i)}$, $\mathbf{W}_\Gamma := \prod_{i=1}^{N} \mathbf{W}_\Gamma^{(i)}$, $\mathbf{W}_\Delta := \prod_{i=1}^{N} \mathbf{W}_\Delta^{(i)}$, $\mathbf{W}_\Pi := \prod_{i=1}^{N} \mathbf{W}_\Pi^{(i)}$, and $\mathbf{W}_I := \prod_{i=1}^{N} \mathbf{W}_I^{(i)}$. Correspondingly, we have $\mathbf{W} = \mathbf{W}_I \bigoplus \mathbf{W}_\Gamma$ and $\mathbf{W}_\Gamma = \mathbf{W}_\Pi \bigoplus \mathbf{W}_\Delta$.

We will consider elements of the product space $\mathbf{W}$ which are discontinuous across the interface. However, the finite element approximation of the elliptic problem is continuous across Γ and we denote the corresponding subspace of $\mathbf{W}$ by $\widehat{\mathbf{W}}$.

We further introduce an interface subspace $\widetilde{\mathbf{W}}_\Gamma \subset \mathbf{W}_\Gamma$, for which certain primal constraints are enforced. Here, the continuous primal subspace, denoted by $\widehat{\mathbf{W}}_\Pi$, is spanned by only the continuous finite element basis functions of the vertex nodes in two dimensions and by the continuous edge average variables over each subdomain edge in three dimensions. For three dimensions, we change the variables to make the edge average degrees of freedom explicit, see [5, Sec 6.2] and [6, Sec 2.3]. From now on, we assume that all the matrices are written in terms of the new variables in three dimensions. The space $\widetilde{\mathbf{W}}_\Gamma$ can be decomposed into $\widetilde{\mathbf{W}}_\Gamma = \widehat{\mathbf{W}}_\Pi \bigoplus \mathbf{W}_\Delta$.

We define an operator $\widetilde{S}_\Gamma$ by: given $\mathbf{u}_\Gamma = \mathbf{u}_\Pi \oplus \mathbf{u}_\Delta \in \widehat{\mathbf{W}}_\Pi \bigoplus \mathbf{W}_\Delta = \widetilde{\mathbf{W}}_\Gamma$, we find $\widetilde{S}_\Gamma \mathbf{u}_\Gamma$ by eliminating the interior variables of the partially assembled system with continuous primal components.

The operator $R_{\Gamma\Delta} : \widetilde{\mathbf{W}}_\Gamma \rightarrow \mathbf{W}_\Delta$, restricts the functions in the space $\widetilde{\mathbf{W}}_\Gamma$ to $\mathbf{W}_\Delta$, and is a block diagonal matrix $diag(R_{\Gamma\Delta}^{(1)}, \cdots, R_{\Gamma\Delta}^{(N)})$, where each $R_{\Gamma\Delta}^{(i)}$ represents the restriction from $\mathbf{W}_\Gamma^{(i)}$ to $\mathbf{W}_\Delta^{(i)}$. Furthermore, $R_\Delta^{(i)} : \mathbf{W}_\Delta \rightarrow \mathbf{W}_\Delta^{(i)}$, is the restriction matrix which extracts the subdomain part, in the space $\mathbf{W}_\Delta^{(i)}$, of the functions in the space $\mathbf{W}_\Delta$, and $R_{\Gamma\Pi}$ restricts the functions in the space $\widetilde{\mathbf{W}}_\Gamma$ to $\widehat{\mathbf{W}}_\Pi$. $R_\Pi^{(i)}$ is the restriction operator from the space $\widehat{\mathbf{W}}_\Pi$ to $\mathbf{W}_\Pi^{(i)}$.

$R_\Gamma = (R_\Gamma^{(1)}, \cdots, R_\Gamma^{(N)})^T$ and $R_{D,\Gamma} = (R_{D,\Gamma}^{(1)}, \cdots, R_{D,\Gamma}^{(N)})^T$ are the restriction and scaled restriction operators from the space $\widehat{\mathbf{W}}_\Gamma$ onto $\widetilde{\mathbf{W}}_\Gamma$, respectively. Here $R_\Gamma^{(i)}$ maps a vector in $\widehat{\mathbf{W}}_\Gamma$ to a vector in $\mathbf{W}_\Gamma^{(i)}$. Each column of $R_\Gamma^{(i)}$ with a nonzero entry corresponds to an interface node, $x \in \partial\Omega_{i,h} \cap \Gamma_h$, shared by the subdomain Ω_i and its neighboring subdomains. Multiplying each such column of $R_\Gamma^{(i)}$ with $\delta_i^\dagger(x)$ gives us $R_{D,\Gamma}^{(i)}$, where $\delta_i^\dagger(x)$ is related to the number of subdomains to which a node belongs, defined in [10, Formula (6.2)].

The reduced interface problem can be written as: find $\mathbf{u}_\Gamma \in \widehat{\mathbf{W}}_\Gamma$ such that $R_\Gamma^T \widetilde{S}_\Gamma R_\Gamma \mathbf{u}_\Gamma = \mathbf{g}_\Gamma$, where $\mathbf{g}_\Gamma$ is the load vector reduced to Γ.

The two-level BDDC preconditioned equation is of the form

$$M^{-1} R_\Gamma^T \widetilde{S}_\Gamma R_\Gamma \mathbf{u}_\Gamma = M^{-1} \mathbf{g}_\Gamma,$$

where the preconditioner $M^{-1} = R_{D,\Gamma}^T \widetilde{S}_\Gamma^{-1} R_{D,\Gamma}$ has the following form (see [6, Formula (33)]) with the columns of Φ, being minimal energy extensions of the primal variables:

$$R_\Gamma^T D_\Gamma \left\{ \sum_{i=1}^{N} R_{\Gamma\Delta}^{(i)^T} \begin{pmatrix} \mathbf{0} & R_\Delta^{(i)^T} \end{pmatrix} \begin{pmatrix} A_{II}^{(i)} & A_{\Delta I}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ R_\Delta^{(i)} \end{pmatrix} R_{\Gamma\Delta}^{(i)} + \Phi S_\Pi^{-1} \Phi^T \right\} D_\Gamma R_\Gamma.$$

Denote by E_D and P_D, the average and jump operators (see [10, Formula (6.4) and (6.38)]), on the space $\widetilde{\mathbf{W}}_\Gamma$, respectively. Central to obtaining the condition number estimate for the preconditioned two-level BDDC operator is a bound for the E_D operator (see [8, Theorem 25]). Since $E_D + P_D = I$ (see [10, Lemma 6.10]), we only need to find a bound for the P_D operator.

A bound for the P_D operator in two dimensions is given in [9], provided that the coefficient $\rho(x)$ of (1) varies moderately in each subdomain. In our theory, we also assume that each subdomain is a union of shape-regular coarse triangles and that the number of such triangles forming an individual subdomain is uniformly bounded. Moreover, we assume that the triangulation of each subdomain is quasi uniform. For the three dimensional case, we need one more requirement for $\rho(x)$ since we only use the edge average constraints, namely that for all pairs of subdomain Ω_i and Ω_j, which have a vertex but not an edge in common, there exists an acceptable edge path (see [10, Definition 6.26]) between the two subdomains. With this assumption, we have a good estimate for the P_D operator (see [10, Lemma 6.36]): under our assumptions, we have in two and three dimensions:

$$\mathbf{u}_\Gamma^T M \mathbf{u}_\Gamma \le \mathbf{u}_\Gamma^T R_\Gamma^T \widetilde{S}_\Gamma R_\Gamma \mathbf{u}_\Gamma \le C\left(1+\log(H/h)\right)^2 \mathbf{u}_\Gamma^T M \mathbf{u}_\Gamma,\ \forall \mathbf{u}_\Gamma \in \widehat{\mathbf{W}}_\Gamma.$$

2 A three-level BDDC method

In the three-level case, we will not factor the coarse problem matrix S_Π by a direct solver. Instead, we will solve the coarse problem approximately by using an idea similar to the two-level preconditioner.

We decompose Ω into N subregions Ω^j with diameters $\hat{H}^j$, $j = 1, \cdots, N$. Each subregion Ω^j has N_j subdomains Ω_i^j with diameter H_i^j. Let $\hat{H} = \max_j \hat{H}^j$ and $H = \max_{i,j} H_i^j$, for $j = 1,\ \cdots,\ N$, and $i = 1, \cdots, N_j$. We introduce the subregional Schur complements:

$$S_\Pi^{(j)} = \sum_{i=1}^{N_j} R_\Pi^{(i)^T} \left\{ A_{\Pi\Pi}^{(i)} - \begin{pmatrix} A_{\Pi I}^{(i)} & A_{\Pi\Delta}^{(i)} \end{pmatrix} \begin{pmatrix} A_{II}^{(i)} & A_{I\Delta}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} \end{pmatrix}^{-1} \begin{pmatrix} A_{\Pi I}^{(i)^T} \\ A_{\Pi\Delta}^{(i)^T} \end{pmatrix} \right\} R_\Pi^{(i)},$$

and note that the coarse problem matrix S_Π can be assembled from the $S_\Pi^{(j)}$.

Let $\widehat{\Gamma}$ be the interface between the subregions; $\widehat{\Gamma} \subset \Gamma$. We denote the set of interior primal variables in each subregion by $\widehat{I}_H$, and the set of interface primal variables on the boundary of the subregions by $\widehat{\Gamma}_H$.

We denote the vector space corresponding to the primal variables of the subregion Ω^i by $\mathbf{W}_c^{(i)}$. We define the subregion spaces $\widehat{\mathbf{W}}_{c,\widehat{\Gamma}}$, $\widetilde{\mathbf{W}}_{c,\widehat{\Gamma}}$, $\widehat{R}_{\widehat{\Gamma}}^{(i)}$, $\widehat{R}_{\widehat{D},\widehat{\Gamma}}^{(i)}$, $\widehat{R}_{\widehat{\Gamma}}$, and $\widehat{R}_{\widehat{D},\widehat{\Gamma}}$, as for the subdomains but on the subregion level.

We introduce an operator $\widetilde{T}$:

$$\widehat{R}_{\widehat{\Gamma}}^T \widetilde{T} \widehat{R}_{\widehat{\Gamma}} = \sum_{i=1}^{N} \widehat{R}_{\widehat{\Gamma}}^{(i)^T} \big(S_{\Pi_{\widehat{\Gamma}\widehat{\Gamma}}}^{(i)} - S_{\Pi_{\widehat{\Gamma}\widehat{I}}}^{(i)} S_{\Pi_{\widehat{I}\widehat{I}}}^{(i)^{-1}} S_{\Pi_{\widehat{\Gamma}\widehat{I}}}^{(i)^T}\big) \widehat{R}_{\widehat{\Gamma}}^{(i)}. \tag{2}$$

and define our three-level preconditioner $\widetilde{M}^{-1}$ by

$$R_\Gamma^T D_\Gamma \left\{ \sum_{i=1}^{N} R_{\Gamma\Delta}^{(i)^T} \begin{pmatrix} \mathbf{0} & R_\Delta^{(i)^T} \end{pmatrix} \begin{pmatrix} A_{II}^{(i)} & A_{\Delta I}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ R_\Delta^{(i)} \end{pmatrix} R_{\Gamma\Delta}^{(i)} + \Phi \widetilde{S}_\Pi^{-1} \Phi^T \right\} D_\Gamma R_\Gamma.$$

Here $\widetilde{S}_{\Pi}^{-1}$ is an approximation of S_{Π}^{-1} and is defined as follows: given any right hand side $\boldsymbol{\Psi}$, let $\mathbf{y} = S_{\Pi}^{-1}\boldsymbol{\Psi}$ and $\widetilde{\mathbf{y}} = \widetilde{S}_{\Pi}^{-1}\boldsymbol{\Psi}$. We first reduce the original coarse problem S_{Π} to the subregion interface problem. We do not solve the interface problem exactly but replace $\mathbf{y}_{\widehat{\Gamma}}$, the interface part of $\mathbf{y}$, by

$$\widetilde{\mathbf{y}}_{\widehat{\Gamma}} = \widehat{R}_{\widehat{D},\widehat{\Gamma}}^{T}\widetilde{T}^{-1}\widehat{R}_{\widehat{D},\widehat{\Gamma}}\mathbf{h}_{\widehat{\Gamma}},$$

where $\mathbf{h}_{\widehat{\Gamma}}$ is the load vector reduced to $\widehat{\Gamma}$.

3 Condition number estimate for the new preconditioner

We first collect a number of results which are needed in our theory. We discuss, in detail, only the two-dimensional case.

Lemma 1. *(Two dimensions) Let V_i^H be the standard continuous piecewise linear finite element function space for a subregion Ω^i with a quasi-uniform coarse mesh with mesh size H. And let $V_{i,j}^h$, $j = 1, \cdots, N_i$ be the space for a subdomain Ω_j^i with a quasi-uniform fine mesh with mesh size h. Moreover, each subdomain is a union of coarse triangles with vertices on the boundary of the subdomain. Given $u \in V_i^H$, let $\hat{u} \in V_i^H$ interpolate u at each coarse node and be the discrete $V_{i,j}^h$-harmonic extension in each subdomain Ω_j^i constrained only at the vertices of Ω_j^i, $j = 1, \cdots, N_i$. Then, there exist two positive constants C_1 and C_2, which are independent of $\hat{H}$, H, and h, such that*

$$C_1(1+\log\frac{H}{h})\left(\sum_{j=1}^{N_i}|\hat{u}|_{H^1(\Omega_j^i)}^2\right) \leq |u|_{H^1(\Omega^i)}^2 \leq C_2(1+\log\frac{H}{h})\left(\sum_{j=1}^{N_i}|\hat{u}|_{H^1(\Omega_j^i)}^2\right).$$

We use [2, Lemma 4.2] to prove Lemma 1. Since we assume that the fine triangulation of each subdomain is quasi uniform, we can then obtain uniform constants C_1 and C_2 in Lemma 1 which work for all the subregions. In addition, a similar result for three dimensions can be obtained with [1, Lemma 4.2].

We define the subregion interface averages operator $\widehat{E}_{\widehat{D}} : \widetilde{\mathbf{W}}_{c,\widehat{\Gamma}} \rightarrow \widehat{\mathbf{W}}_{c,\widehat{\Gamma}}$, by $\widehat{E}_{\widehat{D}} = \widehat{R}_{\widehat{\Gamma}}\widehat{R}_{\widehat{D},\widehat{\Gamma}}^{T}$, which computes averages across the subregion interface $\widehat{\Gamma}$ and then distributes the averages to the boundary points of the subregions.

The interface average operator $\widehat{E}_{\widehat{D}}$ has the following properties:

Lemma 2.

$$\widehat{E}_{\widehat{D}}\mathbf{w}_{\widehat{\Gamma}} = \widehat{R}_{\widehat{\Gamma}}^{T}\widehat{R}_{\widehat{D},\widehat{\Gamma}}\mathbf{w}_{\widehat{\Gamma}} = \mathbf{w}_{\widehat{\Gamma}}, \text{ for any } \mathbf{w}_{\widehat{\Gamma}} \in \widehat{\mathbf{W}}_{c,\widehat{\Gamma}}.$$

Lemma 3.

$$|\widehat{E}_{\widehat{D}}\mathbf{w}_{\widehat{\Gamma}}|_{\widetilde{T}}^2 \leq C\left(1+\log\frac{\hat{H}}{H}\right)^2|\mathbf{w}_{\widehat{\Gamma}}|_{\widetilde{T}}^2,$$

for any $\mathbf{w}_{\widehat{\Gamma}} \in \widetilde{\mathbf{W}}_{c,\widehat{\Gamma}}$, where C is a positive constant independent of $\hat{H}$, H, and h. Here $\widetilde{T}$ is defined in (2).

See [11] for a proof in two dimensions and [12] for a proof in three dimensions. As we mentioned before, we use constraints on the averages over edges in three dimensions. These constraints lead to a considerably more complicated coarse problem which needs new technical tools in the proof of Lemma 3. This is the main difference in the analysis between two and three dimensions.

Lemma 4. *Given any* $\mathbf{u_\Gamma} \in \widehat{\mathbf{W}}_\Gamma$*, let* $\mathbf{\Psi} = \Phi^T D_\Gamma R_\Gamma \mathbf{u}_\Gamma$*. We have,*

$$\mathbf{\Psi}^T S_\Pi^{-1} \mathbf{\Psi} \le \mathbf{\Psi}^T \widetilde{S}_\Pi^{-1} \mathbf{\Psi}^T \le C \left(1 + \log \frac{\hat{H}}{H}\right)^2 \mathbf{\Psi}^T S_\Pi^{-1} \mathbf{\Psi}.$$

Lemma 5. *Given any* $\mathbf{u_\Gamma} \in \widehat{\mathbf{W}}_\Gamma$*,*

$$\mathbf{u}_\Gamma^T M^{-1} \mathbf{u}_\Gamma \le \mathbf{u}_\Gamma^T \widetilde{M}^{-1} \mathbf{u}_\Gamma \le C \left(1 + \log \frac{\hat{H}}{H}\right)^2 \mathbf{u}_\Gamma^T M^{-1} \mathbf{u}_\Gamma.$$

We finally have

Theorem 1. *The condition number for the system with the three-level preconditioner* $\widetilde{M}^{-1}$*is bounded by* $C(1 + \log \frac{\hat{H}}{H})^2 (1 + \log \frac{H}{h})^2$.

4 Using Chebyshev iterations

Another approach to the three-level BDDC methods is to use a preconditioned Chebyshev method with a fixed number of iterations to solve the reduced coarse level subregion interface problem. The preconditioner is $\widehat{R}^T_{\widehat{D},\widehat{\Gamma}} \widetilde{T}^{-1} \widehat{R}_{\widehat{D},\widehat{\Gamma}}$. Denoting the corresponding new coarse problem matrix by $\widehat{S}_\Pi$, the new preconditioner $\widehat{M}^{-1}$ is defined by:

$$R_\Gamma^T D_\Gamma \left\{ \sum_{i=1}^N R_{\Gamma\Delta}^T \begin{pmatrix} \mathbf{0} & R_\Delta^{(i)^T} \end{pmatrix} \begin{pmatrix} A_{II}^{(i)} & A_{\Delta I}^{(i)} \\ A_{\Delta I}^{(i)} & A_{\Delta\Delta}^{(i)} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{0} \\ R_\Delta^{(i)} \end{pmatrix} R_{\Gamma\Delta} + \Phi \widehat{S}_\Pi^{-1} \Phi^T \right\} D_\Gamma R_\Gamma.$$

Denoting by λ_j the eigenvalues of $\mathbf{N}01\mathcal{K}$, we need two input parameters l and u for the Chebyshev iterations, where l and u are estimates for the minimum and maximum values of λ_j, respectively, see [4]. From our analysis above, we know that $\min_j \lambda_j = 1$ and $\max_j \lambda_j \le C(1 + \log \frac{\hat{H}}{H})^2 (1 + \log \frac{H}{h})^2$. We can use the conjugate gradient method to obtain an estimate for the largest eigenvalue at the beginning of the computation to choose a proper u.

Let $\alpha = \frac{2}{l+u}$, $\mu = \frac{u+l}{u-l}$ and $Q = I - \alpha \mathbf{N}01\mathcal{K}$. Denote by σ_j the eigenvalues of Q.

If we choose u such that $\lambda_j < l + u$, we find that $1 - \frac{\cosh\left(k \cosh^{-1}(\mu\sigma_j)\right)}{\cosh\left(k \cosh^{-1}(\mu)\right)} > 0$, and we then have the following lemmas.

Lemma 6. *Given any $\mathbf{u_\Gamma} \in \widehat{\mathbf{W}}_\Gamma$, let $\mathbf{\Psi} = \Phi^T D_\Gamma R_\Gamma \mathbf{u}_\Gamma$ and select u such that $\lambda_j < u + l$. There then exist two functions $C_1(k)$ and $C_2(k)$ that*

$$C_1(k)\mathbf{\Psi}^T S_\Pi^{-1}\mathbf{\Psi} \le \mathbf{\Psi}^T \widehat{S}_\Pi^{-1}\mathbf{\Psi}^T \le C_2(k)\mathbf{\Psi}^T S_\Pi^{-1}\mathbf{\Psi},$$

where $C_1(k)$ and $C_2(k)$ are the minimum and maximum values, over all j, of $\left(1 - \dfrac{\cosh(k\cosh^{-1}(\mu\sigma_j))}{\cosh(k\cosh^{-1}(\mu))}\right)$.

Lemma 7. *Given any $\mathbf{u_\Gamma} \in \widehat{\mathbf{W}}_\Gamma$,*

$$C_1(k)\mathbf{u}_\Gamma^T M^{-1}\mathbf{u}_\Gamma \le \mathbf{u}_\Gamma^T \widehat{M}^{-1}\mathbf{u}_\Gamma \le C_2(k)\mathbf{u}_\Gamma^T M^{-1}\mathbf{u}_\Gamma,$$

where $C_1(k)$ and $C_2(k)$ are defined in Lemma 6.

We finally have

Theorem 2. *The condition number of the preconditioned operator using the three-level preconditioner $\widehat{M}^{-1}$is bounded by $C\dfrac{C_2(k)}{C_1(k)}(1+\log\dfrac{H}{h})^2$, where $C_1(k)$ and $C_2(k)$ are defined in Lemma 6 and $\dfrac{C_2(k)}{C_1(k)} \to 1$ as $k \to \infty$.*

5 Numerical experiments

Table 1. Eigenvalue bounds and iteration counts with the preconditioner $\widetilde{M}^{-1}$.

$\frac{\widehat{H}}{H} = 4,\ \frac{H}{h} = 4$			$\frac{H}{h} = 4$, 4 × 4 subregions			$\frac{\widehat{H}}{H} = 4$, 4 × 4 subregions		
Subreg.	Iter.	Cond. #	$\frac{\widehat{H}}{H}$	Iter.	Cond. #	$\frac{H}{h}$	Iter.	Cond. #
4 × 4	11	1.8096	4	11	1.8096	4	11	1.8096
8 × 8	11	1.8145	8	12	1.8536	8	14	2.4934
12 × 12	12	1.8159	12	12	1.8742	12	16	2.9758
16 × 16	12	1.8162	16	12	1.8912	16	17	3.3473
20 × 20	12	1.8164	20	12	1.9062	20	18	3.6546

We have applied our two three-level BDDC algorithms to the model problem (1). Here we only give results for two dimensions. We decompose the unit square into $\widehat{N} \times \widehat{N}$ subregions and each subregion into $N \times N$ subdomains with the side-length $\widehat{H} = 1/\widehat{N}$ and $H = \widehat{H}/N$, respectively. Equation (1) is discretized, in each subdomain, by conforming piecewise linear elements with a finite element diameter h. The preconditioned conjugate gradient iteration is stopped when the norm of the residual has been reduced by a factor of 10^{-8}.

We have carried out two different sets of experiments. All the experimental results are fully consistent with our theory. In the first set of experiments, we use

Table 2. Eigenvalue bounds and iteration counts with the preconditioner $\widehat{M}^{-1}$, 4×4 subregions, $\frac{\widehat{H}}{H} = 16$ and $\frac{H}{h} = 4$.

			$u = 3.2$						$u = 6$		
k	Iter.	$C_1(k)$	λ_{min}	λ_{max}	Cond. #	k	Iter.	$C_1(k)$	λ_{min}	λ_{max}	Cond. #
1	20	0.4762	0.4829	2.7110	5.6141	1	24	0.2857	0.2899	1.8287	6.3086
2	13	0.8410	0.8540	1.8820	2.2038	2	16	0.6575	0.6670	2.3435	3.5134
3	11	0.9548	0.9981	1.9061	1.9098	3	12	0.8524	0.9286	1.9628	3.1136
4	11	0.9872	1.0019	1.8663	1.8629	4	12	0.9377	0.9795	1.9850	2.0266
5	11	0.9964	1.0006	1.8551	1.8541	5	12	0.9738	0.9983	1.9403	1.9437

the first preconditioner $\widetilde{M}^{-1}$ and take the coefficient $\rho = 1$ in half of the subregions and $\rho = 101$ in the neighboring subregions in a checkerboard pattern. Table 1 gives the iteration counts and condition number estimates with a change of the number of subregions, the number of subdomains, and the size of the subdomain problems.

In the second set of experiments, we use the second preconditioner $\widehat{M}^{-1}$ and take the coefficient $\rho \equiv 1$. We use the PCG to estimate the largest eigenvalue of $\mathbf{N}01\mathcal{K}$ which is approximately 3.2867. For 64×64 subdomains and $\frac{H}{h} = 4$, we have a condition number estimate of 1.8380 for the two-level BDDC. We then select different values of u, the upper bound estimate of the eigenvalues for the preconditioned system, and k to see how the condition number changes. We also evaluate $C_1(k)$ for $k = 1, 2, 3, 4, 5$. From Table 2, we find that the smallest eigenvalue is bounded from below by $C_1(k)$ and that the condition number estimate approaches 1.8380, the value in the two-level case, as k increases. From these results, we see that if we can obtain precise estimate for the largest eigenvalue of $\mathbf{N}01\mathcal{K}$, we need fewer Chebyshev iterations to obtain a condition number, similar to that of the two-level case. However, the iteration count is not very sensitive to the choice of u.

Acknowledgments The author is grateful to Professor Olof Widlund for suggesting this problem and all the help and time he has devoted to my work. The author also thanks Professor Jing Li for personal communications and useful notes.

References

1. S. C. Brenner and Q. He, *Lower bounds for three-dimensional nonoverlapping domain decomposition algorithms*, Numerische Mathematik, (2003).
2. S. C. Brenner and L.-Y. Sung, *Lower bounds for nonoverlapping domain decomposition preconditioners in two dimensions*, Math. Comp., 69 (2000), pp. 1319–1339.
3. C. R. Dohrmann, *A preconditioner for substructuring based on constrained energy minimization*, SIAM J. Sci. Comput., 25 (2003), pp. 246–258.

4. G. H. Golub and M. L. Overton, *The convergence of inexact Chebyshev and Richardson iterative methods for solving linear systems*, Numer. Math., 53 (1988), pp. 571–593.
5. A. Klawonn and O. Widlund, *Dual-Primal FETI methods for linear elasticity*, Tech. Rep. 855, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, September 2004.
6. J. Li and O. B. Widlund, *FETI-DP, BDDC, and block Cholesky methods*, Tech. Rep. 857, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 2004.
7. J. Mandel and C. R. Dohrmann, *Convergence of a balancing domain decomposition by constraints and energy minimization*, Numer. Linear Algebra Appl., 10 (2003), pp. 639–659.
8. J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Appl. Numer. Math., 54 (2005), pp. 167–193.
9. J. Mandel and R. Tezaur, *On the convergence of a dual-primal substructuring method*, Numer. Math., 88 (2001), pp. 543–558.
10. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Springer Series in Computational Mathematics, Springer, 2005.
11. X. Tu, *Three-level BDDC in two dimensions*, Tech. Rep. 856, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 2004.
12. ———, *Three-level BDDC in three dimensions*, Tech. Rep. 862, Department of Computer Science, Courant Institute of Mathematical Sciences, New York University, New York, 2005.

MINISYMPOSIUM 8: Analysis, Development and Implementation of Mortar Elements for 3D Problems in Mechanics

Organizer: Patrick Le Tallec[1]

Ecole Polytechnique patrick.letallec@polytechnique.fr

Mortar methods have been introduced as a weak coupling strategy between subdomains with different scales, with nonconforming meshes, or between subproblems solved with different approximation methods. Despite the optimal error convergence obtained with the original elements, there are major numerical difficulties in applying this method to general 3D problems as encountered in real industrial applications. Such applications require some development of more flexible elements when dealing with general interfaces, a more refined analysis when dealing with large number of subdomains, extensions to handle nonlinear problems such as contact, or efficient numerical techniques to handle general nonlinear situations. This minisymposium addresses such issues.

Two-scale Dirichlet-Neumann Preconditioners for Boundary Refinements

Patrice Hauret[1] and Patrick Le Tallec[2]

[1] Manufacture Française des Pneumatiques Michelin, Centre de Technologies de Ladoux, 63040 Clermont-Ferrand Cedex 09, France.
[2] Laboratoire de Mécanique des Solides, CNRS UMR 7649, Département de Mécanique, Ecole Polytechnique, 91128 Palaiseau Cedex, France.

Summary. The present work introduces simple Dirichlet-Neumann preconditioners for the solution of elasticity problems in presence of numerous small disjoint geometric refinements on the boundary of the domain, situation which typically occurs in the tire industry. Moreover, the condition number of the preconditioned system is proved to be independent of the number and the size of the small details on the boundary. Finally, as an enhancement, a second proposed preconditioner makes use of a coarse space counterbalancing the effect of essential boundary conditions on the small details, and a simple numerical academic test illustrates the increased efficiency. Further details on the motivation as well as complete proofs can be found in [4, 5].

1 Introduction

Let $\Omega \subset \mathbb{R}^d$ be the reference configuration of a body, partitioned into a coarse region Ω_0 where the properties of the material are rather smooth and where a coarse approximation should be sufficient, and into small disjoint boundary regions denoted by $(\Omega_k)_{1\leq k\leq K}$ where a fine discretization is required (e.g. with geometric refinements, fine behavior of the material). Such a situation typically occurs for tires, with the internal structure and the surface sculptures respectively playing the role of the coarse and fine zones. Let us denote by Γ_D a part of the boundary of Ω where displacements are prescribed and by $\Gamma_N = \partial\Omega \setminus \Gamma_D$ its complementary part. Denoting by $H^1_*(\Omega) := \{v \in H^1(\Omega)^d, v|_{\Gamma_D\cap\partial\Omega} = 0\}$ the space of admissible displacements, our model elastostatic problem consists in finding $u \in H^1_*(\Omega)$ such that:

$$a(u,v) := \int_\Omega \mathbf{E}_{ijkl}\varepsilon(u)_{kl}\varepsilon(v)_{ij} = \int_\Omega f\cdot v + \int_{\Gamma_N} g\cdot v =: l(v), \quad \forall v \in H^1_*(\Omega).$$

Here $\mathbf{E}$ denotes the fourth order elasticity tensor, $f \in L^2(\Omega)^d$ and $g \in L^2(\Gamma_N)^d$ the loading forces, and $\varepsilon(v) = \dfrac{1}{2}(\nabla v + (\nabla v)^t)$ is the linearized strain tensor. Considering that the solution must be computed with a multi-scale approach in order to respect the characteristics of the problem, the strategy proposed in this paper consists in using:

(a) mortar formulations [2, 13] on the interfaces $\Gamma_{0k} = \partial\Omega_0 \cap \partial\Omega_k$ enabling the use of independent approximations in the coarse and fine regions respectively,
(b) efficient Dirichlet-Neumann preconditioners [9], which we adapt so that the computational cost of the full algebraic problem remains independent (or is at most weakly dependent) of the number and the size of the fine subdomains $(\Omega_k)_{1\le k\le K}$.

The sequel is organized as follows. After the introduction of a mortar formulation (section 2), we propose two possible Dirichlet-Neumann preconditioners and state their two-scale properties (section 3). In particular, the second enhanced preconditioner makes use of a coarse space counterbalancing the effect of essential boundary conditions imposed on the boundary sculptures. A simple numerical test shows its increased efficiency for a simple academic problem. A broader perspective on the subject as well as complete proofs are given in [4, 5].

2 Non-conforming formulation

For every $0 \le k \le K$, let $(\mathcal{T}_{k;h_k})_{h_k>0}$ be a sequence of meshes of the substructure Ω_k, h_k denoting the maximal diameter of its elements. The corresponding finite-element spaces of order q are denoted by $(V_{k;h_k})_{h_k>0} \subset H^1_*(\Omega_k)$. As in [3, 7], interface bubbles can be added on the fine subdomains for stability purposes, when using a discontinuous mortar formulation. As a consequence, we introduce the potentially enriched spaces of displacements $X_{k;h_k} = V_{k;h_k} \oplus B_{k;h_k}$ for every $1 \le k \le K$ and $X_{0;h_0} = V_{0;h_0}$. For each interface Γ_{0k}, $W_{k;h_k}$ will stand for the trace of the local space $X_{k;h_k}$ on this interface. In order to impose a weak displacement continuity between Ω_0 and Ω_k, a space of Lagrange multipliers $M_{k;h_k}$ is introduced on the mesh $\mathcal{T}_{k;h_k}$ over Γ_{0k}. Actually, various choices of continuous or of discontinuous polynomial functions of degree r can be used [2, 10, 12, 8, 6, 7] but in any case, they must satisfy the following fundamental assumptions:

Assumption 1 [Coercivity]. Let $u_0 \in H^1(\Omega_0)^d$ and $u_k \in H^1(\Omega_k)^d$ be rigid motions, i.e. $\varepsilon(u_0) = 0$ in $L^2(\Omega_0)^{d\times d}$ and $\varepsilon(u_k) = 0$ in $L^2(\Omega_k)^{d\times d}$, satisfying the weak continuity requirement $\displaystyle\int_{\Gamma_{0k}} (u_0 - u_k)\cdot\mu = 0$ for every $\mu \in M_{k;h_k}$. Then $u_0 = u_k$ almost everywhere on Γ_{0k}.

Assumption 2 [Inf-sup condition]. There exists a mapping $\pi_k : L^2(\Gamma_{0k}) \to W_{k;h_k}$ such that for all $v \in L^2(\Gamma_{0k})$,

$$\int_{\Gamma_{0k}} (\pi_k v)\cdot\mu = \int_{\Gamma_{0k}} v\cdot\mu, \quad \forall\mu \in M_{k;h_k},$$

satisfying $\|\pi_k v\|_{k,\frac{1}{2}} \le C\,\|v\|_{k,\frac{1}{2}}$. The mesh dependent norm $\|\cdot\|_{k,\frac{1}{2}}$ introduced above is defined as in [1, 11] by

$$\|v\|^2_{k,\frac{1}{2}} = \sum_{K\in\mathcal{T}_{k;h_k}} diam(K\cap\Gamma_{0k})^{-1}\int_{K\cap\Gamma_{0k}} v^2.$$

Assumption 3 [Accuracy]. The total degree r of Lagrange multipliers is bounded from below by $r \geq q-1$, q being the total degree of the displacement shape functions.

Then, the mortar formulation of the problem of interest can be written as finding $u = (u_0, u_1, ..., u_K) \in \prod_{k=0}^{K} X_{k;h_k}$ and $\lambda = (\lambda_1, ..., \lambda_K) \in \prod_{k=1}^{K} M_{k;h_k}$ satisfying for every $v \in \prod_{k=0}^{K} X_{k;h_k}$ and $\mu \in \prod_{k=1}^{K} M_{k;h_k}$,

$$\begin{array}{llll} a_0(u_0, v_0) & + \sum_{k=1}^{K} b_{0k}(v_0, \lambda_k) & = l_0(v_0) & \\ a_k(u_k, v_k) & - \quad b_k(v_k, \lambda_k) & = l_k(v_k), & 1 \leq k \leq K \\ b_{0k}(u_0, \mu_k) & - \quad b_k(u_k, \mu_k) & = 0, & 1 \leq k \leq K. \end{array} \tag{1}$$

The above problem uses the standard notation for the (bi)linear forms $a_k(u_k, v_k) = \int_{\Omega_k} \mathbf{E}_{ijmn} \varepsilon(u_k)_{mn} \varepsilon(v_k)_{ij}$, $l_k(v_k) = \int_{\Omega_k} f \cdot v_k + \int_{\Gamma_N \cap \partial\Omega_k} g \cdot v_k$, $b_{0k}(v_0, \mu_k) = \int_{\Gamma_{0k}} v_0 \cdot \mu_k$ and $b_k(v_k, \mu_k) = \int_{\Gamma_{0k}} v_k \cdot \mu_k$.

3 Two-scale preconditioners

We respectively denote by $\mathbf{A}_0$, $\mathbf{B}_{0k}$ and $\mathbf{B}_k$ the matrix representation of the bilinear forms a_0, b_{0k} and b_k. Similarly, L_0 and L_k are the vectors representing the linear forms l_0 and l_k. After elimination of the Lagrange multipliers λ_k in the first equation of (1), the system (1) becomes

$$\begin{cases} \mathbf{S}_0 U_0 = \overline{L_0}, \\ \mathbf{K}_k \begin{pmatrix} U_k \\ \Lambda_k \end{pmatrix} = \begin{pmatrix} L_k \\ -\mathbf{B}_{0k} U_0 \end{pmatrix}, \quad 1 \leq k \leq K, \end{cases} \tag{2}$$

where $\mathbf{S}_0 = \mathbf{A}_0 - \sum_{k=1}^{K} \mathbf{B}_{0k}^t R_k \mathbf{K}_k^{-1} R_k^t \mathbf{B}_{0k}$ is the Schur complement matrix, and $\overline{L_0} = L_0 - \sum_{k=1}^{K} \mathbf{B}_{0k}^t R_k \mathbf{K}_k^{-1} \begin{pmatrix} L_k \\ 0 \end{pmatrix}$ the corresponding right hand side. In these definitions, the local stiffness matrix $\mathbf{K}_k$ and restriction operator R_k are given by

$$\mathbf{K}_k = \begin{pmatrix} \mathbf{A}_k & -\mathbf{B}_k^t \\ -\mathbf{B}_k & 0 \end{pmatrix}, \quad R_k \begin{pmatrix} U_k \\ \Lambda_k \end{pmatrix} = \Lambda_k.$$

An iterative solver can be efficiently used to solve (2) if one is able to define a preconditioner $\tilde{\mathbf{S}}_0$ of the exact Schur complement $\mathbf{S}_0$ which is spectrally equivalent to $\mathbf{S}_0$, with constants independent of the number and the size of the small subdomains. When $L_0, .., L_K$ are given, the application of such a preconditioner consists in the following operations:

(a) Compute $\overline{L_0}$ by solving Dirichlet problems on the small subdomains prescribing zero displacements on the interfaces $(\Gamma_{0k})_{1\leq k\leq K}$,
(b) Solve the extended Neumann problem $\tilde{\mathbf{S}}_0\tilde{U}_0 = \overline{L_0}$,
(c) Compute $(\tilde{U}_k, \tilde{\Lambda}_k)$ over each Ω_k by solving the Dirichlet problem:

$$\mathbf{K}_k \begin{pmatrix} \tilde{U}_k \\ \tilde{\Lambda}_k \end{pmatrix} = \begin{pmatrix} L_k \\ -\mathbf{B}_{0k}\tilde{U}_0 \end{pmatrix}.$$

The most natural and rather efficient preconditioner consists in simply using $\tilde{\mathbf{S}}_0 = \mathbf{A}_0$. This is a standard Dirichlet-Neumann preconditioner for which we prove [5]:

Proposition 1. *Assuming that $\mathbf{A}_0$ is invertible, i.e. $\Gamma_D \cap \partial\Omega_0$ has a positive measure, the following spectral equivalence holds for all U_0:*

$$W_{1,h}\left\langle \mathbf{S}_0U_0, U_0\right\rangle \leq \left\langle \mathbf{A}_0U_0, U_0\right\rangle \leq \left\langle \mathbf{S}_0U_0, U_0\right\rangle,$$

with:

$$\frac{1}{W_{1,h}} = 1 + C\left(\max_{k\in I_1}\frac{C_k}{c_0} + \max_{k\in I_2}\frac{C_k\,\ell_0}{\alpha_0\,\ell_k}\right),$$

where I_1 (resp. I_2) is the set of indices $k \geq 1$ such that Ω_k is not fixed on its boundary (resp. is fixed on a part of its boundary), the positive constants c_k and C_k are such that $c_k\left|\xi\right|^2 \leq \mathbf{E}_{ijmn}\xi_{mn}\xi_{ij} \leq C_k\left|\xi\right|^2$ over Ω_k for every symmetric matrix $\xi \in \mathbb{R}^{d\times d}$, α_0 is the coercivity constant of the bilinear form a_0 and $\ell_k = diam(\Omega_k)$. The constant $C > 0$ is independent of the number K and the size of the subdomains.

This simple choice will lack efficiency in two simple situations:

(a) a fine subdomain Ω_k $(k \geq 1)$ has a small size $\ell_k \ll \ell_0$ and is fixed on a part of its boundary $(k \in I_2)$; in this situation, because of its size, the substructure will have a rather large stiffness to interface rigid body displacements,
(b) a fine subdomain Ω_k $(k \geq 1)$ has several stiff modes involving interface motions (rigid links, incompressibility).

Assuming that these directions of localized interface stiffness are few (this is indeed the case for interface rigid body motions), we denote by N_k their number. We then propose a modification of the previous preconditioner enabling us to correct the lack of efficiency.
For all $k \geq 1$ such that Ω_k is fixed on a part of its boundary, we denote by $(e_k^i)_{1\leq i\leq N_k}$ (with $N_k = 6$ in general) the interface rigid motions of Γ_{0k} or rigid links and introduce

$$\mathring{W}_k = span\{e_k^i, i = 1,..,N_k\}.$$

To each interface rigid body motion e_k^i, we associate its local a_k-harmonic extension $(u_k^i, \lambda_k^i) \in X_{k;h_k} \times M_{k;\delta_k}$, the solution of

$$\begin{cases} a_k(v, u_k^i) - \displaystyle\int_{\Gamma_{0k}} v\cdot\lambda_k^i = 0, \quad \forall v \in X_{k;h_k}, \\ -\displaystyle\int_{\Gamma_{0k}} u_k^i\cdot\mu = -\displaystyle\int_{\Gamma_{0k}} e_k^i\cdot\mu, \quad \forall\mu \in M_{k;\delta_k}. \end{cases} \tag{3}$$

These solutions span two small local spaces

$$\overset{\circ}{X}_k = span\{u_k^i, i = 1, .., N_k\} \subset X_{k;h_k},$$

$$\overset{\circ}{M}_k = span\{\lambda_k^i, i = 1, .., N_k\} \subset M_{k;\delta_k}.$$

If $k \geq 1$ is such that Ω_k is not fixed on its boundary, we adopt

$$\overset{\circ}{W}_k = \overset{\circ}{M}_k = \{0\}.$$

Then, instead of finding U_0 such that $\mathbf{S}_0 U_0 = \overline{L_0}$, we propose to compute $u_0 \in X_{0;h_0}$, $(u_k) \in (\overset{\circ}{X}_k)_{1 \leq k \leq K}$, $(\lambda_k) \in (\overset{\circ}{M}_k)_{1 \leq k \leq K}$, the solution of the coupled problem

$$\begin{cases} a_0(u_0, v_0) + \displaystyle\sum_{k=1}^{K} \int_{\Gamma_{0k}} v_0 \cdot \lambda_k = \overline{l_0}(v_0), \quad \forall v_0 \in X_{0;h_0}, \\ a_k(u_k, v_k) - \displaystyle\int_{\Gamma_{0k}} v_k \cdot \lambda_k = 0, \quad \forall v_k \in \overset{\circ}{X}_k, \quad 1 \leq k \leq K, \\ -\displaystyle\int_{\Gamma_{0k}} u_k \cdot \mu_k = -\int_{\Gamma_{0k}} u_0 \cdot \mu_k, \quad \forall \mu_k \in \overset{\circ}{M}_k, \quad 1 \leq k \leq K. \end{cases} \tag{4}$$

This amounts to reducing the local substructure response to the harmonic extension of its stiff interface modes, which belongs to $\overset{\circ}{X}_k$. We introduce the matrix $\mathbf{I}_{0k} = \mathbf{\Lambda}_k^T \mathbf{B}_{0k}$ where $\mathbf{\Lambda}_k^T = \left[\Lambda_k^1, .., \Lambda_k^{N_k}\right]^T$ is the matrix built with the multipliers computed in (3), and the restriction $\mathring{\mathbf{A}}_k$ of the displacement stiffness matrix $\mathbf{A}_k$ to the local space $\overset{\circ}{X}_k$

$$\left(\mathring{\mathbf{A}}_k\right)_{ij} = (U_k^i)^T \mathbf{A}_k U_k^j = a_k(u_k^j, u_k^i) = \int_{\Gamma_{0k}} u_k^j \cdot \lambda_k^i, \tag{5}$$

where (3) has been used. Exploiting (5) to reformulate (4)-2,(4)-3, the system (4) can be rewritten after some algebraic elimination as

$$\tilde{\mathbf{S}}_0 U_0 = \overline{L_0}, \tag{6}$$

with a new approximate Schur complement given by

$$\begin{aligned} \tilde{\mathbf{S}}_0 &= \mathbf{A}_0 + \sum_{k=1}^{K} \mathbf{I}_{0k}^T \mathring{\mathbf{A}}_k^{-t} \mathbf{I}_{0k} \\ &= \mathbf{A}_0 + \sum_{k=1}^{K} \mathbf{B}_{0k}^T \mathbf{\Lambda}_k \mathring{\mathbf{A}}_k^{-t} \mathbf{\Lambda}_k^T \mathbf{B}_{0k}. \end{aligned} \tag{7}$$

The complexity of its inversion is much smaller than solving $\mathbf{S}_0 U_0 = \overline{L_0}$ because each local problem (3) used in the construction of $\tilde{\mathbf{S}}_0$ only involves a subspace of displacements of dimension N_k. Moreover, we prove in [5] that:

Proposition 2. *For all U_0, the following spectral equivalence holds*

$$W_{1,h} \langle \mathbf{S}_0 U_0, U_0 \rangle \leq \left\langle \tilde{\mathbf{S}}_0 U_0, U_0 \right\rangle \leq \langle \mathbf{S}_0 U_0, U_0 \rangle,$$

with

$$\frac{1}{W_{1,h}} = C \left(1 + \max_{1 \leq k \leq K} \frac{C_k}{c_0}\right).$$

The constant $C > 0$ is independent of the number K and the size of the subdomains.

4 Numerical illustration

Let us consider here a two-scale beam (as represented in figure 1) whose both tips are clamped. The material is elastic, isotropic, homogeneous in each substructure, and the displacements under loading are computed by a preconditioned conjugate gradient method. Figure 2 illustrates the advantage of the enhanced Dirichlet-Neumann preconditioner when two small substructures are clamped. In conformity with the announced results, the gain in efficiency is independent of the ratio of Young moduli between the fine and coarse zones. Moreover, a factor 3 improvement is achieved in the number of iterations, and roughly speaking in the time of computation. Finally, it is shown in [5] that such a preconditioner can be used as an efficient quasi-tangent operator in the nonlinear framework as soon as the boundary geometrical details are sufficiently soft.

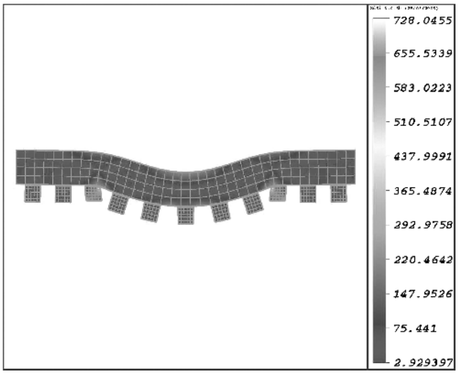


Fig. 1. Maximal stress distribution on a deformed configuration of our two-scale model problem where two of the details are clamped on their lower face.

5 Conclusion

The domain-decomposition based preconditioners proposed here achieve scale-independent performances. They should be extended to cases where the details overlap the coarse region as in a fictitious domain approach, and also to cases where the details are not disjoint but constitute a continuous belt along the boundary.

Acknowledgement. The authors gratefully acknowledge the financial support of Michelin Tire Company during the completion of this work at the Center of Applied Mathematics (CMAP), Ecole Polytechnique, and Professor François Jouve for providing his finite element code in which the presented ideas have been implemented.

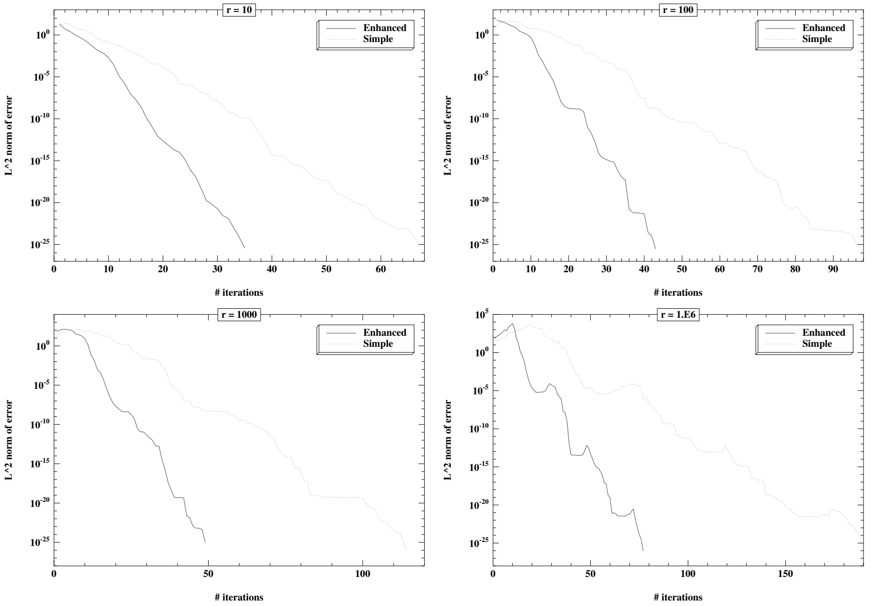


Fig. 2. Convergence of the simple and enhanced Dirichlet-Neumann algorithms for different values of the ratio r of Young moduli between the fine and coarse subdomains.

References

1. A. Agouzal and J.-M. Thomas, *Une méthode d'éléments finis hybrides en décomposition de domaines*, Math. Model. Numer. Anal., 29 (1995), pp. 749–764.
2. C. Bernardi, Y. Maday, and A. T. Patera, *Domain decomposition by the mortar element method*, in Asymptotic and Numerical Methods for Partial Differential Equations with Critical Parameters, H. K. ans M. Garbey, ed., N.A.T.O. ASI, Kluwer Academic Publishers, 1993, pp. 269–286.
3. F. Brezzi and D. Marini, *Error estimates for the three-field formulation with bubble stabilization*, Math. Comp., 70 (2000), pp. 911–934.
4. P. Hauret, *Méthodes numériques pour la dynamique des structures non-linéaires incompressibles à deux échelles (Numerical methods for the dynamic analysis of two-scale incompressible nonlinear structures)*, PhD thesis, CMAP, Ecole Polytechnique, 2004.
5. P. Hauret and P. L. Tallec, *Dirichlet-Neumann preconditioners for elliptic problems with small disjoint geometric refinements on the boundary*, Tech. Rep. 552, CMAP, Ecole Polytechnique, September 2004.
6. ———, *A stabilized discontinuous mortar formulation for elastostatics and elastodynamics problems, part I: abstract framework*, Tech. Rep. 553, CMAP, Ecole Polytechnique, September 2004.

7. ———, *A stabilized discontinuous mortar formulation for elastostatics and elastodynamics problems, part II: discontinuous Lagrange multipliers*, Tech. Rep. 554, CMAP, Ecole Polytechnique, September 2004.
8. C. Kim, R. Lazarov, J. Pasciak, and P. Vassilevski, *Multiplier spaces for the mortar finite element method in three dimensions*, SIAM J. Numer. Anal., 39 (2001), pp. 519–538.
9. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.
10. P. Seshaiyer, *Non-conforming hp finite element methods*, PhD thesis, University of Maryland, Baltimore County, Balitmore, MD, 1998.
11. B. I. Wohlmuth, *Hierarchical a posteriori error estimators for mortar finite element methods with Lagrange multipliers*, SIAM J. Numer. Anal., 36 (1999), pp. 1636–1658.
12. ———, *A mortar finite element method using dual spaces for the Lagrange multiplier*, SIAM J. Numer. Anal., 38 (2000), pp. 989–1012.
13. ———, *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, vol. 17 of Lecture Notes in Computational Science and Engineering, Springer, Berlin, 2001.

A Numerical Quadrature for the Schwarz-Chimera Method

J.-B. Apoung Kamga[1] and Olivier Pironneau[2]

[1] Laboratoire J.L. Lions, Université Paris VI, Paris, France.
apoung@ann.jussieu.fr

[2] Institut Universitaire de France Curie, Paris, France.
Olivier.Pironneau@upmc.fr

Summary. Chimera [7] happens to be a version of Schwarz's method and of Lions' space decomposition method (SDM). It was analyzed by Brezzi et al [1] but an estimate was missing for numerical quadrature. We give it here with new numerical tests.

1 Introduction

Consider a Hilbert space V, a continuous bilinear form $a(u,\hat{u})$ symmetric with a coercivity constant $\alpha > 0$, and f regular enough for well posedness of

$$a(u,\hat{u}) = (f,\hat{u}) \quad \forall \hat{u} \in V. \tag{1}$$

We assume that $V = V_1 + V_2$ and that $V_1 \cap V_2$ is of nonzero measure (i.e., overlapping) where each V_i is a closed subspace of V. We will need also two continuous symmetric bilinear forms $b_i(u,\hat{u}),\ i = 1,2$ coercive enough so that

$$\sum_1^2 b_i(\hat{u}_i,\hat{u}_i) + a(\hat{u}_i,\hat{u}_i) \geq a(\sum_1^2 \hat{u}_i, \sum_1^2 \hat{u}_i) \quad \forall \hat{u}_i \in V_i. \tag{2}$$

A typical example is the Dirichlet problem for $-\Delta u = f$ in $\Omega = \Omega_1 \cup \Omega_2$ and such that $\Omega_1 \cap \Omega_2 \neq \emptyset$; denote by $S_i = \partial\Omega_i \cap \Omega_j,\ j \neq i$. Then set

$$V_i = \{v \in L^2(\Omega): \quad v|_{\Omega_i} \in V(\Omega_i), \quad v|_{\Omega-\Omega_i} = 0\}. \tag{3}$$

Algorithm 1 (Schwarz)
Begin loop with a chosen $v_i^0 \in V_i$, and $n = 0$.
Find v_i^{n+1} such that $v_i^{n+1} - v_j^n \in V_i, \quad i,j = 1,2,\ j \neq i$ by solving

$$a(v_i^{n+1},\hat{v}_i) = (f,\hat{v}_i) \quad \forall \hat{v}_i \in V_i. \tag{4}$$

End loop.

The convergence has been analyzed by P.L. Lions [4, 5, 6] in a general setting. To be more precise, we present the following alternative; it uses $b_i(u,v) = b(u,v) = (\beta u, v),\ i = 1,2$ for some positive scalar β and two arbitrary functions $u_i^0 \in V_i$.

Algorithm 2 (SDM)

Begin loop with $n = 0$:

Find $u_i^{n+1} \in V_i$ by solving

$$\begin{aligned} b(u_1^{n+1} - u_1^n, \hat{u}_1) + a(u_1^{n+1} + u_2^n, \hat{u}_1) = (f, \hat{u}_1) \quad \forall \hat{u}_1 \in V_1, \\ b(u_2^{n+1} - u_2^n, \hat{u}_2) + a(u_1^n + u_2^{n+1}, \hat{u}_2) = (f, \hat{u}_2) \quad \forall \hat{u}_2 \in V_2. \end{aligned} \tag{5}$$

End loop.

When $\beta = 0$ Algorithm 2 is identical to Algorithm 1 with $u_i^{n+1} = v_i^{n+1} - v_j^n,\ i,j = 1,2,\ j \neq i$. If the decomposition is done with m subregions with $m \geq 2$ then u^{n+1} is found by solving

$$b(u_i^{n+1} - u_i^n, \hat{u}_i) + a(u_i^{n+1} - u_i^n + \sum_{j=1}^m u_j^n, \hat{u}_i) = (f, \hat{u}_i) \quad \forall \hat{u}_i \in V_i. \tag{6}$$

Theorem 1. *(Hecht et al. [3]) We assume (1-2). Then Algorithm (6) is convergent in the following sense: as $n \to \infty$, $u_i^n \to u_i^*$ with $u_1^* + u_2^* = u$ the solution of (1) and the decomposition is uniquely defined by*

$$\begin{aligned} &(\beta + A)u_1 = \frac{1}{2}(\beta + A)(u + u_1^0 - u_2^0) \quad in \quad \Omega_1 \cap \Omega_2, \quad u_1|_{S_1} = 0, \quad u_1|_{S_2} = u, \\ &(\beta + A)u_2 = \frac{1}{2}(\beta + A)(u + u_2^0 - u_1^0) \quad in \quad \Omega_1 \cap \Omega_2, \quad u_2|_{S_2} = 0, \quad u_2|_{S_1} = u, \\ &Au_i = f \quad in \quad \Omega_i \backslash \Omega_1 \cap \Omega_2, \quad u_i|_{\partial\Omega_i} = 0. \end{aligned} \tag{7}$$

2 Discretization

Let $\mathcal{T}_{1h}$ (resp $\mathcal{T}_{2h}$) be a triangulation of Ω_1 (resp Ω_2), quasi-uniform [2], in the sense that, if h_M and h_m are the maximum and minimum edges in $\mathcal{T}_{1h}$, and H_M and H_m are the maximum and minimum edges in $\mathcal{T}_{2h}$, then there exists two constants C_{1T} and C_{2T} such that $h_M \leq C_{1T} h_m$ and $H_M \leq C_{2T} H_m$. Without loss of generality we can also assume, that $h_M \leq H_M$. For clarity we assume that the Ω_i are polygonal and that $a(\cdot,\cdot)$ represents the Laplace operator with Dirichlet conditions. Let V_{1h} and V_{2h} be two Lagrange conforming continuous finite element approximation spaces of order p of the spaces $V_1 = H_0^1(\Omega_1)$ and $V_2 = H_0^1(\Omega_2)$. Then the discrete version of Algorithm 2 is to find for i=1,2, $u_{ih}^{n+1} \in V_{ih}$ such that $\forall v_{ih} \in V_{ih}$

$$\int_{\Omega_i} (\beta(u_{ih}^{n+1} - u_{ih}^n) v_{ih} + \nabla(u_{1h}^{n+1} + u_{2h}^n) \nabla v_{ih}) = \int_{\Omega_i} f v_{ih}.$$

Theorem 2. *(Hecht et al. [3]) Assume that the solution of (1) is in $H^{p+1}(\Omega)$ for some $p \geq 1$. Assume that in (7) $u_i|_{\Omega_i} \in H^{p+1}(\Omega_i)$. If $u_h = \lim(u_{1h}^n + u_{2h}^n)$ is computed with Lagrange conforming finite elements of order p, then $\|u - u_h\|_{1,\Omega} \leq Ch^p(|u_1|_{p+1,\Omega_1} + |u_2|_{p+1,\Omega_2})$.*

3 Numerical Quadrature

As such, the scheme is too costly to implement because it requires the intersection of triangulations. Recall that the quadrature formula with integration points at the vertices is exact for polynomials of degree less than or equal to one. In particular, for a given triangle $\hat{T}$ one has

$$\int_{\hat{T}} g\,dxdy = \frac{|\hat{T}|}{3} \sum_{i=1,2,3} g(q_i) \quad \forall g \in P_1(\hat{T}). \tag{8}$$

Hence we introduce the following quadrature rule. $(\nabla u, \nabla v)_h :=$

$$\sum_{T\in\mathcal{T}_{1h}} \frac{|T|}{3} \sum_{i=1,2,3} \frac{\nabla(u_{|T})\cdot\nabla v}{I_{\Omega_1}+I_{\Omega_2}}|_{q_i(T)} + \sum_{K\in\mathcal{T}_{2h}} \frac{|K|}{3} \sum_{j=1,2,3} \frac{\nabla(v_{|K})\cdot\nabla u}{I_{\Omega_1}+I_{\Omega_2}}|_{q_j(K)}, \tag{9}$$

where $I_{\Omega_j}(x) = 1$ if $x \in \Omega_j$ and zero otherwise ($j = 1, 2$). The notation $\nabla(u_{|T})$ is used to indicate that we first restrict the function u to T, and then we compute its gradient (which is actually constant in T). A similar interpretation holds for $\nabla(v_{|K})$. With such definitions we propose to solve the discrete problems:
- Find $u_{ih}^{n+1} \in V_{ih}$ such that $\forall v_{ih} \in V_{ih}$

$$\begin{array}{ll} b(u_{1h}^{n+1} - u_{1h}^{n}, \hat{u}_{1h}) + a_h(u_{1h}^{n+1} + u_{2h}^{n}, \hat{u}_{1h}) = (f, \hat{u}_{1h}) & \forall \hat{u}_{1h} \in V_{1h}, \\ b(u_{2h}^{n+1} - u_{2h}^{n}, \hat{u}_{2h}) + a_h(u_{1h}^{n} + u_{2h}^{n+1}, \hat{u}_{2h}) = (f, \hat{u}_{2h}) & \forall \hat{u}_{2h} \in V_{2h}. \end{array} \tag{10}$$

Clearly these define u_{ih}^{n+1} uniquely. At convergence the problem solved is
- Find $u_{ih} \in V_{ih}$ such that $\forall \hat{u}_{ih} \in V_{ih}$

$$a_h(u_{1h} + u_{2h}, \hat{u}_{1h} + \hat{u}_{2h}) = (f, \hat{u}_{1h} + \hat{u}_{2h}). \tag{11}$$

Under a mild assumption on the triangulations this discrete problem has a unique solution at least when linear elements are used ($p = 1$):

$$\textbf{each vertex of } \mathcal{T}_{1h} \textbf{ is internal to a triangle } K \textbf{ of } \mathcal{T}_{2h}, \textbf{ and conversely}. \tag{12}$$

This is because of the coercivity of the bilinear form and of the uniqueness of the decomposition $u_h = u_{1h} + u_{2h}$:

Theorem 3. *(Brezzi et al. [1]) Assume (12) holds. If two functions $u_{ih} \in V_{ih}$, $i = 1, 2$ coincide on a connected subset $\mathcal{X}$ of $\Omega_1 \cap \Omega_2$, then both u_{ih} are linear (not just piecewise linear) in $\mathcal{X}$. Furthermore $a_h(u_{1h} + u_{2h}, u_{1h} + u_{2h}) \geq c\|u_{1h} + u_{2h}\|^2$ for all $u_{ih} \in V_{ih}$.*

One more property is needed, the continuity of a_h, and then we can apply Strang's lemma and obtain the following estimate:

Proposition 1. *(Hecht et al. [3]) Assume that the triangulations of Ω_1 and Ω_2 are compatible in the sense that they give a coercive bilinear form. Assume that a_h is uniformly continuous for all h. Then the error between the approximate problem (11) and the continuous one is $\|u - u_h\| < Ch(|u_1|_{2,\Omega_1} + |u_2|_{2,\Omega_2})$.*

4 Continuity of the Approximate Bilinear Form

4.1 The One-dimensional Case

We begin with the one-dimensional case because the proof is easier to follow. The same argument will be extended to two dimensions.

Proposition 2. *In one dimension the constant of continuity C in*

$$|\nabla u_H + \nabla u_h|_h \le C|\nabla u_H + \nabla u_h|$$
$$\textit{satisfies} \qquad C^2 \le \frac{1}{2}\max\{\max_{i\in K}\frac{|x_{i+1}-x_i|}{|x_i - X_{j(i)}|}, \max_{i\in L}\frac{|X_{i+1}-X_i|}{|X_i - x_{j(i)}|}\}, \tag{13}$$

where K (resp. L) is the set of i such that $j(i)$ exists with $X_{j(i)} \in [x_i, x_{i+1}]$ (resp $x_{j(i)} \in [X_i, X_{i+1}]$). Consequently C is bounded by the square root of half the largest interval length divided by the smallest distance between two vertices.

Proof. For any real valued function f, $\max\limits_{u_h,u_H} f(\nabla u_H + \nabla u_h) \le \max\limits_{U_h,U_H} f(U_H + U_h)$ where u_h, u_H are real valued continuous-piecewise linear functions on their meshes and U_H, U_h are piecewise constant vector valued on their meshes, because every ∇u is a U and the opposite is not true when boundary conditions exist at both ends. Denote $V = U_H + U_h$. As V is piecewise constant, by definition

$$4|V|_h^2 = \sum_i |x_{i+1}-x_i|(|V|(x_i^+)^2 + |V|(x_{i+1}^-)^2) + \sum_j |X_{j+1}-X_j|(|V|(X_j^+)^2 + |V|(X_{j+1}^-)^2),$$

$$2|V|_0^2 = \sum_{i,j\in K} |X_j - x_i|(|V|(X_j^-)^2 + |V|(x_i^+)^2) + \sum_{i,j\in L} |x_i - X_j|(|V|(X_j^+)^2 + |V|(x_i^-)^2) \tag{14}$$
$$+\sum_{i\in I} |x_{i+1}-x_i|(|V|(x_i^+)^2 + |V|(x_{i+1}^-)^2) + \sum_{j\in J} |X_{j+1}-X_j|(|V|(X_j^+)^2 + |V|(X_{j+1}^-)^2), \text{ where}$$

I, J are the set of intervals completely inside an interval of the other mesh, i.e.

$$I = \{i : \exists j \ s.t. \ [x_i, x_{i+1}] \subset [X_j, X_{j+1}\}, \ J = \{j : \exists i \ s.t. \ [X_j, X_{j+1}] \subset [x_i, x_{i+1}]\}$$

Denote by N the set of values of V_k of V right or left of x_i or X_j. As $f(V) = |V|_h^2/|V|_0^2$ we see that it is of the type $f(V) = \sum\limits_{k\in N} \alpha_k |V|_k^2 \ / \ \sum\limits_{k\in N} \beta_k |V|_k^2$ with α_i equal to a fourth of $x_{i+1} - x_i$ or $X_{i+1} - X_i$, and β_i equal half of $x_{i+1} - x_i$ or $X_{i+1} - X_i$ or $x_i - X_{j(i)}$ or $X_i - x_{j(i)}$ a sum of two of those. Of course it is important to notice that all values appear both in the nominator and denominator. With a change of variable this is also

$$f(W) = \frac{\sum \frac{\alpha_k}{\beta_k} W_k^2}{\sum W_k^2}. \text{ Then} \qquad \max f(W) = \max_k \frac{\alpha_k}{\beta_k}.$$

♯

Now that this is established we can address much more simply the problem of finding $\max \alpha_k/\beta_k$: it is the largest ratio of coefficients multiplying $V(x_i^{\pm})$ or $V(X_j^{\pm})$ in the expressions for $|V|_h$ and $|V|_0^2$, i.e. in (14).

4.2 The Two-dimensional Case

A similar argument applies in two dimensions. Assume we have two triangulations with triangles $\{T_k\}_1^N$ and $\{t_k\}_1^n$ respectively and vertices Q_i and q_i. Recall that

$$|V|_h^2 = \frac{1}{6}\sum_{k=1}^{N}\sum_{j=1,2,3}|V_{T_k}(Q_{i_j})|^2|T_k| + \frac{1}{6}\sum_{k=1}^{n}\sum_{j=1,2,3}|V_{t_k}(q_{i_j})|^2|t_k|, \tag{15}$$

where $i_j,\ j = 1,2,3$ are the numbers of the 3 vertices for each triangle. On the other hand the exact value $|V|_0^2$ is

$$|V|_0^2 = \sum_{k,l}\sum_{j=1,2,3}|V_{R_{kl}}(\xi_{kl})|^2|R_{kl}|, \tag{16}$$

where $R_{kl} = T_k \cap t_l$ and ξ_{kl} is any point in R_{kl}.
For each Q_{i_j} (resp q_{i_j}) in (15) there is a R_{kl} which contains it. For these R let us choose in (16) $\xi_{kl} = Q_{i_j}$ and q_{i_j}. Then for every term in $|V|_h^2$ there is a corresponding term in $|V|_0^2$:

$$\frac{1}{6}|V_{T_k}(Q_{i_j}|^2|T_k| \text{ corresponds to } |V_{T_k}(Q_{i_j}|^2|T_k \cap t_l|, \tag{17}$$

where l is such that $Q_{i_j} \in t_l$; and similarly with q_{i_j}.
However in this construction we will assign as many ξ to R as the number of vertices it contains. So the safest is to divide the second term in (17) by 3.
Notice that some R do not contain any vertex; if we leave these aside we obtain

$$\frac{|V|_h^2}{|V|_0^2} \le \frac{1}{2}\max_{k,l}\{\frac{\max\{|T_k|,|t_l|\}}{|T_k \cap t_l|}\ :\ T_k \cap t_l \text{ contains at least one vertex }\}. \tag{18}$$

So we have proved the following

Proposition 3. *In two dimensions, the constant of continuity between the approximate norm $|\nabla u_H + \nabla u_h|_h$ and the exact one is proportional to the square root of the biggest ratio of area between a triangle T and one of its polygons $T \cap t$ where t is a triangle of the other triangulation containing a vertex of T.*

The proof is similar, except that in the exact norm there are terms which do not exist in the approximate norm; but these are positive and appear in the denominator of the expression which bounds C.

Remark 1. Consider the case where each triangle of the mesh h has no more than one vertex of the mesh H inside. Assume that this vertex is near the center of the triangle (or segment in one-D). Assume that all angles between two intersecting edges are bounded away from 0 and π when $h, H \to 0$ and that H/h and h/H do not tend to 0. Then C is strictly posivite in the limit. However it is difficult in practice to insure that no angle tend to zero when the mesh is refined.

5 Numerical Test

In all the numerical tests that follow, errors are evaluated on the intersected mesh, using exact quadrature formula. The problem to solve is $-\Delta u = f$ in Ω, $u = g$ on $\partial\Omega$. Data are chosen so that $u(x,y) = \sin(x) \times \sin(y)$.

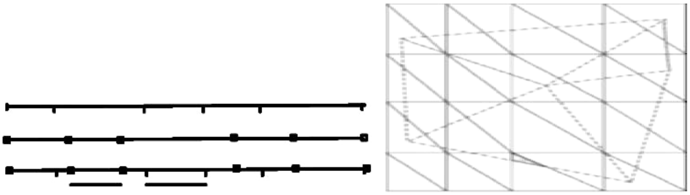


Fig. 1. Left: Two meshes in 1D and the intersected mesh. Two intervals have been singled out as they are strictly inside an interval of the other mesh; the continuity constant is proportional to the ratio of the smallest interval in the intersected mesh to the biggest interval in both mesh neighbors to the smallest one. Right: The continuity constant is proportional to the smallest polygon containing a vertex (shown with a texture) divided by the area of the biggest neighbor triangle in both meshes. Notice that some edges pass right through a vertex in this example, so if one mesh is shifted slightly the continuity constant estimate suddenly deteriorates.

5.1 Exact quadrature

This formula is introduced so as to give an exact computation for integral like $\int_{T_h \cap T_H} \Phi\Psi$. Where Φ and Ψ see Fig 2 below are $P1$-lagrange functions on the triangle T_h and T_H respectively. It is based on the intersection of the two meshes. Ω_1 is a circle of radius 1 centered at $(0,0)$ and Ω_2 is the square $(-0.5, 0.5)^2$. Ω_2 is going to be meshed with uniform triangles so that by dyadic refinement, order of convergence can be easily evaluated see Table 1.

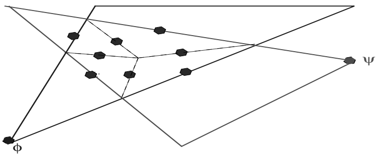


Fig. 2. Quadrature for exact evaluation of $\int_{T_h \cap T_H} \Phi\Psi$.

5.2 First quadrature formula

Table 1 displays the results when (9) is used. Notice that by taking $u \in V_h, v \in V_h$, we do not recover the ordinary approximated bilinear form for the Laplace equation on the domain Ω_1. So for a parallel implementation of (10), instead, we must find $u^{n+1} \in V_{0h}$ such that (here $b \equiv 0$), $\forall \hat{v} \in V_{0h}(\Omega_1)$

$$(\nabla u_1^{n+1}, \nabla \hat{v})_h = (f, \hat{v}) - (\nabla u_2^n, \nabla \hat{v})_{hH} - \frac{1}{2}(\nabla u_1^n, \nabla \hat{v})_h + \frac{1}{2}(\nabla u_1^n, \nabla \hat{v})_H.$$

Here $(\cdot,\cdot)_h, (\cdot,\cdot)_H$ do not need quadrature. For the numerical experiments, we have taken $\Omega_2 = (-2,3) \times (-3,2)$ and $\Omega_1 = (-\frac{4}{3}, \frac{5}{3}) \times (-\frac{5}{3}, \frac{4}{3})$.

5.3 Second quadrature formula

In our works, we have also tried, for $u_1, v_1 \in V_h,\ u_2, v_2 \in V_H$

$$
\begin{aligned}
(\nabla u_1, \nabla v_2)_{hH,\Omega_1\cap\Omega_2} &:= \sum_{K\in\mathcal{K}_H} \frac{|K|}{3} \sum_{j=1,2,3} \left(\nabla(u_1)\cdot\nabla(v_2|_K)\right)(q_j(K)),\\
(\nabla u_2, \nabla v_1)_{Hh,\Omega_1\cap\Omega_2} &:= \sum_{T\in\tau_h} \frac{|T|}{3} \sum_{j=1,2,3} \left(\nabla(u_2)\cdot\nabla(v_1|_T)\right)(q_j(T)).
\end{aligned}
\tag{19}
$$

5.4 Schwarz algorithm with quadrature

Finally, to compare with Schwarz' algorithm, let $\pi_{hH} : V_h \mapsto V_H$ and $\pi_{Hh} : V_H \mapsto V_h$ be the P^1 interpolation operators. Then the Schwarz method is implemented as

$$
\begin{cases}
(\nabla(u^{n+1} + \pi_{Hh}v^n), \nabla\hat{u})_h = (f, \hat{u})_h & \forall \hat{u} \in V_{0h},\\
(\nabla(v^{n+1} + \pi_{hH}u^n), \nabla\hat{v})_H = (f, \hat{v})_H & \forall \hat{v} \in V_{0H}.
\end{cases}
\tag{20}
$$

		$u-(u_1+u_2)$			
$N1$	$N2$	L^2 error	rate	∇L^2 error	rate
Exact Quadrature					
10	5	$1.54E-02$	–	$2.25E-01$	–
20	10	$3.78E-03$	2.02	$1.11E-01$	1.02
40	20	$8.24E-04$	2.2	$5.03E-02$	1.15
First Quadrature					
3	5	$4.64E-01$	–	$1.00E-00$	–
6	10	$8.18E-02$	2.50	$5.44E-01$	0.89

		$u-(u_1+u_2)$			
$N1$	$N2$	L^2 error	rate	∇L^2 error	rate
Second Quadrature					
10	5	$1.85E-02$	–	$2.32E-01$	–
20	10	$5.66E-03$	1.71	$1.16E-01$	1.00
40	20	$1.03E-03$	2.45	$5.34E-02$	1.12
Schwarz overlapping					
10	5	$1.68E-02$	–	$2.29E-01$	–
20	10	$3.49E-03$	2.26	$1.09E-01$	1.06
40	20	$9.15E-04$	1.93	$5.13E-02$	1.09

Table 1. Numerical L^2 errors, and convergence rate, for P1 polynomials with different quadrature formula. $N_i, i = 1,2$ is the number of vertices on the boundary of the domain Ω_i.

Conclusion

The results show that the first quadrature formula has optimal errors numerically but the results are very sensitive to the position of the grid points. Good results are obtained with the second quadrature formula, which is also easy to implement in 3D but no error analysis is yet available.

References

1. F. Brezzi, J.-L. Lions, and O. Pironneau, *Analysis of a Chimera method*, C.R. Math. Acad. Sci. Paris, (2001), pp. 655–660.
2. P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.
3. F. Hecht, J.-L. Lions, and O. Pironneau, *Domain decomposition algorithm for computed aided design*, in Applied Nonlinear Analysis, A. S. et al., ed., Kluwer Academic-Plenum Publishers, New York, 1999, pp. 185–198.
4. P.-L. Lions, *On the Schwarz alternating method. I.*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM, pp. 1–42.
5. ———, *On the Schwarz alternating method. II.*, in Domain Decomposition Methods, T. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1989, SIAM, pp. 47–70.
6. ———, *On the Schwarz alternating method. III: a variant for nonoverlapping subdomains*, in Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989, T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM, pp. 202–223.
7. J. L. Steger, *The Chimera method of flow simulation.* Workshop on applied CFD, University of Tennessee Space Institute, August 1991.

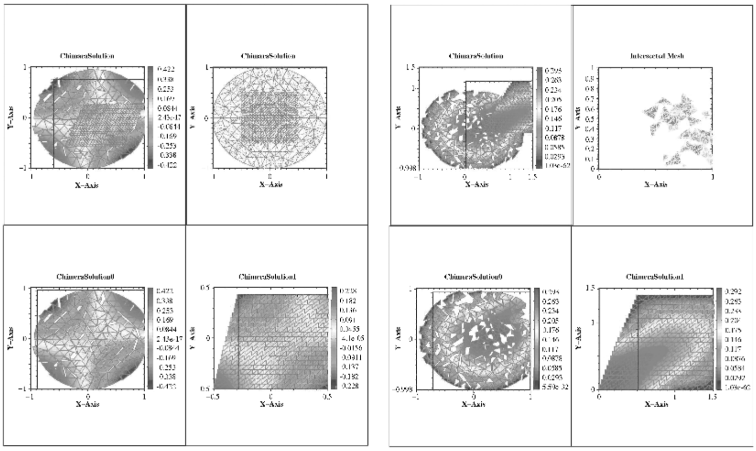


Fig. 3. Left set: Chimera test case with exact quadrature formula. (Top: solution on the entire domain. Bottom: solution on each subdomain.) Right set: Chimera test of $\Delta u = 1$ on Ω, $u = 0$ on $\partial\Omega$ with second quadrature formula. (Top right: intersected mesh. Bottom: solution on each domain.)

A New Variant of the Mortar Technique for the Crouzeix-Raviart Finite Element

Talal Rahman[1] and Xuejun Xu[2]

[1] BCCS, Bergen Center for Computational Science, Thormøhlensgt. 55, N-5008 Bergen, Norway. talal@bccs.uib.no

[2] LSEC, Institute of Computational Mathematics, Chinese Academy of Sciences, P.O. Box 2719, Beijing, 100080, People's Republic of China. xxj@lsec.cc.ac.cn

Summary. We propose a new variant of the mortar method for the lowest order Crouzeix-Raviart finite element for the approximation of second order elliptic boundary value problems on nonmatching meshes.

1 Introduction

The mortar technique (cf. [3, 1]) is the class of domain decomposition method that allows for nonmatching meshes for solving partial differential equations. To ensure that the overall discretization involving the nonmatching meshes makes sense, an optimal coupling between the meshes is required. In a standard mortar technique, this condition is realized by applying the condition of weak continuity on the solution, called the mortar condition, saying that the jump of the solution along the interface between two meshes is orthogonal to some suitable test space. Since its first introduction, the mortar technique has been studied extensively, see [2, 6, 8, 10, 5], and the references therein.

In order to apply the mortar condition, it is necessary to know the function on the interface. For the conforming P1 finite element, it is enough to know the nodal values along the interface. However, for the nonconforming P1 finite element (the lowest order Crouzeix-Raviart finite element), where the degrees of freedom are associated with the edge midpoints, see Fig. 1, the function on the interface depends on the nodal values corresponding to interface nodes and some subdomain interior nodes lying closest to the interface, cf. [6]. The purpose of this paper is to modify the mortar condition, so that the new method will use only the nodal values on the interface. This is a clear advantage compared to the standard method, especially in 3D. The approach can also be seen as the mortar method with an approximate constraint, see [4] for instance.

We propose our new mortar variant in Section 2, and present its matrix formulation in Section 3. An additive Schwarz preconditioner similar to the one in [7] for

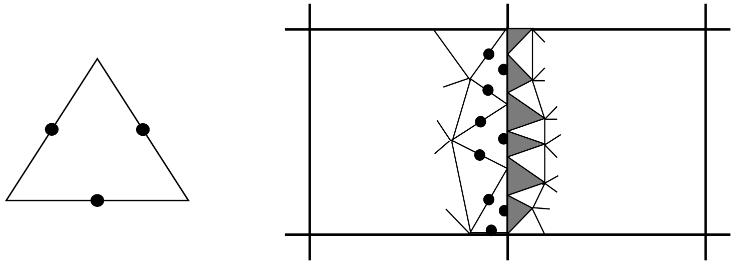


Fig. 1. The lowest order Crouzeix-Raviart (CR) finite element (left) and two nonmatching grids (right). CR basis functions associated with the nodes on the mortar side, denoted by dots (in the interior) and semi-dots (on the mortar), have nonzero support on the nonmortar side, denoted by the shaded triangles.

the new mortar variant is formulated in Section 4, and finally some numerical results are presented in Section 5.

2 The new mortar variant

Let $\Omega \subset R^2$ be a simply connected bounded domain, partitioned (conformingly) into a collection of nonoverlapping polygonal subdomains, $\Omega_i, i = 1, \ldots, N$, such that $\overline{\Omega} = \bigcup_i \overline{\Omega}_i$. We consider the problem: Find $u^* \in H_0^1(\Omega)$ such that

$$a(u^*, v) = f(v), \;\; v \in H_0^1(\Omega), \tag{1}$$

where $a(u, v) = \sum_{i=1}^{N} \int_{\Omega_i} \nabla u \cdot \nabla v \; dx$ and $f(v) = \sum_{i=1}^{N} \int_{\Omega_i} fv \; dx$. With each subdomain Ω_i, we associate a quasi-uniform triangulation $\mathcal{T}_h(\Omega_i)$ of mesh size h_i. The resulting triangulation can be nonmatching across subdomain interfaces.

Let $X_h(\Omega_i)$ be the nonconforming P1 (Crouzeix-Raviart) finite element space defined on the triangulation $\mathcal{T}_h(\Omega_i)$ of Ω_i, consisting of functions which are piecewise linear in each triangle $\tau \subset \Omega_i$, continuous at the interior edge midpoints of Ω_{ih}^{CR}, and vanishing at the edge midpoints of $\partial\Omega_{ih}^{CR} \cap \partial\Omega$ lying on the boundary $\partial\Omega$. Here, Ω_{ih}^{CR} and $\partial\Omega_{ih}^{CR}$ represent the sets of edge midpoints, i.e., the Crouzeix-Raviart nodal points, of Ω_i and $\partial\Omega_i$, respectively. In the same way, we use Ω_{ih} and $\partial\Omega_{ih}$ (without the superscript CR) to denote the corresponding sets of triangle vertices.

Since the triangulations on Ω_i and Ω_j do not match on their common interface Γ_{ij}, the functions in $X_h(\Omega) = \prod_i X_h(\Omega_i)$ are discontinuous at the edge midpoints along the interface. In the standard mortar technique, see [6], the condition of weak continuity, called the mortar condition, is therefore imposed. In this paper, we introduce a new variant of the mortar condition. Let $\gamma_{m(i)} \subset \partial\Omega_i$ and $\delta_{m(j)} \subset \partial\Omega_j$ be the mortar and the nonmortar side of the interface Γ_{ij}, respectively. Let $u_h \in X_h$, where $u_h = \{u_i\}_{i=1}^{N}$. A function $u_h \in X_h$ satisfies the mortar condition on $\delta_{m(j)} = \Gamma_{ij} = \gamma_{m(i)}$, if

$$Q_m I_m u_i = Q_m u_j, \tag{2}$$

where I_m is an interpolation operator, to be defined in the next paragraph, and Q_m is the L^2-projection operator $Q_m : L^2(\Gamma_{ij}) \to M^{h_j}(\delta_{m(j)})$ defined as $(Q_m u, \psi)_{L^2(\delta_{m(j)})} = (u, \psi)_{L^2(\delta_{m(j)})}, \quad \forall \psi \in M^{h_j}(\delta_{m(j)})$, where $M^{h_j}(\delta_{m(j)}) \subset L^2(\Gamma_{ij})$ is the test space of functions which are piecewise constant on the triangulation of $\delta_{m(j)}$, and $(\cdot,\cdot)_{L^2(\delta_{m(j)})}$ denotes the L^2 inner product on $L^2(\delta_{m(j)})$. For the function u_j on the nonmortar, the interior degrees of freedom do not affect the mortar matching condition for $M^{h_j}(\delta_{m(j)})$ containing piecewise constant functions. We note that, for the standard mortar method, I_m is simply the identity.

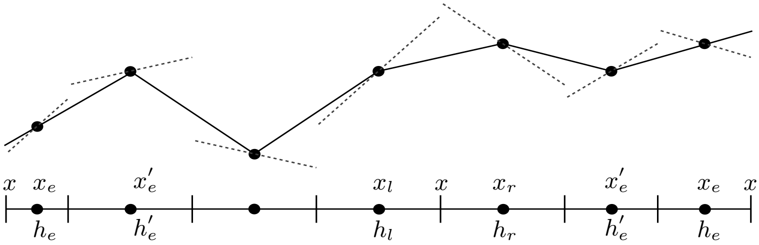


Fig. 2. Showing $u|_{\gamma_m}$ by dotted lines, and $I_m u|_{\gamma_m}$ by the solid line.

Let $\mathcal{T}_{\frac{h}{2}}(\gamma_m)$ be the triangulation associated with the mortar γ_m, which is obtained as a result of dividing the edges of $\mathcal{T}_h(\gamma_m)$ in two. Let $W_{\frac{h}{2}}(\gamma_m)$ be the conforming space of piecewise linear continuous functions on the triangulation $\mathcal{T}_{\frac{h}{2}}(\gamma_m)$. The functions of this space are defined by their values at the set $\overline{\gamma}_{m\frac{h}{2}}$ of all edge endpoints of $\mathcal{T}_{\frac{h}{2}}(\gamma_m)$. It is easy to see that $\overline{\gamma}_{m\frac{h}{2}} = \gamma_{mh}^{CR} \cup \overline{\gamma}_{mh}$, where γ_{mh}^{CR} and $\overline{\gamma}_{mh}$ are respectively the sets of edge midpoints and edge endpoints of $\mathcal{T}_h(\gamma_m)$. We now define the operator $I_m : X_h(\gamma_m) \to W_{\frac{h}{2}}(\gamma_m)$ below.

Definition 1. *For $u \in X_h(\gamma_m)$, $I_m u \in W_{\frac{h}{2}}(\gamma_m)$ is defined by the nodal values as*

$$I_m u(x) = \begin{cases} u(x), & x \in \gamma_{mh}^{CR}, \\ \dfrac{h_r}{h_l + h_r} u(x_l) + \dfrac{h_l}{h_l + h_r} u(x_r), & x \in \gamma_{mh}, \\ u(x_e) + \dfrac{h_e}{h_e + h_e'}(u(x_e) - u(x_e')) & x \in \partial\gamma_{mh}. \end{cases} \tag{3}$$

Here, x_l and x_r are the left and the right neighboring edge midpoints of x, respectively. Correspondingly, h_l and h_r are the left- and the right edge lengths. x_e and h_e are the midpoint and the length of the edge of $\mathcal{T}_h(\gamma_m)$, touching $\partial\gamma_m$. The edge midpoint x_e' and the edge length h_e' correspond to the neighboring edge.

The interpolation is done basically by first joining the edge midpoints with piecewise straight lines, and then stretching the two end straight lines to the end of the mortar γ_m, cf. Fig. 2. It is not difficult to see that the operator I_m preserves all linear functions on the mortar.

$V_h \subset X_h$ is a subspace of functions which satisfy the mortar condition for all $\delta_m \subset \mathcal{S}$. Since functions of V_h are not continuous, we use the broken bilinear form $a_h(\cdot,\cdot)$ defined according to $a_h(u,v) = \sum_{i=1}^{N} a_i(u,v) = \sum_{i=1}^{N} \sum_{\tau \in \mathcal{T}_h(\Omega_i)} (\nabla u, \nabla v)_{L^2(\tau)}$. The discrete problem takes the following form: Find $u_h^* = \{u_i\}_{i=1}^{N} \in V_h$ such that

$$a_h(u_h^*, v_h) = f(v_h), \quad \forall v_h \in V_h. \tag{4}$$

If the h_i's are of the same order h, then the following error estimate can be shown.

Theorem 1. *For all $u \in V_h$,*

$$\| u^* - u_h^* \|_{L^2(\Omega)} + h|u^* - u_h^*|_{H_h^1(\Omega)} \leq ch^2 \| u^* \|_{H^2(\Omega)} \tag{5}$$

3 Matrix Formulation

Like in the standard mortar case, each basis function of V^h is associated with an edge midpoint either in the interior of a subdomain or on a mortar, and not on any nonmortar. Let $\varphi_k^{(i)}$ denote a standard nodal basis function of $X_h(\Omega_i)$, associated with an edge midpoint $x_k \in \overline{\Omega}_{ih}^{CR}$. The basis functions of V^h can be defined as follows. If $x_k \in \Omega_{ih}^{CR}$, a subdomain interior node, then ϕ_k is identical with $\varphi_k^{(i)}$. If $x_k \in \gamma_{m(i)h}^{CR}$, a mortar node, then $\phi_k(x) = \varphi_k^{(i)}(x)$ on $\overline{\Omega}_i$, while on $\overline{\delta}_{m(j)}$, where $\gamma_{m(i)} = \delta_{m(j)}$, $\phi_k(x) = Q_m(I_m\varphi_k^{(i)})(x)$ at $x \in \delta_{m(j)h}^{CR}$. ϕ_k is zero at the remaining edge midpoints of $\overline{\Omega}_j$, and zero everywhere on the remaining subdomains. Using the basis functions of V_h, the problem (4) can be rewritten in the matrix form as

$$\mathbf{A}\mathbf{u}^* = \mathbf{f}, \tag{6}$$

where $\mathbf{u}^*$ is a vector of nodal values of u_h^*, and $\mathbf{A}$ is a matrix generated by the bilinear form $a_h(.,.)$ on $V_h \times V_h$. We shall now see how this matrix can be obtained from the local matrices $\hat{\mathbf{E}}_i$ generated by $a_i(.,.)$ on $X_h(\Omega_i) \times X_h(\Omega_i)$.

Observing that $a_h(.,.) = \sum_{i=1}^{N} a_i(.,.)$, where $a_i(.,.) = a_h(.,.)|_{\Omega_i}$, we can calculate the elements of $\mathbf{A}$ from their local contributions restricted to individual subdomains Ω_i. In order to calculate the local contribution $a_i(.,.)$, we use only those basis functions that have nonzero supports on $\overline{\Omega}_i$. These basis functions are exactly the ones associated with the nodes of Ω_{ih}^{CR}, $\gamma_{m(i)h}^{CR}$ $(\gamma_{m(i)} \subset \partial\Omega_i)$, and the set $\gamma_{m(j)h}^{CR}$ $(\gamma_{m(j)} = \delta_{m(i)} \subset \partial\Omega_i)$ of neighboring mortar edge midpoints except those on $\partial\Omega$. Let Λ_i be the set of all these nodes, see Fig. 3 for an illustration.

Let $\mathbf{P}_i$ be the restriction matrix which is a permutation of a rectangular identity matrix, such that $\mathbf{P}_i\mathbf{u}$ returns the vector of all coefficients of $\mathbf{u}$, associated with the nodes of Λ_i. $\mathbf{P}_i^T$ is the corresponding extension matrix. Let $\mathbf{E}_i$, associated with the subdomain Ω_i, be the matrix generated by $a_i(.,.)$ on $span\{\phi_k : x_k \in \Lambda_i\} \times span\{\phi_l : x_l \in \Lambda_i\}$. Using these three types of matrices, we can assemble the global matrix as $\mathbf{A} = \sum_{i=1}^{N} \mathbf{P}_i^T \mathbf{E}_i \mathbf{P}_i$.

We note that $\mathbf{E}_i = \{a_i(\phi_k, \phi_l)\}$, for $x_k, x_l \in \Lambda_i$, and $\hat{\mathbf{E}}_i = \{a_i(\varphi_k^{(i)}, \varphi_l^{(i)})\}$, for $x_k, x_l \in \overline{\Omega}_{ih}^{CR}$. If $x_k, x_l \in \Omega_{ih}^{CR} \cup \gamma_{m(i)h}^{CR}$, then $a_i(\phi_k, \phi_l) = a_i(\varphi_k^{(i)}, \varphi_l^{(i)})$. If $x_k \in \gamma_{m(j)h}^{CR}$, then the calculation of an element of $\mathbf{E}_i$ involving ϕ_k, requires the values of $Q_m(I_m \varphi_k^{(j)})(x_o)$ at the nodes $x_o \in \delta_{m(i)h}^{CR}$, since by definition $\phi_k = \sum_{x_o \in \delta_{m(i)h}^{CR}} Q_m(I_m \varphi_k^{(j)})(x_o)\varphi_o^{(i)}$ in $\overline{\Omega}_i$. In the following, we derive these coefficients $\{Q_m(I_m \varphi_k^{(j)})(x_o)\}$ from the mortar condition.

For a mortar γ_m, let $\mathbf{I}_m$ be the matrix representation of the interpolation operator $I_m : X_h(\gamma_m) \rightarrow W_{\frac{h}{2}}(\gamma_m)$, whose columns correspond to the nodes γ_{mh}^{CR} (edge midpoints of $\mathcal{T}_h(\gamma_m)$), and the rows correspond to the nodes $\overline{\gamma}_{m\frac{h}{2}}$ (edge endpoints of $\mathcal{T}_{\frac{h}{2}}(\gamma_m)$), along the mortar γ_m.

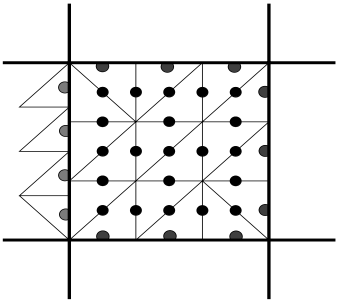


Fig. 3. Showing Ω_i with one nonmortar side, and the corresponding set Λ_i of edge midpoints shown as dots (in the interior) and semi-dots (on the mortars).

We assume that the subdomain Ω_i has only one nonmortar side $\delta_{m(i)}$, cf. Fig. 3; the extension to more than one nonmortar edge is straightforward. Let the master matrix be $\mathbf{M}_{\gamma_{m(j)}} = \{(I_m \varphi_k^{(j)}, \psi_o)_{L^2(\delta_{m(i)})}\}$, and the slave matrix be $\mathbf{S}_{\delta_{m(i)}} = \{(\varphi_l^{(i)}, \psi_o)_{L^2(\delta_{m(i)})}\}$, where $x_k \in \gamma_{m(j)h}^{CR}$ and $x_l, x_o \in \delta_{m(i)h}^{CR}$. Let ξ_n be the basis function of $W_{\frac{h}{2}}(\gamma_{m(j)})$, associated with the edge endpoints $x_n \in \overline{\gamma}_{m(j)\frac{h}{2}}$. Then

$$\mathbf{O}_{m(i)} = \mathbf{S}_{\delta_{m(i)}}^{-1} \mathbf{M}_{\gamma_{m(j)}} = \mathbf{S}_{\delta_{m(i)}}^{-1} \mathbf{N}_{\gamma_{m(j)}} \mathbf{I}_m$$

is the matrix representation of the mortar projection $Q_m I_m : X_h(\gamma_{m(j)}) \rightarrow M^{h_i}(\delta_{m(i)})$, where $\mathbf{N}_{\gamma_{m(j)}} = \{(\xi_n, \psi_o)_{L^2(\delta_{m(i)})}\}$ with $x_n \in \overline{\gamma}_{m(j)\frac{h}{2}}$ and $x_o \in \delta_{m(i)h}^{CR}$. The columns of this matrix $\mathbf{O}_{m(i)}$ correspond to the nodes $x_k \in \gamma_{m(j)h}^{CR}$, containing exactly the coefficients $\{Q_m(I_m \varphi_k^{(j)})(x_o)\}$. We note that $\mathbf{S}_{\delta_{m(i)}}$ is a diagonal matrix containing the lengths of the edges along $\delta_{m(i)}$ as entries.

Now define the matrix $\mathbf{O}_i = diag(\mathbf{I}, \mathbf{O}_{m(i)})$, where $\mathbf{I}$ is the identity matrix corresponding to the nodes of Ω_{ih}^{CR} and $\gamma_{m(i)h}^{CR}$, and $\mathbf{O}_{m(i)}$ is the mortar projection matrix corresponding to the nodes of $\gamma_{m(j)h}^{CR}$. Then it is easy to see that $\mathbf{E}_i = \mathbf{O}_i^T \hat{\mathbf{E}}_i \mathbf{O}_i$. Finally, we have $\mathbf{A} = \sum_{i=1}^{N} \mathbf{P}_i^T \mathbf{O}_i^T \hat{\mathbf{E}}_i \mathbf{O}_i \mathbf{P}_i$. In the same way, we get $\mathbf{f} = \sum_{i=1}^{N} \mathbf{P}_i^T \mathbf{O}_i^T \hat{\mathbf{f}}_i$.

4 An additive Schwarz method

In this section, we design an additive Schwarz method for the problem (4), which is an extension of the algorithm in [7] for the standard mortar case, to the new mortar variant. The method is defined using the general framework for additive Schwarz methods (cf. [9]). We decompose V_h as $V_h = V^{\mathcal{S}} + V^0 + \sum_{i=1}^{N} V^i$. For $i = 1, \ldots, N$, V^i is the restriction of V_h to Ω_i, with functions vanishing at subdomain boundary edge midpoints $\partial\Omega_{ih}^{CR}$ as well as on the remaining subdomains.

$V^{\mathcal{S}}$ is a space of functions given by their values on the skeleton edge midpoints $\mathcal{S}_h^{CR} = \bigcup_{\gamma_m} \gamma_{mh}^{CR}$, i.e. $V^{\mathcal{S}} = \{v \in V_h : v(x) = 0,\ x \in \overline{\Omega}_h^{CR} \setminus \mathcal{S}_h^{CR}\}$. Due to its construction, any two basis functions of this space, those associated with the edge midpoints on the same mortar side or two neighboring mortar sides may have a common support. In case of the two neighboring mortar sides, however, this common support becomes nonexistent if we make sure that the triangulation in each subdomain does not contain any corner triangle. A corner triangle is a triangle having more than one edge on the subdomain boundary. As a result, the corresponding stiffness matrix takes the form of a block diagonal matrix with each block belonging to one mortar side only.

The coarse space V^0, a special space having a dimension equal to the number of subdomains, is defined using the function $\chi_i \in X_h(\Omega_i)$ associated with the subdomain Ω_i. χ_i is defined by its nodal values as: $\chi_i(x) = 1/\sum_j \rho_j(x)$ at $x \in \overline{\Omega}_{ih}^{CR}$, where the sum is taken over the subdomains Ω_j to which x belongs, and $\rho_j = 1$, $\forall j$. Note that the ρ_j's may represent physical parameters with jumps across interfaces, see [7]. V^0 is given as the span of its basis functions, $\Phi_i, i = 1, \ldots, N$, i.e., $V^0 = span\{\Phi_i : i = 1, \ldots, N\}$, where Φ_i associated with Ω_i, is defined as follows.

$$\Phi_i(x) = \begin{cases} 1, & x \in \Omega_{ih}^{CR}, \\ \rho_i \chi_i(x), & x \in \gamma_{m(i)h}^{CR}, \\ \rho_i Q_m(I_m \chi_j)(x), & x \in \delta_{m(i)h}^{CR},\ \delta_{m(i)} = \gamma_{m(j)}, \\ \rho_i Q_m(I_m \chi_i)(x), & x \in \delta_{m(j)h}^{CR},\ \delta_{m(j)} = \gamma_{m(i)}, \\ \rho_i \chi_j(x), & x \in \gamma_{m(j)h}^{CR},\ \gamma_{m(j)} = \delta_{m(i)}, \\ 0, & x \in \partial\Omega_{ih}^{CR} \cap \partial\Omega, \end{cases} \tag{7}$$

and $\Phi_i(x) = 0$ at all other x in $\overline{\Omega}_h^{CR}$. We use exact bilinear forms for all our subproblems. The projection like operators $T^i : V_h \to V^i$ are defined in the standard way, i.e., for $i \in \{\mathcal{S}, 0, \ldots, N\}$ and $u \in V_h$, $T^i u \in V^i$ is the solution of $a_h(T^i u, v) = a_h(u, v),\ v \in V^i$. Let $T = T^{\mathcal{S}} + T^0 + T^1 + \ldots + T^N$. The problem (4) is now replaced by the preconditioned system

$$Tu_h^* = g, \tag{8}$$

where $g = T^{\mathcal{S}} u_h^* + \sum_{i=0}^{N} T^i u_h^*$. Let c and C represent constants independent of the mesh sizes $h = \inf_i h_i$ and $H = \max_i H_i$, then the following theorem holds.

Theorem 2. *For all $u \in V_h$,*

$$c\frac{h}{H}a_h(u,u) \leq a_h(Tu,u) \leq Ca_h(u,u). \tag{9}$$

The theorem can be shown in the same way as the proof in [7], which uses the general theory for Schwarz methods, cf. [9]. It follows from the theorem that the condition number of the operator T grows as $\frac{H}{h}$.

5 Numerical results

For the experiment, we consider our model problem to be defined on a unit square domain, Ω, with the forcing function f chosen so that the exact solution u is equal to $\sin(\pi x)\sin(\pi y)$. The domain Ω is initially divided into $3^2 = 9$ square subdomains (subregions). Each subdomain is then discretized uniformly using, in a checkerboard fashion, either $2m^2$ or $2n^2$ right angle triangles of equal size, where m and n are fixed and $m \neq n$ resulting in nonmatching grids across all interfaces.

Table 1. Condition number estimates (κ_2), PCG-iteration counts (#iter), and L^2-norm (error_{L^2}) and H^1-seminorm (error_{H^1}) of the error in each case.

$\{m,n\}$	Standard CR Mortar				Proposed CR Mortar			
	κ_2	#iter	error_{L^2}	error_{H^1}	κ_2	#iter	error_{L^2}	error_{H^1}
$\{06,05\}$	28.85	25	0.002020	0.065293	30.11	23	0.002484	0.078409
$\{12,10\}$	63.44	35	0.000497	0.032843	60.90	31	0.000667	0.038768
$\{24,20\}$	134.18	49	0.000123	0.016479	122.55	45	0.000175	0.019321

A comparison between the standard and the proposed mortar technique for the Crouzeix-Raviart finite element is shown in Table 1. The Preconditioned Conjugate Gradients (PCG) method has been used to solve the resulting algebraic systems with their respective additive Schwarz preconditioners. As seen from the table, the numerical results agree with the theory. The proposed method exhibits a similar behavior as that of the standard method.

Acknowledgement. We are extremely thankful to the referees of this paper for their valuable comments. We are thankful also to Prof. Petter Bjørstad for presenting this work at the DD16 conference in New York for us.

References

1. F. B. Belgacem, *The mortar element method with Lagrange multipliers.* Université Paul Sabatier, Toulouse, France, 1994.

2. F. B. Belgacem and Y. Maday, *The mortar element method for three dimensional finite elements*, RAIRO Mathematical Modelling and Numerical Analysis, 31 (1997), pp. 289–302.
3. C. Bernardi, Y. Maday, and A. T. Patera, *A New Non Conforming Approach to Domain Decomposition: The Mortar Element Method*, vol. 299 of Pitman Res. Notes Math. Ser., Pitman, 1994, pp. 13–51.
4. S. Bertoluzza and S. Falletta, *The mortar method with approximate constraint*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003, pp. 357–363.
5. D. Braess and W. Dahmen, *The mortar element method revisited—what are the right norms?*, in Thirteenth international conference on domain decomposition, N. Debit, M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, eds., ddm.org, 2001, pp. 27–40.
6. L. Marcinkowski, *The mortar element method with locally nonconforming elements*, BIT Numerical Mathematics, 39 (1999), pp. 716–739.
7. T. Rahman, X. Xu, and R. Hoppe, *Additive Schwarz methods for the Crouzeix-Raviart mortar finite element for elliptic problems with discontinuous coefficients*, Numer. Math., 101 (2005), pp. 551–572.
8. P. Seshaiyer and M. Suri, *Uniform hp convergence results for the mortar finite element method*, Math. Comput., 69 (2000), pp. 521–546.
9. B. F. Smith, P. E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
10. B. I. Wohlmuth, *A mortar finite element method using dual spaces for the Lagrange multiplier*, SIAM J. Numer. Anal., 38 (2000), pp. 989–1012.

Part III

Contributed Presentations

A New Probabilistic Approach to the Domain Decomposition Method

Juan A. Acebrón[1] and Renato Spigler[2]

[1] Departament d'Enginyeria Informàtica i Matemàtiques, Universitat Rovira i Virgili, Av. Països Catalans, 26, 43007 Tarragona, Spain. juan.acebron@urv.cat
[2] Dipartimento di Matematica, Università di "Roma Tre",
Largo S. Leonardo Murialdo 1, 00146 Roma, Italy. spigler@mat.uniroma3.it

Summary. A hybrid numerical scheme based on a probabilistic method along with a classical domain decomposition is proposed for solving numerically linear elliptic boundary-value problems. Full decoupling can be accomplished by computing a few values of the solution inside the domain by Monte Carlo or quasi-Monte Carlo techniques, and interpolating at the nodal points where the solution has been obtained previously. Thus, this method appears to be fault-tolerant as well as suited for time decomposition. Some examples are shown to illustrate performance and scalability.

1 Introduction

Domain decomposition methods are nowadays considered among the most natural ways to exploit parallel architectures in solving boundary-value problems for partial differential equations (PDEs). The main idea consists of decoupling the original problem into several sub-problems, and was proposed originally in the seminal work of H. A. Schwarz in 1870. More precisely, the given domain is divided into a number of subdomains, and the task of the numerical solution on such separate subdomains are then assigned to different processors. However, the computation cannot run independently for each subdomain, because they are coupled together through an internal interface, where the solution is unknown. Therefore, for every computational time step, processors have to exchange data along these interfaces, slowing down the overall performance.

In fact, due to the global character of the PDE, the solution cannot be obtained at a single point inside the domain prior to solving the full problem. Consequently, certain iterations are required across the chosen (or prescribed) interfaces, in order to determine approximate values of the solution sought inside the original domain. There exists two approaches for a domain decomposition depending on whether the domains are overlapping or not overlapping, see [5, 15, 16], e.g. Given the domain, problems are coupled, some additional numerical work is needed, and therefore, it is doubtful whether full scalability can be attained as the number of the subdomains (hence, of the processsors) increases unboundedly.

In order to overcome such a drawback, a new method has recently been proposed [1, 2]. The core of the method is based on combining a certain probabilistic method suited for solving elliptic and parabolic partial differential equations with a classical domain decomposition method, namely a *probabilistic domain decomposition* method (PDD). This approach allows us to obtain the solution in some points, internal to the domain, without first solving the entire problem. In fact, this can be done by means of the probabilistic representation of the solution. The basic idea is to compute only a few values of the solution on certain chosen interfaces, and then interpolate to get continuous approximations. These can be used as boundary values to decouple the problem into sub-problems, see Fig. 1. Each such sub-problem can then be solved independently on a separate processor. Clearly, neither communication among the processors, nor iteration across the interfaces is needed.

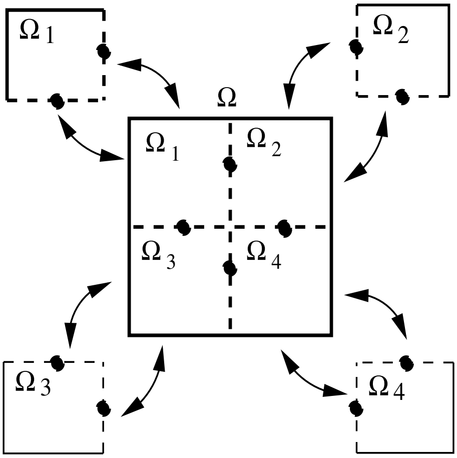


Fig. 1. Sketchy diagram illustrating the numerical method, splitting the initial domain Ω into four subdomains, $\Omega_1, \Omega_2, \Omega_3, \Omega_4$.

Solving boundary-value problems for linear elliptic and parabolic partial differential equations numerically by the probabilistic representation of their solutions, essentially by a Monte Carlo method, is known for a long time, see [4, 11]. It is based on the well-known Feymann-Kac formula, which is an extremely powerful representation, and inherently parallel since it allows for obtaining the solution at single points inside the domain. However, it can hardly be used because requires evaluating accurately the first exit point along with the first exit time from the domain, as well as solving numerically a boundary value problem for a stochastic differential equation. Both issues, however, can be managed reasonably well resorting to several powerful numerical and asymptotic techniques, extracted from probability theory, number theory and statistical physics, see [3, 9, 10].

2 Generalities

For the purpose of illustration, let us confine ourselves to the case of the Dirichlet problem for a linear elliptic equation,

$$Lu - c(\mathbf{x})u = f(\mathbf{x}), \quad \mathbf{x} \in \Omega \subset \mathbf{R}^2, \qquad u|_{\partial\Omega} = g, \tag{1}$$

where $L := \sum_{i,j=1}^{2} a_{ij}(\mathbf{x})\partial_i\partial_j + \sum_{i=1}^{2} b_i(\mathbf{x})\partial_i$ is a linear elliptic operator with smooth coefficients, $c(\mathbf{x}) \geq 0$, with the boundary $\partial\Omega$ of the domain Ω also smooth, as well as the boundary data, g, and the source term, f. The probabilistic representation is given by,

$$u(\mathbf{x}) = E^L_{\mathbf{x}}\left[g(\beta(\tau_{\partial\Omega}))e^{-\int_0^{\tau_{\partial\Omega}} c(\beta(s))\,ds} - \int_0^{\tau_{\partial\Omega}} f(\beta(t))\,e^{-\int_0^t c(\beta(s))\,ds}\,dt\right], \tag{2}$$

see e.g.[7, 12]. Here $\beta(t)$ is the (vector-valued) stochastic process associated to the elliptic operator L, which solves the system of (Ito type) stochastic differential equations (SDEs)

$$d\beta = \mathbf{b}(\mathbf{x})dt + \sigma(\mathbf{x})dW(t). \tag{3}$$

Here $W(t)$ represents the 2-dimensional standard brownian motion (also called Wiener process), and $\tau_{\partial\Omega}$ is the first passage (or hitting) time of the path $\beta(t)$ started at the point $\mathbf{x}$ to $\partial\Omega$. As usual, the dependence of β on the chance variable, running on the underlying probability space, is not displayed. When the operator L is the Laplace operator, Δ, the stochastic process $\beta(t)$ reduces to the standard 2-dimensional brownian motion. The drift vector, $\mathbf{b}$ in (3) is the same one appearing in the operator L, that is $\mathbf{b}(\mathbf{x}) = (b_1, b_2)^T$, while the diffusion matrix, σ, is related to the coefficients a_{ij} by the relation $\sigma\sigma^T = a \equiv (a_{i,j})_{i,j=1,2}$.

The representation formula in (2) is used to obtain a few values of the solution at some points inside the domain Ω. The expected value is approximated by an arithmetic mean (which is known to provide the best estimator) over N realizations of the process β, at the price of a (statistical) error on the order of $N^{-1/2}$. The main problem with using a Monte Carlo method rests on this fact, which entails a rather poor accuracy, unless N is taken extremely large. An alternative, however, does exist, and consists of resorting to sequences of quasi-random numbers [14], which have been used succesfully in mathematical finance, and recently applied in solving stochastic differential equations with high efficiency [3]. Using quasi-random numbers allows for speeding up the calculations in comparison with the classical Monte Carlo method based on pseudorandom numbers, since now the statistical error becomes of order N^{-1}. Such a method is called quasi- Monte Carlo.

Since evaluating the solution via a probabilistic method may require a large number of realizations (large sample size) in order to reduce the associated statistical error, we compute the solution in only a few points along the interfaces between subdomains. Such points will be used as nodal points to interpolate the solution at the interfaces.

Apart from the statistical error, there are however other sources of numerical error which affect the evaluation of $u(\mathbf{x})$ by means of (2), besides that due to the finite sample size mentioned above. These are due to: (i) the truncation error made in the numerical solution of the SDEs in (3); (ii) the uncertainty of estimating first exit times; (*iii*) the numerical quadrature errors in (2). Estimating precisely the first exit times and the first exit points (which are also needed for problems with $c(\mathbf{x})$ and $f(\mathbf{x})$ not identically zero) has often been overlooked in the existing literature. An efficient way to locate accurately the first exit time is based on the use of an exponential timestepping, see [9, 10]. All these sources of error have been analyzed in [1, 2].

At the present time, machines working in the petaflops regime, and endowed with hundred of thousands or even millions of processors are planned for the near future, and taking full advantage of massive parallel computing would be highly desirable. With such machines, the issue of *scalability* remains open, at least in some cases. As was pointed out in [13], Schwarz-type DD methods are not truly scalable, at least in the theoretical sense, since their parallel efficiency in solving elliptic problems is subject to degradation as the number of processors, p, goes to infinity, indeed when p is in the thousands. It seems however that things go better, in practice, for a number of reasons, described in [13].

In addition, failure of even few processors is very likely to occur frequently [8]. Therefore, algorithms which are scalable and *fault-tolerant* at the same time would be extremely important if not mandatory.

The method proposed here seems to be free of the aforementioned drawbacks. In fact, decoupling is complete, and it was shown in [1, 2] that scalability is attained with respect to an arbitrary number of subdomains and processors, and that the algorithm is naturally fault-tolerant. The latter property rests on two ingredients, one due to the intrinsic parallelizability of the Monte Carlo methods, and to the full decoupling that can be realized.

3 Numerical examples

Below we show some examples to illustrate the numerical method proposed. It is worth to stress that, even though the “pivotal” values generated by Monte Carlo or by quasi-Monte Carlo are quite inaccurate (unless an extremely large sample, N, of realizations is used), and the Chebyshev interpolation adds some additional error, the numerical error inside each subdomain is dominated by the boundary errors, due to the maximum principle, and that it decays rapidly inside.

Note that a comparison with a true deterministic DD method has not yet been done; we show only a comparison with “parallel finite differences”. However, we do not expect our algorithm can necessarily outperform any given deterministic DD method, but, rather, that our approach might win over others regarding full scalability and fault-tolerance.

All codes have been implemented using OpenMP, which is a standard parallelization library, designed for shared memory computer architectures. We have simply used a 16 processor IBM Power 3 machine, with 375 MHz clock, and with a peak performance of 24 GFLOPS.

Example 1. We consider the Dirichlet problem [2]

$$\frac{y^2+1}{2}u_{xx} + \frac{x^2+1}{2}u_{yy} + x\,u_x + y^2\,u_y - (x^3+y^2)u = \\ P\cos(2x+y) + Q\sin(2x+y) \qquad \text{in } \Omega = (0,1)\times(0,1), \tag{4}$$

$$P = 1 + x(4+x+2x^2) + 2x\,y + x(4+x)y^2 + y^3, \\ Q = -\frac{1}{2}[-2 + x^4 + 2x^5 + 2x^3\,y + y(5-4y+6y^2) + x^2(1+y+6y^2)] \tag{5}$$

with the boundary data

$$u(x,y)|_{\partial\Omega} = \left[(x^2+y)\sin(2x+y)\right]_{\partial\Omega}, \tag{6}$$

the solution being $u(x,y) = (x^2+y)\sin(2x+y)$.

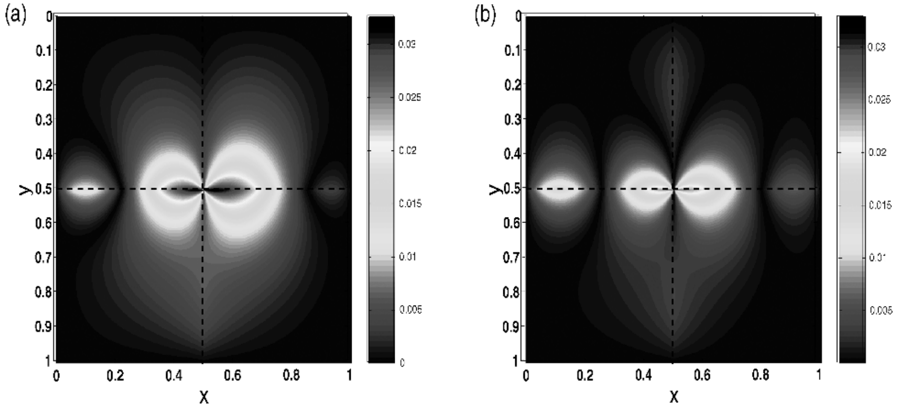


Fig. 2. Example 2. Pointwise numerical error in: (a) the PDD algorithm, and (b) the quasi-PDD algorithm. Parameters are $N = 10^4$, $\Delta x = \Delta y = 2\times 10^{-3}$, $\Delta t = 10^{-3}$.

Table 1. CPU time in seconds for example 1

Processors	PFD	PDD_{Total}	$PDD_{Monte\,Carlo}$	PDD_{FD}
4	9200.107	2087.947	3.492	2084.015
9	4098.381	489.684	3.872	485.484
16	2638.937	175.168	3.365	171.508

In Fig. 2*a* and 2*b*, the pointwise numerical error is shown, for the PDD with pseudorandom sequence of numbers, and quasi-random, respectively. Here only two nodes on each interface have been used. It should be remarked that the quasi-PDD algorithm outperforms the PDD algorithm. The second column (PFD) in Table 1 shows the total computational time (in seconds) spent by the parallel finite difference algorithm using $p = 4$, 9, and 16 processors, which corresponds to 4, 9, and 16 subdomains. The corresponding time spent by the PDD algorithm is shown in the third column. In the last two columns, that quantity is split into two parts, i.e, that required by the Monte Carlo simulation, and that needed by the local solvers. The two methods are compared for approximately the same maximum error, 10^{-3}. In both algorithms the CPU time decreases as p increases, and this trend is more dramatic in the PDD algorithm. Moreover, the CPU time decreases for each given number of processors, passing from PFD to PDD, and this behavior is more pronounced, when the number of processors is longer.

Example 2. Consider the so-called Stommel model, which is a two dimensional model for ocean circulation, and is given by

$$u_{xx} + u_{yy} + \beta\, u_x = -\alpha \sin(\pi y/2) \qquad \text{in } \Omega = (0,1) \times (0,1), \tag{7}$$

with the boundary data $u(x,y)|_{\partial\Omega} = 0$, and $\alpha = 10$, $\beta = 1$. In this example an analytical solution is unknown, and to quantify the numerical error, an accurate numerical solution obtained solving the elliptic equation by a multigrid method has been used instead. As for the previous example, contour plots are shown in Fig. 3 of the numerical error.

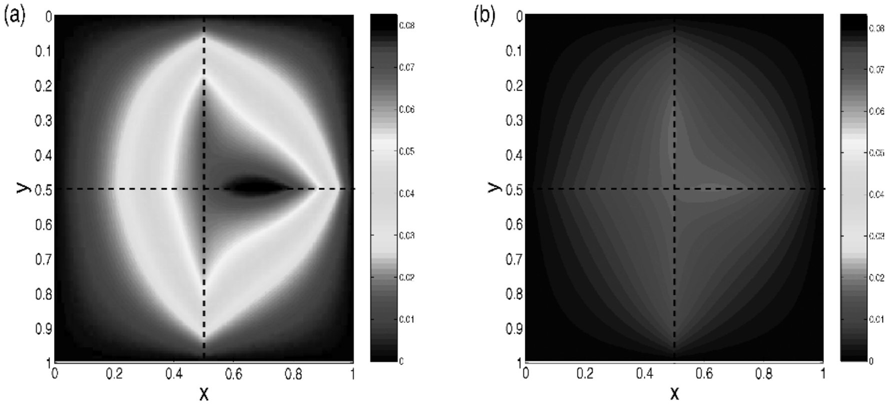


Fig. 3. Example 2. Pointwise numerical error in: (a) the PDD algorithm, and (b) the quasi-PDD algorithm. Parameters are as in Fig. 2

4 Conclusions

A hybrid method based on a probabilistic approach along with a classical domain decomposition for the numerical solution of elliptic partial differential equation, has been described. The associated stochastic differential equation is solved by Monte Carlo simulation only at very few points of a given interface internal to the domain. Then, the solution at the interface is constructed interpolating by using the values at these points. Consequently, a full splitting into several subdomains, to be handled by separate processors acting simultaneously, can be accomplished. Since it has been shown in the literature that the ensuing error may dominate, an essential ingredient of the algorithm consists of a suitable boundary treatment. Moreover, it has been shown that efficiency can be increased by adopting sequences of "quasi-random numbers" (instead of the more customary pseudorandom numbers).

It is worth noticing that a comparison with true deterministic domain decomposition algorithms has not been made yet. A parallel finite differences algorithm allows for an automatic distribution of the computational load among the prescribed subdomains. We do not expect that our code necessarily to be competitive with the existing deterministic codes, but, rather, that it might compete as because of its scalability and fault-tolerance properties. In fact, our approach allows for a complete decoupling among processors, without degrading the overall performance due to strong interprocessors communication. Therefore, it appears to be well suited for grid computing and today's supercomputers with hundreds of thousands of processors or more.

The availability of such a large number of processors in supercomputers, and the desire to put every available processor to work, suggests that we should think about developing new strategies of parallelization, which exploit now the time variable [6]. The method proposed here can be generalized to treat for parabolic partial differential equations, where the time evolution of the solution should now be taken into account. In fact, the probabilistic method can now be used as well to evaluate the solution of a parabolic partial differential equation in any given point and time inside the spatio-temporal domain.

References

1. J. A. Acebrón, M. P. Busico, P. Lanucara, and R. Spigler, *Domain decomposition solution of elliptic boundary-value problems via Monte Carlo and quasi-Monte Carlo methods*, SIAM J. Sci. Comput., 27 (2005), pp. 440–457.
2. ———, *Probabilistically-induced domain decomposition methods for elliptic boundary-value problems*, J. Comput. Phys., 210 (2005), pp. 421–438.
3. J. A. Acebrón and R. Spigler, *Fast simulations of stochastic dynamical systems*, J. Comput. Phys., 208 (2005), pp. 106–115.
4. R. E. Caflisch, *Monte Carlo and quasi Monte Carlo methods*, Acta Numerica, (1998), pp. 1–50.
5. M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci.Comput., 15 (1994), pp. 604–620.
6. 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 (2003), pp. 1397–1434.
7. M. I. Freidlin, *Functional integration and partial differential equations*, vol. 109 of Annals of Mathematics Studies, Princeton University Press, 1985.
8. A. Geist, *Progress towards petascale virtual machines*, in 10th European PVM/MPI User's Group Meeting, Venice, Italy, September/October 2003, Proceedings, J. Dongarra, D. Laforenza, and S. Orlando, eds., vol. 2840 of Lecture Notes in Computer Science, Berlin, 2003, Springer, pp. 10–14.
9. K. M. Jansons and G. D. Lythe, *Efficient numerical solution of stochastic differential equations using exponential timestepping*, J. Statist. Phys., 100 (2000), pp. 1097–1109.

10. ———, *Exponential timestepping with boundary test for stochastic differential equations*, SIAM J. Sci. Comput., 24 (2003), pp. 1809–1822.
11. M. H. Kalos and P. A. Whitlock, *Monte Carlo methods: Volume I: Basics*, John Wiley & Sons, New York, 1986.
12. I. Karatzas and S. E. Shreve, *Brownian motion and stochastic calculus*, vol. 113 of Graduate Texts in Mathematics, Springer, second ed., 1991.
13. D. E. Keyes, *How scalable is domain decomposition in practice?*, in Proceedings of the 11th International Conference on Domain Decomposition Methods, C.-H. Lai, P. E. Bjørstad, M. Cross, and O. B. Widlund, eds., DDM.org, 1999, pp. 286–297.
14. H. Niederreiter, *Random number generation and quasi Monte-Carlo methods*, SIAM, Philadelphia, PA, 1992.
15. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.
16. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

An Adapted Coarse Space for Balancing Domain Decomposition Methods in Nonlinear Elastodynamics

Mikaël Barboteu

University of Perpignan, 52 Avenue Paul-Alduy, 66860 Perpignan, France.
barboteu@univ-perp.fr

This work is devoted to a scalable domain decomposition method to solve nonlinear elastodynamic problems. Large nonlinear elastodynamic problems represent an appropriate application field for substructuring methods which are efficient on parallel computer with the proviso of using specific preconditioner techniques well adapted to the mechanical modeling. Accordingly, we develop an adapted balancing domain decomposition method [4, 6] appropriate for solving this kind of systems. By using the theoretical framework of Schwarz additive decomposition method [6, 7] and by using arguments developed in [1], we propose a two level Neumann-Neumann preconditioner based on the construction of a coarse space of *lower energy* modes adapted to finite deformation problems of a dynamic process.

In section 1, nonlinear elastodynamic problems and the domain decomposition frameworks are recalled. The section 2 is devoted to the definition of an adapted coarse space by using Schwarz additive formulation. The construction of the two level Neumann-Neumann preconditioner is detailed in section 3. In section 4, we test the efficiency of this improved balancing domain decomposition method for the numerical solutions of an academic nonlinear dynamic problem.

1 Nonlinear elastodynamic problems and domain decomposition frameworks

Dynamic deformable body systems in large deformations are governed by nonlinear time dependent equations. A typical nonlinear elastodynamic problem defined in a reference configuration can take the following variational form,

$$\begin{cases} \text{Find } \mathbf{u} \in L^2(]0;T[;U_0) \quad \text{such that for each } t \in]0;T[, \\ \displaystyle\int_\Omega \rho \ddot{\mathbf{u}}(t).\mathbf{v} + \int_\Omega \mathbf{\Pi}(t) : \nabla\mathbf{v} - \int_\Omega \mathbf{f}(t).\mathbf{v} - \int_{\partial_g \Omega} \mathbf{g}(t).\mathbf{v} = 0, \quad \forall \mathbf{v} \in U_0 \end{cases} \tag{1}$$

where ρ denotes the mass density; $\mathbf{\Pi}$ is the first Piola-Kirchoff tensor and $\mathbf{f}$ and $\mathbf{g}$ are the external force densities. A dot superscript indicates the time derivative. The set $U_0 = \{\mathbf{v} \in H^1(\Omega)^{dim}; \mathbf{v} = \mathbf{0} \text{ on } \partial_0\Omega\}$ represents the space of kinematically admissible displacement fields.

The problem (1) can be solved by an energy conservative time integration scheme [3] which is appropriate due to the long term time integration accuracy and stability. In the following, we consider a collection of discrete times $(t_p)_{p=1...P}$ which define a partition of the time interval $[0;T] = \bigcup_{p=1}^{P} [t_p; t_{p+1}]$ with $t_{p+1} = t_p + \Delta t$ and $\Delta t = \dfrac{T}{P}$. By using a second order time integration scheme (adapted midpoint scheme) [3], the weak form (1) integrated between the times t_p and t_{p+1} gives the following system,

$$\begin{cases} \text{Find } \mathbf{u}_{p+1} \in U_0 \quad \text{such that} \\ \dfrac{1}{\Delta t} \displaystyle\int_{\Omega} \rho(\dot{\mathbf{u}}_{p+1} - \dot{\mathbf{u}}_p).\mathbf{v} + \int_{\Omega} \mathbf{\Pi}_{algo} : \nabla \mathbf{v} - \int_{\Omega} \mathbf{f}_{p+\frac{1}{2}}.\mathbf{v} - \int_{\partial_g \Omega} \mathbf{g}_{p+\frac{1}{2}}.\mathbf{v} = 0, \end{cases} \tag{2}$$

where $\Box_{p+\frac{1}{2}} = \dfrac{1}{2}(\Box_p + \Box_{p+1})$ and $\Box_p$ denotes the approximation of $\Box(t_p)$. The energy conservative scheme (2) used in this work, is characterized by the tensor $\mathbf{\Pi}_{algo}$ proposed by Gonzalez [3]. After a fully discretization step (time and space), we obtain the nonlinear system defined by

$$\frac{1}{\Delta t} \mathcal{M}(\dot{\mathbf{u}}_{p+1} - \dot{\mathbf{u}}_p) + \mathcal{G}_{algo}(\mathbf{u}_{p+1}, \mathbf{u}_p) - \mathbf{q}_{p+\frac{1}{2}} = \mathbf{0} \tag{3}$$

where $\mathcal{M}$ comes from the discretization of $\dfrac{1}{\Delta t} \displaystyle\int_{\Omega} \rho(\dot{\mathbf{u}}_{p+1} - \dot{\mathbf{u}}_p).\mathbf{v}$ and $\mathcal{G}_{algo}$ is due to the discretization of the hyperelastic part $\displaystyle\int_{\Omega} \mathbf{\Pi}_{algo} : \nabla \mathbf{v}$ and $\mathbf{q}_{p+\frac{1}{2}}$ comes from the discretization of the external forces $\displaystyle\int_{\Omega} \mathbf{f}.\mathbf{v} + \int_{\partial_g \Omega} \mathbf{g}.\mathbf{v}$. The nonlinear system (3) can be solved by an iterative linearization scheme indexed by i which leads to the solution of linear systems:

$$\mathbf{Ka}_{i,p+1} \Delta \mathbf{u}_{i,p+1} = -\frac{1}{\Delta t} \mathcal{M}(\dot{\mathbf{u}}_{i,p+1} - \dot{\mathbf{u}}_p) - \mathcal{G}_{algo}(\mathbf{u}_{i,p+1}, \mathbf{u}_p) + \mathbf{q}_{p+\frac{1}{2}} \tag{4}$$

$$\text{with} \quad \mathbf{Ka}_{i,p+1} = \frac{2}{\Delta t^2} \mathbf{Ma} + \mathbf{K}_{i,p+1} \quad \text{and} \quad \Delta \mathbf{u}_{i,p+1} = \mathbf{u}_{i+1,p+1} - \mathbf{u}_{i,p+1}$$

where $\mathbf{Ma} = \partial_{\dot{\mathbf{u}}_{p+1}} \mathcal{M}$ represents the mass matrix and $\mathbf{Ka}_{i,p+1} = \partial_{\mathbf{u}_{p+1}} \mathcal{G}_{algo}$ the hyperelastic tangent matrix. We highlight the fact that the matrix $\mathbf{Ka}_{i,p+1}$ of system (4) is **nonsymmetric**; the nonsymmetry comes from the form of tensor $\mathbf{\Pi}_{algo}$ (see [3]).

The linear systems (4) can be solved by a domain decomposition method [6] which has to be adapted to the nonsymmetry but also to the presence of inertia terms. Before giving the adaptations to nonlinear dynamic problems (sections 2 and 3), we present briefly now the principal features of the balancing domain decomposition method [4, 6]. We choose to adopt a primal Schur complement method written in terms of displacement variables. The basic idea in nonoverlapping domain decomposition methods is to split the domain Ω of study into N small nonoverlapping subdomains Ω^n and interfaces Γ^n $(n = 1 \dots N)$. The Schur complement method consist then in reducing the global system to an interface problem by a block Gaussian elimination of the internal degrees of freedom. The interface problem takes the following variational form:

$$\exists \bar{\mathbf{u}} \in \bar{V} \;\; \text{such that} \quad < \mathbf{S}_{i,p+1} \bar{\mathbf{u}}, \bar{\mathbf{v}} > = < \bar{\mathbf{f}}_{i,p+1}, \bar{\mathbf{v}} > \;\; \forall \bar{\mathbf{v}} \in \bar{V} = tr(V)|_{\Gamma}, \tag{5}$$

where V is the discrete set defined from the space U_0 and $\Gamma = \bigcup_{n=1}^{N} \Gamma^n$. The matrices $\mathbf{S}_{i,p+1} = \sum_{n=1}^{N} \mathbf{R}^n \mathbf{S}^n_{i,p+1} (\mathbf{R}^n)^t$ denote the global Schur complement matrices defined on Γ; $(\mathbf{R}^n)^t$ is the restriction operator which goes from Γ to Γ^n. The local Schur complement matrices $\mathbf{S}^n_{i,p+1}$ are defined on Γ^n by

$$\mathbf{S}^n_{i,p+1} = \bar{\mathbf{Ka}}^n_{i,p+1} - (\mathbf{B}^n_{i,p+1})^t (\mathring{\mathbf{Ka}}^n_{i,p+1})^{-1} \mathbf{B}^n_{i,p+1}. \tag{6}$$

To do that, we have considered the subdomain stiffness matrix formulated by $\mathbf{Ka}^n_{i,p+1} = \begin{pmatrix} \mathring{\mathbf{Ka}}^n_{i,p+1} & \mathbf{B}^n_{i,p+1} \\ (\mathbf{B}^n_{i,p+1})^t & \bar{\mathbf{Ka}}^n_{i,p+1} \end{pmatrix}$. The blocks $\mathring{\mathbf{Ka}}^n_{i,p+1}$ and $\bar{\mathbf{Ka}}^n_{i,p+1}$ correspond respectively to the internal and interface degrees of freedom. The matrix $\mathbf{B}^n_{i,p+1}$ represents the contribution connecting Γ^n to Ω^n. The interface problem (5) can be solved by a GMRES method (nonsymmetric cases) with the multilevel Neumann-Neumann preconditioner [6, 1]. This iterative technique requires the formation of the matrix vector products $\mathbf{S}\bar{\mathbf{p}}$ and $\mathbf{M}^{-1}\bar{\mathbf{r}}$ by solving independent auxiliary Dirichlet and Neumann problems on the local subdomains and a global coarse problem defined on a space of singular (rigid body) motions. The adaptation of the balancing method to solving linear systems from nonlinear elastodynamic problems [2] can be realized by using the theoretical framework of Schwarz additive decomposition method introducing an adapted coarse space.

2 Schwarz additive formulation: towards a definition of an adapted coarse space

The two-level Neumann-Neumann preconditioner may be interpreted as an additive Schwarz algorithm [6]. This method consists in decomposing the interface space $\bar{V}$ into a coarse and a fine component: $\bar{V}_G$ and $\bar{V}_f$. The coarse space $\bar{V}_G = \sum_{n=1}^{N} D^n Z^n$ (D^n is a given partition of unity defined on the interface with $\sum_{n=1}^{N} D^n R^n = Id|_{\bar{V}}$) can be defined by adding local low energy components and the fine space $\bar{V}_f$ is defined by duality as follows:

$$\bar{V}_f = \sum_{n=1}^{N} D^n \bar{V}^n_f \quad \text{where } \bar{V}^n_f = \{\bar{\mathbf{v}}^n_f \in \bar{V}^n, \ < \mathbf{S}\mathbf{R}^n \bar{\mathbf{v}}^n, \mathbf{R}^n \mathbf{z}^n >= 0, \ \ \forall \mathbf{z}^n \in Z^n\}. \tag{7}$$

The key point is the construction of the local spaces Z^n of rigid motions. This construction must, if that is necessary, regularize the local Neumann problems but more specifically eliminate the low energy modes (like rigid body motions) in the solution of local Neumann problems. For more details on the presentation of the Schwarz additive method, we can refer to [6, 7] for symmetric cases and [1] for nonsymmetric cases.

With finite deformations and dynamic problems, some low energy modes cannot be detected in the factorization step of the local tangent matrices of Neumann problems [2]. These complications come from the finite deformations modeling but also from the regularizing contribution of the mass matrix. Moreover, we also need to improve the continuity between subdomains by taking into account specific modes (like corners modes). So we have to introduce a specific construction of these lower energy modes.

In the following, we present in detail the construction of the coarse space $\bar{V}_G$ for nonlinear dynamic problems. The orthogonality relation used in (7) characterizing the space $\bar{V}_f^n$ permits us to obtain information in order to define the local coarse space Z^n. Indeed the expansion of this orthogonality relation by using $\mathbf{S}_{i,p+1} = \sum_{n=1}^{N} \mathbf{R}^n \mathbf{S}^n_{i,p+1} (\mathbf{R}^n)^t$ involves terms only from subdomains that are neighbors of the n^{th} and one term from the n^{th} itself,

$$\underbrace{\sum_{l=1}^{neigh(n)} < \bar{\mathbf{v}}^n \ , \ (\mathbf{R}^n)^t \mathbf{R}^l (\mathbf{S}^l_{i,p+1})^t (\mathbf{R}^l)^t \mathbf{R}^n \mathbf{z}^n >}_{(8i)} + \underbrace{< \bar{\mathbf{v}}^n \ , \ (\mathbf{S}^n_{i,p+1})^t \mathbf{z}^n >}_{(8ii)} = 0 \quad (8)$$

The relation (8) can be verified by setting the terms ($8i$) and ($8ii$) to zero. Let us now see what the use of these relations imply:

- use of the term ($8ii$): this term may be eliminated by including the kernel of $(\mathbf{S}^n_{i,p+1})^t$ in the local space Z_n, $(Ker(\mathbf{S}^n_{i,p+1})^t \subset Z^n)$. Such a choice leads to the same simplification as obtained with the kernel of $\mathbf{S}^n$ for the more common symmetric case [6]. For nonsymmetric cases, this choice leads to the introduction of Dual Rigid Modes (DRM) defined through the kernel of $(\mathbf{S}^n_{i,p+1})^t$ (see [1] for details). >From a pratical point of view and according to the form $\mathbf{S}^n_{i,p+1}$ given in (6), the dual rigid modes (noted by $\mathbf{v}^n_{G\alpha}$) defined on Ω^n can be calculated by using the local matrices $(\mathbf{Ka}^n_{i,p+1})^t$ and by using the solution of the following Neumann systems:

$$\mathbf{v}^n_{G\alpha} \in V^n \ \text{ such that } \ (\mathbf{Ka}^n_{i,p+1})^t \mathbf{v}^n_{G\alpha} = \mathbf{0}, \quad \alpha = 1, nbDRM^n. \qquad (9)$$

where $nbDRM^n$ represents the total number of dual rigid modes of subdomains Ω^n. One can easily prove that the modes $\mathbf{v}^n_{G\alpha} \in Ker(\mathbf{Ka}^n_{i,p+1})^t$ are connected to the elements $\mathbf{z}^n \in Ker(\mathbf{S}^n_{i,p+1})^t$ by the relation $\mathbf{z}^n = \bar{\mathbf{v}}^n_{G\alpha}$ (where $\bar{\mathbf{v}}^n_{G\alpha}$ represents the contribution of $\mathbf{v}^n_{G\alpha}$ on Γ^n).

- contribution of the term ($8i$): a simple manner to cancel the term ($8i$) is to fix all the terms of the sum to zero; the elements $\mathbf{z}^n$ of Z^n could then be characterized as the solution of $(\mathbf{R}^n)^t \mathbf{R}^l (\mathbf{S}^l_{i,p+1})^t (\mathbf{R}^l)^t \mathbf{R}^n \mathbf{z}^n = 0$. That makes it possible to ensure the continuity of the coarse space elements through the interface Γ^n of Ω^n by considering the contributions relating to corners, edges and faces of the neighbouring subdomains Ω^l. Indeed, the elements $\mathbf{z}^n$ can be found respectively by these following relations :

$$\mathbf{z}^l \in Z^l \ \text{ such that } \ (\mathbf{S}^l_{i,p+1})^t \mathbf{z}^l = \mathbf{0} \quad \forall \ l = 1, neigh(n) \qquad (10)$$

$$\mathbf{z}^n \in Z^n \ \text{ such that } \ (\mathbf{R}^l)^t \mathbf{R}^n \mathbf{z}^n = \mathbf{z}^l \quad \forall \ l = 1, neigh(n) \qquad (11)$$

The use of the ($8ii$) and (10) makes it possible to calculte the dual rigid modes of the subdomains Ω^n and its neighbors Ω^l; furthermore the relation (11) represents the continuity constraint of dual modes through the interface (corners, edges and faces) between Ω^n and Ω^l. This last point makes it possible to connect this approach with the balancing domain decomposition method by constraints [5]; the enforcement of these kind of constraint leads to expensive computational cost. An inexpensive way, inspired by [7] and [5] would be to impose only the continuity at the corners of the subdomains Ω^n. That can be done by the computation of the $nbDCM^n$ Dual Corner Modes (DCM) of subdomains Ω^n by enforcing the same arbitrary Dirichlet boundary value for the corner interface degrees of freedom for all concerned subdomains Ω^n and Ω^l (where $nbDCM^n$ is the total number of DCM).

In conclusion, the coarse space Z^n can be generated by considering the $nbDRM^n$ dual rigid modes defined by solutions of the systems (12) and particulary by the $nbDCM^n$ dual corner modes given by the systems (13) (see the next section 3 for more details on the computation of these modes).

3 Adaptation of the 2-level Neumann-Neumann preconditionner

According to the definition of the coarse space introduced in section 2, we propose an adaptive construction of the two level Neumann-Neumann preconditioner based on the following steps:

(a) Preliminary step : We identify the local internal degrees of freedom $\{Pr^n_\alpha;\ \alpha = 1, nbDRM^n\}$ which cancel all $nbDRM^n$ rigid motions of subdomain Ω^n. This can be realized during the factorization of the stiffness matrix $(\mathbf{K}^n_e)$ coming from the linear elastostatic system associated to the nonlinear elastodynamic problem (3).

(b) For each Newton iteration i and for each time step p

   a) We construct the local regularized matrices $\widetilde{\mathbf{Ka}}^n_{i,p+1}$ by using the degrees of freedom $\{Pr^n_\alpha\}$ detected in step (1). These matrices can be written by using the contributions from the internal and interface degrees of freedom; then only the internal contribution $\mathring{\mathbf{Ka}}^n_{i,p+1}$ of the matrix $\mathbf{Ka}^n_{i,p+1}$ is regularized by using the matrix $\mathring{\mathbf{Q}}^n_\alpha$:

$$\widetilde{\mathring{\mathbf{Ka}}}^n_{i,p+1} = \mathring{\mathbf{Ka}}^n_{i,p+1} + \mathring{\mathbf{Q}}^n_\alpha \quad \text{where} \quad (\mathring{\mathbf{Q}}^n_\alpha)_{jk} = \begin{cases} BV & \text{if } j = k = Pr^n_\alpha \\ 0 & \text{otherwise} \end{cases}$$

   where BV is an arbitrary big value (like 10^{30} for example). This regularization is not necessary for dynamic problems due to the contribution of the inertia terms $\dfrac{2}{\Delta t^2}\mathbf{Ma}$ which ensures that the matrices $\mathbf{Ka}^n_{i,p+1}$ are non-singular. On the other hand, we need to construct the regularized matrices $\widetilde{\bar{\mathbf{Ka}}}^n_{i,p+1}$ in order to impose the boundary conditions on the corners degrees of freedom noted by $\{Pc^n_\gamma;\ \gamma = 1, nbDCM^n\}$:

$$\widetilde{\bar{\mathbf{Ka}}}^n_{i,p+1} = \bar{\mathbf{Ka}}^n_{i,p+1} + \bar{\mathbf{Q}}^n_\gamma \quad \text{where} \quad (\bar{\mathbf{Q}}^n_\gamma)_{jk} = \begin{cases} BV & \text{if } j = k = Pc^n_\gamma \\ 0 & \text{otherwise} \end{cases}$$

   b) We compute the dual rigid modes $\{\mathbf{v}^n_{G\alpha};\ \alpha = 1, nbDRM^n\}$ by solving the (regularized) local Neumann problems set in the space V^n of subdomain displacements functions,

$$\begin{pmatrix} \widetilde{\mathring{\mathbf{Ka}}}^{n}_{i,p+1} & \mathbf{B}^{n}_{i,p+1} \\ (\mathbf{B}^{n}_{i,p+1})^{t} & \bar{\mathbf{Ka}}^{n}_{i,p+1} \end{pmatrix}^{t} \begin{pmatrix} \mathring{\mathbf{v}}^{n}_{G\alpha} \\ \bar{\mathbf{v}}^{n}_{G\alpha} \end{pmatrix} = \begin{pmatrix} \mathring{\mathbf{e}}^{n}_{\alpha} \\ \mathbf{0} \end{pmatrix}; \ \alpha = 1, nbDRM^{n} \qquad (12)$$

where the j^{th} component $(\mathring{\mathbf{e}}^{n}_{\alpha})_j$ of the vector $\mathring{\mathbf{e}}^{n}_{\alpha}$ is equal to the arbitrary big value BV if $j = Pr^{n}_{\alpha}$ and to the value zero if not.

c) We compute the dual corner modes $\{\mathbf{v}^{n}_{G\gamma};\ \gamma = 1, nbDCM^{n}\}$ by solving local Neumann problems in which the continuity of modes on corners can be realized by enforcing the same arbitrary Dirichlet boundary value (1 for example) on the corners interface degrees of freedom $\{Pc^{n}_{\gamma};\ \gamma = 1, nbDCM^{n}\}$ for all concerned subdomains Ω^{n} :

$$\begin{pmatrix} \mathring{\mathbf{Ka}}^{n}_{i,p+1} & \mathbf{B}^{n}_{i,p+1} \\ (\mathbf{B}^{n}_{i,p+1})^{t} & \widetilde{\bar{\mathbf{Ka}}}^{n}_{i,p+1} \end{pmatrix}^{t} \begin{pmatrix} \mathring{\mathbf{v}}^{n}_{G\gamma} \\ \bar{\mathbf{v}}^{n}_{G\gamma} \end{pmatrix} = \begin{pmatrix} \mathbf{0} \\ \bar{\mathbf{e}}^{n}_{\gamma} \end{pmatrix}; \ \gamma = 1, nbDCM^{n} \qquad (13)$$

where the j^{th} component $(\bar{\mathbf{e}}^{n}_{\gamma})_j$ of the vector $\bar{\mathbf{e}}^{n}_{\gamma}$ is equal to the arbitrary big value BV if $j = Pc^{n}_{\gamma}$ and to the value zero if not.

(c) We define the local coarse space by:

$$Z^{n} = \text{vect}\left(\{\bar{\mathbf{v}}^{n}_{G\alpha};\ \alpha = 1, nbDRM^{n}\},\ \{\bar{\mathbf{v}}^{n}_{G\gamma};\ \gamma = 1, nbDCM^{n}\}\right).$$

With this construction of low energy modes, the two-level Neumann-Neumann preconditioner is classically defined for each time step p and each Newton iteration i by

$$\mathbf{M}^{-1}_{i,p+1} = \mathbf{P}_G + \sum_{n=1}^{N} (\mathbf{I} - \mathbf{P}_G)\mathbf{D}^{n}_{i,p+1}(\widetilde{\mathbf{S}}^{n}_{i,p+1})^{-1}(\mathbf{D}^{n}_{i,p+1})^{t}(\mathbf{I} - \mathbf{P}_G)^{t}, \qquad (14)$$

where $(\widetilde{\mathbf{S}}^{n}_{i,p+1})^{-1}$ is the regularized Schur inverse matrix and $\mathbf{P}_G$ denotes the orthogonal $\mathbf{S}$-projection of $\bar{V}$ on $\bar{V}_G$.

4 A nonlinear dynamic problem: the cantilever beam

In this section, we illustrate numerically the previous adaptations in the case of a two-dimensional nonlinear elastodynamic problem. The application relates to the dynamic evolution of a cantilever beam in plane displacements. To do that, we consider an elastic beam clamped at one of its tips and an external time independent loading g on the opposite tip. The compressible material response considered is governed by an Ogden constitutive law. The mesh and its deformed configurations during the time are presented in figure 1. >From this numerical experiment, we analyse the scalability of the interface solver (GMRES) with some versions of the two-level Neumann-Neumann preconditioners. The considered preconditioners are :
- the nonsymmetric Neumann-Neumann preconditioner given in [1] (curve ▼) without nonlinear dynamic adaptations,
- the nonsymmetric Neumann-Neumann preconditioner given in [2] (curve ■) presented in section 3 but without dual corner modes (step (c)),
- the improved nonsymmetric Neumann-Neumann preconditioner introduced in section 3 (curve ●) with all the features (steps (a), (b) and (c)).

The figure 2 gives the average number of GMRES iterations (per Newton iterations) for a beam decomposed into 2, 5, 10, 20, 40, 80 and 160 subdomains (see

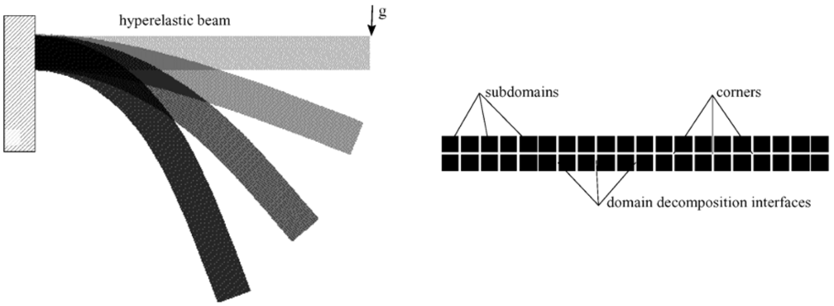
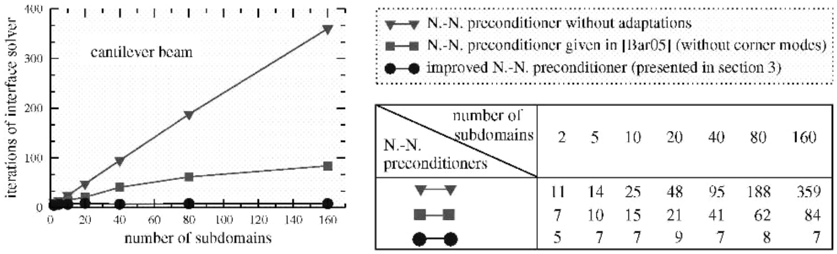


Fig. 1. Deformed sequence of a cantilever beam and substructuration of the beam in 40 subdomains.



N.-N. preconditioners \ number of subdomains	2	5	10	20	40	80	160
▼—▼	11	14	25	48	95	188	359
■—■	7	10	15	21	41	62	84
●—●	5	7	7	9	7	8	7

Fig. 2. Numerical scalability with Neumann-Neumann (N.-N.) preconditioners.

figure 1 for a decomposition into 40 subdomains). We observe that the number of iterations obtained with the 2-level Neumann-Neumann preconditioner without adaptations (curve ▼) grows with the number of subdomains. So the interface solver with this preconditioner loses its classical scalability. We remark that the preconditioner (curve ■) given in [2] (without corners modes) permits a large decrease in the iteration but that the dependance on the number of subdomains is already present. On the other hand, the improved Neumann-Neumann preconditioner (curve ●) leads to a recovery of the numerical scalability properties i.e. the independence of the number of iterations with respect to the number of subdomains. Furthermore, the performance of this preconditioner is practically the same as the ones obtained for linear elastostatic problems. Indeed, the average number of iterations is equal to 7 for a decomposition into 160 subdomains (see table in figure 2) and if we consider the associated linear elastostatic problem the number of iterations is equal to 6. In order to validate these performances, we must test this preconditioner on other less academic simulations.

References

1. P. Alart, M. Barboteu, P. L. Tallec, and M. Vidrascu, *Additive Schwarz method for nonsymmetric problems: application to frictional multicontact problems*, in Thirteenth international conference on domain decomposition, N. Debit,

M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, eds., ddm.org, 2001.

2. M. Barboteu, *Construction du préconditionneur Neumann-Neumann de décomposition de domaine de niveau 2 pour les problèmes élastodynamiques en grandes déformations*, C. R. Acad. Sci., 340 (2005), pp. 171–176.
3. O. Gonzalez, *Exact energy and momentum conserving algorithms for general models in non linear elasticity*, Comput. Meth. Appl. Mech. Engrg., 190 (2000), pp. 1763–1783.
4. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.
5. J. Mandel and C. R. Dohrmann, *Convergence of a balancing domain decomposition by constraints and energy minimization*, Numer. Linear Algebra Appl., 10 (2003), pp. 639–659.
6. P. L. Tallec, *Domain decomposition methods in computational mechanics*, in Computational Mechanics Advances, J. T. Oden, ed., vol. 1 (2), North-Holland, 1994, pp. 121–220.
7. P. L. Tallec, J. Mandel, and M. Vidrascu, *A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems*, SIAM J. Numer. Anal., 35 (1998), pp. 836–867.

On Nonlinear Dirichlet–Neumann Algorithms for Jumping Nonlinearities

Heiko Berninger, Ralf Kornhuber, and Oliver Sander

Freie Universität Berlin, Fachbereich Mathematik und Informatik, Arnimallee 14, D-14195 Berlin, Germany.

Summary. We consider a quasilinear elliptic transmission problem where the nonlinearity changes discontinuously across two subdomains. By a reformulation of the problem via a Kirchhoff transformation, we first obtain linear problems on the subdomains together with nonlinear transmission conditions and then a nonlinear Steklov–Poincaré interface equation. We introduce a Dirichlet–Neumann iteration for this problem and prove convergence to a unique solution in one space dimension. Finally we present numerical results in two space dimensions suggesting that the algorithm can be applied successfully in more general cases.

1 Introduction

Let $\Omega \subset \mathbb{R}^n$ be a bounded Lipschitz domain divided into two non-overlapping subdomains Ω_1, Ω_2 with the interface $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$.

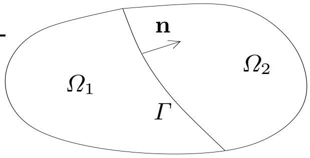


Fig. 1. Non-overlapping partition of the domain Ω.

Given $f \in L^2(\Omega)$, k_1, $k_2 \in L^\infty(\mathbb{R})$ with $k_i \geq \alpha > 0$ for $i = 1, 2$ we consider the following quasilinear elliptic transmission problem:

find a function p in Ω, $p_{|\Omega_i} = p_i \in H^1(\Omega_i)$, $i = 1, 2$, $p_{|\partial\Omega} = 0$, such that

$$-\operatorname{div}(k_i(p_i)\nabla p_i) = f \qquad \text{on } \Omega_i, \;\; i=1,2 \tag{1}$$
$$p_1 = p_2 \qquad \text{on } \Gamma \tag{2}$$
$$k_1(p_1)\nabla p_1 \cdot \mathbf{n} = k_2(p_2)\nabla p_2 \cdot \mathbf{n} \qquad \text{on } \Gamma\,. \tag{3}$$

Observe that the nonlinearities k_i need not be differentiable. Hence, Newton-type linearization suffers from a lack of smoothness. However, by a standard *Kirchhoff transformation* [1], we can reformulate the two nonlinear pdes (1) as linear *Poisson equations* in both subdomains. Based on this observation, we introduce a nonlinear Dirichlet–Neumann algorithm for (1)–(3) which requires the solution of two linear problems in each iteration step but does *not* involve any further linearization. We present a convergence analysis including sufficient conditions for convergence in one space dimension. As a by-product, we obtain an existence and uniqueness result for (1)–(3). Numerical computations suggest that the algorithm can also be applied successfully to higher-dimensional problems. Related Robin methods will be treated in a forthcoming paper (see also [2]).

This paper is organized as follows. In Section 2, we apply the Kirchhoff transformation to (1)–(3) in order to derive an interface problem with linear problems on the subdomains and nonlinear transmission conditions. The transformed problem can be rewritten as a nonlinear Steklov–Poincaré interface equation. In analogy to the linear case, the nonlinear Dirichlet–Neumann algorithm can be regarded as a preconditioned Richardson iteration applied to the nonlinear Steklov–Poincaré equation. In Section 3, we present a convergence theorem in 1D generalizing related results in the linear case [4]. Finally, in Section 4, we illustrate the numerical properties of the nonlinear Dirichlet–Neumann method in a nontrivial two-dimensional setting.

2 An elliptic problem with jumping nonlinearities

We introduce (for $i=1,2$) the spaces

$$V_i := \{v_i \in H^1(\Omega_i) \,|\, v_{i|\partial\Omega\cap\partial\Omega_i} = 0\}, \quad V_i^0 := H_0^1(\Omega_i), \quad \Lambda := H_{00}^{1/2}(\Gamma)$$

and for $w_i, v_i \in V_i$ the forms

$$a_i(w_i,v_i) := (\nabla w_i, \nabla v_i)_{\Omega_i}\,, \qquad b_i(w_i,v_i) := (k(w_i)\nabla w_i, \nabla v_i)_{\Omega_i}\,,$$

where $(\cdot,\cdot)_{\Omega_i}$ stands for the L^2 inner product on Ω_i. The norm in $H^1(\Omega_i)$ will be denoted by $\|\cdot\|_{1,\Omega_i}$, the norm in Λ with $\|\cdot\|_\Lambda$. We point out that much of what we present in the first two sections are generalizations of the linear theory given in [4]. The notation here is used accordingly.

Let R_i, $i=1,2$, be any continuous extension operator from Λ to V_i. Then the variational formulation of problem (1)–(3) reads as follows:

find $p_i \in V_i$, $i=1,2$, such that

$$b_i(p_i,v_i) = (f,v_i)_{\Omega_i} \qquad \forall v_i \in V_i^0, \;\; i=1,2 \tag{4}$$
$$p_{1|\Gamma} = p_{2|\Gamma} \qquad \text{in } \Lambda \tag{5}$$
$$b_1(p_1,R_1\mu) - (f,R_1\mu)_{\Omega_1} = -b_2(p_2,R_2\mu) + (f,R_2\mu)_{\Omega_2} \quad \forall \mu \in \Lambda\,. \tag{6}$$

We now introduce new variables u_i, $i=1,2$, by Kirchhoff transformations κ_i (cf. [1]):

$$u_i(x) := \kappa_i(p_i(x)) = \int_0^{p_i(x)} k_i(q)\,dq \quad \text{a.e. in } \Omega_i\,,$$

which yields $k_i(p_i)\nabla p_i = \nabla u_i$. Further properties of κ_i are listed in the following

Proposition 1. *$\kappa_i : \mathbb{R} \to \mathbb{R}$ is a.e. differentiable with $\kappa_i' = k_i$, strictly monotonically increasing and Lipschitz continuous with Lipschitz constant $\|k_i\|_\infty$. κ_i^{-1} is Lipschitz continuous with Lipschitz constant $\|k_i^{-1}\|_\infty^{-1}$.*
Furthermore there exist positive constants c, C with

$$c\,\|p_i\|_{1,\Omega_i} \le \|\kappa_i(p_i)\|_{1,\Omega_i} \le C\,\|p_i\|_{1,\Omega_i} \quad \textit{and}$$

$$c\,\|p_{i|\Gamma}\|_\Lambda \le \|\kappa_i(p_i)_{|\Gamma}\|_\Lambda \le C\,\|p_{i|\Gamma}\|_\Lambda\,.$$

Thus with this transformation, problem (4)–(6) becomes:
find $u_i \in V_i$, $i = 1, 2$, such that

$$a_i(u_i, v_i) = (f, v_i)_{\Omega_i} \qquad \forall v_i \in V_i^0,\ \ i = 1, 2 \tag{7}$$
$$\kappa_1^{-1}(u_{1|\Gamma}) = \kappa_2^{-1}(u_{2|\Gamma}) \qquad \text{in } \Lambda \tag{8}$$
$$a_1(u_1, R_1\mu) - (f, R_1\mu)_{\Omega_1} = -a_2(u_2, R_2\mu) + (f, R_2\mu)_{\Omega_2} \quad \forall \mu \in \Lambda\,. \tag{9}$$

For a given $\lambda \in \Lambda$, we now consider for $i = 1, 2$ the harmonic extensions $u_i^0 = H_i(\kappa_i(\lambda)) \in V_i$ of the Dirichlet boundary value $\kappa_i(\lambda)$ on Γ. (From now on, the brackets are mostly left out to simplify the notation.) Furthermore let $u_i^* = \mathcal{G}_i f$ be the solutions of the subproblems (7) with homogeneous Dirichlet data $u_{i|\partial\Omega}^0 = 0$. Due to the linearity of the local problems (7), the functions $u_i = H_i\kappa_i\lambda + \mathcal{G}_i f$ satisfy (7)–(9) if and only if

$$\begin{aligned} &a_1(H_1\kappa_1\lambda, R_1\mu) + a_2(H_2\kappa_2\lambda, R_2\mu) = \\ &\qquad (f, R_1\mu)_{\Omega_1} - a_1(\mathcal{G}_1 f, R_1\mu) + (f, R_2\mu)_{\Omega_2} - a_2(\mathcal{G}_2 f, R_2\mu) \quad \forall \mu \in \Lambda\,. \end{aligned} \tag{10}$$

Since the extension operators R_i, $i = 1, 2$, can be chosen arbitrarily, we set $R_i = H_i$. Denoting by $\langle\cdot,\cdot\rangle$ the duality pairing between Λ' and Λ, we recall the definition of the Steklov–Poincaré operators $S_i : \Lambda \to \Lambda'$:

$$\langle S_i\eta, \mu\rangle = a_i(H_i\eta, H_i\mu) \quad \forall \eta, \mu \in \Lambda, \quad i = 1, 2$$

and furthermore the functional $\chi = \chi_1 + \chi_2 \in \Lambda'$:

$$\langle \chi_i, \mu\rangle = (f, H_i\mu)_{\Omega_i} - a_i(\mathcal{G}_i f, H_i\mu) \quad \forall \mu \in \Lambda, \quad i = 1, 2\,.$$

Now (10) can be written as the nonlinear Steklov–Poincaré interface equation

$$\text{find } \lambda \in \Lambda : \qquad (S_1\kappa_1 + S_2\kappa_2)\lambda = \chi \tag{11}$$

or equivalently:

$$\text{find } \lambda_2 \in \Lambda : \qquad (S_1\kappa_1\kappa_2^{-1} + S_2)\lambda_2 = \chi \tag{12}$$

if we set $\lambda_2 = \kappa_2\lambda$. Note that if $\kappa_2^{-1} : \Lambda \to \Lambda$ is Lipschitz continuous the convergence of a sequence of iterates λ_2^k to λ_2 implies the convergence of $\lambda^k = \kappa_2^{-1}\lambda_2^k$ to λ. We state the main result of this section:

Proposition 2. *Solving problem (4)–(6) is equivalent to solving the nonlinear Steklov–Poincaré equations (11) or (12).*

We point out at this stage that the reformulation of the problem (4)–(6) by Kirchhoff transformation is not only a powerful tool for the analysis of the problem that will be subject of the following two sections but also for its numerical treatment due to the linearity of transformed subproblems. In more complicated cases like the time-discretized Richards equation, a Kirchhoff transformation allows a reformulation of the quasilinear subproblems as convex minimization problems which can be solved efficiently and robustly using monotone multigrid methods [3].

3 Nonlinear Dirichlet–Neumann iteration

Now we consider the Dirichlet–Neumann algorithm applied to our problem (4)–(6). However, since it turns out that for a rigorous analysis the damping has to be carried out in the transformed variables, we state it for the transformed version (7)–(9):

Given $\lambda_2^0 \in \Lambda$, successively find $u_1^{k+1} \in V_1$ and $u_2^{k+1} \in V_2$ for each $k \geq 0$ such that

$$a_1(u_1^{k+1}, v_1) = (f, v_1)_{\Omega_1} \qquad \forall v_1 \in V_1^0 \tag{13}$$

$$u_{1|\Gamma}^{k+1} = \kappa_1 \kappa_2^{-1}(\lambda_2^k) \qquad \text{in } \Lambda \tag{14}$$

$$a_2(u_2^{k+1}, v_2) = (f, v_2)_{\Omega_2} \qquad \forall v_2 \in V_2^0 \tag{15}$$

$$a_2(u_2^{k+1}, H_2\mu) - (f, H_2\mu)_{\Omega_2} = -a_1(u_1^{k+1}, H_1\mu) + (f, H_1\mu)_{\Omega_1} \quad \forall \mu \in \Lambda\,. \tag{16}$$

Then, with some damping parameter $\theta \in (0,1)$, the new iterate is

$$\lambda_2^{k+1} := \theta\, u_{2|\Gamma}^{k+1} + (1-\theta)\lambda_2^k\,. \tag{17}$$

Considering the harmonic extensions $H_i u_{i|\Gamma}^{k+1}$ and the solutions $\mathcal{G}_i f$ of the problems (7) with homogeneous boundary data for $i = 1, 2$, the intermediate iterates are obtained by

$$u_1^{k+1} = H_1 \kappa_1 \kappa_2^{-1} \lambda_2^k + \mathcal{G}_1 f \quad \text{and} \quad u_2^{k+1} = H_2 u_{2|\Gamma}^{k+1} + \mathcal{G}_2 f\,.$$

Thus equation (16) provides

$$a_1(H_1\kappa_1\kappa_2^{-1}\lambda_2^k, H_1\mu) + a_2(H_2 u_{2|\Gamma}^{k+1}, H_2\mu) = \sum_{i=1}^{2} (f, H_i\mu)_{\Omega_i} - a_i(\mathcal{G}_i f, H_i\mu) \quad \forall \mu \in \Lambda\,,$$

which is the same as

$$\langle S_2 u_{2|\Gamma}^{k+1}, \mu\rangle = \langle -S_1\kappa_1\kappa_2^{-1}\lambda_2^k + \chi, \mu\rangle \quad \forall \mu \in \Lambda$$

and regarding (17) altogether yields

$$S_2(\lambda_2^{k+1} - \lambda_2^k) = \theta(\chi - (S_1\kappa_1\kappa_2^{-1} + S_2)\lambda_2^k) \quad \text{in } \Lambda\,. \tag{18}$$

Consequently the damped Dirichlet–Neumann algorithm applied to (7)–(9) is a preconditioned Richardson procedure for the nonlinear Steklov–Poincaré formulation (12) with S_2 as a preconditioner.

Note that an analogous formulation for the interface equation (11) cannot be obtained due to the nonlinearity of $S_2\kappa_2$. However, (18) can be treated just as in the linear case if we apply the following generalization of an abstract convergence result in [4, pp. 118/9]. Let X be a Hilbert space, let $Q_1 : X \to X'$ be a not necessarily linear operator and let $Q_2 : X \to X'$ be linear and invertible. With the definition $Q := Q_1 + Q_2$ and for given $G \in X'$, we consider the problem

$$\text{find } \lambda \in X : \quad Q\lambda = G \tag{19}$$

together with the corresponding Richardson iteration

$$\lambda^{k+1} = \lambda^k + \theta(G - Q\lambda^k)\,. \tag{20}$$

Theorem 1. *Let Q_2 be continuous and coercive, i.e. there are positive constants β_2 and α_2 such that*

$$\langle Q_2\eta, \mu\rangle \le \beta_2\|\eta\|_X\|\mu\|_X \;\; \forall \eta, \mu \in X, \quad \langle Q_2\eta, \eta\rangle \ge \alpha_2\|\eta\|_X^2 \;\; \forall \eta \in X\,.$$

Let Q_1 be Lipschitz continuous, i.e. there is a constant $\beta_1 > 0$ such that

$$\langle Q_1\eta - Q_1\mu, \lambda\rangle \le \beta_1\|\eta - \mu\|_X\|\lambda\|_X \quad \forall \eta, \mu, \lambda \in X\,. \tag{21}$$

Suppose there exists a constant $\kappa^* > 0$ such that

$$\langle Q_2(\eta - \mu), Q_2^{-1}(Q\eta - Q\mu)\rangle + \langle Q\eta - Q\mu, \eta - \mu\rangle \ge \kappa^*\|\eta - \mu\|_X^2 \quad \forall \eta, \mu \in X\,. \tag{22}$$

Then (19) has a unique solution $\lambda \in X$. Furthermore for any given $\lambda^0 \in X$ and any $\theta \in (0, \theta_{\max})$ with

$$\theta_{\max} := \frac{\kappa^*\alpha_2^2}{\beta_2(\beta_1 + \beta_2)^2}\,,$$

the sequence given by (20) converges in X to λ.

The proof is an application of Banach's fixed point theorem and can be carried out along the lines of the one given in [4, pp. 118/9], see also [2].

Remark 1. Note that condition (22) reduces to a much simpler expression if Q_2 is symmetric. In the linear case (22) is just the coerciveness of Q_1. In our nonlinear case, (22) states a uniform monotonicity of Q_1 of the form

$$\langle Q_1\eta - Q_1\mu, \eta - \mu\rangle \ge \frac{\kappa^*}{2}\|\eta - \mu\|_X^2 \quad \forall \eta, \mu \in X\,. \tag{23}$$

Now, it is well known that in the particular situation of (12) and (18) both Steklov–Poincaré operators S_1 and S_2 are symmetric, continuous and coercive. Thus in order to apply Theorem 1 to the case $X = \Lambda$, $G = \chi$, $Q_2 = S_2$ and $Q_1 = S_1\kappa_1\kappa_2^{-1}$, we have to make sure that the conditions (21) and (23) are satisfied for $Q_1 = S_1\kappa_1\kappa_2^{-1}$. So we arrive at the following

Theorem 2. *The nonlinear Steklov–Poincaré equation (12) has a unique solution λ_2 in Λ to which the nonlinear Dirichlet–Neumann scheme (13)–(17) converges for sufficiently small θ and any $\lambda_2^0 \in \Lambda$ if the following two conditions are satisfied:*
$\kappa_1\kappa_2^{-1} : \Lambda \to \Lambda$ is Lipschitz continuous, i.e., there is a constant $L(\kappa_1\kappa_2^{-1}) > 0$ such that

$$\|\kappa_1\kappa_2^{-1}\eta - \kappa_1\kappa_2^{-1}\mu\|_\Lambda \le L(\kappa_1\kappa_2^{-1})\|\eta-\mu\|_\Lambda \quad \forall \eta,\mu\in\Lambda\,, \tag{24}$$

and $S_1\kappa_1\kappa_2^{-1} : \Lambda \to \Lambda'$ *is a uniformly monotone operator, i.e. there is a constant* $\alpha_1 > 0$ *such that*

$$\langle S_1(\kappa_1\kappa_2^{-1}\eta - \kappa_1\kappa_2^{-1}\mu), \eta-\mu\rangle \ge \alpha_1\|\eta-\mu\|_\Lambda^2 \quad \forall \eta,\mu\in\Lambda\,. \tag{25}$$

Proposition 3. *The conditions (24) and (25) are satisfied in one space dimension.*

Proof. Let $\Omega_1 = [a,b]$, $\Omega_2 = [b,c]$, with $\Gamma = \{b\}$ and $a < b < c$. Then we have $\Lambda = H_{00}^{1/2}(\Gamma) = H^{1/2}(\Gamma) \cong (\mathbb{R}, |\cdot|)$ and condition (24) follows from Proposition 1.

Let $L(\kappa_1^{-1})$ and $L(\kappa_2)$ be the Lipschitz constants of the real functions κ_1^{-1} and κ_2 according to Proposition 1. In order to prove (25), let $\eta,\mu,\lambda\in\mathbb{R}$. The harmonic extension $H_1(\lambda)$ is the affine function $x \mapsto \dfrac{\lambda}{b-a}x - \dfrac{\lambda}{b-a}a$. As κ_1^{-1} and κ_2 are monotonically increasing, we then have

$$\begin{aligned}
&\langle S_1(\kappa_1\kappa_2^{-1}\eta - \kappa_1\kappa_2^{-1}\mu), \eta-\mu\rangle \\
&\qquad = \int_a^b \nabla H_1(\kappa_1\kappa_2^{-1}\eta - \kappa_1\kappa_2^{-1}\mu)\nabla H_1(\eta-\mu)\,dx \\
&\qquad = \int_a^b \frac{\kappa_1\kappa_2^{-1}\eta - \kappa_1\kappa_2^{-1}\mu}{b-a}\cdot\frac{\eta-\mu}{b-a}\,dx \\
&\qquad = \frac{(\kappa_1\kappa_2^{-1}\eta - \kappa_1\kappa_2^{-1}\mu)(\eta-\mu)}{b-a} \\
&\qquad \ge \frac{1}{(b-a)L(\kappa_1^{-1})L(\kappa_2)}\,|\eta-\mu|^2\,.
\end{aligned}$$

♯

Remark 2. Unfortunately, in higher dimensions condition (25) is violated since $\langle S_1(\kappa_1\kappa_2^{-1}\eta - \kappa_1\kappa_2^{-1}\mu), \eta-\mu\rangle$ can be negative. A counterexample in 2D is easily obtained by considering a harmonic function u with $\dfrac{\partial u}{\partial \mathbf{n}}\cdot u \le c < 0$ on a subset of Γ with positive Hausdorff measure (see [2]).

4 Numerical example

In this section, we apply our nonlinear Dirichlet–Neumann method to a problem in two space dimensions. We consider the transmission problem (1)–(3) on the Yin Yang domain Ω shown in Figure 2. We denote the white subdomain together with the grey circle B_1 by Ω_1 and the grey subdomain with the white circle B_2 by Ω_2.

We select the data

$$f(x) = (-1)^i \quad \text{on } B_i,\ i=1,2, \qquad f(x) = 0 \quad \text{elsewhere}$$

and the nonlinearities

$$k_i(p_i) = \begin{cases} K_h \max\{p_i^{3\lambda_i+2}, c\} & \text{for } p_i \le -1 \\ 1 & \text{for } p_i \ge -1\,. \end{cases}$$

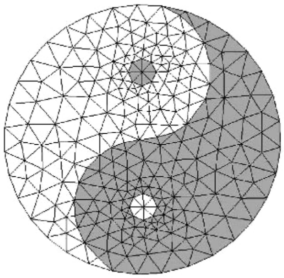


Fig. 2. Yin Yang domain Ω.

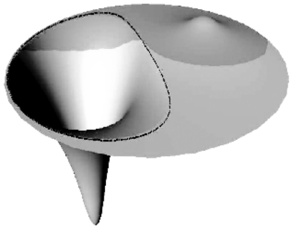


Fig. 3. Solution p on Ω with free boundary (black line).

This choice is motivated by the well-known state equations of Brooks–Corey and Burdine (cf. [5]) for the hydraulic conductivity of a saturated/unsaturated porous media with different soils. In this way, our model problem can be regarded as a stationary Richards equation. Note that $p_i < -1$ characterizes the unsaturated region which is separated by a free boundary from the linear, saturated regime occuring for $p_i \geq -1$. The parameters λ_1 and λ_2 in Ω_1 resp. Ω_2 are called the pore size distribution factors. We choose them in an extreme manner as $\lambda_1 = 1.0$ (very coarse sand) and $\lambda_2 = 0.1$ (fine clay). The factor $K_h = 0.002$ is a realistic hydraulic conductivity in the case of full saturation. The parameter $c = 0.1 > 0$ is introduced to enforce ellipticity.

The choice of the data f which results in a strong sink in B_1 and a strong source in B_2 and our special choice of $\overline{\Omega}_1$ and $\overline{\Omega}_2$ ensure that the free boundary has a nontrivial intersection with the interface $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$ (see the numerical solution as shown in Figure 3). Since we apply the Dirichlet–Neumann scheme (13)–(17), we hereby make sure that step (14) is nonlinear.

We discretize the problem on the two subdomains using piecewise linear finite element spaces. The linear problems on the subdomains are solved by a linear multigrid method. Figure 4 shows the convergence rate ρ measured in the energy norm

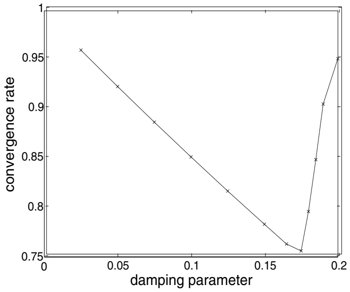


Fig. 4. ρ vs. damping parameter θ.

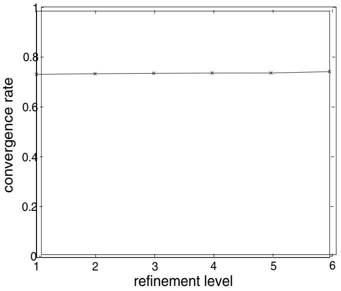


Fig. 5. ρ vs. refinement level.

(for the transformed variables u_i^k) with respect to the damping parameter θ. We use a grid hierarchy of six levels resulting from a uniform mesh refinement of the coarse grid depicted in Figure 2. In this way, we obtain about 235,000 nodes on the finest mesh. Figure 5 shows the convergence rate over the refinement levels. The damping parameter $\theta_{opt} = 0.175$ is obtained from Figure 4. For this optimal choice, we observe mesh-independence of the convergence speed.

References

1. H. W. Alt and S. Luckhaus, *Quasilinear elliptic-parabolic differential equations*, Math. Z., 183 (1983), pp. 311–341.
2. H. Berninger, *Domain Decomposition Methods for Problems with Jumping Nonlinearities*, PhD thesis, Freie Universität Berlin, 2006. In preparation.
3. R. Kornhuber, *On constrained Newton linearization and multigrid for variational inequalities*, Numer. Math., 91 (2002), pp. 699–721.
4. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.
5. M. T. van Genuchten, *A closed-form equation for predicting the hydraulic conductivity of unsaturated soils*, Soil Sci. Soc. Am. J., 44 (1980), pp. 892–898.

Preconditioners for High Order Mortar Methods based on Substructuring

Silvia Bertoluzza[1] and Micol Pennacchio[1]

Istituto di Matematica Applicata e Tecnologie Informatiche - C.N.R., via Ferrata,
1 - 27100 Pavia, Italy. {Silvia.Bertoluzza,Micol.Pennacchio}@imati.cnr.it

Summary. A class of preconditioners for the Mortar Method based on substructuring is studied. We generalize the results of Achdou, Maday and Widlund [1], obtained for the case of order one finite elements, to a wide class of discretization spaces including finite elements of any orders. More precisely, we show that the condition number of the preconditioned matrix grows at most polylogarithmically with the number of degrees of freedom per subdomain.

1 Introduction

We consider the mortar method, a nonconforming version of domain decomposition methods, that allows for different discretizations and/or methods in different subdomains. Consequently, in an adaptive strategy, refinement can be carried out in each subdomain independently and it is possible to use in each subdomain the best suited method.

Here we face the problem of the efficient solution of the linear system arising from such discretization in order to make these techniques more competitive for real life applications. After elimination of the degrees of freedom internal to the subdomains, we need to find the traces of the solution on the subdomain boundaries, i.e. to solve the Schur complement system. The approach that we follow is the substructuring one, proposed by Bramble, Pasciak and Schatz [5] in the framework of conforming domain decomposition. Such an approach was already applied to the Mortar method by Achdou, Maday and Widlund in [1] for the case of order one finite elements. This consists in considering a suitable splitting of the nonconforming discretization space in terms of "edge" and "vertex" degrees of freedom and then using the related block-Jacobi type preconditioners. In this work we generalize the results of [1] to a wide class of discretization spaces (including finite elements of any order) showing that the condition number of the preconditioned matrix grows at most polylogarithmically with the number of degrees of freedom per subdomain, analogously to what happens for the order one case. Finally, we present numerical tests showing the scalability of the method for Q_1 and Q_2 finite elements.

2 The Mortar Method

We first briefly introduce the Mortar method with its main properties (see [2, 6]) and we focus, for simplicity, on the following simple model problem, even if the results of this paper can easily be extended to a more general situation. Let $\Omega \subset \mathrm{R}^2$ be a polygonal domain and $f \in L^2(\Omega)$, then we look for u satisfying

$$-\sum_{i,j=1}^{2} \frac{\partial}{\partial \mathbf{x}_j}\left(a_{ij}(\mathbf{x})\frac{\partial u}{\partial \mathbf{x}_i}\right) = f \text{ in } \Omega, \qquad u = 0 \text{ on } \partial\Omega. \tag{1}$$

The matrix $a(\mathbf{x}) = (a_{ij}(\mathbf{x}))_{i,j=1,2}$ is assumed to be, for almost all $\mathbf{x} \in \Omega$, symmetric positive definite with eigenvalues uniformly bounded from above and from below.

In order to solve (1) we decompose the computational domain Ω as the union of L subdomains Ω_ℓ, $\Omega = \bigcup_{\ell=1,\dots,L} \Omega_\ell$, which, for the sake of simplicity we assume to be quadrilateral (in general the constants in the inequalities will depend on the number of edges of the subdomains as well as on their aspect ratio). We follow the notation of [4]: we set

$$\Gamma_{\ell n} = \partial\Omega_n \cap \partial\Omega_\ell, \qquad S = \cup \Gamma_{\ell n} \tag{2}$$

and we denote by $\gamma_\ell^{(i)}$ $(i = 1, \dots, 4)$ the i-th side of the ℓ-th domain so that $\partial\Omega_\ell = \bigcup_{i=1}^{4} \gamma_\ell^{(i)}$.

Definition 1. *We say that a decomposition is* geometrically conforming *if each edge $\gamma_\ell^{(i)}$ coincides with $\Gamma_{\ell n}$ for some n. If the decomposition is not geometrically conforming, then each interior edge $\gamma_\ell^{(i)}$ will be in general split as the union of several segments $\Gamma_{\ell n}$:*

$$\gamma_\ell^{(i)} = \bigcup_{n \in I_\ell^{(i)}} \Gamma_{\ell n}, \tag{3}$$

where $I_\ell^{(i)} = \{n : \ |\partial\Omega_n \cap \gamma_\ell^{(i)}| \neq 0\}$.

We now consider a non conforming domain decomposition method, based on this splitting of the domain Ω, for the solution of problem (1). First we introduce the corresponding functional setting, hence let

$$X = \prod_\ell \{u_\ell \in H^1(\Omega_\ell) | \ \ u_\ell = 0 \text{ on } \partial\Omega \cap \partial\Omega_\ell\}, \qquad T = \prod_\ell H_*^{1/2}(\partial\Omega_\ell),$$

with $H_*^{1/2}(\Omega_\ell) = H^{1/2}(\partial\Omega_\ell)$ if $\partial\Omega_\ell \cap \partial\Omega = \emptyset$ and $H_*^{1/2}(\partial\Omega_\ell) = \{\eta \in H^{1/2}(\partial\Omega_\ell),\ \eta|_{\partial\Omega_\ell \cap \partial\Omega} \equiv 0\} \sim H_{00}^{1/2}(\partial\Omega_\ell \setminus \partial\Omega)$ otherwise. We denote by $\|\cdot\|_{1/2,\partial\Omega_\ell}$ the related norm, and by $\|\cdot\|_{-1/2,\ell}$ the norm of the corresponding dual space.

On X and T we introduce the following broken norm and semi-norm: $\|u\|_X = \left(\sum_\ell \|u\|^2_{1,\Omega_\ell}\right)^{1/2}$, $|u|_X = \left(\sum_\ell |u|^2_{1,\Omega_\ell}\right)^{1/2}$, $\|\eta\|_T = \left(\sum_\ell \|\eta_\ell\|^2_{1/2,\partial\Omega_\ell}\right)^{1/2}$.

For each ℓ, let $\mathcal{V}_h^\ell$ be a family of finite dimensional subspaces of $H^1(\Omega_\ell) \cap C^0(\bar{\Omega}_\ell)$, depending on a parameter $h = h_\ell > 0$ and satisfying an homogeneous boundary condition on $\partial\Omega \cap \partial\Omega_\ell$.

Let $T_h^\ell = \mathcal{V}_h^\ell|_{\partial\Omega_\ell}$, and, for each edge $\gamma_\ell^{(i)}$ of the subdomain Ω_ℓ,

$$T_{\ell,i} = \{\eta : \eta \text{ is the trace on } \gamma_\ell^{(i)} \text{ of some } u_\ell \in \mathcal{V}_h^\ell\}$$
$$T_{\ell,i}^0 = \{\eta \in T_{\ell,i} : \eta = 0 \text{ at the vertices of } \gamma_\ell^{(i)}\}.$$

We set

$$X_h = \prod_{\ell=1}^L \mathcal{V}_h^\ell \ \subset X, \qquad T_h = \prod_{\ell=1}^L T_h^\ell \ \subset T \tag{4}$$

and we define a composite bilinear form $a_X : X \times X \longrightarrow \mathrm{R}$ as follows:

$$a_X(u,v) = \sum_\ell \int_{\Omega_\ell} \sum_{i,j} a_{ij}(\mathbf{x}) \frac{\partial u_\ell}{\partial \mathbf{x}_i} \frac{\partial v_\ell}{\partial \mathbf{x}_j} \, d\mathbf{x}. \tag{5}$$

The bilinear form a_X is clearly not coercive on X. In order to obtain a well posed problem, we will then consider proper subspaces of X consisting of functions that satisfy a suitable *weak continuity* constraint defined, according to the Mortar method, by choosing a splitting of the skeleton S as the disjoint union of a certain number of subdomain sides $\gamma_\ell^{(i)}$ called "multiplier sides". We denote by $I \subset \{1,\ldots,L\} \times \{1,\ldots,4\}$ the set of indices (l,i) such that $\gamma_l^{(i)}$ is a multiplier side, while $I^* \subset \{1,\cdots,L\} \times \{1,\cdots,4\}$ will denote the index-set corresponding to "trace sides" ("mortars" or "master sides" in the usual terminology).

For each $m = (\ell,i) \in I$ let a finite dimensional multiplier space M_h^m (also depending on the parameter h) on γ_m,

$$M_h^m \subset L^2(\gamma_m), \qquad dim(M_h^m) = dim(T_h^{m,0}),$$

be given. We set:

$$M_h = \{\eta \in H^{-1/2}(S), \ \forall m \in I \ \ \eta|_{\gamma_m} \in M_h^m\} \sim \prod_{m \in \mathrm{i}} M_m.$$

The *constrained* approximation and trace spaces X_h and $\mathcal{T}_h$ are then defined as follows:

$$\begin{aligned} X_h &= \{v_h \in X_h, \int_S [v_h]\lambda \, ds = 0, \forall \lambda \in M_h\} \\ \mathcal{T}_h &= \{\eta \in T_h, \int_S [\eta]\lambda \, ds = 0, \forall \lambda \in M_h\}. \end{aligned} \tag{6}$$

The elements of X_h can be obtained by applying to any element of X_h a correction operator $\mathcal{P}_h : X_h \to X_h$, whose action consists in suitably modifying its argument to impose the constraint; we remark that $\mathcal{P}_h$ is a projector.

Thus we can introduce the following discrete problem:

Problem 1. Find $u_h \in X_h$ such that for all $v_h \in X_h$

$$a_X(u_h, v_h) = \int_\Omega f v_h.$$

It is not difficult to choose the class M_h of multipliers in such a way to guarantee ellipticity uniformly with respect to the mesh-size parameter h and to the number L of subdomains.

Then it can be proved that for all $h > 0$, Problem (1) admits a unique solution u_h which satisfies the following error estimate [4]:

$$\|u - u_h\|_X \lesssim \left(\inf_{v_h \in X_h} \|u - v_h\|_X + \inf_{\lambda \in M_h} \left\| \frac{\partial u}{\partial \nu} - \lambda \right\|_{-1/2,S} \right) \qquad (7)$$

with $\| \cdot \|_{-1/2,S}$ denoting the norm of T', dual of T.

3 Substructuring Preconditioners for the Mortar Element Method

In this section we focus on a class of preconditioners for the linear system arising from the discretization by the Mortar method. We will follow the "substructuring" approach first introduced in [5] and already studied in the case of the Mortar Finite Element method in [1]. The main idea of these preconditioners consists in distinguishing three types of degrees of freedom: *interior* degrees of freedom (corresponding to basis functions vanishing on the skeleton and supported on one sub-domain), *edge* degrees of freedom, and *vertex* degrees of freedom. Consequently we can split the functions $u \in \mathcal{X}_h$ as the sum of three suitably defined components: $u = u^0 + u^E + u^V$. Moreover, when expressed in a basis related to such a splitting, substructuring preconditioners can be written in a block diagonal form.

Let us now examine in details how the splitting is constructed. Given any discrete function $w = (w_\ell)_{\ell=1,\cdots,L} \in X_h$ we can split it in a unique way as the sum of an *interior* function $w^0 \in \mathcal{X}_h^0$ and a discrete lifting, performed subdomainwise of its trace $\eta(w) = (w^\ell|_{\Omega_\ell})_{\ell=1,\cdots,L}$ which by abuse of notation we will denote by $R_h(w)$ (rather than using the heavier notation $R_h(\eta(w))$):

$$w = w^0 + R_h(w), \qquad w^0 \in \mathcal{X}_h^0, \qquad (8)$$

with $R_h(w) = (R_h^\ell(w_\ell))_{\ell=1,\ldots,K}$, $R_h^\ell(w_\ell)$ being the unique element in $\mathcal{V}_h^\ell$ satisfying

$$R_h^\ell(w_\ell) = w_\ell \qquad \text{on } \Gamma_\ell,$$

$$\int_{\Omega_\ell} \sum_{i,j} a(\mathbf{x}) \frac{\partial}{\partial \mathbf{x}_i} \frac{\partial}{\partial \mathbf{x}_j} R_h^\ell(w_\ell) v_h^\ell \, d\mathbf{x} = 0, \quad \forall v_h \in \mathcal{V}_h^\ell \cap H_0^1(\Omega_\ell).$$

Thus the spaces X_h of unconstrained functions and $\mathcal{X}_h$ of constrained functions can be split as direct sums of an interior and of a (respectively unconstrained or constrained) trace component:

$$X_h = X_h^0 \oplus R_h(T_h), \qquad \mathcal{X}_h = \mathcal{X}_h^0 \oplus R_h(\mathcal{T}_h). \qquad (9)$$

We can easily verify that $a_X : X_h \times X_h \to \mathrm{R}$ satisfies

$$a_X(w,v) = a_X(w^0,v^0) + a_X(R_h(w),R_h(v)) := a_X(w^0,v^0) + s(\eta(w),\eta(v)),$$

where the *discrete Steklov-Poincaré* operator $s : T_h \times T_h \to \mathrm{R}$ is defined by

$$s(\xi,\eta) := \sum_{\ell} \int_{\Omega_\ell} (a(\mathbf{x})\nabla R_h^\ell(\xi)) \cdot \nabla R_h^\ell(\eta).$$

We note that the problem of preconditioning the matrix A associated with the discretization of a_X, reduces to finding good preconditioners for the matrices A_0 and S corresponding respectively to the bilinear forms $a_X : \mathcal{X}_h^0 \times \mathcal{X}_h^0 \longrightarrow \mathrm{R}$ and $s : T_h \times T_h \longrightarrow \mathrm{R}$. The matrix A_0 is block diagonal since the coupling between subdomains is taken into account only by the Steklov-Poincaré operator. The blocks of A_0 (which are in fact stiffness matrices corresponding to standard Dirichlet solvers) are widely studied in the literature; here we concentrate only on the preconditioning of the discrete Steklov-Poincaré operator S.

3.1 The splitting of the trace space

The space of constrained skeleton functions $\mathcal{T}_h$ defined in (6) can be further split as the sum of *vertex* and *edge* functions. More specifically, if we denote by $\mathfrak{L} \subset \prod_{\ell=1}^{L} H_*^{1/2}(\partial\Omega_\ell)$ the space

$$\mathfrak{L} = \{(\eta_\ell)_{\ell=1,\cdots,L},\ \eta_\ell \text{ is linear on each edge of } \Omega_\ell\}, \tag{10}$$

then we can define the space of constrained *vertex* functions as

$$\mathcal{T}_h^V = \mathcal{P}_h \mathfrak{L} \tag{11}$$

with $\mathcal{P}_h$ the correction operator imposing the constraint. In the following we will make the (not restrictive) assumption $\mathfrak{L} \subset T_h$, which yields $\mathcal{T}_h^V \subset \mathcal{T}_h$.

We then introduce the space of constrained *edge* functions $\mathcal{T}_h^E \subset \mathcal{T}_h$ defined by

$$\mathcal{T}_h^E = \{\eta = (\eta_\ell)_{\ell=1,\cdots,L} \in \mathcal{T}_h,\ \eta_\ell(A) = 0,\ \forall \text{ vertex } A \text{ of } \Omega_\ell\} \tag{12}$$

and it is quite simple to check that a function in $\mathcal{T}_h^E$ is uniquely defined by its value on trace edges, the value on multiplier edges being forced by the constraint.

Thus, it can be easily verified that

$$\mathcal{T}_h = \mathcal{T}_h^V \oplus \mathcal{T}_h^E \tag{13}$$

and that each $\eta \in \mathcal{T}_h$ can be decomposed in a unique way as

$$\eta = \eta^V + \eta^E, \quad \text{with} \quad \eta^V \in \mathcal{T}_h^V \text{ and } \eta^E \in \mathcal{T}_h^E.$$

The preconditioner

The preconditioner that we consider for S is of block-Jacobi type with blocks related to edges and vertexes. More specifically we can assemble the preconditioner $\hat{s}$ as

$$\begin{aligned}&\hat{s} : \mathcal{T}_h \times \mathcal{T}_h \longrightarrow \mathrm{R}\\ &\hat{s}(\eta, \xi) = b^V(\eta^V, \xi^V) + b^E(\eta^E, \xi^E)\end{aligned} \tag{14}$$

with

$$b^V : \mathcal{T}_h^V \times \mathcal{T}_h^V \longrightarrow \mathrm{R} \qquad \text{such that} \qquad b^V(\eta^V, \eta^V) \simeq s(\eta^V, \eta^V)$$

and

$$b^E : \mathcal{T}_h^E \times \mathcal{T}_h^E \longrightarrow \mathrm{R} \qquad\qquad b^E(\eta, \xi) = \sum_{(\ell,i)\in I^*} b_{\ell,i}(\eta_\ell, \xi_\ell)$$

where for any trace side $\gamma_\ell^{(i)}$, $(\ell, i) \in I^*$, $b_{\ell,i} : T_{\ell,i}^0 \times T_{\ell,i}^0 \longrightarrow \mathrm{R}$ is a symmetric bilinear form satisfying for all $\eta \in T_{\ell,i}^0$

$$b_{\ell,i}(\eta, \eta) \simeq \|\eta\|_{H_{00}^{1/2}(\gamma_\ell^{(i)})}.$$

Denoting by H_l the diameter of Ω_l and writing conventionally $H/h = \max_l \{H_l/h_l\}$ then, under suitable regularity assumptions on the subdomains and on the spaces considered (see [3]), we can prove the following theorem providing bounds for the condition number of the preconditioned matrix.

Theorem 1. *Let S and $\hat{S}$ be the matrices obtained by discretizing respectively s and $\hat{s}$. Then it holds*

$$Cond(\hat{S}^{-1}S) \lesssim \left(1 + \log\left(\frac{H}{h}\right)\right)^4. \tag{15}$$

Moreover, if the decomposition is geometrically conforming then

$$Cond(\hat{S}^{-1}S) \lesssim \left(1 + \log\left(\frac{H}{h}\right)\right)^2. \tag{16}$$

The proof of Theorem 1 follows essentially the pattern of the proofs of the analogous results in [5, 1]; due to space constraint, we do not present it but we refer to [3].

4 Numerical tests

Finally, we have performed numerical experiments to test the scalability of the method for Q_1 and Q_2 finite elements. The model problem is the Poisson equation on the unit square Ω with homogeneous Dirichlet boundary conditions. A uniform, geometrically conforming, decomposition of Ω in $K = N \times N$ equal square subdomains of size $H \times H$ with $H = 1/N$ is considered. In each subdomains Ω_k, a uniform mesh $\mathcal{T}^k$ is built and Q_1, Q_2 finite elements are used in each square.

In order to study the dependence on H (size of the subdomains) and on h (finest meshsize of the finite element spaces), we set $n_k = n$ for all k; hence $h_k = h = H/n$ and $H/h = n$. This corresponds to a non–conforming implementation of the standard domain decomposition method. Then, we tested the preconditioners for several combinations of N and n with n in the range $[5, 40]$ and N in the range $[4, 32]$.

The preconditioned conjugate gradient iteration was stopped when the residual norm had decreased by a factor of 10^{-5} and the experiment were carried out in MATLAB.

Table 1 shows the number of conjugate gradient iterations for reducing the residual of a factor 10^{-5} for Q_1 (left) and Q_2 (right) finite elements respectively. For the edge block of the preconditioner we considered the square root of the stiffness matrix associated to the discretization of the operator $-d^2/dx^2$ by P_1 and P_2 finite elements on each edge with homogeneous Dirichlet boundary conditions at the extrema.

The results are in close agreement with the theory: the condition number of the preconditioned matrix grows at most polylogarithmically with the number of degrees of freedom per subdomain, as indicated by theorem (1).

K= N^2	n=5 # iter.	n=10 # iter.	n=20 # iter.	n=40 # iter.
16	23	25	26	27
64	24	26	27	28
144	24	26	27	29
256	24	26	27	29
400	24	26	27	28
576	24	26	27	28
784	24	26	27	28
1024	23	26	27	28

K= N^2	n=5 # iter.	n=10 # iter.	n=20 # iter.	n=40 # iter.
16	25	25	27	29
64	27	28	30	31
144	27	28	30	31
256	27	28	30	32
400	27	28	30	31
576	27	28	30	31
784	27	28	30	31
1024	27	28	30	31

Table 1. Number of conjugate gradient iterations needed for reducing the residual of a factor 10^{-5}, for different combinations of the number $K = N^2$ of subdomains and n elements per edge (n^2 elements per subdomains) and for Q_1 finite elements (left) and Q_2 finite elements (right).

A complete set of numerical tests showing the scalability of the method for Q_1 and Q_2 finite elements can be found in [3].

Acknowledgement. This work was partially supported by the IHP Project *Breaking Complexity*, contract HPRN-CT-2002-00286.

References

1. Y. Achdou, Y. Maday, and O. B. Widlund, *Iterative substructuring preconditioners for mortar element methods in two dimensions*, SIAM J. Numer. Anal., 36 (1999), pp. 551–580.
2. C. Bernardi, Y. Maday, and A. T. Patera, *A New Non Conforming Approach to Domain Decomposition: The Mortar Element Method*, vol. 299 of Pitman Res. Notes Math. Ser., Pitman, 1994, pp. 13–51.
3. S. Bertoluzza and M. Pennacchio, *Analysis of substructuring preconditioners for mortar methods in an abstract framework*, Appl. Math. Lett., (2006).
4. S. Bertoluzza and V. Perrier, *The mortar method in the wavelet context*, Math. Model. Numer. Anal., 35 (2001), pp. 647–673.
5. J. H. Bramble, J. E. Pasciak, and A. H. Schatz, *The construction of preconditioners for elliptic problems by substructuring, I*, Math. Comp., 47 (1986), pp. 103–134.
6. B. I. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, vol. 17 of Lecture Notes in Computational Science and Engineering, Springer, Berlin, 2001.

Adaptive Smoothed Aggregation in Lattice QCD

James Brannick[1], Marian Brezina[1], David Keyes[3], Oren Livne[5], Irene Livshits[2], Scott MacLachlan[1], Tom Manteuffel[1], Steve McCormick[1], John Ruge[1], and Ludmil Zikatanov[4]

[1] Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO 80309, USA. brannick@newton.colorado.edu, mbrezina@math.cudenver.edu, maclachl@colorado.edu, tmanteuf@colorado.edu, stevem@colorado.edu, jruge@colorado.edu

[2] Department of Mathematical Sciences, Ball State University, Muncie, IN 47306, USA. ilivshits@bsu.edu

[3] Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10027, USA. david.keyes@columbia.edu

[4] Department of Mathematics, The Pennsylvania State University, University Park, PA 16802, USA. ludmil@psu.edu

[5] Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT 84112, USA. livne@sci.utah.edu

Summary. The linear systems arising in lattice quantum chromodynamics (QCD) pose significant challenges for traditional iterative solvers. The Dirac operator associated with these systems is nearly singular, indicating the need for efficient preconditioners. Multilevel preconditioners cannot, however, be easily constructed for these systems becasue the Dirac operator has multiple locally distinct near-kernel components (the so-called slow-to-converge error components of relaxation) that are generally both oscillatory and not known a priori. This paper presents heuristic arguments and numerical results demonstrating that the recently developed adaptive smoothed aggregation (αSA) [2] methodology can be used to overcome the challenges posed by these systems.

1 Introduction

In the field of particle physics, the “Standard” model accounts for the interactions between the elementary particles that make up matter. The Standard model is completely described by two theories: the Electroweak theory for weak interactions, and the widely accepted QCD theory for strong interactions. The interactions between these constituents of matter are well understood for Electroweak theory, where they can be analyzed analytically using perturbation theory. For strong interactions between fermions (quarks), the coupling forces are so strong that a perturbation theory analysis becomes increasingly complex and ultimately breaks down. In the early 1970's, Wilson proposed simulating these strong interactions numerically using Lattice Gauge Theory (LGT), effectively discretizing QCD [4]. LGT is now the primary means for modeling such strong interactions. However, a major obstacle remains: current LGT simulations require enormous computations that become prohibitively expensive for physically interesting choices of parameters (e.g., quark mass and temperature of the physical system), even on today's supercomputers. Hence, the understanding of strong interactions is still very limited.

The majority of the computations in these numerical simulations is dedicated to solving the linear Dirac systems arising from discretization of a coupled system of PDEs on a four-dimensional space-time lattice. This is due to the fact that, in current state-of-the-art simulations, the solvers used for these systems are limited to Krylov methods with preconditioners that are suboptimal for interesting choices of the physical parameters [1, 3]. Developing an appropriate preconditioner for these systems has been a topic of intense research for many years. In the 1990's, various multigrid approaches were explored [1, 3]. More recently, in [6], the use of an alternating Schwarz preconditioner was studied.

This paper considers a simplified 2D Schwinger model exhibiting similar challenges to those of the four-dimensional problem of interest. We explore the use of an adaptive smoothed aggregation [2] iterative solver for these systems. The remaining sections are organized as follows. In §2, we present the 2D Hermitian Dirac-Wilson operator. In §3, we discuss the properties of the 2D operator. Numerical results demonstrating the effectiveness of our approach are given in §4. In §5, we give some concluding remarks.

2 2D Hermitian Dirac-Wilson formulation

Following [7], we describe here the 2D Dirac-Wilson operator, H, and the attendant system of linear equations, $H(\mathbf{u})\mathbf{f} = \mathbf{b}$. The values of the gauge variables, $\mathbf{u}$, are given as $e^{i\theta}$ and, thus, are unitary, complex, and scalar valued. The distribution of the phase angles, θ, depends on the "temperature" of the physical system, prescribed by parameter β. For $\beta \to \infty$, corresponding to a cold temperature of the physical system, the distribution is *smooth* $(\lim_{\beta\to\infty} \mathbf{u} \equiv 1)$. For realistic values of β, between 2 and 6, the system temperature is said to be hot, and the phases are randomly distributed.

The domain of interest is a 2D periodic $N \times N$ uniform lattice (grid), where the lattice sites (gridpoints) are distance $h = 1$ apart. Larger systems are thus obtained by enlarging N, and not by moving lattice points closer together. The fermionic degrees of freedom are defined at the gridpoints on the lattice, while the gauge variables are defined on the lattice edges, as shown in Figure 1. At every gridpoint, x, the unknown function is a vector of length 2:

$$\mathbf{f}(x) = \begin{pmatrix} \mathbf{f}(x, s=1) \\ \mathbf{f}(x, s=2) \end{pmatrix} = \begin{pmatrix} \mathbf{v}(x) \\ \mathbf{w}(x) \end{pmatrix},$$

where $s = 1, 2$ are spin indices, with spin corresponding to angular momentum.

Let $\hat{m}$ be the unit vector in coordinate directions $m = 1, 2$ and $\mathbf{u}(x, m)$ be the gauge variable located at the link associated with gridpoints x and $x + \hat{m}$. Then the action of H is defined in terms of the covariant difference operators,

$$(\nabla_m^+ \mathbf{f})(x, s) = \mathbf{u}(x, m)\mathbf{f}(x + \hat{m}, s) - \mathbf{f}(x, s) \tag{1}$$

$$(\nabla_m^- \mathbf{f})(x, s) = \mathbf{f}(x, s) - \mathbf{u}(x - \hat{m}, m)^* \mathbf{f}(x - \hat{m}, s), \tag{2}$$

and the Pauli matrices,

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}, \quad \sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix}, \quad \sigma_3 = -i\sigma_1\sigma_2 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

The action of the Pauli matrices are denoted by $\sigma_m \mathbf{f}$. Explicitly,

$$(\sigma_m \mathbf{f})(x, s) = \sum_{s'} (\sigma_m)_{s,s'} \mathbf{f}(x, s').$$

With these definitions, H is defined implicitly by its action on $\mathbf{f}(x, s)$ at lattice site x:

$$(H\mathbf{f})(x, s) = \sigma_3 \Bigg[\sum_{m=1}^{2} \frac{1}{2h} \sigma_m (\nabla_m^+ + \nabla_m^-) \mathbf{f}(x, s) + \frac{1}{2h} \sum_{m=1}^{2} (-\nabla_m^+ + \nabla_m^-) \mathbf{f}(x, s) + \rho \mathbf{f}(x, s) \Bigg], \tag{3}$$

where ρ is the relative quark mass and $h = 1$; h is included here to emphasize the comparison to familiar matrices from PDEs, and is used later in §3 to scale the matrix. Note that, due to the σ_3 term in (3), H is Hermitian and indefinite.

The corresponding system of linear equations can be written in the following two-by-two block form:

$$\begin{pmatrix} -\rho I - A & B \\ B^* & \rho I + A \end{pmatrix} \begin{pmatrix} \mathbf{v} \\ \mathbf{w} \end{pmatrix} = \begin{pmatrix} \mathbf{b}_1 \\ \mathbf{b}_2 \end{pmatrix}.$$

Operators $A, B \in \mathbb{C}^{n \times n}$, $n = N \times N$, are defined by their actions on the components of $\mathbf{f}$, corresponding to spin indices $s = 1, 2$. For example, using the local numbering in Figure 1, the actions of A and B on $\mathbf{v}$ are given as follows:

$$(A\mathbf{v})_0 = \frac{1}{2h}(\overline{\mathbf{u}}_- \mathbf{v}_- + \mathbf{u}_+ \mathbf{v}_+ + \overline{\mathbf{u}}^- \mathbf{v}^- + \mathbf{u}^+ \mathbf{v}^+) - \frac{2}{h}\mathbf{v}_0, \tag{4}$$

$$(B\mathbf{v})_0 = \frac{1}{2h}(\mathbf{u}_+ \mathbf{v}_+ - \overline{\mathbf{u}}_- \mathbf{v}_- - i\mathbf{u}^+ \mathbf{v}^+ + i\overline{\mathbf{u}}^- \mathbf{v}^-). \tag{5}$$

The right side, $\mathbf{b}$, is called the *fermionic source vector* and is equal to one at a given gridpoint x and is zero elsewhere.

3 Spectral properties of the Dirac system

In this section, we analyze the properties of the 2D Hermitian Dirac-Wilson operator, H, for different values of the relative quark mass, ρ, and temperature parameter, β.

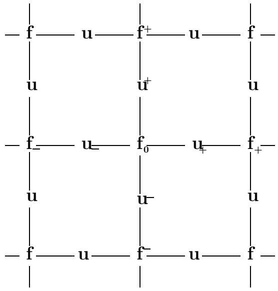


Fig. 1. Local numbering of the unknowns, $\mathbf{f}$, and gauge field coefficients, $\mathbf{u}$, used in the definition of the operators A and B.

In particular, we consider the dependence of the conditioning of H on ρ, recalling that, as ρ approaches its physical value, H becomes nearly singular.

We begin with an analysis of H in the absence of an external field (i.e., $\mathbf{u} \equiv 1$), referred to as the "free" case. For $\mathbf{u} \equiv 1$, the covariant difference operators defined in (1) and (2) reduce to the standard first-order forward and backward difference operators. It is easy to see that, for $\mathbf{u} \equiv 1$, A and B in (4) and (5) have zero-row-sum. Hence, H is singular for $\rho = 0$, and H becomes nearly singular for $0 < \rho \ll 1$, the physical value of ρ for $\mathbf{u} \equiv 1$.

Recall that H is indefinite. Considering a (preconditioned) conjugate gradient algorithm, the equivalent system of normal equations,

$$H^* H \mathbf{f} = H^* \mathbf{b}, \tag{6}$$

can be solved instead. Given an eigenpair, $(\lambda, \mathbf{x})$, of H, we have $H^* H \mathbf{x} = \lambda^2 \mathbf{x}$, so the normal form also becomes increasingly ill-conditioned as $\rho \rightarrow 0$, since the maximum eigenvalue remains $O(1)$ for all ρ. Clearly, CG is inefficient as a stand-alone solver for this system. However, classical multigrid provides a suitable preconditioner in this "free" case, as can be easily seen by relating H^2 to a decoupled two-by-two system of partial differential equations (PDEs) as follows.

Consider the equivalent system, where the problem is rescaled such that $h = \dfrac{1}{N-1} \rightarrow 0$ on a fixed domain. For $\mathbf{u} \equiv 1$, $A = \dfrac{h}{2} \Delta_h$ and $B^* B = B B^* = -\Delta_h$, where Δ_h denotes the five-point discrete Laplacian obtained using second-order centered differences. Thus, $H^* H$ has the following two-by-two block diagonal form:

$$H^*H = \begin{pmatrix} -\rho I - A & B \\ B^* & \rho I + A \end{pmatrix}^2 = \begin{pmatrix} (\rho I + \frac{h}{2}\Delta_h)^2 - \Delta_h & 0 \\ 0 & (\rho I + \frac{h}{2}\Delta_h)^2 - \Delta_h \end{pmatrix}.$$

Denoting the diagonal blocks by C, for $\rho = 0$ we have $C = -\Delta_h(I - \frac{h^2}{4}\triangle_h)$. Since $\sigma(-\Delta_h) \subseteq [0, \frac{8}{h^2}]$, we have that $\sigma(I - \frac{h^2}{4}\Delta_h) \subseteq [1, 3]$, implying C is spectrally equivalent to $-\Delta_h$. The theory of equivalent preconditioning [5] then suggests that preconditioning C with a standard multigrid method for the Laplacian would be efficient. We note that, in practice, if we apply an AMG-preconditioned CG, we observe good solver performance for physical values of the quark mass (i.e., $0 < \rho \ll 1$) as well.

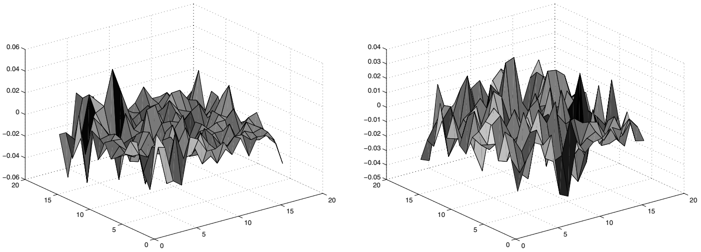


Fig. 2. Real (left) and complex (right) algebraically smooth error of H^2 for $m_q = .05, \beta = 3$ and $N = 16$. This error was computed using 200 Gauss-Seidel iterations on $H^2\mathbf{f} = \mathbf{0}$ with a random initial guess for $\mathbf{f}$.

We now focus on the more challenging case with the presence of the gauge field, when H^2 is no longer a block diagonal system. Unfortunately, in the presence of an external field, where $\mathbf{u}$ is random (e.g., $\beta \in [2, 6]$), H^2 does not appear to be related to any standard system of PDEs. However, similar to the free case, H is indefinite and H^2 becomes ill conditioned as ρ approaches its critical value. More precisely, take $\rho = 0$ for H defined in (3) and let $\hat{H} = \sigma_3 H$. $\hat{H}$ is then non-Hermitian and has eigenvalues with positive *real part*. The critical value of the quark mass, ρ_{cr}, is then defined by $\rho_{cr} = \min_i |\mathbb{R}(\lambda_i(\hat{H}))|$. For physical values of ρ, the *mass gap*, $m_q := \rho - \rho_{cr}$, tends to zero, and H^2 becomes near-singular. This is the primary reason that all existing local algorithms grow in computational complexity for the Dirac system as the relative quark mass approaches its physical value, a phenomenon referred to as *critical slowing down.*

An additional difficulty, not encountered in the free case, is that the near-kernel components of H and, hence, H^*H, are locally oscillatory. This is demonstrated in Figure 2, where plots of the real and imaginary parts of a near-kernel component, computed using 200 iterations of Gauss-Seidel on the homogeneous problem, are given. Our experiments indicate that this oscillatory local character of the near-kernel components is dependent on the distribution of the gauge field, which is itself randomly specified.

To successfully solve the Dirac system for the random case, it is imperative that our iterative solver be able to efficiently attenuate such error components. Standard geometric and algebraic multigrid methods typically construct coarse-level corrections based on the assumption that the error not effectively reduced by the multigrid relaxation procedure is locally constant or, in general, smooth in the geometric sense, and would thus not be immediately suitable as a solver for this random case.

Smoothed aggregation multigrid (SA) [8] was designed to allow efficient attenuation of error in a subspace characterized locally by a given set of error components, regardless of whether these are smooth or oscillatory in nature. The Dirac system poses an additional difficulty for the iterative solver, in that an a priori knowledge of these near-kernel components is not available. For this reason, we use the recently developed adaptive version of the smoothed aggregation multigrid method (αSA, [2]), which allows its setup procedure to identify the requisite error components and modify the method to ensure they can be efficiently eliminated. The αSA setup procedure is a multilevel scheme based on the power method for the error propagation operator of the method itself. In the interest of brevity, we refer for details of the method to [2].

4 Numerical results

This present section reports on numerical results obtained by applying various solvers to the 2D Dirac system defined in §2. We solve the equivalent normal system of equations, (6), reformulated as a two-by-two block *real* system,

$$\begin{bmatrix} X & -Y \\ Y & X \end{bmatrix} \begin{bmatrix} \mathbf{x} \\ \mathbf{y} \end{bmatrix} = \begin{bmatrix} \mathbf{c}_1 \\ \mathbf{c}_2 \end{bmatrix}, \tag{7}$$

where X, Y are real-valued matrices that satisfy $H^*H = X + iY$, $\mathbf{f} = \mathbf{x} + i\mathbf{y}$, and $H^*\mathbf{b} = \mathbf{c}_1 + i\mathbf{c}_2$.

Results of our numerical experiments are given in Table 1. The experiments were carried out on the problem with $\mathbf{b} = \mathbf{0}$ and random initial guesses for $[\mathbf{x}, \mathbf{y}]^T$. For the αSA preconditioner, we used 3 level $V(2,2)$-cycles with an SOR-type relaxation. We note that, due to the more aggressive coarsening used in smoothed aggregation multigrid, $V(2{,}2)$ cycles are commonly used even when solving the Poisson equation. For the reported results, eight near-kernel components were computed in the adaptive setup and used to define the transfer operators of the αSA preconditioner. In the αSA preconditioned conjugate gradient (CG) results, one iteration of the resulting SA cycle was used as a preconditioner in the CG iteration. The Krylov solver used for comparison in Table 1 was a standard diagonally preconditioned CG iteration.

In current state-of-the-art QCD simulations, typical problem sizes are $N = 32$ and 64. As already mentioned, the asymptotic convergence factors of existing solvers applied to the Dirac system quickly tend towards one for critical choices of mass and temperature. This is demonstrated in Table 1 for diagonally preconditioned CG. Further, the numerical results in Table 1 suggest that when using eight computed near-kernel components in defining the SA interpolation operators, the asymptotic

	.01	.05	.1	.3
2	.25 / .98	.27 /.96	.24 /.91	.22 /.83
3	.29 /.98	.27 /.94	.26 /.92	.27 /.84
5	.28 /.96	.29 /.95	.26 /.92	.25 /.81

	.01	.05	.1	.3
2	.33 /.99	.31 /.96	.31 /.94	.31 /.85
3	.42 /.98	.42 /.97	.40 /.93	.31 /.86
5	.31 /.99	.31 /.96	.29 /.92	.28 /.83

Table 1. Average convergence factors for αSA preconditioned CG and diagonally preconditioned CG applied to (7) with $N = 32$ (left) and $N = 64$ (right), for various choices of the mass gap, m_q, and several values of the temperature parameter, β.

convergence factor of αSA preconditioned CG remains uniformly bounded away from one for $\beta \in [2.0, 6.0]$ and $\rho \rightarrow \rho_{cr}$. This is the main result of this paper, and the first such result to date.

To obtain a more complete picture of the overall effectiveness of our multigrid iteration, we examine also *operator complexity*, defined as the number of nonzero entries stored in the operators on all levels divided by the number of nonzero entries in the finest-level matrix. The operator complexity can be viewed as indicating how expensive the entire V-cycle is compared to performing only the finest-level relaxations of the V-cycle. Even though we use eight near-kernel components to define prolongation, the operator complexity in our experiments stayed bounded by 3.0. These low values result from the fact that the problem size is aggressively reduced in forming the smoothed aggregation coarse problems.

One drawback of using the adaptive version of SA is the nontrivial cost associated with identifying the error components on which we base the transfer operators in SA. Of course, the hope is that this cost is optimal in that it is proportional to the number of degrees of freedom in the problem. Our experiments suggest that this is so. Further, a QCD simulation requires solving the system of equations for many right sides, whereas the adaptive SA setup is performed only once, with the resulting method used to solve the system with many right sides. For the results reported here, even with the cost of the adaptive setup, the resulting SA method is more efficient than a diagonally preconditioned CG algorithm when solving with a small number ($O(1)$) of right sides. Indeed, for the experiments considered in Table 1, only four right sides need be solved to justify the cost of the adaptive setup for the smallest value of mass gap ($m_q = .01$), i.e., the most ill conditioned system. For example, with $\beta = 3$ and $m_q = .01$, the CPU time required for the adaptive setup was 13.7 seconds and the CPU time needed to reduce the relative residual by a factor of 10^5 for a single right side, using αSA preconditioned CG, was 0.8 seconds. Solving the same system using diagonally preconditioned CG required 4.7 sec CPU time.

5 Conclusions

Our experiments demonstrate that αSA provides an efficient preconditioner for the 2D lattice QCD problems considered. We note that the cost of each iteration critically depends on the ratio of the number of degrees of freedom on the fine level to that on the coarser level. The coarsening for our 2D problem was very aggressive, leading to acceptable operator complexities even with eight adaptively computed near-kernel components. It remains to be verified whether these favorable 2D results carry over to the full 4D case.

Acknowledgement. The authors thank Rich Brower and Claudio Rebbi of the Boston University Physics Department for their many insightful comments on this work and for supplying the gauge variable data used in the numerical tests. This work was sponsored by the Department of Energy under grant numbers DE-FC02-01ER25479, DE-FC02-04ER25595 and DE-FG03-99ER25217, Lawrence Livermore National Laboratory under contract number B533502, Sandia National Laboratories under contract number 15268, and the National Science Foundation under grant number DMS-0410318.

References

1. A. Brandt, *Multigrid methods in lattice field computations*, Nuclear Phys. B Proc. Suppl., 26 (1992), pp. 137–180.
2. M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA)*, SIAM J. Sci. Comput., 25 (2004), pp. 1896–1920.
3. R. C. Brower, R. G. Edwards, C. Rebbi, and E. Vicari, *Projective multigrid for Wilson fermions*, Nucl. Phys., B366 (1991), pp. 689–709.
4. M. Creutz, *Quarks, Gluons and Lattices*, Cambridge Univ. Press, Cambridge, 1982.
5. V. Faber, T. A. Manteuffel, and S. V. Parter, *On the theory of equivalent operators and application to the numerical solution of uniformly elliptic partial differential operators*, Adv. in Appl. Math., 11 (1989), pp. 109–163.
6. M. Lüscher, *Lattice QCD and the Schwarz alternating procedure*, tech. rep., CERN, Theory Division, 2004.
7. C. Rebbi, *hwilson2d, Code description*, tech. rep., Boston University, 2003.
8. P. Vaněk, J. Mandel, and M. Brezina, *Algebraic multigrid by smooth aggregation for second and fourth order elliptic problems*, Computing, 56 (1996), pp. 179–196.

Spectral Element Agglomerate AMGe *

Timothy Chartier[1], Robert Falgout[2], Van Emden Henson[2], Jim E. Jones[4], Tom A. Manteuffel[3], Steve F. McCormick[3], John W. Ruge[3], and Panayot S. Vassilevski[2]

[1] Department of Mathematics, Davidson College, P.O. Box 6908, Davidson, NC 28035, USA. tichartier@davidson.edu

[2] Center for Applied Scientific Computing, UC Lawrence Livermore National Laboratory, P.O. Box 808, L-561, Livermore, CA 94551, USA. {rfalgout,vhenson,panayot}@llnl.gov

[3] Applied Math Department, Campus Box 526, University of Colorado at Boulder, Boulder, CO 80309-0526, USA. {tmanteuf, stevem, jruge}@colorado.edu

[4] Department of Mathematics, Florida Institute of Technology, 150 West University Blvd., Melbourne, FL, 32901, USA. jim@fit.edu

1 Introduction

In recent years, several extensions of the classical AMG method (see [2] and [10]) to handle more general finite element matrices have been proposed (see, e.g., [3], [9], and [7]). Other extensions are related to the so–called smoothed aggregation method; see e.g., [11] and the papers cited therein. For the most recent versions of both the AMG and smoothed aggregation approaches, we refer to [5] and [4]. In this note, under the assumption that one has access to the fine–grid element matrices, we combine the effectiveness of element interpolation given in [3] with a "spectral" approach to selecting coarse degrees of freedom, as proposed in [7]. The method presented here selects the coarse degrees of freedom from the eigenvectors in the lower parts of the spectra of certain small matrices– special Schur complements of assembled neighborhood matrices. These Schur complements are associated with so–called minimal intersection sets, which in turn are derived from the partitioning provided by an algorithm (e.g., from [9]) that creates agglomerated elements. The idea of selecting coarse degrees of freedom from the eigenvectors of small matrices has been used previously in connection with certain aggregation methods; see, e.g., [8] and the report [6].

2 The spectral way of selecting coarse degrees of freedom

Assume we are given the fine-grid symmetric positive (semi-) definite matrix A and have access, for a given set of finite-elements $\{\tau\}$, to the symmetric semi-definite

*This work was performed in part under the auspices of the U. S. Department of Energy by University of California Lawrence Livermore National Laboratory under contract W-7405-Eng-48.

fine-grid element matrices A_τ. Here, we consider each element τ to be a subset (list) of fine–grid degrees of freedom, or *dofs*. We let $\mathcal{D}$ denote the set of fine dofs, and identify $\mathcal{D}$ with the index set $\{1, 2, \ldots, n\}$. For any vector $\mathbf{w}$, we use the notation $\mathbf{w}_\tau$ to denote the restriction of $\mathbf{w}$ to the subset τ of $\mathcal{D}$. Armed with these notational conventions, we note that the original matrix A is assembled from the element matrices in the usual way, that is, for any vector $\mathbf{v}$, one has

$$\mathbf{v}^T A \mathbf{v} = \sum_\tau \mathbf{v}_\tau^T A_\tau \mathbf{v}_\tau.$$

Let $\mathbf{V} = V(\mathcal{D})$ denote the vector space (or the space of discrete functions) of vectors with indices from $\mathcal{D}$; that is, one can identify $\mathbf{V}$ with the vector space $\mathbf{R}^n$. Based on an agglomeration algorithm (e.g., as originally proposed in [9]; see also [12]) one generates a set of agglomerated elements $\{T\}$ from the fine–grid elements $\{\tau\}$. Every agglomerated element T consists of a number of connected fine–grid elements and every fine–grid element τ belongs to exactly one agglomerated element T.

Note that every agglomerated element T can also be considered as a set of fine degrees of freedom, namely, as the union of the fine degrees of freedom that belong to the fine–grid elements τ that form T.

One can partition the fine degrees of freedom (dofs) into non–overlapping sets $\{\mathcal{I}\}$, based on the relationships between agglomerated elements and dofs. This relationship is described by the incidence matrix $\mathcal{E}$, defined as

$$\mathcal{E}_{ij} = \begin{cases} 1, \text{ if dof } j \text{ is in the agglomerated element } i, \\ 0, \text{ otherwise.} \end{cases}$$

Note that $\mathcal{E} \in \mathbf{R}^{n_E \times n}$, where n_E is the number of agglomerated elements and n is the number of fine dofs. Consider next $\mathcal{Z} = \mathcal{E}^T \mathcal{E} \in \mathbf{R}^{n \times n}$, and observe that $\mathcal{Z}_{ij} = |\{T : \; i \in T \text{ and } j \in T\}|$, where $|\cdot|$ indicates cardinality. That is, $\mathcal{Z}_{ij}$ equals the number of agglomerated elements containing both dofs i and j. We then split the set of dofs $\{1, \ldots, n\}$ into non-overlapping sets $\{\mathcal{I}_k\}_{k=1}^{\ell}$ (called minimal intersection sets) with the property that i and j are in one and the same set $\mathcal{I}_k$ if and only if $\mathcal{Z}_{ij} = \mathcal{Z}_{ii} = \mathcal{Z}_{jj}$.

With the minimal intersection sets $\mathcal{I}$, one is able to define a change of basis from "nodal" to "spectral" dofs through the following process. For every minimal intersection set $\mathcal{I}$, define the neighborhood $\mathcal{N}(\mathcal{I}) = \cup \tau_\mathcal{I}$, where the union consists of all fine elements $\tau_\mathcal{I}$ that share a dof from $\mathcal{I}$. Let the assembled local matrix be denoted $A_{\mathcal{N}(\mathcal{I})}$ and compute its Schur complement $S_\mathcal{I}$ by eliminating the dofs outside $\mathcal{I}$. (Note that, in the case where $\mathcal{I}$ is a single dof and $A_{\mathcal{N}(\mathcal{I})}$ is a semidefinite matrix, this Schur complement may be the zero matrix). Next, we compute all the eigenvalues of $S_\mathcal{I}$ and the associated eigenvectors $\{\mathbf{q}_{\mathcal{I};\,k}\}$, $k = 1, \ldots, |\mathcal{I}|$. Whenever necessary, we extend the eigenvectors by zero outside $\mathcal{I}$. If $S_\mathcal{I} = [0]$ we use the standard unit vectors $\mathbf{q}_{\mathcal{I};\,k} := \mathbf{e}_i$ for $i \in \mathcal{I}$. We may observe that the set $\{\mathbf{q}_{\mathcal{I};\,k}\}$, $k = 1, \ldots, |\mathcal{I}|$, for $\mathcal{I}$ running over all the minimal intersection sets, provides an orthogonal basis of $\mathbf{R}^n$. For a given minimal intersection set $\mathcal{I}$, the group of vectors $\{\mathbf{q}_{\mathcal{I};k}\}$ are orthogonal (as eigenvectors of symmetric matrices) and if two vectors belong to groups for different sets $\mathcal{I}$, they have non-intersecting supports; therefore, they are also orthogonal.

For every set $\mathcal{I}$, we split the eigenvectors into two groups, $\mathbf{V}_{\mathcal{I}_c}$ and $\mathbf{V}_{\mathcal{I}_f}$, in the following way. Let the eigenvectors $\{\mathbf{q}_{\mathcal{I};\,k}\}$ be ordered according to the eigenvalues $0 \leq \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_{|\mathcal{I}|}$. For any given integer $1 \leq p_\mathcal{I} \leq |\mathcal{I}|$, the first $p_\mathcal{I}$

eigenvectors define the orthogonal basis of the space $\mathbf{V}_{\mathcal{I}_c}$. This space, representing the set of "c"–dofs (or coarse dofs), corresponds to a lower portion of the spectrum of the Schur complement $S_{\mathcal{I}}$. The remaining eigenvectors span $\mathbf{V}_{\mathcal{I}_f}$. Together, they form an orthogonal decomposition of the space spanned by the eigenvectors for $S_{\mathcal{I}}$ and, taking all such spaces for all $\mathcal{I}$, we observe that

$$\mathbf{R}^n = \mathbf{V} = \mathbf{V}_{\mathcal{I}_c} \bigoplus_{\mathcal{I}} \mathbf{V}_{\mathcal{I}_f}.$$

We denote the direct sums, over all $\mathcal{I}$, of the coarse and fine spaces, respectively, as $\mathbf{V}_c$ and $\mathbf{V}_f$. The motivation for this splitting is given next.

Lemma 1. *The block A_{ff}, representing the restriction of A to the subspace $\mathbf{V}_f$, is well–conditioned if $p_{\mathcal{I}}$ for all $\mathcal{I}$ are sufficiently large, in particular, if $\lambda_{p_{\mathcal{I}}+1}[S_{\mathcal{I}}] > 0$ and $\lambda_{p_{\mathcal{I}}+1}[S_{\mathcal{I}}] \simeq \|A_T\|$ for all neighboring T that contain $\mathcal{I}$.*

Proof. Define the vector norm $\|.\|$ as $\|\mathbf{w}\| = \sqrt{\mathbf{w}^T\mathbf{w}}$. Let $\mathbf{v}_f \in \mathbf{V}_f$, that is, $\mathbf{v} = \mathbf{v}_f$ is a vector with vanishing coarse–grid component. We can split the inner product $\mathbf{v}^T A\mathbf{v}$ over the agglomerates T, using the local matrices A_T, and then rewrite the sum $\sum\limits_T \mathbf{v}_T^T A_T \mathbf{v}_T = \sum\limits_{\mathcal{I}} \sum\limits_{T:\, \mathcal{I}\subset T} C_{\mathcal{I}} \mathbf{v}_T^T A_T \mathbf{v}_T$ for some constants $C_{\mathcal{I}}$ (depending on the number of agglomerates T that share $\mathcal{I}$). Using well known minimization properties of Schur complements of symmetric positive (semi–)definite matrices, one readily obtains the inequalities

$$\min_{\mathcal{I}} C_{\mathcal{I}} \lambda_{p_{\mathcal{I}}+1}[S_{\mathcal{I}}]\ \|\mathbf{v}_f\|^2 \ \le\ \mathbf{v}_f^T A_{ff} \mathbf{v}_f \ \le\ \max_T \|A_T\| \|\mathbf{v}_f\|^2. \quad \text{QED} \qquad (1)$$

♯

Remark 1. The $C_{\mathcal{I}}$ are topological constants (i.e., independent of the matrix). However, we have the option to choose the integers $p_{\mathcal{I}}$ sufficiently large to lead to an improved minimal eigenvalue of A_{ff}. Hence, by selecting the $p_{\mathcal{I}}$ appropriately, we can insure that A_{ff} is well–conditioned. We may also observe that, for the model case where A is the discretization of the 2D finite element Laplacian, both bounds in (1) are mesh independent. The property that A_{ff} is well conditioned gives rise to a special form of the so–called compatible relaxation principle introduced in [1].

In the new (orthogonal) basis, the matrix A has a block, A_{ff}, that is well–conditioned. With the coarse dofs identified, the interpolation matrix P can be computed locally by building, for every agglomerated element T, a prolongator P_T such that fine dofs that are shared by two or more agglomerated elements are interpolated by the coarse dofs from that common set. More specifically, for every dof i, consider its neighborhood $N(i) = \cap\{T :\ i \in T\}$. Then i is interpolated from all coarse dofs that belong to $N(i)$. Further details about this construction are found in [9].

It can be proven, in the manner given in [9], that the locally constructed P satisfies a weak approximation property. That is, for some constant $\eta \ge 1$,

$$\|A\| \|\mathbf{v} - P\mathbf{v}_c\|^2 \le \eta\, \mathbf{v}^T A\mathbf{v} \ \text{ for any } \mathbf{v} = \mathbf{v}_f + \mathbf{v}_c,$$

where $\mathbf{v}_f \in \mathbf{V}_f,\ \mathbf{v}_c \in \mathbf{V}_c$. This implies (see e.g., [3]) an optimal convergence result for a two–grid method based on P and simple Richardson smoothing.

3 Numerical Experiments

We describe here results of numerical experiments designed to illustrate the use of the spectral agglomerate AMGe algorithm. We first stress that, in 2D, the minimal intersection sets are the vertices of the agglomerated elements (all of which naturally become coarse dofs), the interior of the faces of the agglomerated elements (or AEfaces), as well as the interior of the agglomerated elements (or AEs). For either of these two minimal intersection sets, we can form the appropriate Schur complement, S_{AEf} or S_{AE}, and compute the associated eigenvalues and eigenvectors.

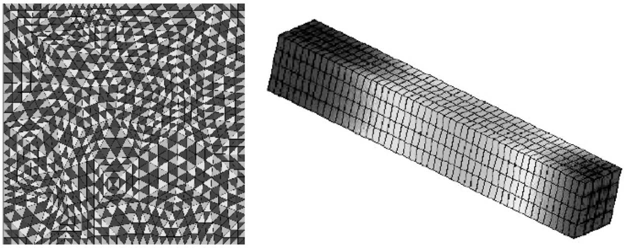


Fig. 1.

Unstructured triangular mesh: 1600 elements (left) Tri-linear hexahedral mesh for 3D elasticity problem (right).

For the sets of eigenvectors of S_{AEf} or S_{AE}, we select an eigenvector with index k to define a coarse dof if, given a tolerance $\tau \in [0, 1)$, the corresponding eigenvalue λ_k satisfies

$$\lambda_k < \tau\lambda\mathrm{max}.$$

We have the option to use different values of τ for the sets of eigenvectors; one, denoted τ_{AEf}, is used on the eigenvectors of S_{AEf}, while another one (denoted by τ_{AE}) is applied to select coarse dofs from the eigenvectors of S_{AE}.

To save some computation in the setup phase, one may choose to set $\tau_{AE} = 0$. (Note that when $\tau = 0$ for both sets, the resulting spectral agglomerate AMGe reduces to the agglomeration based AMGe method from [9]; that is, the coarse dofs are the vertices of the agglomerates only.) In general, we recommend choosing $\tau_{AE} < \tau_{AEf}$.

In 3D, there are additional minimal intersection sets: subsets of the boundary of the faces of the agglomerated elements. We also select these sets as coarse dofs. For thin body elasticity (the particular 3D application we consider in this section), choosing these additional sets as coarse dofs is acceptable in terms of computational complexity.

The problems utilized for this study are:

(1) a **2D anisotropic diffusion** problem, given by

$$-\mathrm{div}((\varepsilon I + \underline{bb}^T)\nabla u) = f,$$

posed in a unit square. Three different selections of the parameter ε are considered, $\varepsilon = 1,\ 0.01,\ 0.001$, which control the strengths of the anisotropy in each experiment.

Table 1. Convergence results for AMGe and the spectral agglomerate AMGe with $\tau_{AE} = \frac{1}{4}\tau_{AEf} = \frac{1}{4}\tau$; 2D anisotropic diffusion.

6400 elements		AMGe	Spectral agglomerate AMGe			
10 levels		$(\tau = 0)$	$\tau = 0.03125$	$\tau = 0.125$	$\tau = 0.25$	$\tau = 0.5$
$\varepsilon = 1$	iterations	12	10	9	8	6
	ρ	0.405	0.432	0.396	0.313	0.176
	grid complexity	1.64	2.77	2.79	2.86	3.13
	operator complexity	1.86	4.73	4.81	5.13	6.83
$\varepsilon = 0.01$	iterations	25	16	11	9	7
	ρ	0.614	0.511	0.378	0.306	0.185
	grid complexity	1.64	2.86	3.11	3.34	3.76
	operator complexity	1.86	5.09	6.31	7.99	14.01
$\varepsilon = 0.001$	iterations	36	17	15	13	13
	ρ	0.721	0.517	0.428	0.388	0.365
	grid complexity	1.64	3.09	3.31	3.50	3.87
	operator complexity	1.86	6.11	7.46	9.32	16.18

The anisotropy is not grid aligned, and its direction is controlled by $\underline{b} = \begin{bmatrix} \cos\theta \\ \sin\theta \end{bmatrix}$, $\theta = \frac{\pi}{4}$. For this 2D problem, results computed on unstructured triangular meshes with 6400 and 25600 elements are presented. A mesh typical of the problem, with 1600 elements, is shown in Fig. 1 (left).

(2) a **3D thin body elasticity** problem posed on a domain $\Omega = (0,1) \times (0,1) \times (0,d)$ and discretized on a number of uniform trilinear hexahedral meshes. The problem is formulated as follows: for a given vector function $\mathbf{f} = (f_i)_{i=1}^3$, find the displacements $\mathbf{u} = (u_i)_{i=1}^3$ such that, for all $\mathbf{x} = (x_i)_{i=1}^3 \in \Omega$

$$\sum_{j=1}^{3} \frac{\partial}{\partial x_j} \left(\sum_{k,l=1}^{3} E_{i,j,k,l} \frac{\partial u_k(\mathbf{x})}{\partial x_l} \right) = f_i(\mathbf{x}),$$

for $i = 1, 2, 3$. Homogeneous boundary conditions are imposed: A Dirichlet condition of $u_i = 0$, $i = 1, 2, 3$ is imposed on the side of the body $\Gamma_D = \{(x, y, z = 0)\}$, and, on the remainder of the boundary $\partial\Omega \setminus \Gamma_D$, we apply a Neumann condition

$$\sum_{j=1}^{3} n_j \left(\sum_{k,l=1}^{3} E_{i,j,k,l} \frac{\partial u_k(\mathbf{x})}{\partial x_l} \right) = 0$$

Table 2. Convergence results for AMGe and the spectral agglomerate AMGe with $\tau_{AE} = \frac{1}{4}\tau_{AEf} = \frac{1}{4}\tau$; 2D anisotropic diffusion.

25600 elements		AMGe	Spectral agglomerate AMGe			
12 levels		$(\tau = 0)$	$\tau = 0.03125$	$\tau = 0.0625$	$\tau = 0.25$	$\tau = 0.5$
$\varepsilon = 1$	iterations	15	22	21	10	6
	ρ	0.517	0.673	0.649	0.429	0.154
	grid complexity	1.62	2.81	2.82	2.92	3.21
	operator complexity	1.86	5.01	5.04	5.62	8.41
$\varepsilon = 0.01$	iterations	32	30	23	12	8
	ρ	0.715	0.747	0.673	0.431	0.202
	grid complexity	1.62	2.90	3.01	3.42	3.89
	operator complexity	1.86	5.43	5.96	9.60	21.92
$\varepsilon = 0.001$	iterations	57	36	28	22	20
	ρ	0.834	0.761	0.698	0.604	0.549
	grid complexity	1.62	3.13	3.25	3.62	4.04
	operator complexity	1.86	6.58	7.36	12.25	28.38

Table 3. Convergence results for spectral agglomerate AMGe with $\tau_{AE} = \frac{1}{4}\tau_{AEf}$; 3D thin body elasticity.

400 elements	Spectral agglomerate AMGe				
$d = 25h$	$\tau = 0.03125$	$\tau = 0.0625$	$\tau = 0.125$	$\tau = 0.25$	$\tau = 0.5$
iterations	29	28	28	15	7
ρ	0.748	0.738	0.727	0.553	0.295
grid complexity	2.20	2.22	2.28	2.50	3.28
operator complexity	2.84	2.90	3.04	3.63	6.94
coarsening levels	7	7	7	7	7

for $i = 1, 2, 3$. Here, $\mathbf{n} = (n_i)_{i=1}^3$ is the outward unit normal to $\partial\Omega$. The coefficients $E_{i,j,k,l}$ are expressed in terms of the Lame coefficients $\lambda = 113$ and $\mu = 81$ as follows:

$$\begin{aligned} E_{1,1,1,1} = E_{2,2,2,2} = E_{3,3,3,3} &= 2\mu + \lambda, \\ E_{i,i,j,j} &= \lambda, \quad i \neq j,\ i, j = 1, 2, 3, \\ E_{i,j,i,j} = E_{j,i,i,j} = E_{i,j,j,i} = E_{j,i,j,i} &= \mu, \quad i \neq j,\ i, j = 1, 2, 3, \\ E_{i,j,k,l} &= 0, \quad \text{all remaining indices.} \end{aligned}$$

We describe results for this problem using several different geometries for the body. Using the basic mesh size $h = 0.25$, we observe results for four different choices of d, namely $d = 25h = 6.25, d = 50h = 12.5, d = 100h = 25$ and $d = 200h = 50$. The geometry for this problem is shown in Fig. 1 (right).

For all experiments, we apply one iteration of symmetric block Gauss–Seidel as the smoother in a V(1,1)–cycle. The blocks in the smoother correspond to the elements of the grid at the given level and hence are overlapping.

Illustrative results of the experiments are given in four tables in this section. For each experiment, we report several quantities: the number of iterations needed to reduce the ℓ^2–norm of the initial residual by 10^{-9}; the asymptotic reduction factor ρ; the grid and operator complexities; and, for the 3D case, the number of coarsening levels used in the problem. The grid and operator complexities, commonly used in AMG, are defined respectively as the total number of dofs on all levels divided by the number of dofs on the finest level and the total number of nonzero entries in the operator matrices for all levels divided by the number of nonzero entries in the fine level matrix A. The tables also indicate the mesh sizes and values for parameters of the algorithm (such as τ).

Table 4. Convergence results for spectral agglomerate AMGe with $\tau_{AE} = \dfrac{1}{4}\tau_{AEf}$; 3D thin body elasticity.

1600 elements $d = 100h$	Spectral agglomerate AMGe $\tau = 0.03125$	 $\tau = 0.0625$	 $\tau = 0.125$	 $\tau = 0.25$	 $\tau = 0.5$
iterations	39	32	28	15	7
ρ	0.860	0.788	0.739	0.555	0.264
grid complexity	2.09	2.12	2.18	2.41	3.25
operator complexity	2.53	2.60	2.73	3.36	6.81
coarsening levels	9	9	9	9	9

The numerical results generally agree with the observations in Remark 1; it is clear that richer coarse spaces produce better convergence factors. Naturally, this gain is obtained at the expense of higher complexities of the method. The spectral agglomerate AMGe method can become a fairly expensive method; it requires a number of local computations: assembling of local neighborhood matrices, computing their respective Schur complements S_{AE} and S_{AEf}, and solving local eigenproblems associated with them. In addition, all the normal costs of the traditional AMG–type methods applies; namely, computing the respective interpolation matrices and the associated coarse–level stiffness matrices. The local dense matrices grow in size with the tolerance τ. This cost is especially noticeable in 3D problems, where the increased complexity leads to significant increases in the time required to solve the problem. steps (e.g., between 5 and 20) The operator complexities can be reduced by using more aggressive agglomeration at the initial level(s).

4 Conclusions

This note describes an algorithm resulting from uniting two ideas introduced and applied elsewhere. For many problems, AMG has always been hampered by complexities whose natures are difficult to discern from the entries of matrix A alone. Element–based interpolation has been effective for some of these problems, but requires access to element matrices on all levels. One way to obtain these has been to perform element agglomeration to form coarse elements, but defining the coarse dofs is often not easy. The spectral approach to coarse dof selection is very attractive due to its elegance and simplicity. The algorithm presented here combines the robustness of element interpolation, ease of coarsening by element agglomeration, and simplicity of defining coarse dofs through the spectral approach. As demonstrated in the numerical results, the method yields a reasonable solver for the problems described. It can, however, be an expensive method due to the number and cost of the local, small dense linear algebra problems; making it a generally competitive method remains an area for further research.

References

1. A. Brandt, *Generally highly accurate algebraic coarsening*, Electron. Trans. Numer. Anal., 10 (2000), pp. 1–20.
2. A. Brandt, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid (AMG) for sparse matrix equations*, in Sparsity and Its Applications, D. J. Evans, ed., Cambridge University Press, Cambridge, UK, 1984, pp. 257–284.
3. M. Brezina, A. J. Cleary, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, and J. W. Ruge, *Algebraic multigrid based on element interpolation (AMGe)*, SIAM J. Sci. Comput., 22 (2000), pp. 1570–1592.
4. M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (αSA)*, SIAM J. Sci. Comput., 25 (2004), pp. 1896–1920.
5. ———, *Adaptive algebraic multigrid methods*, SIAM J. Sci. Comput., 27 (2006), pp. 1261–1286.

6. M. Brezina, C. Heberton, J. Mandel, and P. Vaněk, *An iterative method with convergence rate chosen a priori*, Tech. Rep. 140, Center for Computational Mathematics, University of Colorado at Denver, March 1999.
7. T. Chartier, R. D. Falgout, V. E. Henson, J. E. Jones, T. A. Manteuffel, S. F. McCormick, J. W. Ruge, and P. S. Vassilevski, *Spectral ϱAMGe*, SIAM J. Sci. Comput., 25 (2003), pp. 1–26.
8. J. Fish and V. Belsky, *Generalized aggregation multilevel solver*, Internat. J. Numer. Methods Engrg., 40 (1997), pp. 4341–4361.
9. J. E. Jones and P. S. Vassilevski, *AMGe based on element agglomeration*, SIAM J. Sci. Comput., 23 (2001), pp. 109–133.
10. J. W. Ruge and K. Stüben, *Algebraic multigrid (AMG)*, in Multigrid Methods, S. F. McCormick, ed., vol. 3 of Frontiers in Applied Mathematics, SIAM, Philadelphia, PA, 1987, pp. 73–130.
11. P. Vaněk, J. Mandel, and M. Brezina, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.
12. P. S. Vassilevski, *Sparse matrix element topology with application to AMG and preconditioning*, Numer. Linear Algebra Appl., 9 (2002), pp. 429–444.

Scalable Three-Dimensional Acoustics Using *hp*-finite/infinite Elements and FETI-DP*

D. K. Datta[1], Saikat Dey[1], and Joseph J. Shirron[3]

[1] SFA Inc./NRL, 2200 Defense Highway, Suite 405, Crofton, MD 21114, USA.
`datta,dey`†`@pa.nrl.navy.mil` † Corresponding author.
[2] Metron Inc., 11911 Freedom Dr, Reston, VA 20190, USA. `shirron@metsci.com`

Summary. This paper addresses scalable, parallel *hp*-finite/infinite element-based solution of time-harmonic acoustics problems in three-dimensions. We discuss the application of FETI-DP, an iterative domain-decomposition scheme, to both interior and exterior acoustics problems. We evaluate parallel scalability in terms of number of iterations, wall-clock time, mesh size h, polynomial degree p, number of partitions, and normalized wavenumber. We also discuss the impact of proper selection of the coarse problem on the accuracy of the computed solutions.

1 Introduction

Time-harmonic problems in structural acoustics solve *Helmholtz* equation in bounded (interior) and unbounded (exterior) domains. Examples include propagation in bounded waveguides, scattering and radiation from structures in an infinite fluid domain. Numerical solution of such problems in *medium*-frequency regimes with p-version of finite/infinite elements have been shown to be very effective [1]. p-refinement provides better control of the dispersion (pollution) error enabling increased rate of error convergence compared to h-refinement. hp-approximations for three-dimensional problems in structural acoustics, at mid-to-high frequencies, results in large algebraic systems, $\mathbf{Ax} = \mathbf{b}$, having millions of unknowns. The efficient solution of such problems calls for the application of scalable parallel algorithms.

Due to the indefinite nature of the algebraic systems coupled with poor conditioning from p-approximations and frequency-dependence, direct solution techniques have been favoured for such problems. Unfortunately, the parallelization of facorization-based direct solution strategies offer limited scalability due to the high irregularity of matrix factoring. For large-scale problems of our interest, parallel multi-frontal schemes do not scale well beyond 8 processors. A class of domain-

[*] Research supported by HPCMP.

decomposition algorithms [5] called FETI-DP (see [3] and references therein) have been shown to sustain scalability for increasing number of processors.

Most existing research on applying FETI-type algorithms to exterior acoutsics problems have used the so-called *artificial boundary conditions* and low-degree h-approximations [2]. We evaluate FETI-DP applied to 3D acoustics problems discretized by p-hierarchic finite and infinite elements [1] and highlight the impact of proper selection of the so-called *coarse* problem on solution accuracy.

2 Model Problem

Figure 1 shows the computational domain for a typical acoustics problem where $\Omega_\pm$ denotes the exterior/interior fluid domain, Γ denotes the boundary of the obstacle with outward unit normal ν, and $\Gamma_R = \Omega_+ \cap \Omega_-$ is a separable boundary of radius R. The pressure field ϕ satisfies

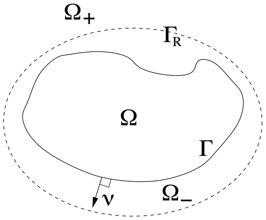


Fig. 1. Computational domain.

$$\Delta\phi + k^2\phi = 0 \quad \text{in } \Omega_\pm, \tag{1}$$

where $k = \omega/c$ is the acoustic wavenumber, ω is the circular frequency of excitation, and c is the speed of sound in the fluid. For exterior domains, we consider the Neumann problem with the boundary conditions

$$\frac{\partial\phi}{\partial\nu} = g \quad \text{on } \Gamma, \quad \lim_{r\to\infty} r\left[\frac{\partial\phi}{\partial r}(r\hat{\mathbf{e}}) - ik\phi(r\hat{\mathbf{e}})\right] = 0 \quad \text{uniformly } \forall\, |\hat{\mathbf{e}}| = 1, \tag{2}$$

where g is the specified Neumann data and the second equation is the Sommerfeld radiation condition prescribing the out-going asymptotic behavior of ϕ. For problems in which an incident wave ϕ_0 scatters from the rigid body enclosed by Γ, we have $g = -\partial\phi_0/\partial\nu$. For interior acoustic problems, we apply a Robin boundary condition $\partial\phi/\partial\nu - ik\phi = h$ on Γ. Both the interior and exterior problems are uniquely solvable for all wavenumbers.

Following standard Galerkin technique results in a weak (variational) form of the interior acoustics problem: Find $\phi \in H^1(\Omega_-)$ such that

$$\mathcal{B}_-(\phi,\psi) + \mathcal{C}(\phi,\psi) = \mathcal{L}_h(\psi)$$

for all $\psi \in H^1(\Omega_-)$, where $\mathcal{B}_-(\phi,\psi) = \int_{\Omega_-} \left(\nabla\phi\cdot\nabla\psi - k^2\phi\psi\right) d\Omega_-$, $\mathcal{C}(\phi,\psi) = \int_\Gamma \psi M\phi\, d\Gamma$, and $\mathcal{L}_h(\psi) = \int_\Gamma h\psi\, d\Gamma$.

Similarly, for exterior problems, we seek test and trial functions $\phi, \psi \in H^1(\Omega_-)$, such that $\phi = \phi_R$ and $\psi = \psi_R$ on Γ_R. The functions ϕ_R and ψ_R with support in Ω_R^+ satisfy the Sommerfeld radiation condition. Weak form of the exterior problem satisfies

$$\mathcal{B}_-(\phi, \psi) + \mathcal{B}_R(\phi_R, \psi_R) = \mathcal{L}_g(\psi),$$

where $\mathcal{L}_g(\psi) = \int_\Gamma g\psi \, d\Gamma$. The bilinear form $\mathcal{B}_R$ is given by

$$\mathcal{B}_R(\phi_R, \psi_R) = \lim_{S \to \infty} \left[\int_{\Omega_{RS}^+} \left(\nabla \phi_R \cdot \nabla \psi_R - k^2 \phi_R \psi_R \right) d\Omega_{RS}^+ - ik \int_{\Gamma_S} \phi_R \psi_R \, d\Gamma_S \right],$$

where Γ_S denotes a separable surface of radius $S > R$ and Ω_{RS}^+ is the annular domain bounded by Γ_R and Γ_S.

Let Δ_h^f be the spatial discretization of the fluid domain Ω_-, with h representing a measure of the spatial mesh size. Let $p_f \geq 1$ and $q_f \geq 0$ be polynomial degrees of finite and radial degree of infinite fluid elements, respectively. Our discrete interior and exterior problems consist of solving

$$\mathcal{B}_-(\phi^{(h,p_f)}, \psi) + \mathcal{C}(\phi^{(h,p_f)}, \psi) = \mathcal{L}_h(\psi) \tag{3}$$

$$\mathcal{B}_-(\phi^{(h,p_f)}, \psi) + \mathcal{B}_R(\phi_R^{(h,p_f,q_f)}, \psi_R) = \mathcal{L}_g(\psi_R), \tag{4}$$

respectively, where $\phi^{(h,p_f)}$, $\phi_R^{(h,p_f,q_f)}$ belong to a finite-dimensional subset of the admissible space of functions for a given h, p_f and q_f (see [1]).

3 Review of FETI-DP

Let the domain Ω_- be subdivided into N subdomains $\Omega_s^i, i = 1, \cdots, N$. Each subdomain is discretized using finite elements and we get the system of equations $K^s u^s = f^s$, where K^s, u^s, and f^s are the finite element left-hand side matrix, the solution and the right-hand side load vector, respectively, for Ω_s. This can be rewritten as

$$\begin{bmatrix} K_{rr}^s & K_{rc}^s \\ {K_{rc}^s}^T & K_{cc}^s \end{bmatrix} \begin{bmatrix} u_r^s \\ u_c^s \end{bmatrix} = \begin{bmatrix} f_r^s \\ f_c^s \end{bmatrix} \tag{5}$$

where the degrees of freedom of a subdomain are divided into two groups r and c referred to as the "interior" and "corner" degrees of freedom, respectively. The c degrees of freedom are created at a global level such that we have ${B_c^1}^T u_c^1 = {B_c^2}^T u_c^2 = \cdots = {B_c^N}^T u_c^N = u_c$, where B_c^s maps the corner degrees of freedom of Ω_s to the set of global corner degrees of freedom. The subdomain equations can now be written as

$$\begin{gathered} K_{rr}^s u_r^s + K_{rc}^s B_c^s u_c = f_r^s \\ \sum_{s=1}^{s=N} B_c^{sT} {K_{rc}^s}^T u_r^s + \sum_{s=1}^{s=N} B_c^{sT} K_{cc}^s B_c^s u_c = \sum_{s=1}^{s=N} B_c^{sT} f_c^s = f_c \end{gathered} \tag{6}$$

At the interfaces of the subdomains, the continuity of subdomain solutions is imposed by the following condition

$$u_b^m - u_b^n = 0 \quad \text{on } \Gamma_{mn} \tag{7}$$

with $\Gamma_{mn} = \partial\Omega_s^m \cap \partial\Omega_s^n$ for $m, n = 1, ..., N$, and $m \neq n$. Note that index b ($b \subset r$) denotes those degrees of freedom which lie on the interface boundary except the corner degrees of freedom c. Recasting the above equation as $\sum_{s=1}^{s=N} B_r^s u_r^s = 0$ where B_r^s is a signed boolean matrix such that $B_r^s u_r^s = \pm u_b^s$, and letting λ denote the lagrange multipliers for enforcing the interface condition (7), leads to the system of equations

$$K_{rr}^s u_r^s + K_{rc}^s B_c^s u_c + B_r^{sT} \lambda = f_r^s, \tag{8}$$

$$\sum_{s=1}^{s=N} B_c^{sT} K_{rc}^{s\;T} u_r^s + \sum_{s=1}^{s=N} B_c^{sT} K_{cc}^s B_c^s u_c = \sum_{s=1}^{s=N} B_c^{sT} f_c^s = f_c, \tag{9}$$

$$\sum_{s=1}^{s=N} B_r^s u_r^s = 0. \tag{10}$$

Elimination of u_r^s and u_c from the above equations gives the interface problem in terms of the dual (lagrange multiplier) solution

$$\big(F_{rr} + F_{rc} K_{cc}^{*\;-1} F_{rc}^{\;T}\big)\lambda = \; d_r - F_{rc} K_{cc}^{*\;-1} f_c^*. \tag{11}$$

Here $F_{rr} = \sum_{s=1}^{s=N} B_r^s K_{rr}^{s\;-1} B_r^{sT}$, $F_{rc} = \sum_{s=1}^{s=N} B_r^s K_{rr}^{s\;-1} K_{rc}^s B_c^s$, $K_{cc}^* = \sum_{s=1}^{s=N} B_c^{sT} K_{cc}^s B_c^s - (K_{rc}^s B_c^s)^T K_{rr}^{s\;-1} K_{rc}^s B_c^s$, $d_r = \sum_{s=1}^{s=N} B_r^s K_{rr}^{s\;-1} f_r^s$, and $f_c^* = f_c - \sum_{s=1}^{s=N} B_c^{sT} K_{rc}^{s\;T} K_{rr}^{s\;-1} f_r^s$. In equation (11) F_{rr} forms a fine-level operator and $F_{rc} K_{cc}^{*\;-1} F_{rc}^{\;T}$ forms a coarse-level operator.

3.1 Coarse space for *p*-approximation and infinite element

Three-dimensional p-approximations offer multiple options for selecting the "coarse" degrees of freedom. For a partitioned domain, let $\Sigma = \bigcup_{m,n=1}^{N} \Gamma_{mn}$, $m \neq n$, denote the closure of the partition boundary as depicted in Figure 2a. We consider as "coarse" candidates only those subdomain basis functions which have support on mesh entities that belong to the boundary of at least three partitions. This implies that only mesh edges and vertices contribute to the coarse space. If M_i^d denotes the i-th mesh entity of dimension d then Figure 2b depicts the closure of all the candidate mesh entities denoted by Σ_e. Knowing that there are $p_f - 1$ edge modes in addition to the linear (vertex) modes, suggests two schemes to pick coarse *dofs*:

C1 Consider only vertex modes. No edge modes.
C2 Consider vertex and edge p-modes.

Additional care is needed when considering a coarse problem space for an exterior discretization consisting of both finite and infinite elements as shown in Figure 2c. Let M_j^3 be a mesh region in Ω_- and let M_i^2 be a corresponding mesh face on the infinite boundary implying $M_i^2 = \partial M_j^3 \cap \Gamma_R$. The finite element approximation spaces consist of $S^p_{\Omega_-}$ - the space of degree p polynomials assiociated with closure of mesh region M_j^3. The infinite element approximation [1] consists of the tensor-product space $\hat{S}^p_{\Omega_-} \otimes S^q_{\Omega_+}$ where $\hat{S}^p_{\Omega_-}$ is the subset of $S^p_{\Omega_-}$ with nonzero support on M_j^2 and $S^q_{\Omega_+}$ is a space of degree $q_f + 1$ polynomials in $1/\rho$ which satisfy the Sommerfeld condition. When selecting the *dofs* for the a mesh vertex that lies on the intersection of Γ_R and Σ_e, as depicted in Figure 2d, the additional $q_f + 1$ radial *dofs* must also be included.

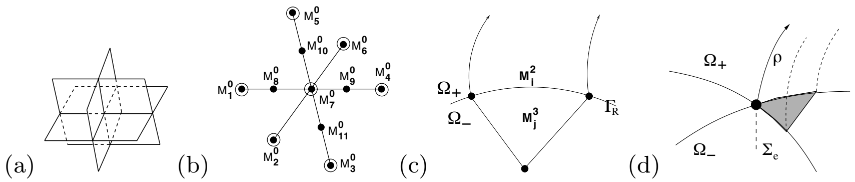


Fig. 2. (a) Partition boundary. (b) Closure of mesh entities for coarse problem. (c) Discretization using finite and infinite elements (d) Mesh vertex belonging to infinite element as well as Σ_e.

4 Numerical Examples

This section gives several numerical examples to evaluate the performance of FETI-DP for helmholtz problem in 3D for various ka, p, and meshes. In the implementation of FETI-DP within STARS3D, the sub-domain matrices K^s_{rr} and the coarse problem matrix K^*_{cc} are assembled in a sparse representation and factored using a sparse multi-frontal solver. For the interface problem (11), a GMRES based iterative solver [4] is used with a lumped preconditioner [3]. The parallel implementation within STARS3D is based on MPI.

Interior Helmholtz problem: Consider the solution of the interior problem in $\Omega \equiv [-1,-4,-1] \times [1,4,1]$ with Robin data data $h = \partial\phi_{\text{ex}}/\partial\nu - ik\phi_{\text{ex}}$, where $\phi_{\text{ex}} = exp(ik\,|\mathbf{r} - \mathbf{r}_0|)/|\mathbf{r} - \mathbf{r}_0|$ is a point source located at $\mathbf{r}_0 = (0,-5,0)$ outside of Ω_-. Three hexahedral meshes, *Mesh A*, *Mesh B* and *Mesh C*, with 8192, 65,536, and 262,144 hexahedrons, respectively, with $p = 1,2,3,4$ are used. Table 1 gives the iteration counts M. Table 2 lists the wall-clock time t_N and parallel efficiency $E_N = \dfrac{Nt_2}{2t_N} \times 100$. These computations were done on a SGI-Altix. As expected, scalability improves as the problem size grows. Note that *Mesh C*, with $p = 3$, has 1.98 million complex degrees of freedom. More than 100% parallel efficiency is observed because of significant drop in total time to factor K^s_{rr} as the domain is divided into subdomains.

Exterior Helmholtz problem: This example considers scattering of plane wave by a rigid obstacle. The incident-wave ϕ_0 is along $(0,0,-1)$. Two different shapes for the obstacle are considered: (1) a sphere of radius a, and (2) a submarine-like structure. The sphere mesh is partitioned based on the coordinate planes,

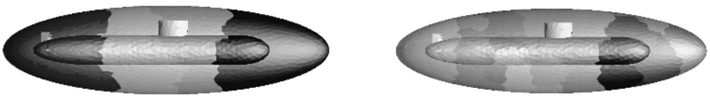
Table 1. Iteration counts for interior helmholtz problem for $Mesh\ A$.

	$p=1$						$p=2$						$p=3$						$p=4$					
$ka \backslash N$	2	4	8	16	32	64	2	4	8	16	32	64	2	4	8	16	32	64	2	4	8	16	32	64
10	5	6	4	6	10	9	6	6	5	7	11	12	6	6	5	7	12	13	6	6	5	7	13	16
14	7	7	5	8	12	12	7	8	6	9	15	19	7	8	6	9	17	22	7	8	6	9	20	29
18	7	8	6	10	22	23	9	9	8	11	35	55	9	9	8	11	38	58	9	9	8	11	40	62

Table 2. Parallel scalability for interior helmholtz problem with $ka = 10$.

	$Mesh\ B, p = 3$		$Mesh\ B, p = 4$		$Mesh\ C, p = 2$		$Mesh\ C, p = 3$	
N	M	$Time(Eff.)$	M	$Time(Eff.)$	M	$STime(Eff.)$	M	$Time(Eff.)$
2	6	656.8	6	3346.0	7	3697.0	-	-
4	9	148.2 (221%)	9	631.7 (265%)	9	1061.8 (174%)	9	3849.5
8	8	48.8 (336%)	8	175.4 (477%)	8	298.9 (309%)	9	1375.3 (140%)
32	24	30.2 (136%)	25	75.6 (276%)	25	141.1 (163%)	25	319.9 (154%)

while the mock-submarine (shown in Figure 3) is partitioned using *METIS* (`www-users.cs.umn.edu/ karypis/metis/`).



(a) (b)

Fig. 3. Partitions for mock-submarine. (a) $N = 4$, (b) $N = 32$.

Table 3 gives the iteration counts for the sphere problem for a mesh with 7896 mesh regions. Here, *unconjugated* infinite elements with radial degree $q_f = 2$ were used. Similar results for the mock-submarine are given in Table 4.

Impact of coarse problem selection To evaluate the impact of the choice of method to select the *degrees of freedom* for the *coarse* problem outlined in Section 3.1, we consider the sphere problem. Consider a finite element approximation with $p_f = 3$ and infinite element radial degree $q_f = 2$. We compare the number of iterations needed to converge to a given tolerance. The impact on the accuracy is evaluated by computing the pointwise maximum (L_∞) relative error in the real and the imaginary parts of the computed scattered field as:

Table 3. Iteration counts for the sphere problem.

	$p=1$			$p=2$			$p=3$			$p=4$		
$ka \setminus N$	2	4	8	2	4	8	2	4	8	2	4	8
5	13	5	3	15	6	7	16	7	7	12	8	10
10	16	5	3	21	7	7	20	8	7	12	9	11

Table 4. Iteration counts for exterior scattering from a rigid mock submarine. *Mesh* H_1 (*Mesh* H_2) has 19024 (60819) regions.

	Mesh $H_1, p=1$			*Mesh* $H_2, p=1$			*Mesh* $H_1, p=2$			*Mesh* $H_2, p=2$		
$N \setminus ka$	1	5	10	1	5	10	1	5	10	1	5	10
4	19	64	415	20	40	162	24	121	236	23	93	386
8	21	93	771	21	67	272	26	190	383	24	140	863
16	21	132	2457	22	92	347	27	243	970	28	169	1403
32	22	153	1906	22	68	716	31	293	2480	30	126	2955

$$|\Re(e)|_\infty = \frac{\max_i |\Re(u_i) - \Re(u_i^0)|}{\max_i |\Re(u_i^0)|}, \quad |\Im(e)|_\infty = \frac{\max_i |\Im(u_i) - \Im(u_i^0)|}{\max_i |\Im(u_i^0)|} \tag{12}$$

where, u^0 is a globally C^0 solution obtained by a direct multi-frontal scheme.

From Table 5 we note that use of all the edge modes from Σ_e (**C2**) makes the approximate FETI-DP solution to converge to the globally C^0 solution within the tolerance used in the iterative solution of the interface problem 11. In contrast, use of only vertex (linear) modes (**C1**) make the FETI-DP solution to have errors that significantly exceed the convergence tolerance used in the iterative solver.

5 Discussion and Conclusion

We have successfully applied the FETI-DP algorithm to *hp*-finite/infinite element discretization of both interior and exterior acoustics problems. We show super-linear scalability for a set of interior acoustics problems. For exterior problems, we demonstrate excellent scalability of FETI-DP except at very high *ka* values. The lack of better scalability at higher wavenumber is tied to numerically dispersive nature of

Table 5. Impact of coarse space selection on iteration counts and accuracy for scattering from rigid sphere at $ka = 1, 10$. Lumped preconditioner and a tolerance $1.0e - 09$ is used for iterative solve. n_c denotes the number of coarse *dofs*.

	$N = 4$			$N = 8$		
	M (n_c)	$\lvert\Re(e)\rvert_\infty$	$\lvert\Im(e)\rvert_\infty$	M (n_c)	$\lvert\Re(e)\rvert_\infty$	$\lvert\Im(e)\rvert_\infty$
$ka = 1$, **C1**	40 (16)	8.34e-3	3.71e-3	40 (48)	8.33e-3	6.21e-3
$ka = 1$, **C2**	40 (36)	2.83e-8	1.54e-8	37 (108)	9.13e-9	9.75e-9
$ka = 10$, **C1**	100 (16)	1.26e-2	1.59e-2	110 (48)	2.64e-2	2.38e-2
$ka = 10$, **C2**	100 (36)	9.08e-9	1.11e-8	102 (108)	8.73e-9	8.31e-9

these approximations and will be addressed with effective augmentation strategies that accelarate convergence further. We have also discussed strategies for selecting the *coarse* problem space and show that for p-approximations it is impertative to include all the high-order mesh edge modes to ensure expected accuracy of the subdomain solutions.

References

1. S. Dey, J. J. Shirron, and L. S. Couchman, *Mid-frequency structural acoustic and vibration analysis in arbitrary, curved, three-dimensional domains*, Computers and Structures, 79 (2001), pp. 617–629.
2. C. Farhat, P. Avery, R. Tezaur, and J. Li, *FETI-DPH: a dual-primal domain decomposition method for acoustic scattering*, J. Comp. Acoust., 13 (2005), pp. 499–524.
3. 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.*, Int. J. Numer. Meth. Engrg., 50 (2001), pp. 1523–1544.
4. V. Frayssé, L. Giraud, S. Gratton, and J. Langou, *A set of GMRES routines for real and complex arithmetics on high performance computers*, Tech. Rep. TR/PA/03/3, CERFACS, 2003.
5. B. F. Smith, P. E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.

A Multilevel Energy-based Quantization Scheme

Maria Emelianenko [1] and Qiang Du [2]

[1] Department of Mathematical Sciences, Carnegie Mellon University, Pittsburgh, PA 15213, USA. masha@cmu.edu
[2] Department of Mathematics, Pennsylvania State University, University Park, PA 16802, USA. qdu@math.psu.edu

Summary. Quantization has diverse applications in many areas of science and engineering. In this paper, we present a new nonlinear multilevel algorithm that accelerates existing numerical methods for finding optimal quantizers. Both a theoretical framework for the convergence analysis and results of some computational experiments are provided.

1 Introduction

A vector quantizer maps N -dimensional vectors in the domain $\Omega \subset \mathbb{R}^N$ into a finite set of vectors $\{\mathbf{z}_i\}_{i=1}^k$. Each vector $\mathbf{z}_i$ is called a code vector or a *codeword*, and the set of all the codewords is called a codebook. A special, yet popular, quantization scheme is given by the Voronoi tessellation associated with some codewords $\{\mathbf{z}_i\}$, also called *generators*.

A Voronoi tessellation for the given generating points $\{\mathbf{z}_i\}_{i=1}^k \subset \Omega$ refers to the tessellation of a given domain Ω by the Voronoi regions $\{V_i\}_{i=1}^k$ where, for each i , the Voronoi region V_i consists of all points in Ω that are closer to $\mathbf{z}_i$ than to the other generating points. For a density function ρ defined on Ω , we define the centroids, or mass centers, of the regions $\{V_i\}_{i=1}^k$ by

$$\mathbf{z}_i^* = \Big(\int_{V_i} \mathbf{y}\rho(\mathbf{y})\, d\mathbf{y} \Big) \Big(\int_{V_i} \rho(\mathbf{y})\, d\mathbf{y} \Big)^{-1} . \tag{1}$$

Then, an *optimal quantization* may be constructed through a *centroidal Voronoi tessellation* (CVT) for which the generators of the Voronoi tessellation themselves are the centroids of their respective Voronoi regions, in other words, $\mathbf{z}_i = \mathbf{z}_i^*$ for all i . Besides providing an optimal least square vector quantizer design in electrical engineering applications [10],[11],[20], the concept of CVT has other diverse applications in many areas of science and engineering, such as image and data analysis, resource optimization, sensor networks, and numerical partial differential equations

[5],[6],[8],[12],[13],[16],[18]. We refer to [5] for a more comprehensive review of the mathematical theory and diverse applications of CVTs.

In the seminal work of Lloyd on the least square quantization [17], one of the algorithms proposed for computing optimal quantizers is an iterative algorithm consisting of the following simple steps: starting from an initial quantization (a Voronoi tessellation corresponding to an old set of generators), a new set of generators is defined by the mass centers of the Voronoi regions. This process is continued until a certain stopping criterion is met.

Given a set of points $\{\mathbf{z}_i\}_{i=1}^k$ and a tessellation $\{V_i\}_{i=1}^k$ of the domain, we may define the *energy functional* or the *distortion value* for the pair $(\{\mathbf{z}_i\}_{i=1}^k, \{V_i\}_{i=1}^k)$ by:

$$\mathcal{H}\Big(\{\mathbf{z}_i\}_{i=1}^k, \{V_i\}_{i=1}^k\Big) = \sum_{i=1}^k \int_{V_i} \rho(\mathbf{y})|\mathbf{y} - \mathbf{z}_i|^2 \, d\mathbf{y} \, .$$

The minimizer of $\mathcal{H}$, that is, the optimal quantizer, necessarily forms a CVT which illustrates the optimization property of the CVT [5]. The terms optimal quantizer and CVT are thus to be used interchangeably in the sequel. It is also easy to see that Lloyd's algorithm is an energy descent iteration, which gives strong indications to its practical convergence.

Lloyd's algorithm has [11],[12],[13], in recent years, sparked an enormous research effort and its variants have been proposed and studied in many contexts for different applications. Efficient algorithms for computing the CVTs play crucial roles for modern application of CVT in large scale scientific and engineering problems such as data communication and mesh generation. In this short paper, we first discuss some convergence theory recently derived in [3] for Lloyd's algorithm to motivate our ongoing work. Then we outline a new multilevel approach to the optimal quantization problem introduced recently in [1],[4] which can be used to accelerate the convergence of Lloyd's algorithm. We discuss the idea of a dynamic nonlinear preconditioner and also give a convergence theorem as well as some numerical results.

2 Convergence properties of Lloyd's iteration

Even with their great success in practice, only limited rigorous results on the convergence properties of Lloyd's iteration have been obtained and many important computational issues remain to be explored [5]. Some important characterizations of convergence for Lloyd's scheme have been obtained recently in [3]. The results stated below demonstrate the global convergence properties of the Lloyd iteration and its relationship to the critical points of the energy functional.

Theorem 2.1 *Any limit point of the Lloyd algorithm is a fixed point of the Lloyd map, and this determines a stationary point of $\mathcal{H}$. The set of limit points share the same distortion value $\mathcal{H}$ for a given iteration.*

Theorem 2.2 *If the iterations in the Lloyd algorithm stay in a compact set where the Lloyd map T is continuous, then the algorithm is globally convergent to a critical point of $\mathcal{H}$.*

We refer to [3] for the proofs and further discussions of related results.

Beyond the study on the global convergence, the characterization of the convergence rate is often also important in practice. For instance, one may inquire if a geometric convergence rate can be established. This is indeed verified in [5] for the constant density function and later in [3] under a strong type of log-concavity conditions, where the established geometric convergence rate r is shown to be of the order of $1-ck^{-2}$, therefore the Lloyd method slows down for large values of k, the total number of generators. Even in the one-dimensional case, both our theoretical estimates and the experiments indicate that the convergence of the Lloyd iterations is at most linear.

3 The new energy-based nonlinear multilevel algorithm

The evidence of slow convergence of the Lloyd iteration and its descent properties motivated our search for a Lloyd iteration based numerical scheme with superior convergence properties. A possible approach to speeding up the convergence of Lloyd's method is to use a domain or space decomposition (or multigrid) strategies ([1],[2],[4],[15],[14]). There are many ways of implementing such an algorithm in the context of CVTs. However, the problem of constructing a CVT is nonlinear in nature and hence cannot be analyzed using standard linear multigrid theory. Without using any type of linearization techniques, we hope to overcome the difficulties of the nonlinearity by essentially relying on the energy minimization.

3.1 Description of the algorithm

Our motivation in using the energy minimization approach was the optimality property of the CVTs mentioned above. The optimality property implies that at the optimal quantizer $\nabla\mathcal{H}=0$.

Since the energy functional is in general non-convex, we use a dynamic nonlinear preconditioner to relate our problem to a convex optimization problem. More precisely, let $R=\mathtt{diag}\{R_i^{-1}\}, i=1,\dots,k+1$ where $R_i=\int_{V_i}\rho(\mathbf{y})\,d\mathbf{y}$ are the masses of the corresponding Voronoi cells. We arrive at an equivalent formulation of the minimization problem: $R\nabla\mathcal{H}=0$, or $\min\|R\nabla\mathcal{H}\|^2$. This *preconditioning* makes the energy functional convex in a large neighborhood of the minimizer and therefore the new formulation has advantages over the original problem. Hence, defining the set of iteration points $\mathbf{W}$ by

$$\mathbf{W}=\{(w_i)|_{i=0}^{k+1}|\quad 0=w_0\le w_i\le w_{i+1}\le w_{k+1}=1,\ \forall 0\le i\le k\}\ ,$$

our new multilevel algorithm is then based on the following nonlinear optimization problem

$$\min_{\mathbf{Z}\in\mathbf{W}}\tilde{\mathcal{H}}(\mathbf{Z}),\ \text{ where } \tilde{\mathcal{H}}(\mathbf{Z}=\{\mathbf{z}_i\}_{i=0}^{k+1})=||R\nabla\mathcal{H}(\{\mathbf{z}_i\}_{i=1}^{k},\{V_i\}_{i=1}^{k})||^2 \tag{2}$$

where $\{V_i\}_{i=1}^{k}$ is the Voronoi tessellation corresponding to the generators $\{\mathbf{z}_i\}_{i=1}^{k}$. For simplicity, consider the CVT on the one-dimensional unit interval $[0,1]$. Let S_k be the space of continuous piecewise linear functions on a uniform mesh with k

sub-intervals and a hierarchical basis $\{\{\psi^i_j\}_{j=1}^{n_i}\}_{i=1}^{H}$ with H the number of levels. Let $\bar{\psi}^i_j = \{\psi^i_j(\frac{m}{k+1})\}_{m=0}^{k+1} \in \mathbb{R}^{k+2}$ and set $\mathbf{W}_i = \text{span}\{\bar{\psi}^i_j\}_{j=1}^{n_i}$. We now present our multilevel successive subspace correction algorithm as follows:

Algorithm 3.1 (Successive correction $V(\nu_1, \nu_2)$ scheme)

```
Input:
  $\Omega$, the domain of interest; $\rho$, a probability distribution on $\Omega$;
  $k$, number of generators;
  $\mathbf{Z} = \{z_i\}_{i=0}^{k+1} \in \mathbf{W}$, the ends plus the initial generators.
Output:
  $\mathbf{Z} = \{z_i\}_{i=0}^{k+1}$, the ends plus generators for CVT $\{V_i\}_{i=1}^{k}$.
Method:
1. For i=H:-1:2, repeat $\nu_1$ times:
     given $\mathbf{Z}$, find $\mathbf{Z} = \mathbf{Z} + \alpha_j^0 \bar{\psi}^i_j \in \mathbf{W}$ sequentially for $1 \le j \le n_i$
     such that $\tilde{\mathcal{H}}(\mathbf{Z} + \alpha_j^0 \bar{\psi}^i_j) = \min_{\alpha_j} \tilde{\mathcal{H}}(\mathbf{Z} + \alpha_j \bar{\psi}^i_j)$.
   Endfor
2. $\mathbf{Z}$ ←CoarseGridSolve($\mathbf{Z}$)
3. For i=2:1:H, repeat $\nu_2$ times:
     given $\mathbf{Z}$, find $\mathbf{Z} = \mathbf{Z} + \alpha_j^0 \bar{\psi}^i_j \in \mathbf{W}$ sequentially for $1 \le j \le n_i$
     such that $\tilde{\mathcal{H}}(\mathbf{Z} + \alpha_j^0 \bar{\psi}^i_j) = \min_{\alpha_j} \tilde{\mathcal{H}}(\mathbf{Z} + \alpha_j \bar{\psi}^i_j)$.
   Endfor
4. Repeat steps 1 to 3 until some stopping criterion is met.
```

Each step of the above algorithm involves solving a system of nonlinear equations which plays the role of a relaxation. The solution at the current iterate is updated after each nonlinear solve by the Gauss-Seidel type procedure, hence the resulting scheme is sequential in nature. Here ν_1, ν_2 denote the number of Gauss-Seidel iterations used at each level. Although $\nu_{1,2} = 1$ is sufficient in theory, larger values need to be used in practice due to the numerical error in solving the nonlinear system. The values $\nu_{1,2} \le 3$ usually suffice for the optimization to reach saturation. More general algorithms and convergence results will be given in future work. It is worth noting that in the one-dimensional case the set of basis functions

$$Q_i = [\bar{\psi}^i_1, \ldots, \bar{\psi}^i_{n_i}]^T \in R^{n_i \times k}$$

used at each iteration can be pre-generated using the recursive procedure: $Q_1 = I_{k\times k}$ and $Q_s = (\Pi_{i=1}^{s} P_i) Q_1$ where P_i is the basis transformation from the space $\mathbf{W}_{i+1}$ to $\mathbf{W}_i$ which plays a role of a restriction operator.

Let us now state the following convergence theorem [1]:

Theorem 3.1 *Algorithm 3.1 converges uniformly in* $\mathbf{W}$ *for any density of the type* $\rho(x) = 1 + \varepsilon g(x)$, *where* $g(x)$ *is smooth and* ε *is small. Moreover,* $d_n = \tilde{\mathcal{H}}(u_n) - \tilde{\mathcal{H}}(u)$ *satisfies*

$$d_n \le r d_{n-1},$$

for some constant $r = \dfrac{C}{1+C}$, *where* C *is a constant independent of the number of generators or the number of levels.*

A proof of this result can be derived based on the framework of [19]. Supply $\mathbf{W}$ with the norm $||y||_{1,\mathbf{W}}^2 = \frac{1}{k}\sum_{i=1}^{k+1}(y_i - y_{i-1})^2$, the key steps of the proof include demonstrating the continuity and local convexity of the functional $\tilde{\mathcal{H}}$ with respect to the norm $||\cdot||_{1,\mathbf{W}}$, and establishing a strengthened Cauchy-Schwartz inequality with respect to the space decomposition $\mathbf{W} = \bigoplus_{i=1}^{H} \bar{\mathbf{W}}_i$ where $\bar{\mathbf{W}}_i = \mathbf{W}_i/\mathbf{W}_{i-1}$ for $i > 1$ and $\bar{\mathbf{W}}_1 = \mathbf{W}_1$. The complete proof is given in [1] and is omitted here. It follows that for a suitable choice of decomposition in 1D the asymptotic convergence factor of our multilevel algorithm is independent of the size of the problem and the number of grid levels, which gives a significant speedup in comparison to other methods, like the traditional Lloyd iteration. Moreover, we have

Corollary 3.2 *For the* hat *basis and the constant density function, we may take* $C = 4$ *and thus* $r = 0.8$.

We note that the estimated convergence rate of $r = 0.8$ is merely an upper bound, and the actually convergence rate is much smaller in practice. We justify the above theoretical results in the numerical examples that follow.

3.2 Numerical results

For the V(1,1) multigrid implementation of the new algorithm, we compared our algorithm with the regular Gauss-Seidel performance. We plotted the convergence factor $\rho \approx \frac{z_{n+1} - z_n}{z_n - z_{n-1}}$ for each V(1,1) cycle with respect to k, the total number of generators (grid points) taken for $\rho(x) = 1$.

Figure 1 justifies the fact that the speed of convergence for the proposed scheme does not grow with the number of generators, while Table 1 shows the stabilization of the number of multigrid cycles $V(\nu_1, \nu_2)$ needed to reduce the error to $\varepsilon = 10^{-12}$ in the $\rho(x) = 1$ case. The difference in the number of iterations required for $V(1,1)$ and $V(2,2)$ comes from the approximation error in solving the optimization problem at each level, which decreases quickly as the number of relaxations grows.

The geometric rate of energy and error reduction asserted by the Theorem 3.1 is confirmed by the experiments. Indeed, Figure 2 shows the convergence history of the error (left) of a $V(1,1)$-cycle and the energy (right) vs. total number of relaxations for the $k = 64$, $\rho(x) = 1$ case (in log-normal scale).

The results for other nonlinear densities, though not shown here, are also consistent with the theoretical conclusions reached above (see [1]). Multidimensional extensions are discussed in [4].

4 Applications

CVTs have a rich field of applications in various areas of mathematics as well as engineering. Here we provide a couple of geometric examples to give a flavor of the kind of problems that benefit from the study of this concept. Figure 3 shows

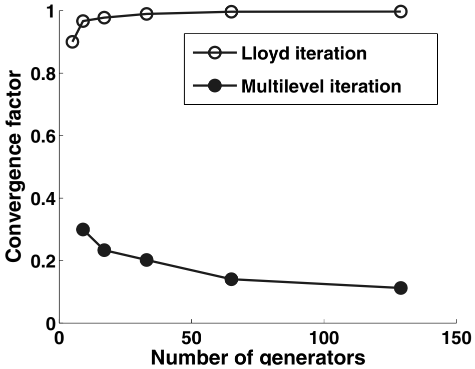


Fig. 1. Convergence factor ρ vs. k for the multigrid and Gauss-Seidel methods.

$k/V(\nu_1,\nu_2)$	V(1,0)	V(0,1)	V(1,1)	V(2,0)	V(0,2)	V(2,2)
3	7	8	6	6	7	4
5	11	11	8	8	8	6
9	13	14	9	9	9	7
17	18	18	12	12	12	8
33	21	20	13	12	13	8
65	21	22	12	12	12	8
129	21	21	12	12	12	8
257	20	23	12	12	13	7
513	20	22	12	11	13	7
1025	19	22	11	11	13	7

Table 1. Number of $V(\nu_1,\nu_2)$ cycles vs. the number of generators.

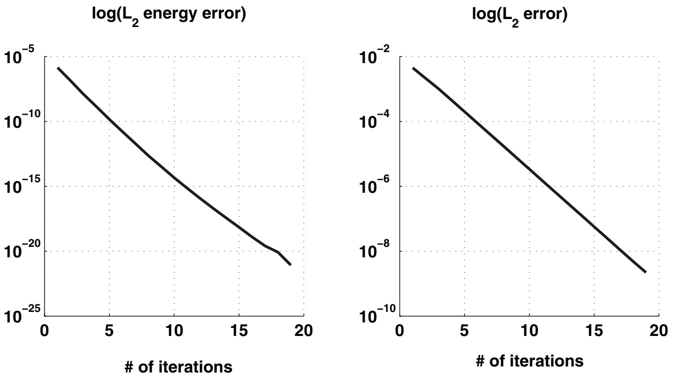


Fig. 2. Error and energy reduction of the $V(1,1)$-cycle.

tessellations of the sphere for different density functions [7] and an example of meshes generated by means of CVT [9].

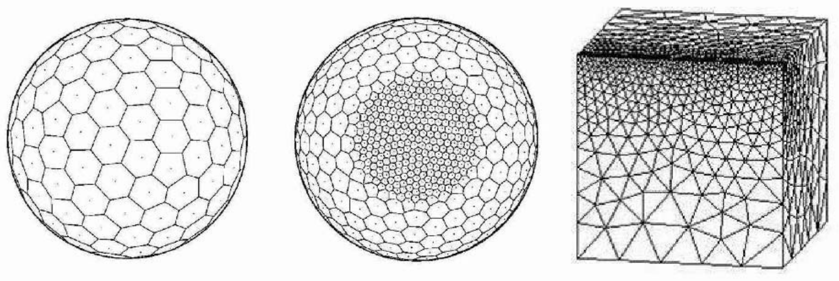


Fig. 3. Examples of CVTs for a sphere and a CVT-based mesh for a cube.

The point distributions generated via CVT can be used for vector quantization, optimal resource allocation, image compression, mesh generation and in many other applications [5] . In many of these applications, the efficiency of the numerical scheme plays a crucial role, so possible new approaches in accelerating existing numerical methods such as the multilevel approach discussed here are very important.

5 Conclusion

A new energy-based multilevel method is introduced for the optimal quantization which adopts dynamic nonlinear preconditioning to take advantage of a nonlinear convex optimization setting. The uniform convergence of the method with respect to the grid size and the number of grid levels and significant speedup compared to Lloyd's method are demonstrated. More work is under way for the multilevel scheme in higher dimensions.

Research support by the NSF through the grants DMR-0205232 and DMS-0409297 is gratefully acknowledged.

References

1. Q. Du and M. Emelianenko, *Uniform convergence of a nonlinear optimization-based multilevel quantization scheme.* preprint, 2005.
2. ———, *Acceleration schemes for computing centroidal Voronoi tessellations*, Numer. Linear Algebra Appl., 13 (2006).
3. Q. Du, M. Emelianenko, and L. Ju, *Convergence properties of the Lloyd algorithm for computing the centroidal Voronoi tessellations*, SIAM J. Numer. Anal., (2006). To appear.
4. Q. Du, M. Emelianenko, and L. Zikatanov, *An energy-based multilevel quantization scheme in multidimension.* In preparation, 2006.
5. Q. Du, V. Faber, and M. Gunzburger, *Centroidal Voronoi tessellations: applications and algorithms*, SIAM Review, 41 (1999), pp. 637–676.
6. Q. Du and M. Gunzburger, *Grid generation and optimization based on centroidal Voronoi tessellations*, Applied Mathematics and Computation, (2002), pp. 591–607.

7. Q. Du, M. Gunzburger, and L. Ju, *Constrained centroidal Voronoi tessellations on general surfaces*, SIAM J. Scientific Computing, 24 (2003), pp. 1488–1506.
8. Q. Du, M. Gunzburger, L. Ju, and X. Wang, *Centroidal Voronoi tessellation algorithms for image processing*, J. Math. Imaging & Vision, (2006). To appear.
9. Q. Du and D. Wang, *Tetrahedral mesh generation and optimization based on centroidal Voronoi tessellations*, Int. J. for Numer. Meth. Engng, 56 (2003), pp. 1355–1373.
10. A. Gersho, *Asymptotically optimal block quantization*, IEEE Trans. Inform. Theory, 25 (1979), pp. 373–380.
11. R. M. Gray and D. L. Neuhoff, *Quantization*, IEEE Trans. Inform. Theory, 44 (1998), pp. 2325–2383.
12. L. Ju, Q. Du, and M. Gunzburger, *Probabilistic methods for centroidal Voronoi tessellations and their parallel implementations*, Parallel Comput., 28 (2002), pp. 1477–1500.
13. L. Ju, M. Gunzburger, and Q. Du, *Meshfree, probabilistic determination of points, support spheres, and connectivities for meshless computing*, Comput. Methods in Appl. Mech. and Engrg., 191 (2002), pp. 1349–1366.
14. Y. Koren and I. Yavneh, *Adaptive multiscale redistribution for vector quantization*, SIAM J. Sci. Comput., (2004). accepted.
15. Y. Koren, I. Yavneh, and A. Spira, *A multigrid approach to the 1-D quantization problem*, Tech. Rep. CS-2005-08, Technion - Israel Institute of Technology, Department of Computer Science, 2005.
16. Y. Linde, A. Buzo, and R. M. Gray, *An algorithm for vector quantizer design*, IEEE Trans. Comm., 28 (1980), pp. 84–95.
17. S. P. Lloyd, *Least square quantization in PCM*, IEEE Trans. Infor. Theory, 28 (1982), pp. 129–137.
18. J. MacQueen, *Some methods for classification and analysis of multivariate observations*, in Proceedings of the 5th Berkeley Symposium on Mathematical Statistics and Probability, L. M. LeCam and J. Neyman, eds., vol. 1, Berkeley, CA, 1967, University of California Press, pp. 281–297.
19. X.-C. Tai and J. Xu, *Global convergence of subspace correction methods for some convex optimization problems*, Math. Comp., 71 (2002), pp. 105–124.
20. A. V. Trushkin, *On the design of an optimal quantizer*, IEEE Trans. Infor. Theory, 39 (1993), pp. 1180–1194.

A Cousin Formulation for Overlapped Domain Decomposition Applied to the Poisson Equation

Drazen Fabris [1] and Sergio Zarantonello [2]

[1] Santa Clara University, School of Engineering, Department of Mechanical Engineering, 500 El Camino Real, Santa Clara, CA 95053, USA.
dfabris@scu.edu

[2] Department of Applied Mathematics, Santa Clara University and 3DGeo Development Inc., 4633 Old Ironsides Drive, Santa Clara, CA 95054, USA.
szarantonello@scu.edu

Summary. In this paper we present a domain decomposition method for the solution of a linear elliptic equation as a direct two-step procedure (in contrast to iterative-based Schwarz and Aitken-Schwarz procedures). First, local solutions to the nonhomogeneous equation are calculated on overlapping subdomains, and second, connection functions, which correct for mismatches on the subdomain intersections and solve the homogeneous equation in each subdomain, are obtained by solving a Cousin-like problem. The procedure is applied to Poisson's equation as a model linear equation. The calculation of the connection functions is achieved through a classical orthogonal decomposition of the solution to Laplace's equation and can be achieved through progressive solutions in each direction for separable boundary conditions. The procedure is also applicable to more complicated domains with non-separable boundary conditions. A few examples will be given. The connection functions can be calculated to high precision and require minimal overlap of the subdomains.

1 Introduction

In the classical Schwarz procedure applied to a domain decomposition with overlapping subdomains the final solution is achieved through an iterative procedure updating the boundary conditions that are internal to the subdomains [10] (see also optimized variants [4]). The classical Schwarz technique is limited since the number of iterations scales geometrically with the number of subdomains and the rate of convergence depends strongly on the overlap. Recent work has shown that each successive iteration can be represented as a linear operator on the solution and that the final solution can be determined from an extrapolation of the linear convergence rate. This Aitken-Schwarz procedure has been extended to non-linear problems as a Steffensen-Schwarz procedure [2, 5, 6]. In this paper we consider Poisson's equation in overlapped regular and irregular geometries as a model system for future applications to the Navier-Stokes equations. The matching requirement is based on equating the two local solutions functionally in the overlap region, a stricter requirement than in the Schwarz procedure. This formulation leads to a Cousin-like problem [7] and, for this linear problem, results in a constrained system for coupled solutions of the homogeneous form of the given equation.

First, we require a local Poisson solve on each subdomain. The mismatch between any two local solutions satisfies the homogeneous form of the equation on the subdomain intersections and poses a Cousin-like problem. Second, the Cousin problem is solved to obtain connection functions that solve the homogeneous form of the equation and smoothly patch together the local solutions. The connection functions are determined based on boundary information. This technique is similar to the superposition, filtering, and patching technique of Israeli *et al.* [8] but uses an exact representation of the connection functions and results in a fully coupled system. In two dimensions, for rectangular computational domains, each connection function can be decomposed into four components corresponding to the boundary values on the four edges of the rectangle. In each direction the solutions are coupled at the local boundaries and lead to a set of simultaneous equations that can be solved for the coefficients in a sine series expansion. Solutions in multiple dimensions can be achieved by recursively treating the connection functions in each dimension. Hence a three-dimensional solution requires three applications of the procedure.

2 Definitions, Cousin's Problem, and Local Solutions

Let Ω be the domain of interest, and let $\{\Omega_i\}$ be an open covering of Ω such that the subdomain boundaries $\partial\Omega_i$ are regular. We consider the linear Poisson equation $\Delta p = f$ in Ω with Dirichlet boundary values $p = b$ on $\partial\Omega$ as model problem. We assume $f \in C(\overline{\Omega})$ and $b \in C(\partial\Omega)$ for real dimension $n \geq 2$.

First, the local Poisson problems are solved

$$\Delta q_i = f \quad \text{in} \quad \Omega_i \quad \text{with} \quad q_i = \tilde{b}_i \quad \text{on} \quad \partial\Omega_i \tag{1}$$

where $\tilde{b}_i$ is a non-unique continuous extension of b on $\partial\Omega_i$, and second, a set of connection functions

$$\Delta h_i = 0 \quad \text{in} \quad \Omega_i \quad \text{with} \quad h_i = \tilde{c}_i \quad \text{on} \quad \partial\Omega_i \tag{2}$$

are calculated which solve a Cousin-like problem generated by the mismatch in the local solutions. The calculation of the $\tilde{c}$'s will be discussed in section 3. We first give a statement of Cousin's Problem tailored to our discussion.

Cousin's Problem. Suppose for every nonempty intersection $\Omega_i \cap \Omega_j \neq \emptyset$ we are given a function $h_{i\,j} \in \mathbf{C}(\overline{\Omega}_i \cap \overline{\Omega}_j)$ such that $\Delta h_{i\,j} = 0$ in $\Omega_i \cap \Omega_j$ and $h_{i\,j} = 0$ on $\partial\Omega_i \cap \partial\Omega_j \cap \partial\Omega$, and such that the following *cocycle* property is satisfied:

$$h_{i\,i} = 0 \qquad h_{i\,j} = -h_{j\,i} \qquad h_{i\,j} + h_{j\,k} = h_{i\,k} \tag{3}$$

The Cousin Problem with data $\{h_{i\,j}\}$ is the problem of finding a set of functions $\{h_i\}$ such that $\Delta h_i = 0$ in Ω_i , $h_i = 0$ on $\partial\Omega_i \cap \partial\Omega$, and $h_{i\,j} = h_j - h_i$ in

$\overline{\Omega}_i \cap \overline{\Omega}_j$. Under conditions of sufficient regularity, this problem can be shown to have a unique solution. For a statement of the classical Cousin problem see Gunning and Rossi [7].

If $\{q_i\}$ is a complete set of exact solutions to the subdomain problems (1), then their differences $h_{i\,j} = q_i - q_j$ satisfy the hypothesis of Cousin's problem. Once Cousin's problem is solved and the functions h_i are found, the global solution p is defined by $p = q_i + h_i$ on each subdomain Ω_i . Since

$$q_i - q_j = h_{i\,j} = h_j - h_i, \tag{4}$$

p is consistently defined and solves the global problem. The key to the procedure is the efficiency of the connection function computations.

Generation of accurate local solutions is critical to overall accuracy of the technique and may limit the final order of accuracy. We use two methods for calculating the local solutions: first, a direct solution via a Schur decomposition [3], and second a pseudo-spectral Fourier method in combination with a boundary regularization procedure to subtract aperiodicity in the boundary conditions [12]. The local solutions can be difficult to compute in that they arise from local problems that are nonperiodic and are subject to the interpolated boundary conditions. A range of spectral methods has been developed to approach this problem [1, 11].

In the first method the solution is determined by the finite difference expansion and, in this case, results in a second order truncation error and a second order accurate solution. The second method is outlined in Zarantonello, Fabris, and Chiappari [12]. In this approach the right hand side is preconditioned by subtracting the aperiodic behavior of f , the right-hand side in Poisson's equation, and then applying a conventional spectral solver. This technique was introduced by Sköllermo [11] and has also been considered by Averbuch *et al.* [1].

3 Examples and Results

Let $\Omega = (a,b) \times (c,d) = \left\{(x,y) \in \mathbf{R^2} \mid a < x < b \; , \; c < y < d\right\}$, be a rectangular domain, and $\{\Omega_n\}$ be a finite open coverings of Ω , consisting of rectangles $\Omega_n = (a_n, b_n) \times (c,d)$ where $a_n < b_{n-1} < a_{n+1} < b_n$. We refer to $\{\Omega_n\}$ as a rectangular domain decomposition of Ω and note that the subdomains overlap and are horizontally aligned, Figure 1. Higher dimensional arrays of overlapping domains can be obtained in a similar manner.

In the case of two subdomains Ω_1 and Ω_2 , let Γ_2^L and Γ_1^R be their respective interior boundaries as shown in Figure 1. The Cousin data consist of the single function $h_{1\,2} = q_1 - q_2$ defined on $\Omega_1 \cap \Omega_2$. Since the local solutions $\{q_i\}$ have boundary values that agree on $\partial\Omega_1 \cap \partial\Omega_2$, the nonzero boundary values of $h_{1\,2}$ are the mismatch of the local solutions on $\Gamma_2^L \cup \Gamma_1^R$. The connection functions, h_1 and h_2 , are solutions to $\Delta h = 0$ and can be directly generated from the non-zero boundary values, and therefore, can be expressed in an orthogonal basis appropriate to the specific domain, in this case an expansion in sines and hyperbolic sines. For the purpose of illustration let

$$h_1(x,y) = \sum_{k=1}^{\infty} R_k \frac{\sinh k\pi(x - a_1)}{\sinh k\pi(b_1 - a_1)} \sin k\pi y \tag{5}$$

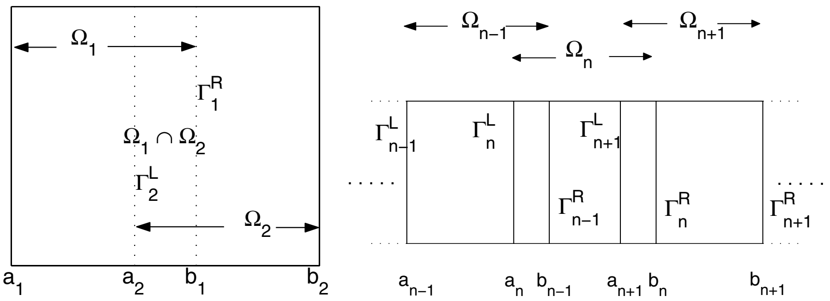


Fig. 1. Two subdomains and a row of subdomains.

and

$$h_2(x,y) = \sum_{k=1}^{\infty} L_k \frac{\sinh k\pi(b_2 - x)}{\sinh k\pi(b_2 - a_2)} \sin k\pi y \tag{6}$$

in the rectangular subdomains. Here $\{L_k\}$ and $\{R_k\}$ are the coefficients derived from the sine expansions of the boundary data on the left and right edges of each rectangular domain. Now, equation (4) is used to identify the boundary data

$$q_1\,|_{\Gamma_2^L} - q_2\,|_{\Gamma_2^L} = h_{12}\,|_{\Gamma_2^L} = h_2\,|_{\Gamma_2^L} - h_1\,|_{\Gamma_2^L} \tag{7}$$

and

$$q_2\,|_{\Gamma_1^R} - q_1\,|_{\Gamma_1^R} = h_{12}\,|_{\Gamma_1^R} = h_1\,|_{\Gamma_1^R} - h_2\,|_{\Gamma_1^R}. \tag{8}$$

Using the data from q_1 and q_2, we expand h_{12} in the same sine basis as (5) and (6)

$$h_{12}(x,y) = \sum_{k=1}^{\infty} \left[T_k \frac{\sinh k\pi(x - a_2)}{\sinh k\pi(b_1 - a_2)} - S_k \frac{\sinh k\pi(b_1 - x)}{\sinh k\pi(b_1 - a_2)} \right] \sin k\pi y. \tag{9}$$

Using the eigenexpansions in (5), (6), and (9), equations (7) and (8) decouple in each wave number and can be solved simultaneously for each pair of expansion coefficients, L_k and R_k, independently.

In the more general case of multiple domains, Ω_n, connected in a row with only overlap of two adjacent domains, the simultaneous system reduces to

$$\begin{bmatrix} D_{1,k} & 1 & 0 & 0 & 0 & .\,. & 0 \\ 1 & A_{2,k} & B_{2,k} & 0 & 0 & .\,. & 0 \\ 0 & C_{2,k} & D_{2,k} & 1 & 0 & .\,. & 0 \\ 0 & 0 & 1 & A_{3,k} & B_{3,k} & .\,. & 0 \\ 0 & 0 & 0 & C_{3,k} & D_{3,k} & .\,. & 0 \\ . & . & . & . & . & .\,. & . \\ 0 & 0 & 0 & . & . & .\,1 & A_{n,k} \end{bmatrix} \begin{bmatrix} R_{1,k} \\ L_{2,k} \\ R_{2,k} \\ L_{3,k} \\ R_{3,k} \\ \cdots \\ L_{n,k} \end{bmatrix} = \begin{bmatrix} S_{1,k} \\ T_{2,k} \\ S_{2,k} \\ T_{3,k} \\ S_{3,k} \\ \cdots \\ T_{n,k}, \end{bmatrix} \tag{10}$$

where

$$A_{n,k} = -\frac{\sinh k\pi(b_n - b_{n-1})}{\sinh k\pi(b_n - a_n)}, \qquad D_{n,k} = -\frac{\sinh k\pi(a_{n+1} - a_n)}{\sinh k\pi(b_n - a_n)}, \tag{11}$$

$$B_{n,k} = -\frac{\sinh k\pi(b_{n-1} - a_n)}{\sinh k\pi(b_n - a_n)}, \qquad C_{n,k} = -\frac{\sinh k\pi(b_n - a_{n+1})}{\sinh k\pi(b_n - a_n)}.$$

In this system each row is normalized by the hyperbolic sine factors in equation (9). In the internal subdomains, the subdomains that overlap with two other subdomains, the connection function contains both left and right coefficients.

Since the subdomains overlap, the determinant of the matrix in (10) is strictly positive and uniformly bounded away from zero. The solution $\{h_n\}$ depends continuously on the data $\{h_{n,n-1}\}$, and the procedure for solving the Cousin Problem is stable. The elements, equation (11), of the matrix in equation (10) are determined purely by the nature of the overlap, and the connection function in any one subdomain depends only on the local solutions in all of the other subdomains, but not on the other connection functions, $R_{j,k} = f_j(\{S_k\}, \{T_k\})$ and $L_{j,k} = g_j(\{S_k\}, \{T_k\})$. In essence, this reduces the iterative problem to a two-step direct solution.

In a computational framework using a finite expansion each function $h_{n,n-1} = q_n - q_{n-1}$ is defined on a collocation grid of dimensions $M_{x;n,n-1} \times M_y$, the sine series (5), (6), and (9) become sine polynomials of order M_y , the sine coefficients are calculated via a Fast Sine Transform, the equation coefficients (11) are precalculated, and (10) reduces to M_y systems of tridiagonal equations in $2N-2$ unknowns.

We consider two test cases to demonstrate the method. In [5] Garbey and Tromeur-Dervout proposed the problem $f(x,y) = 2y(y-1) + 2x^2 - 0.5$, with the exact solution $p(x,y) = (x^2 - 0.25)y(y-1)$ defined on the unit square. This particular case allows an exact solution to be computed with a second order discretization. Results are given in Table 1. Our L_∞ errors are comparable to those of Garbey and Tromeur-Dervout and are on the order of machine precision. In this case the truncation error is exactly zero for the 2^{nd} order technique.

Table 1. First case considered, L_2 and L_∞ relative to the same norms for the solution.

total nodes (subdomains)	spectral err_{L_2}	2^{nd} order err_{L_2}	2^{nd} order err_{L_∞}	err_{L_∞}, Garbey & Tromeur-Dervout
66×66 (2 × 2)	7.8501e-10	4.7668e-14	4.3161e-14	3.8589e-13
66×66 (4 × 4)	4.2206e-9	4.7668e-14	4.9009e-14	4.2577e-14
66×66 (8 × 8)	1.8064e-8	9.5740e-14	6.0633e-14	2.2204e-15
66×66 (16 × 16)	7.2144e-8	2.2889e-13	1.1517e-13	1.1380e-15
258×258 (2 × 2)	3.0977e-12	7.5506e-13	5.6097e-13	1.3513e-11
258×258 (4 × 4	1.6964e-11	1.1543e-12	9.0958e-13	2.8467e-12
258×258 (8 × 8)	7.3314e-11	1.4259e-12	1.1116e-12	1.3563e-12
258×258 (16 × 16)	2.9883e-10	9.0078e-13	6.5830e-13	8.4238e-14

Second, we consider $f(x,y) = 6\,e^{x+y}\,x\,y\,(-3+y+x+x\,y)$, with the exact solution $p(x,y) = 3\,e^{x+y}\,x\,y\,(1-x)(1-y)$. This is Problem 4.1 of Rice *et al.* [9]. Results are shown in Table 2. It is a commonly used example of an analytic problem with homogeneous Dirichlet boundary values. Rice [9] *et al.* consider only a single subdomain setting. The spectral solution is more accurate due to a higher order approximation. Figure 2 shows the solution and error for the third case in Table 2.

The results are given for two local solvers, the spectral and Schur decomposition. We note that the Schur decomposition provides the exact solution for the first problem since the truncation error is exactly zero. The results approach machine

Table 2. Second case considered. The spectral method uses a local solution that is fourth order accurate.

total nodes (subdomains)	spectral err_{L_2}	spectral err_{L_∞}	2^{nd} order err_{L_2}
18×18 (2 × 2)	2.8047e-7	1.3978e-6	3.6947e-4
34×34 (2 × 2)	1.0409e-8	9.6288e-8	8.2776e-5
66×66 (2 × 2)	3.6829e-10	6.3610e-9	1.9237e-5
130×130 (2 × 2)	1.2921e-11	4.0955e-10	4.6102e-6
258×258 (2 × 2)	4.7057e-13	2.5989e-11	1.1266e-6

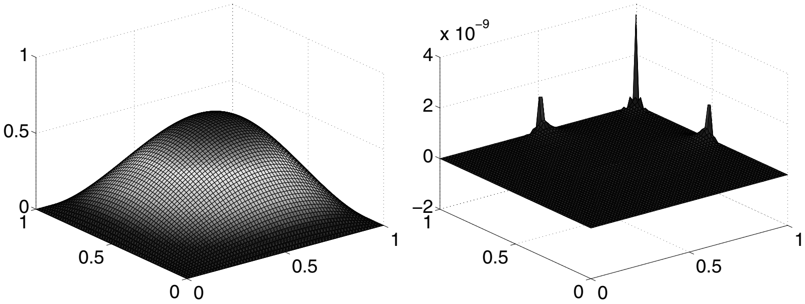


Fig. 2. Solution and error, Rice *et al.* problem.

accuracy and are comparable with or better than the best published results for this particular problem. Figure 3 shows the application of the procedure to an L-shaped domain.

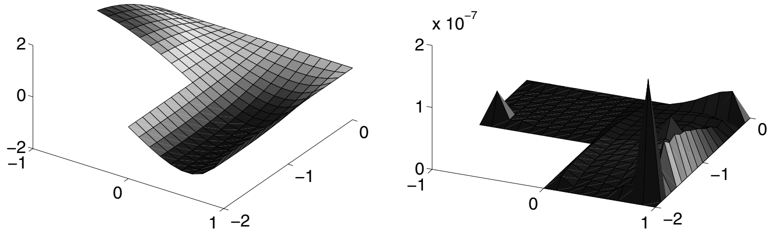


Fig. 3. Two rectangular domains overlapping into an L-shape: solution and error.

4 Conclusions

A direct method for solution of linear elliptic problems through domain decomposition has been presented as applied to the Poisson equation. The procedure calculates local solutions on each domain and computes connection functions as solutions to a Cousin problem that correct for the mismatch on the internal boundaries and overlap domains.

In regular rectangular decompositions, eigenfunction expansions for the connection boundary value problem can be calculated directly with each component separately. In two dimensions (or more) each direction needs to be calculated successively with subsequent Cousin problems generated after calculation of the initial connection functions. In L-shaped and more complicated domains the connection functions can still be calculated but require full coupling of all the local eigenfunctions.

The final procedure is as accurate as the accelerated Schwarz procedure with the order of accuracy determined by the local solution procedure. The benefit of the procedure is the improvement in the calculation of the connection functions, directly through eigenexpansions, that is half as expensive as a second iteration of the Schwarz procedure. Matching the solution in the overlap provides a stronger condition than transmission of boundary data. Furthermore, the procedure allows for the computation of connection solutions in more complicated domains and successively in two or more dimensions.

References

1. A. Averbuch, M. Israeli, and L. Vozovoi, *A fast Poisson solver of arbitrary order accuracy in rectangular regions*, SIAM J. Sci. Comput., 19 (1998), pp. 933–952.
2. J. Baranger, M. Garbey, and F. Oudin-Dardun, *On Aitken like acceleration of Schwarz domain decomposition method using generalized Fourier*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003, pp. 341–348.
3. C. Canuto, M. Y. Hussaini, A. Quarteroni, and T. A. Zang, *Spectral Methods in Fluid Dynamics*, Springer-Verlag, 1988.
4. M. J. Gander, L. Halpern, and F. Nataf, *Optimized Schwarz methods*, in Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan, T. Chan, T. Kako, H. Kawarada, and O. Pironneau, eds., Bergen, 2001, Domain Decomposition Press, pp. 15–28.
5. M. Garbey and D. Tromeur-Dervout, *Aitken-Schwarz method on Cartesian grids*, in Proc. Int. Conf. on Domain Decomposition Methods DD13, N. Debit, M. Garbey, R. Hoppe, J. Périaux, and D. Keyes, eds., CIMNE, 2002, pp. 53–65.
6. ———, *On some Aitken-like acceleration of the Schwarz method*, Internat. J. Numer. Methods in Fluids, 40 (2002), pp. 1493–1513.
7. R. C. Gunning and H. Rossi, *Analytic functions of several complex variables*, Prentice-Hall Series in Modern Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1965.
8. M. Israeli, L. Vozovoi, and A. Averbuch, *Spectral multidomain technique with local Fourier basis*, J. Sci. Comput., 8 (1993), pp. 135–149.

9. J. R. Rice, E. N. Houstis, and W. R. Dyksen, *A population of linear, second order, elliptic partial differential equations on rectangular domains, part I*, Math. Comp., 36 (1981), pp. 475–484.
10. H. A. Schwarz, *Über einen Grenzübergang durch alternierendes Verfahren*, Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, 15 (1870), pp. 272–286.
11. G. Sköllermo, *A Fourier method for numerical solution of Poisson's equation*, Math. Comp., 29 (1975), pp. 697–711.
12. S. Zarantonello, D. Fabris, and S. Chiappari, *The use of a redundant basis for the spectral solution of Poisson's equation.* In preparation.

Solving Frictional Contact Problems with Multigrid Efficiency

Konstantin Fackeldey and Rolf H. Krause

Institute for Numerical Simulation, Wegelerstr. 6, D-53115 Bonn, Germany.
{fackeldey,krause}@ins.uni-bonn.de

1 Introduction

The construction of fast and reliable solvers for contact problems with friction is even nowadays a challenging task. It is well known that contact problems with Coulomb friction have the weak form of a quasi-variational inequality [8, 6, 15]. For small coefficients of friction, a solution can be obtained by means of a fixed point iteration in the boundary stresses [15]. This fixed point approach is often used for the construction of numerical methods, since in each iteration step only a constrained convex minimization problem has to be solved [2, 14]. Unfortunately, the convergence speed of the discrete fixed point iteration deteriorates for smaller meshsizes. Here, we present a new multigrid method which removes the outer fixed point iteration and gives rise to a highly efficient solution method for frictional contact problems with Coulomb friction and other local friction laws in two and three space dimensions. The numerical cost is comparable to those of frictionless contact problems. Our method is based on monotone multigrid methods, see [11], and does not require any regularization of the non-penetration condition or of the friction law. Therefore, the results are highly accurate. Using the basis transformation given in [16], our method can also be applied to two body contact problems.

2 Elastic Contact with Coulomb Friction

In this section, we give the strong and the weak formulation of the contact problem with Coulomb friction between a deformable body and a rigid foundation. We identify the body in its reference configuration with the domain $\Omega \subset \mathbb{R}^d, d = 2, 3$. The boundary $\partial\Omega$ is decomposed into three disjoint parts, Γ_D, Γ_N , and Γ_C . The actual zone of contact is assumed to be contained in Γ_C but is not known in advance. We assume $\text{meas}_{d-1}(\Gamma_D) > 0$ and denote tensor and vector quantities by bold symbols, e.g., $\boldsymbol{v}$, and the components by v_i , $1 \leq i, j \leq d$ and $(\cdot)_{,j} = \partial/\partial x_j(\cdot)$. The summation convention is enforced on indices $1 \leq i, j \leq d$. We define the usual Sobolev space of displacements with weak derivative in $\boldsymbol{L}^2$ by $\boldsymbol{H}^1(\Omega) := (H^1(\Omega))^d$ and set $\boldsymbol{H}_D := \{v \mid v \in \boldsymbol{H}^1(\Omega), v_{|_{\Gamma_D}} = \mathbf{0}\}$. We consider linear elastic material, i.e., the stresses $\boldsymbol{\sigma} = (\sigma_{ij})_{i,j=1}^d$ are given by Hooke's law $\sigma_{ij}(\boldsymbol{u}) := E_{ijml}\, u_{l,m}$. Here, Hooke's tensor $\mathbf{E} = (E_{ijml})_{i,j,l,m=1}^{d}$, $E_{ijlm} \in L^{\infty}(\varOmega)$, $1 \leq i,j,l,m \leq d$ is sufficiently smooth, symmetric and positive definite. On $\partial\varOmega$ the normal and tangential displacements are defined by $u_n = \boldsymbol{u}\cdot\boldsymbol{n}$ and $\boldsymbol{u}_T = \boldsymbol{u} - u_n \cdot \boldsymbol{n}$, where $\boldsymbol{n}$ is the outer normal vector. Similarly, $\sigma_n = n_i \sigma_{ij} n_j$ and $(\boldsymbol{\sigma}_T)_i = \sigma_{ij} n_j - \sigma_n \cdot \boldsymbol{n}$ are the normal and tangential stresses, respectively. Let $g : \mathbb{R}^d \supset \varGamma_C \to \mathbb{R}$ be a a continuous function giving the distance to the foundation, taken in the normal direction with respect to the reference configuration. Then, for small deformations, we can say that the body $\varOmega$ does not penetrate the rigid foundation if we have $u_n(x) \leq g(x)$ for all $x \in \varGamma_C$, see e.g., [8].

At the points, where the body comes into contact with the foundation, friction may occur. Here we use the Coulomb law of friction. It states that the force, which is needed to move a body lengthwise over a rigid foundation, is proportional to the force pushing the body perpendicular onto the foundation. The boundary value problem constituting the elastic contact problem with friction consists of the equilibrium condition (1) in $\varOmega$, the boundary conditions (2) and (3) on $\varGamma_D$ and $\varGamma_N$, the contact conditions (5), (6) on $\varGamma_C$ and the Coulomb law of friction (7), (8) on $\varGamma_C$. In equations (7), (8), $|\cdot|$ is the Euclidean norm on $\mathbb{R}^{d-1}$. Here, we assume sufficiently smooth data and for the coefficient of friction holds $\mathcal{F} \in L^{\infty}(\varGamma_C)$ and $\mathcal{F} \geq \mathcal{F}_0 > 0$ on $\varGamma_C$.

$$-\sigma_{ij}(\boldsymbol{u})_{,j} = f_i \quad \text{in } \varOmega \tag{1}$$
$$\boldsymbol{u} = 0 \quad \text{on } \varGamma_D \tag{2}$$
$$\sigma_{ij}(\boldsymbol{u}) \cdot n_j = p_i \quad \text{on } \varGamma_N \tag{3}$$
$$\sigma_n \leq 0 \tag{4}$$

$$\boldsymbol{u}\cdot\boldsymbol{n} \;\leq\; g \tag{5}$$
$$(\boldsymbol{u}\cdot\boldsymbol{n} - g)\,\sigma_n \;=\; 0 \tag{6}$$
$$\boldsymbol{u}_T = 0 \Rightarrow |\boldsymbol{\sigma}_T| < \mathcal{F}\,|\sigma_n| \tag{7}$$
$$\boldsymbol{u}_T \neq 0 \Rightarrow \boldsymbol{\sigma}_T \;=\; -\mathcal{F}|\sigma_n|\,\frac{\boldsymbol{u}_T}{|\boldsymbol{u}_T|} \tag{8}$$

By (7) and (8), Coulomb's law of friction is a local friction law, since the frictional response at $x \in \varGamma_C$ depends only on the stress developed at x. We can divide all points in the actual zone of contact into sticking and sliding points. A point x is called sticky, if no tangential displacement occurs, i.e., if $\boldsymbol{u}_T(x) = \boldsymbol{0}$. It is called sliding, if $\boldsymbol{u}_T(x) \neq \boldsymbol{0}$. We remark, that in (8) for $d = 2$ and $\boldsymbol{u}_T \neq \boldsymbol{0}$, we have $\boldsymbol{u}_T/|\boldsymbol{u}_T| \in \{-1, 1\}$. This is in contrast to the case $d = 3$, where $\boldsymbol{u}_T/|\boldsymbol{u}_T| \in S^2 = \{\boldsymbol{v} \mid \boldsymbol{v} \in \mathbb{R}^2, |\boldsymbol{v}| = 1\}$. In order to give the variational formulation of problem (1)–(8), let us define the bilinear form $a(\boldsymbol{u},\boldsymbol{v}) = \int_{\varOmega} \sigma_{ij}(\boldsymbol{u}) v_{i,j}\, dx$, the linear form $f(\boldsymbol{v}) = \int_{\varOmega} f_i v_i\, dx + \int_{\varGamma_N} p_i v_i\, ds$ and finally the closed convex set $\mathcal{K}$ of admissible displacements by

$$\mathcal{K} := \{\boldsymbol{v} \in \boldsymbol{H}_D \mid v_n \leq g \text{ a.e. on } \varGamma_C\}\,. \tag{9}$$

At the contact boundary the virtual work of the frictional forces is characterized by the nonlinear and non-differentiable functional $j \colon \boldsymbol{H}_D \times \boldsymbol{H}_D \to \mathbb{R}$

$$j(\boldsymbol{u},\boldsymbol{v}) = \int_{\varGamma_C} \mathcal{F}\,|\sigma_n(\boldsymbol{u})|\,|\boldsymbol{v}_T|\, ds\,. \tag{10}$$

Using these definitions, the weak formulation of the boundary value problem (1)-(8) is given by the quasi-variational inequality: find $\boldsymbol{u} \in \mathcal{K}$, such that

$$a(\boldsymbol{u}, \boldsymbol{v} - \boldsymbol{u}) + j(\boldsymbol{u},\boldsymbol{v}) - j(\boldsymbol{u},\boldsymbol{u}) \;\geq\; f(\boldsymbol{v} - \boldsymbol{u})\,, \quad \boldsymbol{v} \in \mathcal{K}\,, \tag{11}$$

see [3, 8, 6]. The functional j is non-convex, non-quadratic and non-differentiable. Thus, standard methods from convex analysis cannot be applied to gain a solution of the quasi-variational inequality (11).

3 A Multigrid Method for a quasi-variational Inequality

In [15, 6], the following fixed point iteration is considered: let $\boldsymbol{u}^0 \in \mathcal{K}$ be given. Then, for $k = 1, 2, \dots$, compute $\boldsymbol{u}^k$ as the unique solution of the variational inequality

$$a(\boldsymbol{u}^k, \boldsymbol{v} - \boldsymbol{u}^k) + j(\boldsymbol{u}^{k-1}, \boldsymbol{v}) - j(\boldsymbol{u}^{k-1}, \boldsymbol{u}^k) \ \geq\ f(\boldsymbol{v} - \boldsymbol{u}^k) \qquad \boldsymbol{v} \in \mathcal{K}\,. \tag{12}$$

Setting $\tau = -\sigma_n(\boldsymbol{u}^{k-1})$ and introducing

$$H_+^{-1/2} := \{v \in H^{-1/2}(\Gamma_C) \,|\, \langle v, w\rangle_{H^{-1/2}\times H^{1/2}} \geq 0\,, w \in H^{1/2}(\Gamma_C), w \geq 0\}\,,$$

(12) defines a mapping $\Psi\colon H_+^{-1/2} \to H_+^{-1/2}$ by $\Psi(\tau) = -\sigma_n(\boldsymbol{u}^k)$. This mapping can be used to establish the existence of a solution to problem (11) by a fixed point argument for sufficiently small $\mathcal{F}$, see [15, 6]. This fixed point iteration can be used for the numerical solution of (11), see [14, 12, 2, 5] and is also the starting point for our method.

In this section, we present a monotone multigrid method by means of which the variational inequality (12) can be solved efficiently. Moreover, we extend our method in a way that the fixed point iteration is removed. The resulting multigrid method can then be applied to the quasi-variational inequality (11) directly. In our numerical experiments, see Section 4, this method has been shown to be an iterative solution method for (11) with multigrid complexity.

Let $(\mathcal{T}_\ell)_{\ell=1}^L$ denote a family of nested and shape regular meshes with meshsize parameter h_ℓ. We use Lagrangian conforming finite elements $\boldsymbol{S}_\ell \subset \boldsymbol{H}_D$ of first order. The set of nodes of $\mathcal{T}_\ell$ is denoted by $\mathcal{N}^{(\ell)}$ and the nodal basis functions of S_ℓ are $\{\lambda_p^{(\ell)}\}_{p\in\mathcal{N}^{(\ell)}}$. The nodes on the possible contact boundary are $\mathcal{C}^{(\ell)} = \Gamma_C \cap \mathcal{N}^{(\ell)}$. As discretization of the convex set $\mathcal{K}$ we take

$$\mathcal{K}_L := \{\boldsymbol{u} \in \boldsymbol{S}_L \mid \boldsymbol{u}(p)\cdot\boldsymbol{n}(p) \leq g(p)\,, p \in \mathcal{C}^{(L)}\}\,.$$

Note, that in general $\mathcal{K}_L \not\subset \mathcal{K}$. We define the discrete normal stresses s_n for $\boldsymbol{u} \in \boldsymbol{S}_L$ on the basis of Greens theorem as the linear residual by

$$(s_n(\boldsymbol{u}))_p := r(\lambda_p^{(L)} \cdot \boldsymbol{n}) := a(\boldsymbol{u}, \lambda_p^{(L)} \cdot \boldsymbol{n}) - \boldsymbol{f}(\lambda_p^{(L)} \cdot \boldsymbol{n})\,, \tag{13}$$

cf. [6]. As discretization for the functional j we choose for $\boldsymbol{u}, \boldsymbol{v} \in \boldsymbol{S}_L$

$$j^L(\boldsymbol{u}, \boldsymbol{v}) := \sum\nolimits_{p\in\mathcal{C}^{(L)}} \mathcal{F}\,|(s_n(\boldsymbol{u}))_p|\,|\boldsymbol{v}_T(p)|\,. \tag{14}$$

Inserting the functional j^L, (12) gives rise to the discrete fixed point iteration: let $\boldsymbol{u}^0 \in \boldsymbol{S}_L$ be given. For $k = 1, 2, \dots$ solve

$$a(\boldsymbol{u}^k, \boldsymbol{v} - \boldsymbol{u}^k) + j^L(\boldsymbol{u}^{k-1}, \boldsymbol{v}) - j^L(\boldsymbol{u}^{k-1}, \boldsymbol{u}^k) \ \geq\ f(\boldsymbol{v} - \boldsymbol{u}^k)\,, \quad \boldsymbol{v} \in \mathcal{K}_L\,, \tag{15}$$

in each step of which a variational inequality has to be solved. We remark that for the contractivity constant C of the discrete fixed point iteration (15) holds

$C = C(h) = \mathcal{O}(h^{-1/2})$, see [5]. Thus, the iteration process slows down for decreasing meshsize h .

As starting point for our new multigrid method let us note that the solution $\boldsymbol{u}^k$ of (12) can equivalently be characterized as the unique minimizer of the convex functional

$$\hat{\mathcal{J}}_{\boldsymbol{u}^{k-1}}(\cdot) = J + j(\boldsymbol{u}^{k-1}, \cdot) + \varphi_{\mathcal{K}} = (\frac{1}{2}a(\cdot,\cdot) - f(\cdot)) + j(\boldsymbol{u}^{k-1}, \cdot) + \varphi_{\mathcal{K}}$$

over $\boldsymbol{H}_D$. Here, $\varphi_{\mathcal{K}}$ is defined by $\varphi_{\mathcal{K}}(\boldsymbol{u}) = +\infty$ for $\boldsymbol{u} \notin \mathcal{K}$ and zero for $\boldsymbol{u} \in \mathcal{K}$. As discretization of $\hat{\mathcal{J}}_{\boldsymbol{u}^{k-1}}$ we choose on the basis of (14) the functional $\hat{\mathcal{J}}^L_{\boldsymbol{u}^{k-1}} := J + j^L(\boldsymbol{u}^{k-1}, \cdot) + \varphi_{\mathcal{K}_L}$. Following [9, 11], we seek the minimizer of $\hat{\mathcal{J}}^L_{\boldsymbol{u}^{k-1}}$ over $\boldsymbol{S}_L$ by successive minimization in direction of a suitable multilevel basis. To this end, we associate with each node $p \in \mathcal{N}^{(\ell)}$, $1 \leq \ell \leq L$, the local subspace $\boldsymbol{V}_p = \operatorname{span}\{\lambda_p^{(\ell)} \cdot \boldsymbol{e}_1(p), \ldots, \lambda_p^{(\ell)} \cdot \boldsymbol{e}_d(p)\}$. For the nodes $p \in \mathcal{C}^{(L)}$ we choose $\boldsymbol{e}_1(p) = \boldsymbol{n}(p)$ and extend $\boldsymbol{e}_1(p)$ to an orthonormal basis of $\mathbb{R}^d$. For all remaining nodes we choose $\{\boldsymbol{e}_i(p)\}_{1 \leq i \leq d}$ to be the canonical basis vectors of $\mathbb{R}^d$ and set $\boldsymbol{V}_p = \operatorname{span}\{\mu_{p,1}^{(\ell)} \cdot \boldsymbol{e}_1(p), \ldots, \mu_{p,d}^{(\ell)} \cdot \boldsymbol{e}_d(p)\}$ with functions $\mu_{p,i}^{(\ell)}$ to be discussed later. We moreover assume an ordering $k = k(p, \ell)$ of all nodes on all levels to be given such that $k(p, r) \leq k(q, s)$ implies $r \geq s$ for $p, q \in \mathcal{N}^{(\ell)}$, $1 \leq \ell \leq L$. Now, we can introduce the multilevel splitting

$$\mathcal{S}^{(L)} = \boldsymbol{V}_{p_1} + \cdots + \boldsymbol{V}_{p_{n_L}} + \boldsymbol{V}_{p_{n_L+1}} + \cdots + \boldsymbol{V}_{p_M} , \tag{16}$$

where we have set $n_L = \#\mathcal{N}^{(L)}$ and the indices $n_L + 1, \ldots, M$ stand for the coarse grid corrections.

We first consider the successive minimization of $\hat{\mathcal{J}}^L_{\boldsymbol{u}^{k-1}}$ with respect to the leading subspaces $\boldsymbol{V}_{p_1} + \cdots + \boldsymbol{V}_{p_{n_L}}$. Let $\boldsymbol{w}^0 \in \mathcal{S}_L$ be given and set $\boldsymbol{w}_0 = \boldsymbol{w}^0$. For $1 \leq m \leq n_L$, the local minimization problem: find $\boldsymbol{v}_m \in \boldsymbol{V}_{p_m}$, such that

$$\hat{\mathcal{J}}^L_{\boldsymbol{u}^{k-1}}(\boldsymbol{v}_m + \boldsymbol{w}_{m-1}) \leq \hat{\mathcal{J}}^L_{\boldsymbol{u}^{k-1}}(\boldsymbol{v} + \boldsymbol{w}_{m-1}) , \quad \boldsymbol{v} \in \boldsymbol{V}_{p_m} , \tag{17}$$

is solved and update $\boldsymbol{w}_m = \boldsymbol{w}_{m-1} + \boldsymbol{v}_m$. Setting $\boldsymbol{w}^1 := \boldsymbol{w}_{n_L}$, by this nonlinear Gauß-Seidel method a sequence $\boldsymbol{w}^\nu$, $\nu = 0, 1, \ldots$, of iterates is defined which converges to the unique minimizer of $\hat{\mathcal{J}}^L_{\boldsymbol{u}^{k-1}}$, see [4, 11] and [7] for a related method using regularization. We refer to [13] for a detailed description how the local problems (17) can be solved efficiently.

The convergent but slow iteration (17) is now accelerated by additional minimization steps in the direction of coarse grid functions with larger support. In contrast to multigrid methods for linear problems, we cannot represent the functional $\hat{\mathcal{J}}^L_{\boldsymbol{u}^{k-1}}$ to be minimized on the coarser grids and therefore have to use nonlinear coarse grid corrections.

To this end, we introduce the smoothed fine grid iterate $\bar{\boldsymbol{u}}_L := \boldsymbol{w}^\nu$, which is obtained after $\nu > 0$ presmoothing steps of the nonlinear Gauß-Seidel method (17). Since the functional $\varphi_{\mathcal{K}_L} + j^L(\boldsymbol{u}^{k-1}, \cdot)$ is non-differentiable and non-quadratic, we restrict the coarse grid corrections to a neighborhood of the fine grid iterate $\bar{\boldsymbol{u}}_L$ where the functional $\hat{\mathcal{J}}^L_{\boldsymbol{u}^{k-1}}$ is smooth, c.f. [10]. We set $\mathcal{K}_{\bar{\boldsymbol{u}}_L} = \{\bar{\boldsymbol{u}}_L + \boldsymbol{v} \,|\, v_i \in [\underline{\psi}_{p,i}, \overline{\psi}_{p,i}], i = 1, \ldots, d\}$. The local obstacles $\underline{\psi}_{p,i}, \overline{\psi}_{p,i}$, can then be used to derive local obstacles for the coarse grid corrections by *monotone restrictions*, see[11]. For sliding nodes we set $\underline{\psi}_{p,1} = \overline{\psi}_{p,1} = 0$ and, for $2 \leq i \leq d$, $\underline{\psi}_{p,i} = -(\bar{\boldsymbol{u}}_L(p))_i, \overline{\psi}_{p,i} =$

$+\infty$, if $(\bar{\boldsymbol{u}}_L(p))_i > 0$ and $\underline{\psi}_{p,i} = -\infty, \overline{\psi}_{p,i} = -(\bar{\boldsymbol{u}}_L(p))_i$, if $(\bar{\boldsymbol{u}}_L(p))_i < 0$. For sticky nodes we set $\underline{\psi}_{p,i} = \overline{\psi}_{p,i} = 0$ for $1 \le i \le d$. We now define on $\mathcal{K}_{\bar{\boldsymbol{u}}_L}$ the functional

$$j_{\bar{\boldsymbol{u}}_L}(\boldsymbol{w}) = \sum\nolimits_{p\in\mathcal{C}^{(L)}} \mathcal{F}\,|(s_n(\boldsymbol{u}^{k-1}))_p|\,|\boldsymbol{w}_T(p)|\,,$$

and the quadratic energy functional $\bar{\mathcal{J}}^L_{\bar{\boldsymbol{u}}_L}$ by

$$\bar{\mathcal{J}}^L_{\bar{\boldsymbol{u}}_L} = \frac{1}{2}(a(\cdot,\cdot) + j''_{\bar{\boldsymbol{u}}_L}(\bar{\boldsymbol{u}}_L)(\cdot,\cdot)) - (f(\cdot) - j'_{\bar{\boldsymbol{u}}_L}(\bar{\boldsymbol{u}}_L)(\cdot) + j''_{\bar{\boldsymbol{u}}_L}(\bar{\boldsymbol{u}}_L)(\bar{\boldsymbol{u}}_L,\cdot))\,.$$

The resulting constrained quadratic problem: find $\boldsymbol{w} = \bar{\boldsymbol{u}}_L + \boldsymbol{c} \in \mathcal{K}_{\bar{\boldsymbol{u}}_L}$ with

$$\bar{\mathcal{J}}^L_{\bar{\boldsymbol{u}}_L}(\boldsymbol{w}) \le \bar{\mathcal{J}}^L_{\bar{\boldsymbol{u}}_L}(\boldsymbol{v})\,, \quad \boldsymbol{v} \in \mathcal{K}_{\bar{\boldsymbol{u}}_L}\,, \tag{18}$$

requires an minimization problem with constraints in both, normal and tangential direction, to be solved. These constraints stem from the non-penetration condition and the friction law, respectively. The additional minimization steps for $m > n_L$ of our splitting (16) are now done with respect to the energy (18). In order to improve the convergence speed of our method, we use *truncated basis functions* $\boldsymbol{\mu}_p^{(\ell)}$, $\ell < L$, which depend on $\bar{\boldsymbol{u}}_L$ and allow for representing the actual guessed contact boundary on coarser grids, see [11]. The resulting multigrid method can be implemented as a $\mathcal{V}$-cycle, see [11, 12]. It does not require any regularization, neither of the non-penetration condition nor of the functional j. Moreover, no algorithmic parameters such as damping or regularization parameters have to be chosen. We now present the two algorithms to be compared in the next section.

Algorithm 1 (Fixed point iteration with multigrid method)

```
Initialize: $\boldsymbol{u}^0 = \boldsymbol{0}$.      for $k = 1, \ldots, k_{\max}$ do
    Find $\boldsymbol{u}^k \in \mathcal{K}_L$ by applying sufficiently many multigrid steps to:
    $a(\boldsymbol{u}^k, \boldsymbol{v} - \boldsymbol{u}^k) + j^L(\boldsymbol{u}^{k-1}, \boldsymbol{v}) - j^L(\mathbf{u}^{k-1}, \boldsymbol{u}^k) \ge f(\boldsymbol{v} - \boldsymbol{u}^k)$,   $\boldsymbol{v} \in \mathcal{K}_L$.
    if $|s_n(\boldsymbol{u}^{k-1}) - s_n(\boldsymbol{u}^k)|/|s_n(\boldsymbol{u}^k)| \le \mathrm{TOL}$ break
    Compute the normal stress $s_n(\boldsymbol{u}^k)$ as linear residual.
end
```

Algorithm 2 (Multigrid method for frictional contact)

```
Initialize: $\boldsymbol{w}^0 = \boldsymbol{u}^0 = \boldsymbol{0}$.
Find $\boldsymbol{u}_L \in \mathcal{K}_L$ by doing sufficiently many of the following multigrid steps:
    for $m = 1, \ldots, n_L$ do
        Find $\boldsymbol{v}_m \in \boldsymbol{V}_{p_m}$, such that
            $\hat{\mathcal{J}}^L_{\boldsymbol{w}_{m-1}}(\boldsymbol{v}_m + \boldsymbol{w}_{m-1}) \le \hat{\mathcal{J}}^L_{\boldsymbol{w}_{m-1}}(\boldsymbol{v} + \boldsymbol{w}_{m-1})$,   $\boldsymbol{v} \in \boldsymbol{V}_{p_m}$.
        Update $\boldsymbol{w}_m = \boldsymbol{w}_{m-1} + \boldsymbol{v}_m$.
    end
    Set $\bar{\boldsymbol{u}}_L = \boldsymbol{w}_{n_L}$ and compute coarse grid correction $\boldsymbol{c}$ with respect to $\bar{\mathcal{J}}_{\bar{\boldsymbol{u}}_L}$
    Set $\boldsymbol{u}_L = \bar{\boldsymbol{u}}_L + \boldsymbol{c}$
```

Algorithm 2 constitutes our new multigrid method for the quasi-variational inequality (11). It has converged in our numerical experiments, even when the exact fixed point iteration method failed, see Section 4.

4 Numerical Results

In this section we present numerical results for a Hertzian contact problem in two space dimensions. We consider a half circle in contact with a rigid foundation. The half circle is centered at $(0, 0.4)$ with radius 0.4 and we have chosen $E = 2000$ and $\nu = 0.28$ as material parameters. We prescribe vertical displacement $u(x, y) = -0.005$ at the upper boundary $\Gamma_D = \{(x, y) \in \Gamma \big| y = 0.4\}$ and set $\Gamma_C = \partial\Omega \backslash \Gamma_D$ and $\boldsymbol{n}(p) = (0, -1)^T$ for $p \in \mathcal{C}^{(\ell)}$. We discretize with linear and bilinear finite elements on triangles and quadrilaterals, respectively and chose $\mathcal{T}_1$ as depicted in the left of Figure 2. A multilevel hierarchy is created up to Level $L = 11$ by successive adaptive refinement. We now investigate the convergence properties of Algorithm 1 and of our Algorithm 2. In Table 1 and the right of Figure 1, the number of outer iteration steps of Algorithm 1 is given. Here, we have set $\mathrm{TOL} = 10^{-9}$ in Algorithm 1. As expected from the theory, for large coefficients of friction and small meshsize parameters we have no convergence of Algorithm 1. Note, that for $\mathcal{F} = 5.0$ our multigrid method 2 converges even for small meshsizes. In order to compare the total amount of work needed using Algorithm 1 and Algorithm 2, respectively, we compare the total number of $\mathcal{V}$- cycles needed to obtain a solution of (12) on each level. For the fixed point iteration we add the number of $\mathcal{V}$-Cycles in each iteration step per level. For our multigrid method we simply take the total number of iterations. In both cases the iteration is stopped, if for two consecutive iterates $\boldsymbol{u}^{\nu}, \boldsymbol{u}^{\nu+1}$ holds $\|\boldsymbol{u}^{\nu} - \tilde{\boldsymbol{u}}^{\nu+1}\|_a \leq 10^{-11}$ with $\|\boldsymbol{u}\|_a^2 = a(\boldsymbol{u}, \boldsymbol{u})$. As starting value we always use $\boldsymbol{u}^0 = 0$. As can be seen from the left picture in Figure 1, our method 2 shows multigrid efficiency (solid line) and is of much higher efficiency than the fixed point iteration (dashed line). The number of needed $\mathcal{V}$-cycles for our method 2 is independent of h. Finally, the middle picture of Figure 2 shows the tangential and normal stresses (larger values) for $\mathcal{F} = 0.4$. As can be seen, sliding and sticky nodes are clearly identified by our method. The implementation of our method is in the framework of the FEM-toolbox UG, see [1] and is also applicable to complicated geometries and unstructured grids, as can be seen in Figure 2. As $3d$ example, here the displacements of a deformed cork in frictional contact with a surrounding bottle, are shown. The reference configuration is the transparent surface. For a more detailed discussion and elastic contact we refer to [13].

$\mathcal{F}\backslash$#dof	94	830	4.67	44.20	334.82	$\mathcal{F}\backslash$#dof	94	830	4.67	4.420	334.82
0.08	1	5	6	6	6	1.1	1	6	10	16	18
0.4	1	5	10	10	10	2.0	1	6	8	19	33
0.7	1	5	8	12	13	5.0	1	2	2	NC	NC
0.9	1	6	9	15	15	8.0	1	2	2	NC	NC
1.0	1	6	10	10	16	40.0	1	2	2	NC	NC

Table 1. Number of exact fixed point iterations on Level $\ell = 3, 5, 7, 9, 11$.

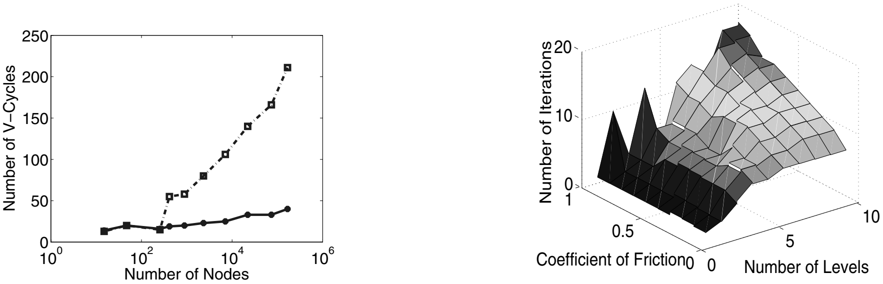


Fig. 1. Left: comparison of exact fixed point iteration Algorithm 1 and multigrid method Algorithm 2. Right: behavior of the fixed point iteration

References

1. 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*, Comp. Vis. Sci., 1 (1997), pp. 27–40.
2. Z. Dostál, J. Haslinger, and R. Kučera, *Implementation of fixed point method for duality based solution of contact problems with friction*, J. Comput. Appl. Math., 140 (2002), pp. 245–256.
3. G. Duvaut and J. L. Lions, *Inequalities in mechanics*, Springer-Verlag, New York, 1976.
4. R. Glowinski, *Numerical methods for nonlinear variational problems*, Series in Computational Physics, Springer, New York, 1984.
5. P. Hild, *On finite element uniqueness studies for Coulomb's frictional contact model*, Int. J. Appl. Math. Comput. Sci., 12 (2002), pp. 41–50.

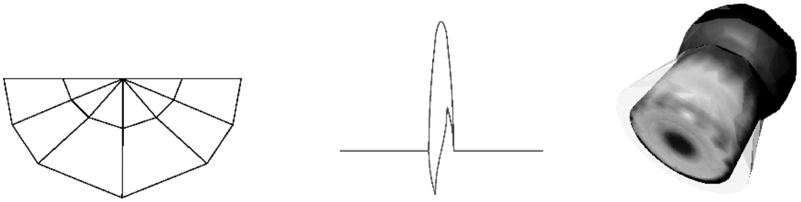


Fig. 2. Left: $\mathcal{T}_1$ Middle: boundary stress Right: $3d$ example

6. I. Hlaváček, J. Haslinger, J. Nečas, and J. Lovíšek, *Solution of Variational Inequalities in Mechanics*, vol. 66 of Applied Mathematical Sciences, Springer-Verlag New York, 1988.
7. F. Jourdan, P. Alart, and M. Jean, *A Gauss-Seidel like algorithm to solve frictional contact problems*, Comput. Methods. Appl. Mech. Engrg., 155 (1998), pp. 31–47.
8. N. Kikuchi and J. T. Oden, *Contact problems in elasticity: a study of variational inequalities and finite element methods*, SIAM, Philadelphia, 1988.
9. R. Kornhuber, *Adaptive Monotone Multigrid Methods for Nonlinear Variational Problems*, Teubner, Stuttgart, 1997.
10. ———, *On constrained Newton linearization and multigrid for variational inequalities*, Numer. Math., 91 (2002), pp. 699–721.
11. R. Kornhuber and R. Krause, *Adaptive multilevel methods for Signorini's problem in linear elasticity*, Comp. Visual. Sci., 4 (2001), pp. 9–20.
12. R. H. Krause, *Monotone multigrid methods for Signorini's problem with friction*, PhD thesis, Freie Universität Berlin, 2001.
13. ———, *Efficient solution of frictional contact problems without regularization.* In preparation, 2006.
14. C. Licht, E. Pratt, and M. Raous, *Remarks on a numerical method for unilateral contact including friction*, Internat. Ser. Numer. Math., 101 (1991), pp. 129–144.
15. J. Nečas, J. Jarušek, and J. Haslinger, *On the solution of the variational inequality to the Signorini problem with small friction*, Boll. Un. Mat. Ital., 17 (1980), pp. 796–811.
16. B. I. Wohlmuth and R. H. Krause, *Monotone multigrid methods on nonmatching grids for nonlinear multibody contact problems*, SIAM J. Sci. Comput., 25 (2000), pp. 324–347.

The Approximate Integration in the Mortar Method Constraint

Silvia Falletta

Dipartimento di Matematica e Applicazioni, Università di Milano-Bicocca, 20126, Milano, Italy. falletta@dimat.unipv.it

Summary. The paper analyzes the approximation of the weak continuity constraint in the mortar method, and provides error estimates in the L^2-norm and H^1 broken norm for generic discretization spaces, treating the presence of cross-points in the geometrical decomposition.

1 Introduction

The mortar element method is a nonoverlapping nonconforming domain decomposition technique for solving PDEs that weakens the continuity constraint of the solution by allowing jumps across the interfaces of the subdomains. Recently it has become of great interest especially for its flexibility in allowing the coupling of different physical models, the use of different discretization schemes and nonmatching grids at the interfaces of the decomposition. An important aspect of such a technique is related to the implementation of the weak constraint across the interfaces. It is in fact well known that the exact computation of the integrals appearing in the jump condition can give rise to nontrivial problems when discrete functions defined on nonmatching grids are involved or when totally heterogeneous discretization spaces are used (as in the case of the wavelet/finite element coupling [2]).
A possible remedy is the use of quadrature formulas to evaluate such integrals. However it has been shown in [4] that if quadrature formulas based on the master or on the slave side of the interface are used, the results are not optimal in terms of the best approximation error and the consistency error respectively. In [6] the authors propose to overcome the above problem by adopting a Petrov-Galerkin approach, namely by choosing a test space in which the quadrature formula is different from the one considered in trial space, and show numerical optimal results. On the other hand, the idea introduced in [2] consists in replacing the classical jump constraint by an approximated one where the trace on the *master edge* is replaced by its projection on a suitably defined auxiliary space. Even if this last approach can be more expensive (the computation of the auxiliary projection requires the solution of a

linear system), it allows us to derive a rigorous analysis of the error and turns out to be applicable in a more general framework than the finite element method. Moreover, in [1] the authors show that the new technique provides an approach to the programming of nonconforming domain decompositions which allows us to create a flexible, easily extendible and usable code. In particular, it is important to point out that, by following the new approach, the introduction of a new type of discretization in an existing program does not require any modification to the pieces of the code already implemented: the programmer should only implement methods of integrating trace functions with functions belonging to the auxiliary space. This is unlike the classical mortar approach, where the realization of the jump condition requires the integration of trace functions with functions belonging to all of the types of the discretizations already in the code. In fact, whatever the exact computation of the integrals or the use of quadrature formulas one decides to use, the programmer should be somewhat familiar with all the libraries implementing the discretizations already in the code, entering and modifying the existing methods with the risk of breaking portions of the code.

We present here an analysis of the mortar method with the introduction of the approximate constraint in a general context, when generic approximation spaces are involved in each subdomain of the decomposition of $\Omega \subset \mathrm{R}^2$, providing L^2-norm and H^1 broken norm error estimates and we show some numerical results comparing the new technique with the classical mortar approach. The extension of such results to the three dimensional case is a work in progress.

2 The Mortar method with approximate constraints

We introduce the mortar method through a very simple model problem, namely the Poisson equation, referring to [5] for more details and for proofs of the main results that we will recall throughout the section.

Let $\Omega \subset \mathrm{R}^2$ be a polygonal domain, and consider the following elliptic problem: given $f \in L^2(\Omega)$, find $u : \Omega \longrightarrow \mathrm{R}$ such that

$$-\Delta u = f, \text{ in } \Omega \qquad u = 0, \text{ on } \partial\Omega. \tag{1}$$

Let $\overline{\Omega} = \bigcup_{\ell=1}^{L} \overline{\Omega}_\ell$, be a fixed decomposition of Ω as the disjoint union of L polygonal subdomains Ω_ℓ and set $\Gamma_{\ell,\ell'} = \partial\Omega_\ell \cap \partial\Omega_{\ell'}$, and $S = \cup \Gamma_{\ell,\ell'}$. We denote by $\gamma_\ell^{(i)}$ the i-th side of the ℓ-th domain, so that we can write $\partial\Omega_\ell = \bigcup_i \gamma_\ell^{(i)}$. We do not fix a priori any restriction on the number of the sides of each polygon, and we assume that the decomposition is *geometrically conforming*, that is each edge $\gamma_\ell^{(i)}$ coincides with $\Gamma_{\ell,n} (= \partial\Omega_\ell \cap \partial\Omega_n)$ for some n, $1 \le n \le L$.
For each ℓ, let $\mathcal{V}_h^\ell$ be a family of finite dimensional subspaces of $H^1(\Omega_\ell) \cap C^0(\bar{\Omega}_\ell)$, depending on a parameter $h = h_\ell > 0$ and satisfying homogeneous boundary conditions on $\partial\Omega \cap \partial\Omega_\ell$, and denote by $X_h = \prod_{\ell=1}^{L} \mathcal{V}_h^\ell$.

According to the mortar approach, in order to impose weak continuity to the solution across the interfaces of the decomposition, we start by choosing the *nonmortars* (or *slave*) *sides* $\gamma_n^{(k)}$. More precisely, since each edge of the conforming decomposition coincides with the intersection of two adjacent subdomains, it is possible to write that $\gamma_n^{(k)} \equiv \gamma_\ell^{(i)} \equiv \varGamma_{\ell n}$ for some indices ℓ and i . Then we choose one side (say $\gamma_\ell^{(i)}$) as the master side and the other as the slave side of the common edge $\varGamma_{\ell n}$, the intersection of the two adjacent master subdomain $\overline{\varOmega}_\ell$ and slave subdomain $\overline{\varOmega}_n$ respectively. Moreover, in order to simplify the notations, we use the compact index $m = (n, k)$ to signify that the integer m is related to the slave side of the interface. Therefore we can rewrite the decomposition of the skeleton as follows:

$$S = \bigcup_m \overline{\gamma}_m \qquad \text{with} \qquad \gamma_m \cap \gamma_{m'} = \emptyset.$$

For $v \in \prod_\ell H^1(\varOmega_\ell)$, let us denote by v^+ and v^- the two $L^2(S)$ functions whose restriction to each edge of the skeleton coincides with the trace on that edge corresponding to the master and to the slave subdomain respectively: $v^+_{|\gamma_m} = v^\ell_{|\gamma_m}$ and $v^-_{|\gamma_m} = v^n_{|\gamma_m}$. On each slave side γ_m we define a multiplier space $M_h^m \subset L^2(\gamma_m)$ and we introduce the following weak continuity constraint which appears in the classical mortar approach:

$$\int_S (v^+ - v^-)\lambda \, ds = 0, \qquad \forall \lambda \in M_h \sim \prod_{m \in I} M_h^m. \tag{2}$$

As already pointed out in the introduction, an important aspect of the mortar technique is related to the implementation of the weak constraint (2) across the interfaces. The problem arises when, within the jump condition, one has to compute the integrals $\int_{\gamma_m} v^+_{|\gamma_m} \lambda_m$ for each interface when $v^+_{|\gamma_m}$ and λ_m belong to different types of discretization. It is in fact well known that the exact computation becomes extremely technical when the intersections of the supports of unrelated triangular meshes have to be computed and when totally heterogeneous functions are involved. It can happen that the integral of the product of unrelated functions cannot be computed exactly, as in the coupling of wavelets and finite elements. The idea proposed in [2] consists in replacing the classical jump constraint by an approximate one where the trace on the master edge v^+ is substituted by its projection on a suitable chosen auxiliary space. More precisely, on each slave side γ_m let us introduce an auxiliary space $U_{\delta,m} \subset L^2(\gamma_m)$ depending on a parameter $\delta = \delta_m$. For all $\zeta \in L^2(\gamma_m)$, let $P_m(\zeta) \in U_{\delta,m}$ be the unique element of $U_{\delta,m}$ such that

$$\int_{\gamma_m} P_m(\zeta)\eta \, ds = \int_{\gamma_m} \zeta\eta \, ds, \qquad \forall \eta \in U_{\delta,m}, \tag{3}$$

and let us define the projection operator $P : L^2(S) \longrightarrow U_\delta = \prod_{m \in I} U_{\delta,m}$ as follows: for $\zeta \in {}^2(S)$, $P(\zeta) = \Big(P_m(\zeta_m)\Big)_m$, with $\zeta_m = \zeta_{|\gamma_m}$.

We remark that the auxiliary space will have to be chosen in such a way that the integrals of the form $\int_{\gamma_m} \zeta\eta \, ds$ are (easily) computable provided that ζ is any trace function on the master side and $\eta \in U_{\delta,m}$.

Therefore, we introduce the following approximate integration

$$\int_S (P(v^+) - v^-)\lambda \, ds = 0, \qquad \forall \lambda \in M_h, \tag{4}$$

where the trace on the master side v^+ is replaced by its projection $P(v^+)$. We point out that now the problem of the computation of the integral $\int_S P(v^+)\lambda$ can be overcome by suitably choosing the auxiliary space $U_{\delta,m}$ in such a way that also the integrals of the form $\int_{\gamma_m} \eta\lambda \, ds$ are (easily) computable provided that $\eta \in U_{\delta,m}$ and λ is a trace function associated to the slave side of the interface; recall that the multiplier space M_h^m is related to the slave side. It is beyond the goal of this paper to deal in details with the proper choice of $U_{\delta,m}$. We refer to [2] for the case of the coupling of wavelets with finite elements and to [5] for more general situations.
Let now $\mathcal{X}_h^* = \{v_h \in X_h, \quad \text{s.t. (4) holds}\}$ be the discrete constrained space. We consider the following problem:

Problem 1. Find $u_h \in \mathcal{X}_h^*$, such that for all $v_h \in \mathcal{X}_h^*$,

$$\sum_{\ell=1}^{L} \int_{\Omega_\ell} \nabla u_\delta \nabla v_\delta = \int_\Omega f v_h.$$

We remark again that Problem 1 is derived from the classical method by simply replacing the jump condition (2) with (4). Moreover, even if this last approach requires the solution of a linear system for the computation of the auxiliary projection, thus resulting in more expense compared with other possible solutions, it allows us to derive a rigorous analysis of the error and it turns out to be applicable in a more general framework than the finite element method. In particular, in [5] we show error estimates for Problem 1 for generic choices of discretization spaces. We recall here for completeness the main result, referring to that paper for more details. Introducing the notation

- $\| \cdot \|_{s,*} = \Big(\sum_\ell \| \cdot \|^2_{H^s(\Omega_\ell)} \Big)^{1/2}$ is the broken H^s- norm for $s \geq 1$,
- is the discretization parameter acting as "mesh sizes" on γ_m,
- $\hat{h} = \max_m \{\}, \quad \check{h} = \min_m \{\}, \quad \hat{\delta} = \max_m \{\delta_m\}, \quad \check{\delta} = \min_m \{\delta_m\}$,

and denoting by $H = \max\{\hat{h}, \hat{\delta}\}$ and $h = \min\{\check{h}, \check{\delta}\}$, let u_h be the approximate solution of (1), and u is the exact solution of (1) assuming $u \in H^s(\Omega)$ for some $s > 1$. Under suitable and fairly standard assumptions on the multiplier space and on the approximation and auxiliary spaces, the following error estimates holds:

$$\|u - u_h\|_{1,*} \lesssim (1 + |log_2 h|)\, H^{s-1} \|u\|_s, \quad \|u - u_h\|_{0,\Omega} \lesssim (1 + |log_2 h|)\, H^s \|u\|_s.$$

We remark that in the analysis we use the trace norm $H^{1/2}$ on the interfaces. This gives rise to the logarithmic factor in the estimates (when cross-points/wire basket are present in the decomposition), but allows us to apply the analysis in a general framework (even when non mesh-dependent spaces as in [3] are involved). Moreover, in the geometrically conforming case, we still get an optimal error estimate (see [2]).

2.1 Numerical results

We conclude the presentation of the method by showing some numerical applications.

Wavelet/finite element coupling

We recall that our approach allows us to overcome one of the drawbacks of wavelet type methods, which perform in a very promising way on academic examples, but whose application to real life problems is seriously limited by the issue of geometry (tensor product-like domains). Moreover, in the wavelet/FEM coupling it is not possible to compute exactly the integral of a wavelet type function times a piecewise polynomial defined on an unstructured grid since wavelets are (in general) not known in closed form. Therefore we apply the technique proposed in this paper and we show some examples of the numerical solution of the Poisson problem (1) when the domain Ω is the reference square $[0,1]^2$ containing holes in different numbers, shapes and positions. Triangular meshes are used to describe the profiles of the holes, so that finite element type discretizations are used in the corresponding subdomains, while wavelet analysis is performed in the presence of tensorial-type meshes (subdomains without holes) (see Figures 1 and 2).

Coupling finite elements with nonmatching grids

In this section, we test the influence of the parameter δ (δ being the step of a uniform mesh defined on the auxiliary space $U_{\delta,m}$) on the behavior of the numerical solution when nonmatching finite element meshes are considered at the interface of the decomposition. In doing this, we compare the classical mortar method and the new technique with the approximate constraint. We recall that the two approaches differ in the computation of the integrals $\int_{\gamma_m} v^+_{|\gamma_m} \lambda_m$ that appear in the constraint: such quantities are computed exactly in the first approach while they are replaced by $\int_{\gamma_m} P_m(v^+_{|\gamma_m})\lambda_m$ in the second one. To fix the ideas we consider a decomposition of $\Omega = [0,1]^2$ into two rectangles $\Omega_1 = [0,.5] \times [0,1]$ and $\Omega_2 = [.5,1] \times [0,1]$ and finite element approximations in both. We always refer to the model problem (1), where the right hand side f is chosen in such a way that the exact solution (plotted in Figure 3) is given by

$$u(x,y) = x(1-x)y(1-y)cos(50(x-.5)y).$$

Table 1 shows the L^2-norm and the H^1-seminorm of the error between the approximate and the exact solution when exact integrals are used in the cases of 256×256 and 1024×1024 number of nodes. Table 2 shows the behavior of the errors for both cases with respect to different values of the parameter δ. In Table 3 we now compare the error behavior of the two methods (the classical approach and the approximate constraint) for different choices of the meshes and for values of the mesh size of the auxiliary space, $\delta = h^\tau$, where h is the maximum mesh size on the master and slave side of the interface. Normally $\tau < 1$ is suitable which allows us to balance the approximation error associated to each subdomain and the contribution that corresponds to the introduction of the auxiliary projection.

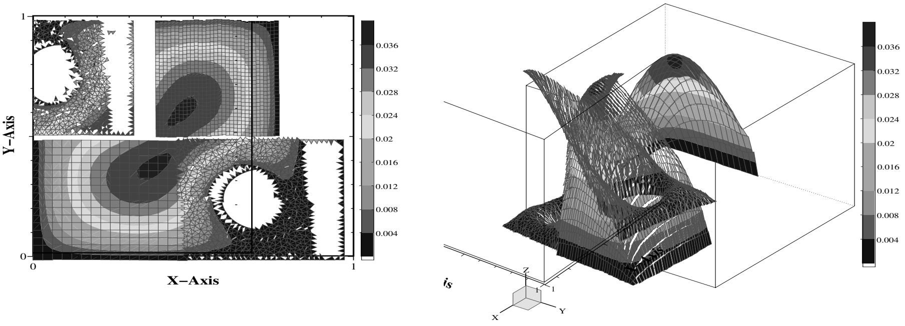


Fig. 1. A 2×2 D.D.: the unit square contains two circular holes in the second and fourth subdomains. Wavelets of level j = 4 in the first subdomain and j = 5 in the third one while finite elements defined on unstructured meshes are used in the other subdomains.

Table 1. Global error in the L^2- norm and H^1- seminorm with respect to the number of nodes with exact computation of integrals.

Nodes	L^2-norm	H^1-seminorm
256×256	0.00201031	0.00204565
1024×1024	0.000503404	0.000511633

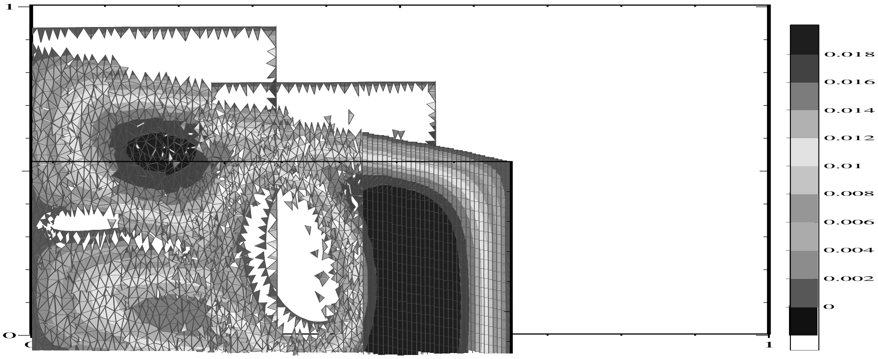


Fig. 2. (a): A 3×1 D.D. The domain contains two holes, the first having a wing profile shape and the second a circular shape. Wavelet discretization space is used in the third subdomain.

Table 2. Behavior of L^2- norm and H^1- seminorm of the error with respect to the parameter δ for the approximate integration.

	Nodes: 256 × 256			Nodes: 1024 × 1024	
	L^2-norm	H^1-seminorm		L^2-norm	H^1-seminorm
δ	Approx. integr.	Approx. integr.	δ	Approx. integr.	Approx. integr.
1/8	0.00245223	0.0170961	1/24	0.000503542	0.000550747
1/10	0.00203049	0.00363632	1/26	0.000503434	0.00051738
1/12	0.00201145	0.00212979	1/28	0.000503409	0.000512209
1/14	0.00201035	0.00204711	1/30	0.000503404	0.000511652
1/16	0.00201031	0.00204565	1/32	0.000503404	0.000511633
1/18	0.00201032	0.00204616	1/34	0.000503404	0.000511644
1/20	0.00201043	0.00205476	1/36	0.000503406	0.000511837
1/30	0.00201031	0.00204575	1/40	0.000503419	0.000516155

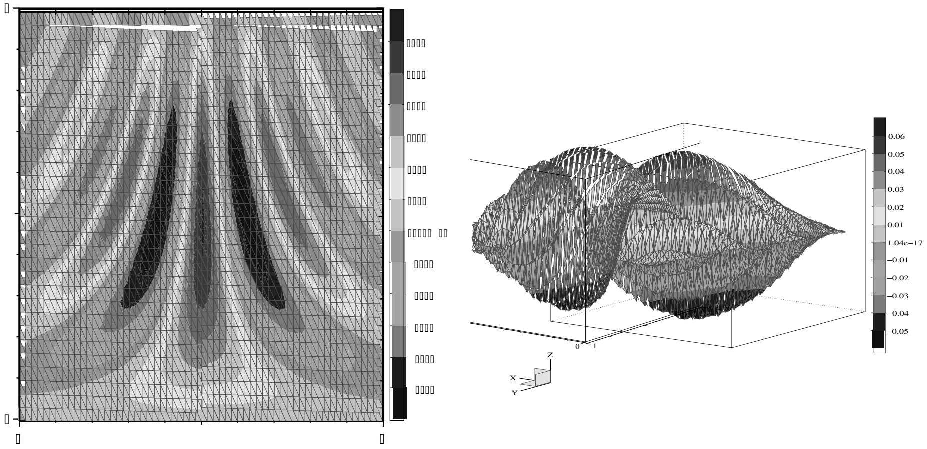


Fig. 3. Analytical solution

Table 3. Comparison between exact and approximate integration

	L^2-norm	L^2-norm	H^1-seminorm	H^1-seminorm
Nodes	Exact integr.	Approx. integr.	Exact integr.	Approx. integr.
100 × 64	0.00151034	0.00151044	0.123656	0.123657
100 × 225	0.001484	0.00148373	0.121936	0.121938
256 × 289	0.000989383	0.000989398	0.0509703	0.0509703
256 × 361	0.000986929	0.000986939	0.0508623	0.0508626
529 × 441	0.000580285	0.000580295	0.0254181	0.0254184
729 × 625	0.000441467	0.000441472	0.0185628	0.018563
729 × 841	0.000439254	0.000439258	0.0184722	0.0184726

2.2 Conclusions

We conclude with some remarks on coupling of finite elements with nonmatching grids in the three dimensional case. By replacing the exact computation of the integral appearing in the jump constraint by an approximate one avoids the difficult task of coding the intersections of the supports of discrete functions living on different meshes of the (bi-dimensional) interfaces. A possible remedy can be to choose Q_1 elements on quadrilateral meshes for the auxiliary space U_δ and the numerical tests performed for the 2D case suggest that the mesh size δ can be chosen coarser than the coarsest of the mesh sizes of the approximation spaces involved in the subdomains. Moreover, the new technique allows us to handle the approximation spaces quite independently from the implementation point of view. The introduction of a new discretization in an existing code turns out to be particularly easy and does not require any modification to the methods already implemented, which is an essential feature of a well designed library.

References

1. S. Bertoluzza and S. Falletta, *An object-oriented implementation of the mortar method with approximate constraint*, Tech. Rep. 10, Istituto di Matematica Applicata e Tecnologie Informatiche, Consiglio Nazionale delle Ricerche, Pavia, Italy, 2004.
2. S. Bertoluzza, S. Falletta, and V. Perrier, *Wavelet/FEM coupling by the mortar method*, in Recent Developments in Domain Decomposition Methods. Proceedings of the Workshop on Domain Decomposition, Zürich, Switzerland, vol. 23 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2002, pp. 119–132.
3. D. Braess and W. Dahmen, *The mortar element method revisited—what are the right norms?*, in Thirteenth international conference on domain decomposition, N. Debit, M. Garbey, R. Hoppe, J. Périaux, D. Keyes, and Y. Kuznetsov, eds., ddm.org, 2001, pp. 27–40.
4. L. Cazabeau, C. Lacour, and Y. Maday, *Numerical quadratures and mortar methods*, in Computational Science for the 21st Century, M.-O. Bristeau, G. Etgen, W. Fitzgibbon, J.-L. Lions, J. Périaux, and M. F. Wheeler, eds., John Wiley & Sons, 1997, pp. 119–128.
5. S. Falletta, *Analysis of the mortar method with approximate integration: the effect of cross-points*, Tech. Rep. 1298, Istitoto Analisi Numerica, Consiglio Nazionale delle Ricerche, Pavia, Italy, 2002.
6. Y. Maday, F. Rapetti, and B. Wohlmuth, *The influence of quadrature formulas in 2D and 3D mortar element methods*, in Recent Developments in Domain Decomposition Methods. Proceedings of the Workshop on Domain Decomposition, Zürich, Switzerland, L. F. Pavarino and A. Toselli, eds., vol. 23 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2002, pp. 203–221.

Fault Tolerant Domain Decomposition for Parabolic Problems

Marc Garbey and Hatem Ltaief

Department of Computer Science, University of Houston, Houston, TX 77204, USA. garbey@cs.uh.edu, ltaief@cs.uh.edu

1 Introduction

The objective of this paper is to present some numerical schemes for the time integration of parabolic problems that can recover from a failure of the computer system. We construct an algorithmic solution of the problem in the context of domain decomposition and distributed computing.

Our model problem is the heat equation:

$$\frac{\partial u}{\partial t} = \Delta u + F(x,t),\ (x,t) \in \Omega \times (0,T),\ u_{|\partial\Omega} = g(x),\ u(x,0) = u_o(x). \quad (1)$$

We suppose that the time integration is done by a first order implicit Euler scheme,

$$\frac{U^{n+1} - U^n}{dt} = \Delta U^{n+1} + F(x, t^{n+1}), \quad (2)$$

and that Ω is partitioned into N subdomains $\Omega_j,\ j = 1..N$. These subdomain problems are distributed among N Processing Units (PUs) or computers. We anticipate that one or several PUs may stall or get disconnected. We complement the distributed architecture of these N PUs, with S additional PUs called spare processing units.

The problem in designing a Fault Tolerant (FT) code that can survive several failures of PUs decomposes as follows:

- (Pb 1) to guarantee that if one or several PUs get disconnected the code can still be executed.
- (Pb 2) to provide an algorithm that can restart the time integration from the data that are available in the distributed memory or file system.

While FT is not so critical for a standard application running for few hours on a medium scale parallel system, it becomes a real issue for long time runs on a large system or a grid computing architecture. In both cases the probability of failures of computing units becomes almost certain, and parallel input/output are not as efficient as ordinary check point procedures may require.

Pb1 is solved by middlewares, like *FT-MPI* [3, 4] for example, which ensure that the application continues while some processors have failed. On the other hand, the application should be FT as well because these middlewares do not guarantee to get the correct numerical solution after a failure.

In this paper we focus on the design of numerical algorithms that solve (Pb2) without using global check pointing.

Global checkpointing does not scale on large parallel system and is impractical on a grid of computers. Indeed, as the number of nodes and the problem size increases, the cost of checkpointing and recovery increases, while the mean time between failures decreases.

The approach we have taken is as follows: spare processors are used to efficiently store the subdomain data of the application during execution in local *asynchronous mode* in their local memory. In case of failure, a spare processor takes over for the failed processor without the entire system having to roll back to a globally consistent checkpoint.

The numerical problem that we address can be defined as follows:

- We assume that spare processors have stored copies of all subdomain data $U_j^{n(j)}, j = 1..N,$ in their local memory. A priori the time step $n(j) \neq n(k)$ for $j \neq k$.
- We look for a (parallel) reconstruction process of U^M at a common time step $M \in (min_j\{n(j)\}, max_j\{n(j)\})$, from subdomains data $U_j^{n(j)}, j = 1..N,$ at different but nearby time steps.

The code can then restart from U^M. Because the time interval between two asynchronous back ups is small, we are looking for a numerical procedure that is completely explicit and does not require the complexity of a standard parameter identification method.

In the next section, we will discuss several algorithmic ideas to solve this problem.

2 Fault Tolerant Algorithms

For the simplicity of the presentation, we will restrict ourselves to the one dimensional heat equation problem $\Omega = (0, 1),$ discretized on a regular Cartesian grid:

$$\frac{U_j^{n+1} - U_j^n}{dt} = \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{h^2} + F_j^{n+1}, \tag{3}$$

and we assume that $dt \sim h$.

However, most of the ideas presented here could be generalized easily to higher space dimension. We will review, one by one, a few numerical methods to reconstruct a uniform approximation of U^M, from disparate data $U^{n(j)}$ in each subdomain Ω_j.

2.1 Interpolation method

Let $M = \frac{1}{N} \Sigma_{j=1..N}\, n(j)$. We look for an approximation of U_j^M in Ω . We assume that we have at our disposal $U^{n(j)}$ and $U^{m(j)}$ at two time steps $n(j) < m(j)$, in each subdomain Ω_j.

Then, if $||U^{n(j)} - U^{m(j)}||_{\Omega_j}$ is below some tolerance number, we may use a second order interpolation/extrapolation in time to get an approximation of U^M.

The numerical error should be of order $((m(j) - n(j))dt)^2$. This simple procedure reduces the accuracy of the scheme and introduces small jumps at the interfaces between subdomains. This method is perfectly acceptable when one is not interested in accurately computing transient phenomena. However, this method is not numerically efficient in the general situation.

2.2 Forward Time Integration

Let us assume that for each subdomain, we have access to $U^{n(j)}$. For simplicity we suppose that $n(j)$ is a monotonically increasing sequence. We further suppose that we have stored in the memory of the spare processors the time history of the artificial boundary conditions $I_j^m = \Omega_j \cap \Omega_{j+1}$ for all previous time steps $n(j) \leq m \leq n(j+1)$. We can then reconstruct with the forward time integration of the original code, the solution $U^{n(N)}$, as follows:

- Processor one advances in time u_1^n from time step $n(1)$ to time step $n(2)$ using boundary conditions I_1^m, $n(1) \leq m \leq n(2)$.
- Then, Processors one and two advance in parallel u_1^n and u_2^n from time step $n(2)$ to $n(3)$ using the interface conditions of neighbors, or the original interface solver of the numerical scheme.
- This process is repeated until the global solution $u^{n(N)}$ is obtained.

This procedure can easily be generalized in the situation of Figure 1 where we do not assume any monotonicity on the sequence $n(j)$. The thick line represents the data that needs to be stored in spare processors, and the interval with circles are the unknowns of the reconstruction process.

The advantage of this method is that we can easily reuse the same algorithm as in the standard domain decomposition method, but restricted to some specific subsets of the domain decomposition. We reproduce then the exact same information U^M as if the process had no failures. The main drawback of the method is that saving the artificial boundary conditions at each time step may slow down the code execution significantly. To illustrate this difficulty, we have implemented a 3D benchmark code with a very simple explicit/Implicit domain decomposition procedure for (1). We refer to [1] for a more sophisticated method along these lines. We use a Krylov method to make the time stepping implicit for each domain and impose explicitly the boundary values on the artificial boundaries. The domain of computation $\Omega = (0,1)^3$ is distributed on a two dimensional grid of $p_x \times p_y$ processors. This two dimensional grid is extended with an additional row of p_x spare processors.

We have implemented a totally asynchronous back up of the subdomains every K time steps. We have also the option to back up in addition all two dimensional artificial interfaces generated in the sequence of $K-1$ consecutive time steps in between two back ups of all subdomains. The communication between active processors and spare processors is done by non-blocking communication, and the back up data are small enough to fit in the main memory of the spare processors.

Figures 2 and 3 report on a numerical experiment with 36 PUs and 6 spare processors on two different architectures. Each processor has a block of $18 \times 18 \times 98$ grid points. The first system used in figure 2 is a Beowulf cluster with dual AMD 32 bit processors and a gigabit ethernet network, while the second system used in figure 3 is a dual Itanium cluster that has a Myrinet network. While the

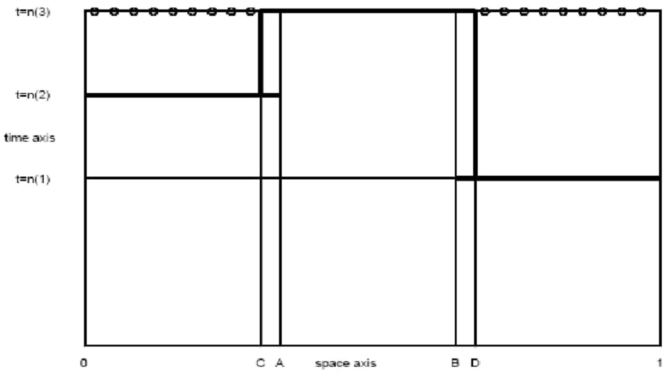


Fig. 1. Illustration of the reconstruction procedure with the forward method.

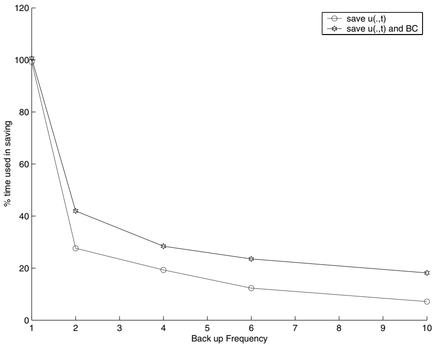


Fig. 2. Overhead with Dual AMD/Gigabit Ethernet

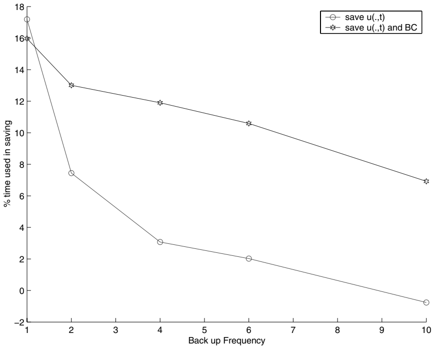


Fig. 3. Overhead with dual Itanium/Myrinet Network

penalty to back up the subdomain on spare processors is particularly high with the system that uses a Gigabit ethernet, it can be seen that saving the artificial boundary conditions in every time step significantly slows down the application on *both* computer architecture systems.

We will now discuss a reconstruction method that produces an approximate solution without using the time series of artificial boundary conditions.

2.3 Backward Integration and Space Marching

Let us suppose now that we asynchronously store only the subdomain data, and not the chronology of the artificial interface condition. To be more specific, we suppose that we have access for each subdomain to the solution at two different time steps $n(i), m(i)$ with $m(i) - n(i) = K >> 1$.

The Forward Implicit scheme provides an explicit formula when we go backwards in time:

$$U_j^n \ = \ U_j^{n+1} \ - \ dt \ \frac{U_{j+1}^{n+1} - 2U_j^{n+1} + U_{j-1}^{n+1}}{h^2} \ - F_j^{n+1}, \tag{4}$$

The existence of the solution is granted by the forward integration in time. A first difficulty is the instability of the numerical procedure and a second is the fact that one is restricted to the cone of dependence as shown in Figure 4.

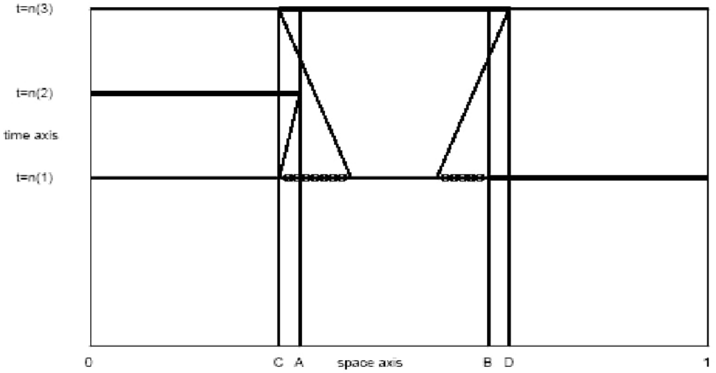


Fig. 4. Illustration in one space dimension of the problem with the third solution.

We have in Fourier modes

$$\hat{U}_k^n \ = \ \delta_k \ \hat{U}_k^{n+1},$$

with

$$\delta_k \sim -\frac{2}{h}(\cos(k \ 2 \ \pi \ h) - 1), \ |k| \leq \frac{N}{2}.$$

The expected error is at most on the order $\frac{\nu}{h^K}$ where ν is the machine precision and K the time step. Therefore, the backward time integration is still accurate up to time step K with

$$\frac{\nu}{h^K} \sim h^2.$$

To stabilize the scheme, one can use the telegraph equation that is a perturbation of the heat equation:

$$\varepsilon \frac{\partial^2 u}{\partial t^2} - \frac{\partial^2 u}{\partial x^2} + \frac{\partial u}{\partial t} = F(x,t), x \in (0,1), t \in (0,T) \tag{5}$$

The asymptotic convergence can be derived from [2] after time rescaling. The general idea is then to use the previous scheme (4) for few time steps and pursue the time integration with the following one

$$\varepsilon \frac{U_j^{n+1} - 2\,U_j^n + U_j^{n-1}}{\tilde{dt}^2} - \frac{U_{j+1}^n - 2\,U_j^n + u_{j-1}^n}{h^2} + \frac{U_j^{n+1} - U_j^n}{\tilde{dt}} = F_j^n. \tag{6}$$

Let us notice that the time step $\tilde{dt}$ should satisfy the stability condition $\tilde{dt} < \varepsilon^{1/2} h$. We take in practice $\tilde{dt} = dt/p$ where p is an integer. The smaller ε, the more unstable the scheme (6) and the flatter the cone of dependence. The smaller ε, the better the asymptotic approximation. We have done a Fourier analysis of the scheme and Figure 5 shows that there is a best compromise for ε to balance the error that comes from the instability of the scheme and the error that comes from the perturbation term in the telegraph equation. We have obtained a similar result in our numerical experiments.

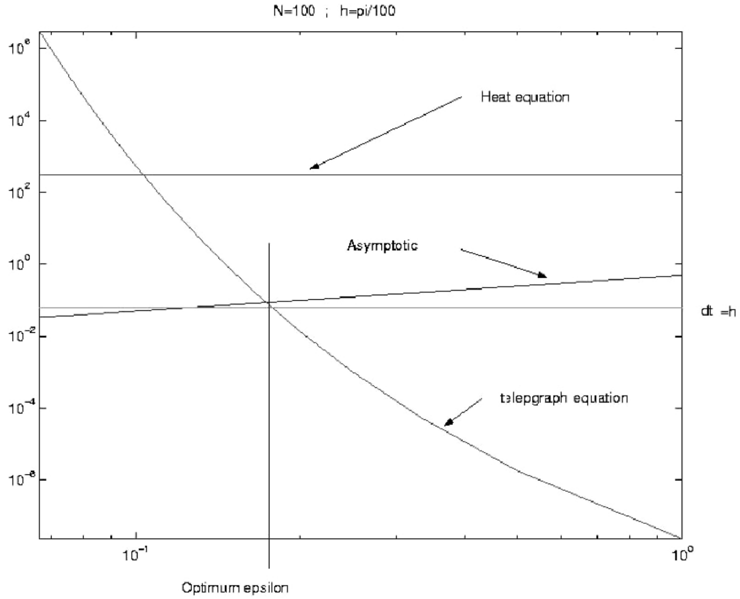


Fig. 5. Stability and error analysis with Fourier

To construct the solution outside the cone of dependencies we have used a standard procedure in the inverse heat problem, the so called space marching method [5]. This method may require a regularization of the solution inside the cone using a convolution

$$\rho_\delta \, * \, u(x\,,\,t),$$

where

$$\rho_\delta = \frac{1}{\delta\sqrt{\pi}} \exp(-\frac{t^2}{\delta^2}).$$

The following space marching scheme:

$$\frac{U_{j+1}^n \; - \; 2\, U_j^n \; + \; U_{j-1}^n}{h^2} = \frac{U_j^{n+1} - U_j^{n-1}}{2\; dt} + F_j^n, \qquad (7)$$

is unconditionally stable, provided $\delta \geq \sqrt{\frac{2\; dt}{\pi}}$.

The last time step $U^{n(i)+1}$ to be reconstructed uses the average

$$U^{n(i)+1} = \frac{U^{n(i)} + U^{n(i)+2}}{2}.$$

We have observed that filtering as suggested in [5] is not necessary in our reconstruction process. Figure 6 illustrates the numerical accuracy of the overall reconstruction scheme that combines (4) and (7) for $\Omega = (-\pi, \pi)$, $dt = h = 0.0314$, $K = 7$ and F is such that the exact analytical solution is $cos(q_1\; x)(sin(q_2\; t) + \frac{1}{2} cos(q_2\; t))$, $q_1 = 2.35, q_2 = 1.37$. In this specific example our method gives better results than the interpolation scheme provided that $K \leq 7$. For larger K we can use the scheme (6) for time steps below $m(i) - 7$. However the precision may deteriorate rapidly in time.

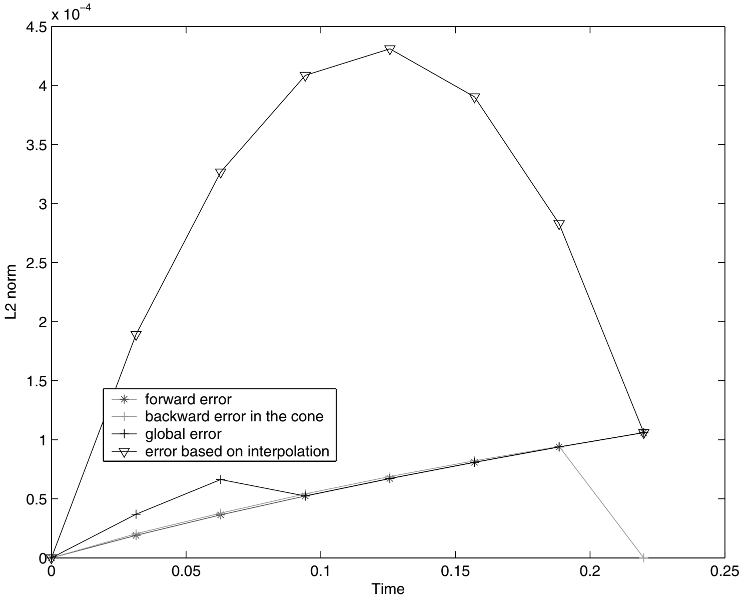


Fig. 6. Numerical accuracy of the overall reconstruction scheme

3 Conclusion

We have presented the problem of FT algorithm for a parabolic operator. We have reviewed several procedures to reconstruct the solution in each subdomain from a set of subdomain solutions given at disparate time steps. This problem is quite challenging because it is very ill posed. We have found a satisfactory solution by combining explicit reconstruction techniques that amounts to a backward integration with some stabilization terms and space marching. We are currently applying these ideas to multi-dimensional parabolic problems.

Acknowledgment. Research reported here was supported by Award 0305405 from the National Science Foundation.

References

1. C. N. Dawson and T. F. Dupont, *Explicit/implicit, conservative domain decomposition procedures for parabolic problems based on block-centered finite differences*, SIAM J. Numer. Anal., 31 (1994), pp. 1045–1061.
2. W. Eckhaus and M. Garbey, *Asymptotic analysis on large time scales for singular perturbation problems of hyperbolic type*, SIAM J. Math. Anal., (1990), pp. 867–883.
3. E. Gabriel, G. E. Fagg, A. Bukovsky, T. Angskun, and J. J. Dongarra, *A fault-tolerant communication library for grid environments*, in Proceedings of the 17th Annual ACM International Conference on Supercomputing, International Workshop on Grid Computing and e-Science, 2003.
4. W. Gropp and E. Lusk, *Fault tolerance in message passing interface programs*, International Journal of High Performance Computing Applications, 18 (2004), pp. 363–372.
5. D. A. Murio, *The Mollification Method and the Numerical Solution of Ill-posed Problems*, John Wiley & Sons, 1993.

Domain Decomposition for Heterogeneous Media

Ivan G. Graham[1] and Patrick O. Lechner[2]

[1] Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, United Kingdom. igg@maths.bath.ac.uk,
[2] Department of Mathematical Sciences, University of Bath, Bath, BA2 7AY, United Kingdom. mappol@maths.bath.ac.uk

Summary. Elliptic problems with multiscale coefficients have been studied to a great extent recently. Preconditioners based on standard domain decomposition methods often perform poorly when the variation of the coefficients inside the subdomains is large. In this paper we study the behaviour of domain decomposition methods based on linear coarsening for such problems and we also propose improved methods which use the notion of multiscale finite elements to define coarsening operators.

1 Problem Description

Typical examples of *elliptic multiscale problems* occur among others in fluid flow in strongly *heterogeneous media* or heat conduction in composite media. Let us therefore consider the *second order partial differential equation of Poisson type*

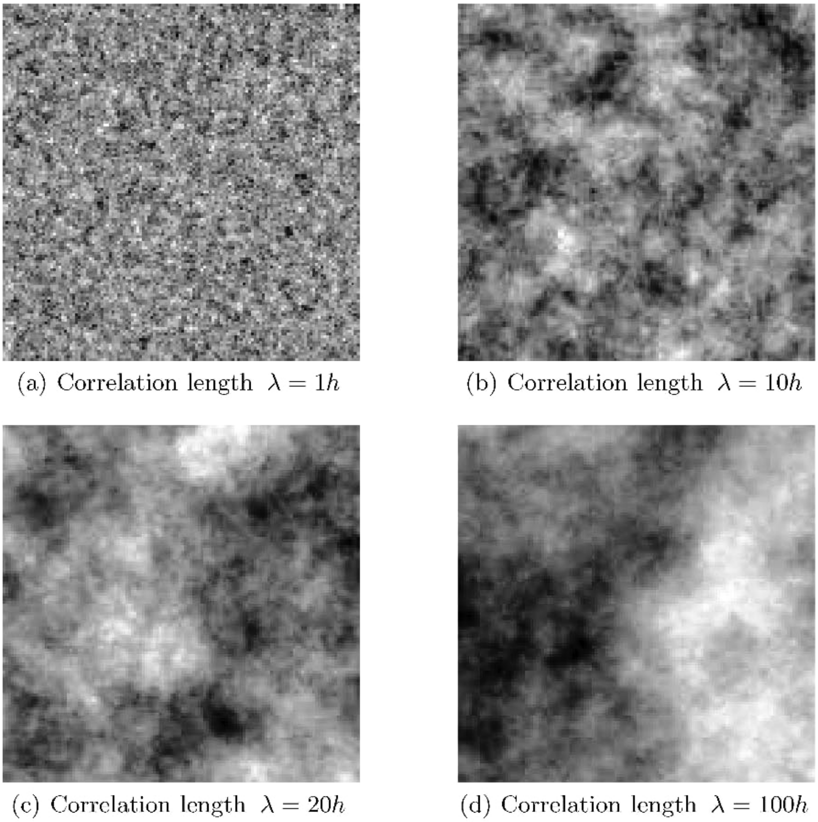
$$-\boldsymbol{\nabla}.\alpha(\mathbf{x})\boldsymbol{\nabla}u(\mathbf{x}) = f(\mathbf{x}) \text{ for } \mathbf{x} \in \Omega, \tag{1}$$

with $\Omega \subset \mathbb{R}^d$, where $\alpha(\mathbf{x})$ is the *conductivity*, which for simplicity is assumed to be scalar valued, symmetric and positive, but which is allowed to vary very strongly, typically as much as $\max_{\mathbf{x},\mathbf{y}\in\Omega}(\alpha(\mathbf{x})/\alpha(\mathbf{y})) \cong 10^9$. Furthermore we assume Ω to be an interval for *d=1*, a polygon for *d=2* and a polyhedron for *d=3*. We consider the Dirichlet problem with $u(\mathbf{x}) = 0$ for all $\mathbf{x} \in \partial\Omega$, the boundary of Ω.
Closely related problems occur in the modeling of groundwater flow. Due to the difficulties in capturing the heterogeneity of rock formations and significant uncertainties away from the limited number of possible observation points, the *permeability field*, which is the (often strongly varying) multiscale field in this application, is then modeled stochastically, and in this case we consider a lognormal model $\alpha(\mathbf{x}) := \exp(Z(\mathbf{x}))$, where $Z(\mathbf{x})$ is a Gaussian random field. Using Monte Carlo Methods on a large sample of reasonable realisations of these fields usually leads to

good numerical results for u.
For physical and practical reasons (see [3]) fields $Z(\mathbf{x})$ of *Ornstein-Uhlenbeck type*, i.e. statistically homogeneous isotropic Gaussian random fields with constant mean, variance σ^2, correlation length λ and covariance function

$$\Sigma(\mathbf{x}, \mathbf{y}) := \sigma^2 \exp\{|\mathbf{x} - \mathbf{y}|/\lambda\}, \tag{2}$$

are considered.
We now discretise (1) using linear finite elements on a uniform triangulation of Ω with element diameter of order h, where the step size is in practice chosen so that $\lambda = Ch$, for C moderate, e.g. $C \approx 10$ (see Figure 1), to achieve an accuarate resolution of the problem without upscaling. Dark regions in these pictures represent areas with high permeability, etc.. The exact values depend on the variance σ^2. For large λ the local variation of the field is reduced in general.


(a) Correlation length $\lambda = 1h$

(b) Correlation length $\lambda = 10h$

(c) Correlation length $\lambda = 20h$

(d) Correlation length $\lambda = 100h$

Fig. 1. Dependence on the correlation length λ on domain $[0, 128h]^2$.

Now let α be a fixed realisation of this random field and let $A(\alpha)$ be the corresponding stiffness matrix with entries $A_{ij}(\alpha) := \int_{\Omega} \alpha \boldsymbol{\nabla}\phi_i . \boldsymbol{\nabla}\phi_j$, where $\{\phi_i\}$ are the piecewise linear nodal basis and let $A(1)$ be the stiffness matrix corresponding to a field with $\alpha(\mathbf{x}) = 1$ for all $\mathbf{x} \in \Omega$. Then it can easily be shown that

$$\kappa\,(A(\alpha)) \leq \max_{\mathbf{x}\,,\,\mathbf{y}\,\subset\Omega} (\alpha(\mathbf{x})/\alpha(\mathbf{y}))\cdot\kappa\,(A(1))\,, \qquad (3)$$

where κ denotes the condition number.
Therefore the convergence of iterative methods, like the *conjugate gradient method*, depends on the global ratio of the coefficient α and can be very slow for strongly varying α .

2 Linear Interpolation Domain Decomposition

The previous observations make it important to find a good preconditioner for the stiffness matrix $A(\alpha)$ and we first of all study an *additive Schwarz domain decomposition method with linear coarse space.* We therefore introduce a coarse grid of size H , which defines triangular subdomains, K_i , that we extend to p overlapping regions, $\hat{K}_i$, $i = 1, ..., p$, with overlap δ , such that the subdomains consist of unions of fine grid elements. Now let R_i be the local restriction of a vector defined for the degrees of freedoms (dof) on the fine mesh to the dof in the interior of $\hat{K}_i$ and let $A_i := R_i A R_i^T$. Also let $\left\{\mathbf{x}_j^H : j = 1, ..., n_c\right\}$ be the set of coarse grid freedoms. Furthermore let $\left\{\phi_i^H\right\}$ be the set of linear interpolation functions with respect to the coarse grid, such that $\phi_i^H(\mathbf{x}_j^H) = \delta_{ij}$, where δ_{ij} denotes the Kronecker delta. Using these functions we introduce an interpolation map $R_L^T := \left[\boldsymbol{\phi}_1^H, ..., \boldsymbol{\phi}_{n_c}^H\right]$, where for $i = 1, ..., n_c$, $\boldsymbol{\phi}_i^H$ is the vector of evaluations of ϕ_i^H at the fine grid freedoms. Finally set $A_L := R_L A R_L^T$.

We can then show (using ideas discussed for example in [4], [2] and [8]), that for the two level linear interpolation additive Schwarz preconditioner $M_L^{-1} := \sum_{i=0}^{p} R_i^T A_i^{-1} R_i + R_L^T A_L^{-1} R_L$, there exists a constant C , such that

$$\kappa\left(M_L^{-1}A\right) \leq C\cdot B(d)\cdot\max_{i}\max_{\mathbf{x}\,,\,\mathbf{y}\,\subset K_i}(\alpha(\mathbf{x})/\alpha(\mathbf{y}))\left(1+\delta^{-1}H\right), \qquad (4)$$

where $B(1) = 1$, $B(2) = (1 + \log(H/h))$ and $B(3) = H/h$.
When α is moderately varying, better estimates which avoid the dependence on H/h can be derived (see [8]). In general, for a medium or large correlation length λ , the dependence on α in (4) may be much better than in (3), since the subdomain ratios $\max_{i}\max_{\mathbf{x}\,,\,\mathbf{y}\,\subset K_i}(\alpha(\mathbf{x})/\alpha(\mathbf{y}))$ can be much smaller than the global ratio $\max_{\mathbf{x}\,,\,\mathbf{y}\,\subset\Omega}(\alpha(\mathbf{x})/\alpha(\mathbf{y}))$ (cf. Fig. 1). Further details can be found in [7].

3 Multiscale Interpolation Domain Decomposition

Estimate (4) still grows linearly with the global maximum ratio of permeability values over all subdomains. This can be improved by replacing the linear interpolation by a more suitable operator. For this we use ideas from *multiscale finite elements (MsFE)* (see [5] and [6]). However we apply them as a way of improving solvers rather than improving accuracy. A similar motivation is discussed in [1]. The basic idea behind this is to use the fine scale structure of the problem in the construction of the coarse grid basis functions, which will then improve the coarse grid solves. For a typical triangular coarse grid element K with nodes $\mathbf{x}_j^K$, $j = 1, 2, 3$, we compute multiscale basis functions ψ_j^K on the subdomain K by solving the partial differential equations

$$-\nabla . \alpha(\mathbf{x})\nabla\psi_j^K(\mathbf{x}) = 0 \text{ for } \mathbf{x} \in K, \tag{5}$$

where we force a Dirichlet boundary condition on ψ_j^K with $\psi_j^K(\mathbf{x}_i^H) = \delta_{ij}$ and fix the behaviour of ψ_j^K on the other parts of the boundary as discussed below. In practice ψ_j^K is approximated by the finite element method on the fine mesh contained inside K .

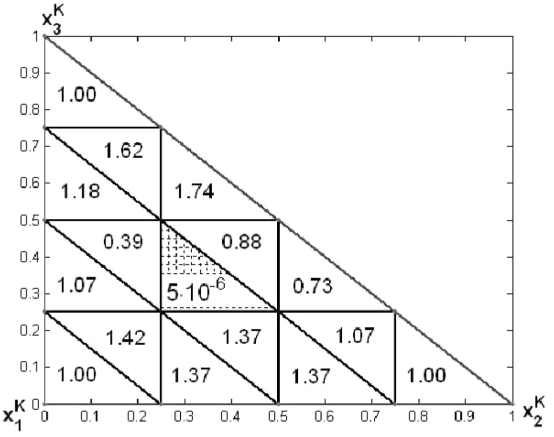


Fig. 2. Values of $H\left|\nabla\psi_3^K(\mathbf{x})\right|$ evaluated on the fine grid elements of subdomain K for one multiscale basis function with $\alpha = 10^6$ on the marked element in the centre and $\alpha = 1$ everywhere else.

In fine grid elements where the permeability is high the corresponding multiscale basis function will have a smaller gradient. This behaviour is illustrated in Fig. 2, where $j = 3$ and the permeability is taken as $\alpha = 10^6$ on one element and 1 on all the others. The ψ_j^K are then combined to define nodal basis functions ψ_i , $i = 1, ..., n_c$, which are defined on all of Ω with $\psi_i\left(\mathbf{x}_j^H\right) = \delta_{ij}$, $j = 1, ..., n_c$, and $\psi_i|_K$ is a linear combination of the ψ_l^K , $l = 1, 2, 3$.

We then replace the *linear interpolation map* R_L^T by a *multiscale interpolation map* $R_{Ms}^T := [\boldsymbol{\psi}_1, ..., \boldsymbol{\psi}_{n_c}]$, where each $\boldsymbol{\psi}_i$ is a vector of evaluations of ψ_i at the fine grid nodes. The choice of the *boundary condition* in (5) can be very important. The simplest choice is to interpolate linearly between the the values $\psi_j(\mathbf{x}_i^H) = \delta_{ij}$ on the edges of K. Numerical results show, that these linear boundary conditions perform very well for small variance σ^2. However for large variance the performance can be improved by using so called *oscillatory conditions* as introduced in ([5], pages 172-173). Solving problems on extended subdomains, so called *oversampling* (useful for improving accuracy, see [5]), has not been found experimentally to improve the convergence rate significantly.
The new two level multiscale interpolation additive Schwarz preconditioner is now given by $M_{Ms}^{-1} := \sum_{i=0}^{p} R_i^T A_i^{-1} R_i + R_{Ms}^T A_{Ms}^{-1} R_{Ms}$.

We compare the performance of this preconditioner with that of a standard linear two-level additive Schwarz preconditioner. Consider therefore a simple two-dimensional problem on $[0,1]^2$ on which we define a uniform triangular fine mesh of size h and a coarse triangular mesh of size H and fix a constant (minimal) overlap $\delta = 2h$. The following tables then compare the iteration numbers and computation times (in brackets; including setup time and iteration time) of the two preconditioners for $h = 1/128$ and $H = 8h$, resp. $H = 16h$ and for different variances σ^2, where in the case of multiscale interpolation, oscillatory boundary conditions were used for the construction of the interpolation functions. Here Table 1 is for fields of Ornstein-Uhlenbeck type with correlation length $\lambda = 10h$ and Table 2 for completely random isotropic fields. Both tables show, that in two dimensions, especially for strongly varying fields, the multiscale interpolation brings a considerable improvement in both iteration numbers and computation times.

Variance	**H = 8h**				**H = 16h**			
$\boldsymbol{\sigma^2}$	**Linear Int.**		**Multisc. Int.**		**Linear Int.**		**Multisc. Int.**	
1	29	(90)	27	(89)	43	(175)	42	(178)
2	33	(100)	29	(93)	45	(180)	43	(180)
4	41	(118)	34	(101)	53	(199)	49	(192)
8	59	(160)	45	(127)	73	(239)	62	(214)
16	109	(277)	70	(184)	144	(388)	90	(269)
24	187	(455)	96	(245)	251	(611)	122	(342)

Table 1. Iteration numbers and computation times (in sec.) for fields of Ornstein-Uhlenbeck type with $\lambda = 10h$.

Variance	**H = 8h**		**H = 16h**	
$\sigma^{\mathbf{2}}$	**Linear Int.**	**Multisc. Int.**	**Linear Int.**	**Multisc. Int.**
1	29 (90)	27 (89)	32 (152)	31 (154)
2	33 (100)	29 (93)	35 (159)	34 (161)
4	49 (137)	34 (103)	44 (178)	40 (175)
8	83 (216)	54 (147)	77 (247)	53 (195)
16	181 (440)	94 (239)	163 (426)	87 (265)
24	279 (664)	118 (320)	301 (714)	129 (356)

Table 2. Iteration numbers and computation times (in sec.) for completely random isotropic fields.

For one dimensional problems, one can show, that M_{Ms}^{-1} is in fact the exact inverse of A , when the subdomains have zero overlap. The same is true in higher dimensions, if we replace the node based version of the multiscale preconditioner by a skeleton based version, which means we compute a multiscale basis function ψ_i for every single freedom of the skeleton (i.e. each fine grid freedom on the boundary of the subdomains), instead of only using the basis functions corresponding to the coarse grid freedoms. Multiscale preconditioners with complexity between the node based and skeleton based methods and their performance are part of our current research and will be discussed in [7].

References

1. J. Aarnes and T. Y. Hou, *Multiscale domain decomposition methods for elliptic problems with high aspect ratios*, Acta Mathematicae Applicatae Sinica (English Series), 18 (2002), pp. 63–76.
2. T. F. Chan and T. P. Mathew, *Domain Decomposition Survey*, Cambridge University Press, 1994, pp. 61–143.
3. G. Dagan and S. P. Neumann, *Subsurface Flow and Transport: A Stochastic Approach*, Cambridge University Press, 1987.
4. M. Dryja and O. B. Widlund, *Domain decomposition algorithms with small overlap*, SIAM J. Sci.Comput., 15 (1994), pp. 604–620.
5. T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media*, J. Comp. Phys., (1997), pp. 169–189.
6. T. Y. Hou, X.-H. Wu, and Z. Cai, *Convergence of a multiscale finite element method for elliptic problems with rapidly oscillating coefficients*, Math. Comp., 68 (1999), pp. 913–943.
7. P. O. Lechner, PhD thesis, Department of Mathematical Sciences, University of Bath, 2006. In preparation.
8. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

Parallel Implicit Solution of Diffusion-limited Radiation Transport

William D. Gropp [1], Dinesh K. Kaushik [1], David E. Keyes [2], and Barry F. Smith [1]

[1] Mathematics and Computer Science Division, Argonne National Laboratory, Argonne, IL 60439, USA. {gropp,kaushik,bsmith}@mcs.anl.gov

[2] Department of Applied Physics and Applied Mathematics, Columbia University, New York, NY 10027, USA. david.keyes@columbia.edu

Summary. We present simulations of diffusion-limited transport in an initially cold medium of two different materials subjected to an impulsive radiative load, using a Newton-Krylov-Schwarz solver. The spatial discretization employs Galerkin finite elements with linear piecewise continuous basis functions over simplices in 2D and 3D. Temporal integration is via a solution-adaptive implicit Euler method. The code shows excellent domain-decomposed scalability on the Teragrid, BlueGene, and System X platforms. Comparing implementations for this application with flop-intensive residual evaluation, we observe that an analytical Jacobian gives better performance (in terms of the overall execution time to solution) than a Jacobian-free approach.

1 Diffusion-limited radiation transport

Under the assumptions of isotropic radiation with no frequency dependence, transport through a material characterized by spatially varying atomic number Z and thermal conductivity of κ can be modeled by the following coupled nonlinear equations, known as flux-limited radiation diffusion [9]:

$$\frac{\partial E}{\partial t} - \nabla \cdot (D_E \nabla E) = \sigma_a (T^4 - E), \quad \frac{\partial T}{\partial t} - \nabla \cdot (D_T \nabla T) = -\sigma_a (T^4 - E) \tag{1}$$

with

$$\sigma_a = \frac{Z^3}{T^3}, \quad D_E(E,T) = \frac{1}{3\sigma_a + \frac{|\nabla E|}{|E|}}, \quad \text{and } D_T(T) = \kappa T^{\frac{5}{2}}. \tag{2}$$

Here, E represents the photon energy density and T is the material temperature. Since the diffusion approximation can allow speeds of propagation faster than speed of light, the above formula for diffusivity D_E includes Wilson's flux limiter

$|\nabla E|/|E|$ [9], a strongly nonlinear effect. Though simple in appearance, this is a challenging problem when atomic number varies sharply, due to the cubic dependence of the source term coefficient on Z .

2 Discretization and algorithmic setting

We employ the Galerkin finite element method using conforming linear P1 elements, triangular for 2D and tetrahedral for 3D [5]. The diffusion coefficients, D_E and D_T , are also expanded in the element basis functions.

In this paper, due to space constraints, we present results from backward Euler time integration only. Comparisons with various higher order time integration methods, including BDF (discussed for radiation diffusion problems in [3]) and implicit Runge-Kutta schemes will be published elsewhere. We evolve the timestep size by limiting the changes in the solution (point by point) according to [10]:

$$\max\left(\frac{|U^{n+1}-U^{n}|}{|U^{n+1}|}\right) \leq \varepsilon_t. \tag{3}$$

We have used $\varepsilon_t = 0.75$ in all of the computational results in this paper. We start with a small value of δt $(= 10^{-5})$ and allow it to evolve with Eqn. (3), except that the timestep grows by no more than 20% per timestep, and an occasional “short” step is imposed to archive the solution for visualization at regularly spaced intervals. The timestep size evolves to about four orders of magnitude larger than the initial timestep size towards the end of the computation, after the radiation pulse has passed beyond the high atomic number zone.

We use the Newton-Krylov-Schwarz (NKS) algorithm [4, 7] to solve the nonlinear problem arising on every timestep of the discretized form of Eqn. (1). Several parameters of NKS must be tuned for optimal performance [4]. Our code is built on PETSc [1]. We use a left-preconditioned inexact Newton method to solve the nonlinear problem on each timestep. The relative tolerance for the nonlinear residual norm reduction in each timestep is 10^{-8} , which is far below discretization error but within easy reach of Newton's method. The linear problems within each Newton step are solved using GMRES with a maximum of 80 iterations and a maximum subspace size of 30 between restarts, or a relative reduction of the left-preconditioned residual by three orders of magnitude. We use a block Jacobi (zero overlap) preconditioner and map each subdomain to a single processor. Though not algorithmically scalable for general elliptic problems, this inexpensive limit of Schwarz preconditioning is adequate for transient problems. We use incomplete factorization (ILU) within each subdomain and allow a single level of fill. This tuning of the NKS method follows [4], where it was effective in overall runtime for a CFD code on a variety of message passing architectures.

3 Results and Discussion

We present 2D and 3D test cases. The computational domain in 2D is the unit square, with a radiation flux incident on the left boundary. The atomic number is location dependent:

$$Z(x,y) = \begin{cases} 10 \text{ for } \frac{1}{3} \leq x \leq \frac{2}{3} \text{ and } \frac{1}{3} \leq y \leq \frac{2}{3}, \\ 1 \quad \text{elsewhere.} \end{cases} \tag{4}$$

The boundary conditions for the Eqns. (1) are set by imposing a constant radiation field at $x = 0$:

$$\mathbf{n} \cdot D_E \nabla E + \frac{E}{2} = 2 \text{ at } x = 0 \text{ and } \mathbf{n} \cdot D_E \nabla E + \frac{E}{2} = 0 \text{ at } x = 1,$$
$$\text{and } \mathbf{n} \cdot \nabla E = 0 \text{ at } y = 0 \text{ and } y = 1,$$

where $\mathbf{n}$ is the outward unit normal to the boundary, as in [8].

Figure 1 plots the material temperature along $y = 0.5$ and $x = 0.5$ cuts for a large range of uniform mesh resolutions, showing asymptotic grid independence. A sufficiently fine mesh is needed to resolve the sharp features inside and around the interior domain of high atomic number. Time evolution of material temperature along the same cuts is presented in Figure 2. The high-Z region at the center of the interior domain takes longest to heat up.

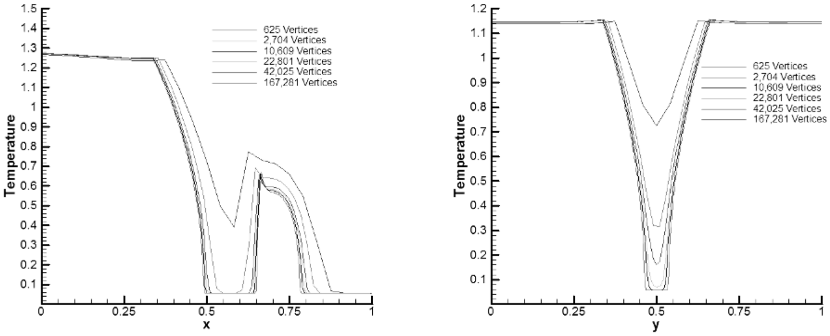


Fig. 1. Material temperature at $y = 0.5$ (left) and at $x = 0.5$ (right) showing mesh independence for the 2D test example at $t = 3$.

The temperature contours showing the propagation of the thermal front from $t = 1$ to $t = 4$ are shown in Figure 3 for the 3D case with a tetrahedral mesh of 237,160 vertices and 1,264,086 elements. This reduces to the 2D case on $z =$ constant planes, as atomic number depends only on the x and y coordinates. This permits comparison to the 2D solution, while providing a fully 3D configuration for demonstrating scaling.

We discuss performance issues of the time-accurate NKS algorithm on the Teragrid cluster (SDSC), BlueGene (Argonne), and System X (Virginia Tech). The Teragrid cluster is made of 1.5 GHz dual Intel Madison processors, each with 4 MB of L2 cache, and 4 GB of memory per node. The IBM BlueGene node contains dual 700 MHz processors with 4 MB shared L3 cache and 512 MB of main memory. We do all the computations on BlueGene in the "co-processor" mode, in which one processor of a node does communication only, since memory and memory bandwidth

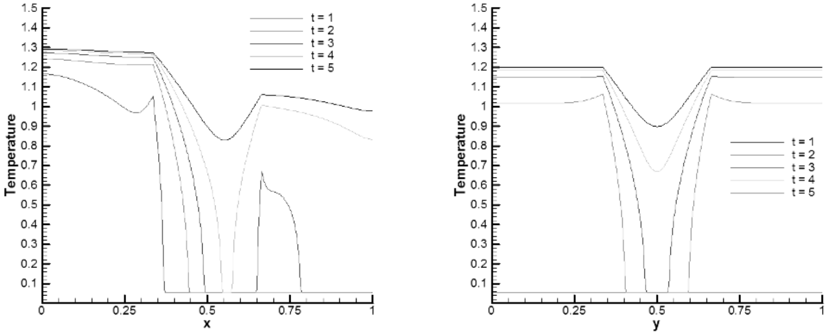


Fig. 2. Material temperature evolution at $y = 0.5$ (left) and at $x = 0.5$ (right) for the 2D test case with 167,281 vertices and 332,928 elements.

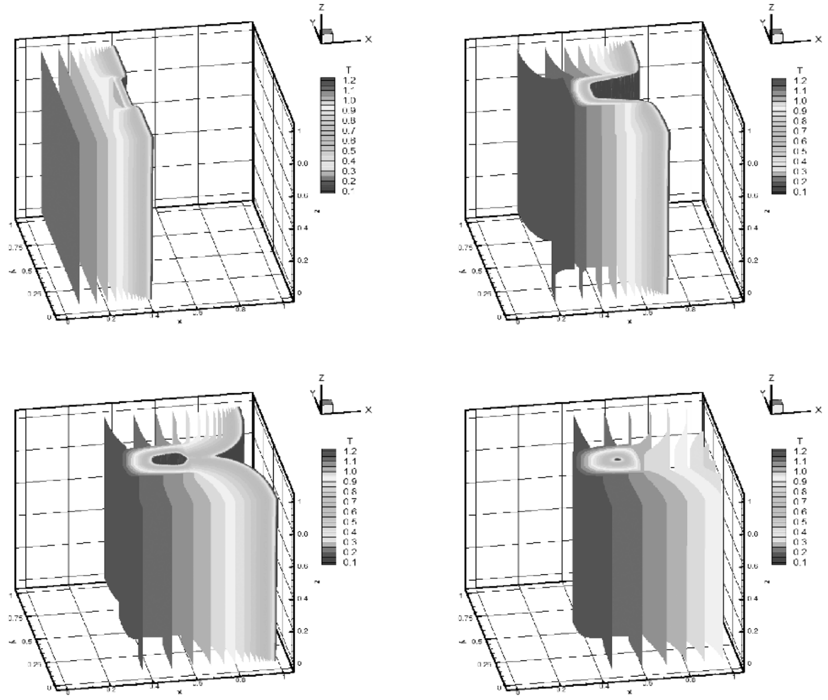


Fig. 3. Evolution of material temperature in time for the 3D test example with a tetrahedral mesh of 237,160 vertices and 1,264,086 elements. The top left figure shows the temperature contours at $t = 1$, the top right at $t = 2$, the bottom left at $t = 3$, and the bottom right at $t = 4$.

are too limited to make effective use of both processors as floating point engines. System X is a cluster of dual 2.3 GHz PowerPC 970FX (Apple Xserve G5) processors with 0.5 MB L2 cache. We do not discuss here the per-processor floating point performance, though it is an important part of performance overall [4]. Though our code scales well, it will require, as in [4], attention to ordering and blocking issues to achieve a high percentage of machine peak on cache-based microprocessors.

3.1 Algorithmic Performance

Figure 4 shows the history of timestep size evolution according to Eqn. 3 and the average number of linear iterations per timestep. Overall execution time is highly sensitive to timestepping strategy for this problem. We are evaluating higher order (BDF [2] and Implicit Runge Kutta [6]) methods from standpoints of computational efficiency and robustness. However, scalability is not highly sensitive to choice of timestepping scheme, so the results of this study on domain-decomposed preconditioning, which cover a wide range of degree of diagonal dominance in the implicit operator, are representative. In the right side plot of Figure 4, the number of linear iteration count rises as the number of subdomains increases since the block Jacobi preconditioner gets weaker as diagonal dominance diminishes for larger timesteps.

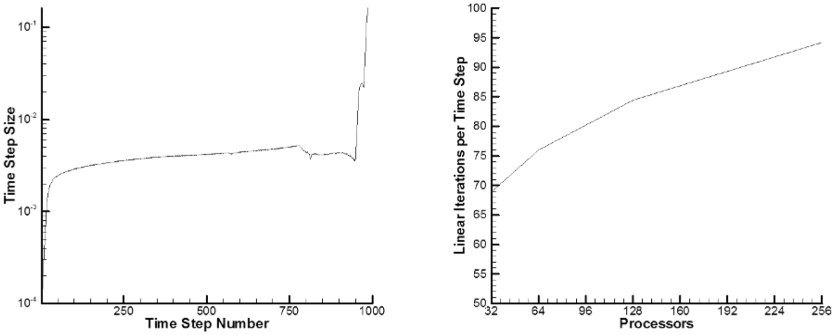


Fig. 4. Time step size history (left) and the average number of linear iterations per timestep (right) for the 3D test case of Figure 3. (The occasional sudden drops in the timestep size are the result of requiring visualization dumps at predefined instants of physical time.)

3.2 Parallel Scalability

We present the execution time and parallel efficiency on up to 256 processors of Teragrid, BlueGene, and System X in Figure 5. The base case for computing the parallel efficiency is chosen such that the problem fits *comfortably* in the available distributed memory. The code shows excellent scaling on the three machines. The superlinear speedup on the Teragrid cluster is primarily due to the superior cache performance offsetting the increased communication time as the problem size per-processor gets smaller. However, this also shows that we have room for doing more per-processor (memory hierarchy) performance optimizations that will benefit the base case the most. We are currently investigating these performance issues.

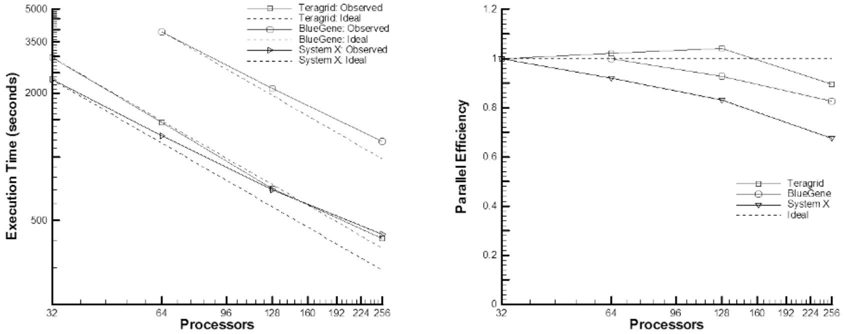


Fig. 5. Execution time (left) and parallel efficiency (right) on 256 processors of Teragrid (1.5 GHz Intel Madison dual processors nodes), BlueGene (700 MHz dual processor nodes), and System X (dual 2.3 GHz PowerPC 970FX processor nodes).

3.3 Analytical vs. Jacobian-free NKS

The diffusion-limited radiation transport equation presents challenging nonlinear behavior. At the same time, it is easy to code the analytical Jacobian (which is used for the preconditioning matrix, as well). The analytical Jacobian has the advantage of possibly better performance (see Table 1) but requires memory to store the matrix explicitly. Another convenient approach is to perform the matrix-vector products (as needed for the Krylov solver) without explicitly forming the Jacobian matrix [7]. This has the obvious advantage of savings in the memory requirements (though the preconditioner matrix may still need to be stored) but requires extra function evaluations. This will compete well with the analytical Jacobian case only when the execution time for function evaluation is significantly (perhaps an order of magnitude) less than the time to compute the analytical Jacobian

In Table 1, we compare the performance of three choices for the time-accurate NKS algorithm:

- **analytical Jacobian** computed to the same order of accuracy as the function in every nonlinear iteration. The preconditioner matrix is chosen to be the same as the Jacobian matrix. The computational cost of this part dominates the execution time: 62% on 256 processors of Teragrid.
- **Jacobian-free** matrix-vector products performed without explicitly forming the Jacobian matrix. However, this matrix (or a cheap approximation to it) is often needed for preconditioning purpose in every nonlinear iteration. The dominating cost here is the function evaluation (about 48% in Table 1)

- **lagged Jacobian-free** matrix-vector products performed the same way as in the previous item but preconditioner matrix is built only once per timestep and reused for all the linear solves with in a timestep. This saves time spent on the preconditioner evaluation but often requires more linear and nonlinear iterations (and thus function evaluations) than the previous choice. One can even freeze the preconditioner evaluation for many timesteps but it should be done only when the step size is small or when there is little change from one step to the next.

We observe that the analytical Jacobian does the best among the three choices in terms of the total wallclock time. The ratio of the cost of one Jacobian evaluation to that of one function evaluation on 256 processors of Teragrid is about thirteen while there is sixteenfold increase in the number of function evaluations for Jacobian-free case (as compared to the analytical Jacobian case). Therefore, the Jacobian-free approach is not competitive in the present scenario even if we assume that the time spent on the preconditioner evaluation is negligible (which may not be the case). However, a short code development cycle and savings in memory must often be considered while choosing Jacobian-free versus analytical approaches.

Table 1. Performance comparison of analytical Jacobian, Jacobian-free, and lagged Jacobian-free NKS methods. In the lagged case, Jacobian is evaluated only once per timestep. For the 3D test of Figure 3 on 256 processors of Teragrid.

Number of	Analytical	Jacobian Free	Lagged Jacobian-free
Time Steps	986	986	986
Nonlinear Iter	4,812	4,812	5,778
Linear Iter	92,843	92,842	92,016
Function Eval	6,140	98,982	99,146

Execution Time of	Analytical	Jacobian Free	Lagged Jacobian-free
Function Eval	25	395	396
Jacobian Eval	254	0	0
PC Eval	0	254	52
Total	412	823	601

4 Conclusions and future work

The time-accurate NKS algorithm scales well on Teragrid, BlueGene, and System X platforms. However, the per-processor performance needs memory hierarchy optimizations. This might make the function evaluation phase relatively cheaper, which in turn, can make the Jacobian-free approach competitive. Higher order time integration poses more difficult nonlinear systems by allowing larger timesteps, but they can be more computationally efficient overall by completing the time marching in fewer steps. Future work will also address more complex computational geometries with irregularly shaped zones of different atomic number, as often encountered in practice.

5 Acknowledgments

We thank Dimitri Mavriplis of the Univ. of Wyoming and Paul Fischer of Argonne Nat. Lab. for many helpful discussions. Computer time was supplied by Argonne Nat. Lab., by NSF's Teragrid facility, and by Virginia Tech.

References

1. S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang, *PETSc home page.* http://www.mcs.anl.gov/petsc, 2001.
2. P. N. Brown, G. D. Byrne, and A. C. Hindmarsh, *VODE: A variable-coefficient ODE solver*, SIAM J. Sci. Stat. Comp., 10 (1989), pp. 1038–1051.
3. P. N. Brown, D. E. Shumaker, and C. Woodward, *Fully implicit solution of large-scale non-equilibrium radiation diffusion with high order time integration*, J. Comput. Phys., 204 (2005), pp. 760–783.
4. W. D. Gropp, D. K. Kaushik, D. E. Keyes, and B. F. Smith, *High performance parallel implicit CFD*, Journal of Parallel Computing, 27 (2001), pp. 337–362.
5. T. J. R. Hughes, *The Finite Element Method: Linear Static and Dynamic Finite Element Analysis*, Dover Publications, Inc., Mineola, NY, 2000.
6. G. Jothiprasad, D. J. Mavriplis, and D. A. Caughey, *Higher order time integration schemes for the unsteady Navier-Stokes equations on unstructured meshes*, in Proceedings of 32nd AIAA Fluid Dynamics Conference and Exhibit, St. Louis, MO, June 2002. AIAA 2002-2734.
7. D. A. Knoll and D. E. Keyes, *Jacobian-free Newton-Krylov methods: a survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397.
8. D. J. Mavriplis, *Multigrid approaches to non-linear diffusion problems on unstructured meshes*, Numerical Linear Algebra with Applications, 8 (2001), pp. 499–512.
9. D. Mihalas and B. Weibel-Mihalas, *Foundations of Radiation Hydrodynamics*, Dover Publications, Inc., Mineola, NY, 1999.
10. M. Pernice and B. Philip, *Solution of equilibrium radiation diffusion problems using implicit adaptive mesh refinement*, SIAM J. Sci. Comput., 27 (2006), pp. 1709–1726.

Adaptive Parareal for Systems of ODEs

David Guibert [1] and Damien Tromeur-Dervout [1]

CDCSP/UMR5208 Institut Camille Jordan, University Lyon 1-CNRS, 69622
Villeurbanne Cedex, France.

Summary. The parareal scheme (resp. PITA algorithm) proposed in [3] (resp. [2]) considers two levels of grids in time in order to split the domain in time-subdomains. A prediction of the solution is computed on the fine grid in parallel. Then at each interface between the time subdomains, the solution makes a jump between the previous initial boundary value (IBV) of the next time-subdomain . A correction of the IBV for the next fine grid iteration is then computed on the coarse grid in time. In this paper, we study adaptivity in the time slice decomposition based on an a posteriori numerical estimation obtained from the time step behavior on coarse grids. The outline of this paper is as follows: in section 1, the original parareal method is recalled and it is shown that it is a particular case of the multiple shooting method of Deuflhard [1]. Then in section 2, the definition of the size of the fineness of the grids is slightly modified in order to introduce adaptivity within the parareal algorithm for the time stepping, the number of subdomains, and the time decomposition. This adaptivity leads to an improvement of the method and enables us to solve moderately stiff nonlinear ODEs problems. Nevertheless for very stiff problems as the Oregonator model, it fails even with the introduced adaptivity. This leads to develop in section 3 an adaptive parallel extrapolation method, based on a posteriori numerical assessment, which obtains results for this stiff problem.

1 Parareal and the BVPSOL multiple shooting method

The principle of the parareal algorithm to solve

$$\frac{dy}{dt} = f(y,t), \quad \forall t \in \Omega_t =]T^0, T^f], \quad \text{with } \; y(t^0) = y_0 \tag{1}$$

[1] This work was funded by the thema “mathématiques” of the Région Rhône-Alpes through 2003-2006 the project: “Développement de méthodologies mathématiques pour le calcul scientifique sur grille”. The second author is partially supported by the GDR-CNRS MOMAS through the project: “Simulation et solveurs multidomaines”

consists in splitting the time domain in m time-slices $\left\{S^i = [t^i, t^{i+1}]\right\}$ of different sizes with $t^0 = T^0$ and $t^m = T^f$. Let Y^i denote the values of the exact solution of problem (1) at the beginning of the time-slice S^i . The principle of the parareal algorithm consists in defining an approximation Y_k^i of these Y^i on a coarse grid. With Y_k^i known, the solution $y_k^i(t)$ on the m time-slices S^i can be computed as:

$$\frac{dy_k^i}{dt} = f(y_k^i, t), \ \ \forall t \in S^i, \ \text{ with } \ y_k^i(t^i) = Y_k^i. \tag{2}$$

These solutions exhibit jumps $\Delta_k^i = y_k^{i-1}(t^i) - Y_k^i, \ \ 1 \le i \le m-1$ at the time-instances t^i . A correction function c_k piecewise C^1 in Ω_t is introduced to update the Y_k^i values with a Newton-type linearized method around y_k ,

$$\frac{dc_k}{dt} = F_y(y_k, t)c_k, \text{ with } c_k(t^0) = 0, \text{ and} \tag{3}$$

$$c_k(t^{i+}) = c_k(t^{i-}) + \Delta_k^i, \ 1 \le i \le m-1. \tag{4}$$

It exists a link between the parareal method and the multiple shooting method (BVPSOL , [1]). As in the parareal method, the multiple shooting method uses a Newton process to supress the jumps of the solution at the end of time slices. If $\left\{T^0 = t^1 < t^2 < ... < t^m = T^f\right\}$ where $m > 2$, represents a decomposition of the time interval and x_j estimates the unknown values at the nodes t_j , then the solution with the initial value x_j on the time slice $S^j = [t^j, t^{j+1}]$ can be written as $y_j(t) = \Phi^{t,t^j} x_j, t \in S^j, j = 1, ..., m-1$ where Φ^{t,t^j} represents the flow trajectory starting from t^j .

For the solution of the problem the sub-trajectories have to be joined continuously and hence at the intermediate nodes the n continuity conditions $F_j(x_j, x_{j+1}) = \Phi^{t,t^j} x_j - x_{j+1} = 0, j = 1, ..., m-1$, have to hold. In addition the n boundary conditions $F_m(x_1, x_m) = r(x_1, x_m) = 0$ must be verified.

$$x = (x_1, \ldots, x_m)^T \in R^{n.m}, F(x) = (F_1(x_1, x_2), \ldots, F_m(x_1, x_m))^T \tag{5}$$

The BVPSOL finds the zeros of F by means of an ordinary Newton correction

$$F'(x_k)\delta x_k = -F(x_k), x_{k+1} = x_k + \Delta x_k, k = 0, 1, ... \tag{6}$$

The corresponding Jacobian matrix has the block cyclic structure:

$$J = F'(x) = \begin{bmatrix} G_1 & -I & & \\ & \ddots & \ddots & \\ & & G_{m-1} & -I \\ A & & & B \end{bmatrix} \tag{7}$$

where A et B are the derivatives of the boundary conditions r with respect to the boundary values (x_1, x_m) and $G_j = \partial\Phi^{t,t_j} x_j / \partial x_j, j = 1, ..., m-1$.

Proposition 1. *If* $B = 0$ *then one iteration of the multiple shooting method is one iterate of the parareal algorithm in which the correction step is a purely sequential process.*

Proof. the solution of eq (6) with the Jacobian matrix system given by eq (7) is reduced to solving [[1],p.319]:

(a) evaluate by recursion over $j = 1, \ldots, m-1$ $E := A + BG_{m-1} \ldots G_1$,
 $u := r + B[F_{m-1} + G_{m-1}F_{m-2} + \ldots + G_{m-1} \ldots G_2 F_1]$
(b) solve the linear (n,n)-system $E\varDelta x_1 = -u$
(c) Execute explicit recursion

$$\varDelta x_{j+1} = G_j \varDelta x_j + F_j,\ j = 1, \ldots, m-1 \tag{8}$$

If $B = 0$ then $E = A$ and $u = r$ thus eq (8) is reduced to the parareal eq (4).

$\sharp$

A simple way of transforming an IBV problem to a BVP is to consider the following statement. Instead of considering the problem on the $[0, T]$ time span, one can consider a time forward integration from 0 to T and then a time backward integration from T to 0 . Then eq (7) becomes:

$$J = F'(x) = \begin{bmatrix} G_1 & -I & & & & & & \\ & \ddots & \ddots & & & & & \\ & & G_{m-1} & -I & & & & \\ & & & \bar{G}_{m-1} & -I & & & \\ & & & & \ddots & \ddots & & \\ & & & & & \bar{G}_1 & -I & \\ A & & & & & & A & \end{bmatrix} \tag{9}$$

where $\bar{G}_j$ is related to the integration with time step $-h$.

Proposition 2. *Consider the IBV problem (1) with* $f(t, y) = -\alpha y$ *,* $\alpha > 0$ *, then the error when solving the BVP (9) with the first order Euler explicit scheme is of the same order as the consistency error of the scheme. More precisely the error produced at the end boundary value is of order* $O(h)$ *.*

Proof. The Euler forward time integration scheme on $]0, T]$ with $h = T/n$ and IBV y_0 gives $y(T) \simeq y_n = (1 - h\alpha)^n y_0$ then the Euler backward time integration on]T,0] with IBV y_n gives $y_{2n+1} = (1 + h\alpha)^n y_n = (1 - (h\alpha)^2)^n y_0$ instead of y_0 . Thus the error with the IBV y_{2n+1} is $O(n\alpha^2 T^2/n^2) = O(h)$ when n is sufficiently large.

$\sharp$

2 Parareal revisited

Test problems with increasing stiffness

Consider three classical IBV problems with increasing stiffness to validate and to give the limitation of the given methodologies. The first model (eq (10)) is a linear problem with oscillations. Thus the linearization of the correction step is exact as the jacobian matrix is constant. The second (eq (11)) is the prey-predator model with the lotka-Volterra 2×2 system of nonlinear ODEs. One can increase the solution oscillations by playing with the multivalued parameter μ. The third ((eq (12)), is the Oregonator model with a 3×3 system of nonlinear ODEs associated with the Belousov-Zhabotinskii (BZ) reaction. This problem can be very stiff for some ranges of the parameter μ.

$$\begin{cases} dx/dt = -\mu_1 x + \mu_2 y \\ dy/dt = \mu_3 x + -\mu_4 y \end{cases} \tag{10}$$

$$\begin{cases} dx/dt = \mu_1 x - \mu_2 xy \\ dy/dt = \mu_3 y - \mu_4 xy \end{cases} \tag{11}$$

$$\begin{cases} \mu_2 dx/dt = \mu_1 y - xy + x(1-x), \\ \mu_3 dy/dt = -\mu_1 y - xy + \mu_4 z, \\ dz/dt \quad = x - z. \end{cases} \tag{12}$$

Introducing adaptivity

A new definition for the grids fineness is introduced below. The fine and coarse grids are not defined by the size of the time steps but by the relative tolerance of the time integrator. The advantage is that adaptivity in the time step can occur in order to overcome strong nonlinearities. For two approximations of order p and $\hat{p}$ of the solution y_1 and $\hat{y}_1$ the error estimate for the less precise is $y_1 - \hat{y}_1$. The time step h is chosen to give $|y_{1i} - \hat{y}_{1i}| \leq max(y_{0i}, y_{1i})Rtol = sc$. The new time step h_{new} is obtained as follows $h_{new} = h.min(facmax, max(facmin, fac.(1/err)^{1/(q+1)}))$ where $q = min(p, \hat{p})$ and $fac, facmin, facmax$ are constant factors to avoid too fast decrease/increase of the time step, and $err = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\frac{y_{1i} - \hat{y}_{1i}}{sc})^2}$. The computational complexity of the modified Rosenbrok(2,3) method (implemented in the ode23s matlab procedure for the test IBV problems eqs (11) and (12)) has a number of function evaluations that increases nearly by 2.3 when rtol is divided by 10. Let α be the reduction cost coefficient of the elapsed time of the time integrator between run on the grid defined by rtol and those on the grid defined by rtol/10. This coefficient α can be considered as a constant. This assumption seems reasonable considering the previous result, excepted for the rejected step which increases nonlinearly when rtol decreases. Let $TIcost(rtol)$ be the elapsed time of the time integrator which solves the problem with a relative tolerance rtol. Let $PararealCost(rtol, rtol/10^l)$ be the computational cost to solve with the modified with the parareal method. This method is applied on P processors using two grids defined by $rtol$ and $rtol/10^l$. Then to perform it iterations, the cost can be roughly approximated by $it \times (1/P + \alpha^l) \times TIcost(rtol)$.

Techniques to evaluate the error and linear problem results

The numerical process to give a measure on the error quality is as follows: first, a fine grid solution is computed with the time integrator using an accuracy $rtol = 10^{-d}$, $d = 8$. Then the solution of the modified parareal method is computed with a fine grid defined by $rtol = 10^{-l}, l < d$. As the time integrator can use different step sizes, the two solutions do not match at the same time interpolation points. The solution of reference is then interpolated to the time mesh generated by the parareal algorithm.

Table 1 (top) shows the effect of the interpolation techniques (linear, cubic, spline) on the measure of error in the maximum norm for the linear problem with 10 subdomains and $rtol = 10^{-7}$ for initializing and for correction. Convergence is guaranteed in at most 10 iterations as the exact IBV is propagated. Good results are obtained with cubic interpolation which will be used to give all results that follow. Table 1 (bottom) shows the convergence of the modified parareal method for linear problem with $rtol = \{10^{-4}, 10^{-5}, 10^{-6}\}$ accuracy for the initial guess and the correction, and $rtol = 10^{-7}$ for fine grid solution.

	linear	cubic	spline
error (parareal iterates)	-4 (2)	-6 (2)	-6 (2)

coarse grid 10^d	d=-4	d=-5	d=-6
error (parareal iterates)	-6(6)	-6(4)	-6(3)

Table 1. Effect of the interpolation technique on the error measurement (top) and convergence of the modified parareal algorithm for eq (10) with $\mu = (-1e-3, 1, 10, -1e-3)$ on 10 subdomains.

Numerical results for nonlinear test problems

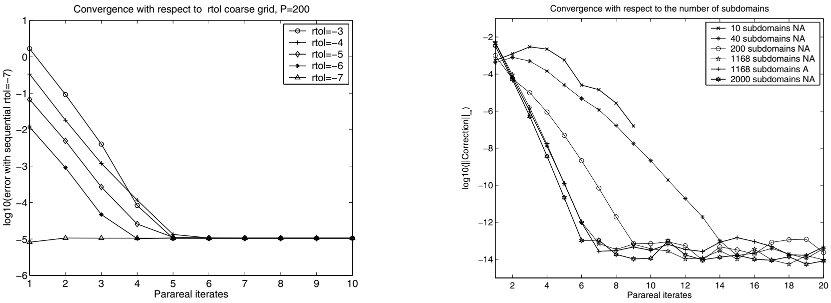


Fig. 1. Modified parareal convergence for Lotka-Volterra Problem on $\mu = (1.5, 1, 3, 1)$ with 200 subdomains with respect to different rtol for the initialization and the correction (left). The convergence of the correction in the parareal method with respect to the number of subdomains with and without adaptivity and $rtol = 10^{-5}$ for the correction (right).

Figure 1 shows the convergence of the method for the Lotka-Volterra problem. For 10 time subdomains, the method blows up for $rtol = 10^{-3}, 10^{-4}$ and finally converges at the 10^{th} iterate. Even with $rtol = 10^{-7}$ convergence takes 7 iterates. For this number of subdomains the method has no interest. Nevertheless, if the number of equal size subdomains is increased to 200, the convergence is reached between 5 and 7 iterates for $rtol = 10^{-6}$ and 10^{-3}, providing a speed up. For 1168 subdomains convergence is obtained in between 2 and 7 iterates for $rtol = 10^{-6}$ to 10^{-3}. Let us notice that the correction can converge to 10^{-14} but the convergence to the solution is limited by the fine grid solver (here $rtol = 10^{-7}$. It is not necessary to reach the machine accuracy for correction to have the effective convergence on the fine grid.

The behavior of the step size is the same for the rtol grids. Moreover, the reduction of the step size indicates directly the stiffness of the solution. The behavior of the time integrator adaptivity on the coarse grid can be useful to introduce adaptivity in the time decomposition. This decomposition is based on the time steps of the time integrator during the coarse grid initialization. Then the size of the subdomain is adapted with respect to strong variations of the step size. Figure 1 shows that the number of subdomains is 1168 defined by the adaptivity (A) gives better results. Nevertheless, this is not the optimal number of subdomains, since 2000 regular subdomains (NA) lead to a faster convergence, and 1168 regular subdomains give quite the same convergence. For the Oregonator problem, the convergence fails even with 1000 subdomains for $rtol = 10^{-6}$ to 10^{-3} and even with time decomposition adaptivity. Theses experiments show that new solutions are needed for stiff and very stiff problems in keeping with the two major features of parareal algorithm: split the time domain in slices then provide a first good initial boundary value at each slice. The correction of the solution based on linearizing and solving a problem with the jacobian of f seems to be sensitive to the behavior of the solution notably when strong nonlinearity effects occur.

3 Adaptive Parallel Extrapolation

The same behavior for the time step adaptation provided by the solver on coarse grids suggests that we can use some combination of solutions like “Richardson extrapolation” based on the solver. Moreover, as the first subdomain has the exact solution, we can compute from the different grid levels exact extrapolation coefficients based on some control points values. Then the extrapolation coefficients can broadcast to the others subdomains where extrapolations are performed. Let us describe the algorithm.

Adaptive Parallel Extrapolation Algorithm:

A) Define some decomposition of the time domain and in each subdomain add some control points $(t_{i,k})_{0 \leq k \leq l}$ which are points in the time slice $[t_{i,0}, t_{i,l}] = [T_i, T_{i+1}]$

B) Evaluate the solution on coarse grids $rtol_1 > ... > rtol_l$.

C) Initialize IBV of time slices for the finest grid with $rtol_f$ with a Richardson extrapolation based on the coarse grids and the first time slice of the fine grid $rtol_f$ as it gives the exact solution. Use the value of the solution at the control points (without the first control point which is a given data and consequently is not a result provided by the time integrator approximation scheme) on this first time slice to define the operators of extrapolation. Let $y^k(t_1, j), 1 \leq j \leq lP, 1 \leq k \leq l$ be the computed solution values on the $rtol_k$ grid at the control point $t_{1,j}$. The extrapolation operator can be computed with the formula as follows:

$$\begin{pmatrix} y^1(t_{1,1}) & y^2(t_{1,1}) & \cdots & y^l(t_{1,1}) \\ y^1(t_{2,1}) & y^2(t_{2,1}) & \cdots & y^l(t_{2,1}) \\ \vdots & \vdots & \ddots & \vdots \\ y^1(t_{l,1}) & y^2(t_{l,1}) & \cdots & y^l(t_{l,1}) \end{pmatrix} \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_l \end{pmatrix} = \begin{pmatrix} y^{l+1}(t_{1,1}) \\ y^{l+1}(t_{2,1}) \\ \vdots \\ y^{l+1}(t_{l,1}) \end{pmatrix} \tag{13}$$

D) Propagate the operator of extrapolation to the other time slices and compute the extrapolated solution as follows:

$$y^{l+1}(t_{0,j}) = \sum_{k=1}^{l} \beta_k y^k(t_{l,j-1}),\ 2 \leq j \leq P \tag{14}$$

E) In order to get the time step behavior lost in the extrapolation compute in parallel the solution on each time-slice for the finest grid. The first time-slice has the exact solution for the finest grid (exact IBV).

F) Apply recursively with a new $rtolf$ grid.

Results on Adaptive Parallel Extrapolation

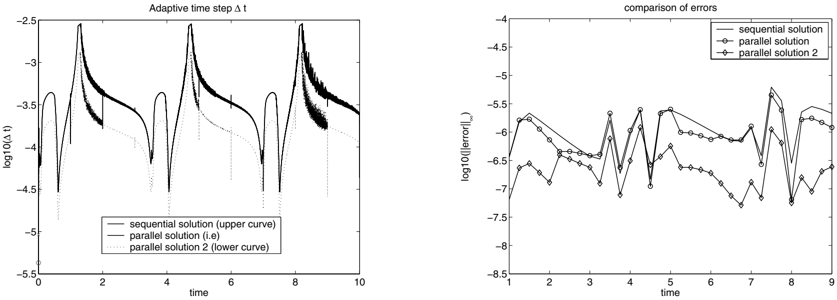


Fig. 2. Comparison of the time step behavior (left) and Error with the solution on a reference grid ($rtol$ = 10−10) (right) between the sequential algorithm and the Adaptive Parallel Extrapolation with 2 level of extrapolation for the Oregonator problem defined by $(\mu = (1e-2, 1e-3, 1e-2, 1)$.

Figure 2 gives the error at the control point between the sequential solution at $rtol = 10^{-10}$ and the Adaptive Parallel Extrapolation with 10 time subdomains and with a fine grid $rtol = 10^{-8}$ for the Oregonator problem. It shows that good approximations of the initial guess for each time slice are obtained even when using two grids with $rtol = 10^{-6}$ and 10^{-5} . The computational cost to define a very good approximation of the time slice initial guess is reduce by a factor nearby 2.3^2 for the considered time integrator. Notice that the behaviors of the time step are the same between the sequential solution and the first level solution of adaptive parallel extrapolation. The recursive application of the solution obtained with the Adaptive Parallel Extrapolation gives globally better results excepted for some localized region. Improvements should be obtained with a local refinement of the grids used for the extrapolation.

Conclusions

The equivalence between the parareal method and the multiple shooting method has been established. Then adaptivity in the parareal parallel ODEs solvers has been introduced in order to apply its concepts to stiff ODEs. Some improvements in the method has been shown by defining the fineness of the grids on the relative tolerance of the time slice integrator and by adapting on the number of subdomains. Nevertheless, for very stiff problems, the linearization of the jacobian in the correction steps makes the method very sensitive to blow up. Another parallel solver has been proposed for stiff ODEs based on Richardson Extrapolation. The extrapolation coefficients are based on the time integrator behavior like in the classical Richardson extrapolation but with an a posteriori estimation based on the correct solution values in the first subdomain at certain control points.

References

1. P. Deuflhard, *Newton Methods for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*, vol. 35 of Series in Computational Mathematics, Springer, 2004.
2. 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 (2003), pp. 1397–1434.
3. J.-L. Lions, Y. Maday, and G. Turinici, *Résolution d'edp par un schéma en temps "pararéel"*, C. R. Acad. Sci. Paris, Série I, 332 (2001), pp. 661–668.

A Fast Helmholtz Solver for Scattering by a Sound-soft Target in Sediment

Quyen Huynh [2] , Kazufumi Ito [1] , and Jari Toivanen [1]

[1] Center for Research in Scientific Computation, Box 8205, North Carolina State University, Raleigh, NC 27695–8205, USA. kito@ncsu.edu, jatoivan@ncsu.edu

[2] Code R21, Littoral Acoustics, Naval Surface Warfare Center, 110 Vernon Ave., Panama City, FL 32407–7001, USA. quyen.huynh@navy.mil

1 Introduction

We consider an efficient numerical method for computing time-harmonic acoustic scattering in a vertically layered media. One application for such problems is the detection of targets buried in a sediment. For this purpose it is useful to have a numerical approximation which can predict reasonably accurately the backscatter by such targets. In this paper, we study scattering by sound-soft targets when the interface between the water and sediment is wavy. Such problems are typically modeled using a Helmholtz equation with varying coefficients.

With higher frequencies a finite element discretization leads to very large systems of linear equations. Often two-dimensional problems have millions of unknowns. It might be possible to solve these problems using a LU factorization with a nested dissection reordering of unknowns, but this approach cannot be used for three-dimensional problems which can have billions of unknows. For this reason, we consider the iterative solution of these problems. We employ an algebraic fictitious domain method [4, 6, 7, 8] which uses a right preconditioned GMRES method.

In a related work [12], it was noted that an iterative method with a separable preconditioner converges fast as long as the media is mainly layered in one direction or frequencies are reasonably low. We will use a separable preconditioner based on the perfectly layered media in our solution procedure. We embed the sound-soft target in a rectangular computation domain with a second order-absorbing boundary condition. Since the media is vertically layered with a wavy interface, our preconditioner coincides with the system matrix except for the rows corresponding to unknowns near-by the interface and the target. Thus, we can reduce the iterations to a small sparse subspace as has been shown in [7, 8]. This reduction makes our preconditioner extremely efficient as our numerical example demonstrates.

2 Model Problem

We are interested in computing the scattering of a time-harmonic acoustic pressure wave by a target which is buried in sediment. We model this situation with a

Helmholtz equation with varying coefficients. Generally, it might be necessary to use elastic equations to model the wave inside the target, but in this investigation, we assume the target to be sound-soft. This means that a Dirichlet boundary condition can be posed on the surface of the target. The sediment is assumed be homogeneous and the surface between the water and the sediment is defined by $x_2 = f(x_1)$, where f is a given function.

We have a radiational wave from a point source in the water which is impinging the sediment and the target Ω. Furthermore, we could have a sensor in the water measuring the scattered wave. For the computations, we truncate the infinite domain to the rectangular domain Π enclosing the target and the source/sensor. Figure 1 shows the set up of our model problem.

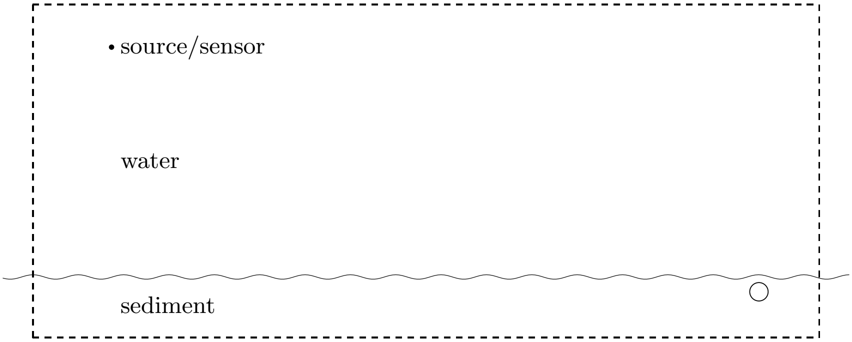


Fig. 1. The geometry of the model problem with a circular target Ω and a rectangular truncated domain Π given by the dashed line.

A time-harmonic acoustic pressure wave p satisfies the Helmholtz equation with varying coefficients

$$\begin{aligned} \nabla \cdot \frac{1}{\rho_1} \nabla p + \frac{k_1^2}{\rho_1} p = g & \qquad \text{in } x_2 > f(x_1), \\ \left. \frac{1}{\rho_1} \frac{\partial p}{\partial n} \right|_{+} = \left. \frac{1}{\rho_2} \frac{\partial p}{\partial n} \right|_{-} & \qquad \text{on } x_2 = f(x_1), \\ \nabla \cdot \frac{1}{\rho_2} \nabla p + \frac{k_2^2}{\rho_2} p = 0 & \qquad \text{in } x_2 < f(x_1) \quad \text{and} \quad x \notin \Omega, \end{aligned} \tag{1}$$

where $k_1 = \dfrac{\omega}{c_1}$ and $k_2 = \dfrac{\omega}{c_2}$ are the wave numbers for the water and the sediment, respectively. The normal of the surface $x_2 = f(x_1)$ is denoted by n. The notation $\Big|_{+}$ refers to the value of a function or derivative when approaching $x_2 = f(x_1)$ from the side $x_2 > f(x_1)$. Similarly $\Big|_{-}$ refers to the value of a function or derivative when approaching $x_2 = f(x_1)$ from the side $x_2 < f(x_1)$. The angular frequency is denoted by ω. The sound speed in the water is c_1 and in the sediment it is c_2. The wave attenuation in the sediment is modeled by the imaginary part of the complex-valued speed c_2 . The densities for the water and the sediment are ρ_1 and ρ_2 , respectively. The right-hand side g is non zero due to the point source.

On the boundary of the sound-soft target Ω we pose a Dirichlet boundary condition

$$p = 0 \qquad \text{on } \partial\Omega. \tag{2}$$

On the artificial boundary $\partial\Pi$ of the truncated rectangular domain Π , we pose a second-order absorbing boundary condition

$$\frac{1}{\rho}\frac{\partial p}{\partial n} - i\frac{k}{\rho}p - i\frac{1}{2k}\frac{\partial}{\partial s}\frac{1}{\rho}\frac{\partial p}{\partial s} = 0 \tag{3}$$

on the faces of $\partial\Pi$ together with the condition $\partial p/\partial n = ik\frac{3}{2}p$ at the corners of $\partial\Pi$. Here, n denotes the unit outward normal vector of $\partial\Pi$ and s denotes the unit tangent vector of $\partial\Pi$. Furthermore, the wave number function k and the density function ρ are defined by

$$k = \begin{cases} k_1, \, x_2 \geq f(x_1) \\ k_2, \, x_2 < f(x_1) \end{cases} \qquad \text{and} \qquad \rho = \begin{cases} \rho_1, \, x_2 \geq f(x_1) \\ \rho_2, \, x_2 < f(x_1). \end{cases}$$

A similar absorbing boundary condition for homogeneous media has been considered in [1].

3 Finite Element Discretization

We discretize the equations (1) together with the Dirichlet boundary condition (2) and the absorbing boundary condition (3) with linear finite elements. We use meshes which are orthogonal and uniform except near the target Ω and the interface. There it is locally perturbed so that the boundary $\partial\Omega$ is approximated well. An algorithm generating such meshes is presented in [3]. An example of a locally perturbed mesh is shown in Figure 2. The meshes have to be sufficiently fine, say, with at least 10 nodes per one wavelength, so that they can approximate properly the oscillatory solution. The discretization leads to a system of linear equations

$$Ap = g, \tag{4}$$

where the matrix A is symmetric and complex-valued, but not Hermitean.

4 Separable Preconditioner

We describe first the construction of our separable preconditioner and after that we consider in Section 5 the algebraic extension of the original system of linear equations (4) to have the same dimension as the preconditioner.

Domain embedding and fictitious domain methods are based on very efficient preconditioners on simple shaped domains. In our particular case the simple shaped domain is the whole rectangle Π , i.e., we neglect the target Ω when constructing the preconditioner. Our separable preconditioner is based on the observation that the density ρ and the wave number k depend only on the x_2 coordinate for the

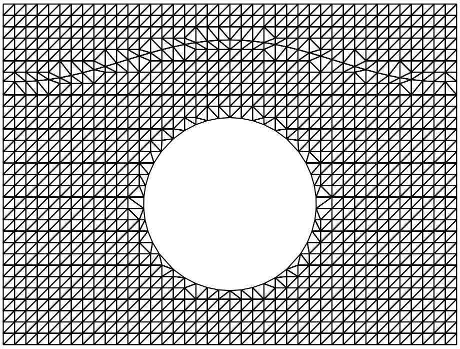


Fig. 2. A part of a locally perturbed mesh for a circular target and a sinusoidal surface of sediment.

perfectly layered media. Due to this we can express our preconditioner in a tensor product form

$$B = A_1 \otimes M_2 + M_1 \otimes (A_2 - \widetilde{M_2}).$$

This preconditioner coincides with the matrix obtained by discretizing the problem (1) without the target $\varOmega$ together with the boundary condition (3) except on a part of the left and right boundary of $\varPi$. The dimension of the matrices A_1 and M_1 is the same as the number of nodes in the x_1 direction and they are given by

$$A_1 = \frac{1}{h} \begin{pmatrix} 1 - ihk/2 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & \ddots & \ddots & \ddots & \\ & & & -1 & 2 & -1 \\ & & & & -1 & 1 - ihk/2 \end{pmatrix}$$

and

$$M_1 = h \begin{pmatrix} 1/2 + i/(2hk) & & & & & \\ & 1 & & & & \\ & & 1 & & & \\ & & & \ddots & & \\ & & & & 1 & \\ & & & & & 1/2 + i/(2hk) \end{pmatrix}.$$

The matrices A_2, M_2, and $\widetilde{M_2}$ correspond to one-dimensional problems in the x_2 direction and their dimension is the number of nodes in the x_2 direction. They can be assembled from the element matrices

$$A_2^e = \frac{1}{h\rho_e} \begin{pmatrix} 1 & -1 \\ -1 & 1 \end{pmatrix}, \quad M_2^e = \frac{h}{2\rho_e} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}, \quad \text{and} \quad \widetilde{M_2^e} = \frac{k_e^2 h}{2\rho_e} \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix},$$

where ρ_e and k_e are the density and the wave number on the element e. Due to the absorbing boundary condition the following terms have to be added to these matrices: add $-ik/(2\rho)$ to the first and last diagonal entry of A_2, add $i/(2k\rho)$

to the first and last diagonal entry of M_2, and add $ik/(2\rho)$ to the first and last diagonal entry of $\widetilde{M_2}$. Systems of linear equations with the matrix B can be solved efficiently using, for example, the fast direct solver considered in [5].

5 Extended Linear System

We now extend the original system of linear equations (4) to have the same size as the preconditioner B. We will accomplish this by using the so-called absorbing extension [9]. The idea is to pose another problem in Ω which is a Helmholtz problem in Ω with an absorbing boundary condition on $\partial\Omega$. Furthermore, we introduce one sided coupling between the problems in $\Pi \setminus \Omega$ and Ω.

After a suitable permutation of rows and columns the preconditioner has the block form

$$B = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix},$$

where the first block row corresponds to the unknowns outside Ω. Thus, B_{11} has the same size as A in (4). We denote the extended system matrix by C. It has the block form

$$C = \begin{pmatrix} A & B_{12} \\ 0 & B_{22} + D \end{pmatrix},$$

where D is a diagonal matrix such that $B_{22} + D$ corresponds to a Helmholtz problem in Ω with a first-order absorbing boundary condition on $\partial\Omega$. In particular, the discretization is based on the orthogonal mesh without local adaptation to the boundary $\partial\Omega$. The extended system of linear equations reads

$$Cu = C\begin{pmatrix} p \\ q \end{pmatrix} = \begin{pmatrix} g \\ 0 \end{pmatrix} = f.$$

The vector q has to be zero, since the matrix block $B_{22} + D$ is non singular, and, thus, p satisfies also the original problem (4). For more details on the extension procedure we refer to [4, 6, 9].

6 Reduction to a Sparse Subspace

We solve the right preconditioned system of linear equations

$$CB^{-1}v = f, \quad u = B^{-1}v. \tag{5}$$

Our sparse subspace X is defined by $X = \text{range}(C - B)$. The jth component x_j of an arbitrary vector x in X can be nonzero only if the jth row of B and C do not coincide. Hence, the subspace X is called sparse. For the problems considered in this paper the dimension of X is very small compared to the size of the linear system (5).

Next we consider the reduction to the sparse subspace in the case of general right-hand vector f. We set $\hat{v} = v - f$ and we then have

$$CB^{-1}\hat{v} = f - CB^{-1}f = -(C - B)B^{-1}f = \hat{f} \in X,$$

where we have used the identity $CB^{-1} = I + (C - B)B^{-1}$. Furthermore, $\hat{v}$ satisfies

$$\left[I + (C - B)B^{-1}\right]\hat{v} = \hat{f} \tag{6}$$

and $\hat{v} \in X$. The reduced equation (6) is well suited for iterating on the subspace X .

If $r \in X$ then the Krylov subspace

$$\mathrm{span}\{r, CB^{-1}r, \cdots, (CB^{-1})^{k-1}\}$$

is a subspace of X . Thus, any iterative method based on the Krylov subspace for the solution of $CB^{-1}v = f$ generates a sequence of approximate solutions v^k in the subspace X provided that the initial iterate is $v^0 = f$. Moreover, the basic operation

$$(C - B)B^{-1}r, \quad r \in X,$$

which is repeated during the iterations requires solutions $B^{-1}r$ with r in the range of $(C - B)^T$. The dimension of this range is usually of the same order as the dimension of X . Hence we apply the partial solution technique [2, 10] for this evaluation. This can reduce the computational cost of these solutions to be order of N floating point operations, where N is the size of the linear system (5).

7 Numerical Example

The geometry of our example problem is a cross cut of the experiment set up in [11]. The interface between water and sediment is given by $x_2 = \cos(360^\circ x_1/(0.75\,\mathrm{m}))$ $(0.0368\,\mathrm{m})$. The target is circular and its diameter is one feet (0.3048 m) and its center is at $(0\,\mathrm{m}, -0.2524\,\mathrm{m})$. Thus, the target is 0.1 m below the median level of the interface. The speed of sound in the water is $c_1 = 1495\,\mathrm{m/s}$ and the speed of the sound in the sediment is $c_2 = (1668 - 16.8i)\,\mathrm{m/s}$. Here, the imaginary part of the speed accounts for wave attenuation. The density for the water and sediment are $\rho_1 = 1000\,\mathrm{kg/m^3}$ and $\rho_2 = 2000\,\mathrm{kg/m^3}$, respectively. The point source is located at $(-10.7\,\mathrm{m}, 3.8\,\mathrm{m})$. We have chosen the frequency to be 20 kHz which corresponds about the wavelength 0.075 m in water.

Our computational domain is $[-12\,\mathrm{m}, 1\,\mathrm{m}] \times [-1\,\mathrm{m}, 4.5\,\mathrm{m}]$ and the mesh is based on 2601×1101 grid. Thus, the mesh step size in the direction of the coordinate axes is $0.005\,\mathrm{m}$ and we have about 15 nodes per wavelength. We have plotted the scattered field intensity level in Figure 3. The computations were performed on a PowerBook G4 with a 1.3 GHz processor and 0.5 Gbytes of memory. The solution required about 5 minutes. The preconditioned GMRES method needed 36 iterations to reduce the norm of the residual by the factor 10^{-6} . The extended linear system (5) has about 2.86 million unknowns while the dimension of the sparse subspace X is 18418. Thus, memory and computational savings due to the use of subspace iterations are indeed extensive.

8 Conclusions and Future Research

We proposed a fast iterative method for computing the scattering in nearly vertically layered media. The main ingredients of our approach leading to computational efficiency are a fast direct solver for a separable preconditioner and a GMRES method

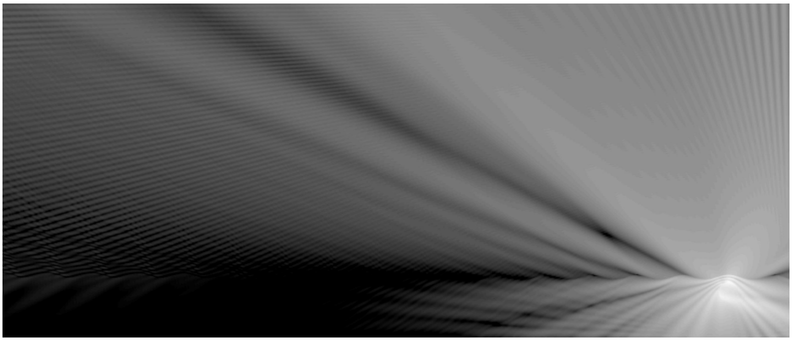


Fig. 3. The scattered field intensity level $\log_{10}|p_s|^2$; the difference between white and black is 60 dB.

iterating on a small sparse subspace. The numerical example demonstrates that problems with millions of unknows can be solved on a contemporary PC in a few minutes.

For considering more practical problems several generalizations have to be made. The proposed method can be extended in a straight forward manner to three-dimensional problems. Typical targets are elastic instead of sound-soft. In such a case one possible approach is to perform a domain decomposition to a small near-field domain and a large far-field domain. For far-field problems similar techniques to the one presented in this paper can used. Due to the small size of near-field problems they can be solved sufficiently fast using more traditional approaches.

References

1. A. Bamberger, P. Joly, and J. E. Roberts, *Second-order absorbing boundary conditions for the wave equation: a solution for the corner problem*, SIAM J. Numer. Anal., 27 (1990), pp. 323–352.
2. A. Banegas, *Fast Poisson solvers for problems with sparsity*, Math. Comp., 32 (1978), pp. 441–446.
3. C. Börgers, *A triangulation algorithm for fast elliptic solvers based on domain imbedding*, SIAM J. Numer. Anal., 27 (1990), pp. 1187–1196.
4. E. Heikkola, T. Rossi, and J. Toivanen, *A domain embedding method for scattering problems with an absorbing boundary or a perfectly matched layer*, J. Comput. Acoust., 11 (2003), pp. 159–174.
5. ———, *Fast direct solution of the Helmholtz equation with a perfectly matched layer or an absorbing boundary condition*, Internat. J. Numer. Methods Engrg., 57 (2003), pp. 2007–2025.
6. ———, *A parallel fictitious domain method for the three-dimensional Helmholtz equation*, SIAM J. Sci. Comput., 24 (2003), pp. 1567–1588.
7. K. Ito and J. Toivanen, *Preconditioned iterative methods on sparse subspaces*, Appl. Math. Letters, 19 (2006), pp. 1191–1197.
8. Y. A. Kuznetsov, *Matrix iterative methods in subspaces*, in Proceedings of the International Congress of Mathematicians, Vol. 1, 2 (Warsaw, 1983), Warsaw, 1984, PWN, pp. 1509–1521.

9. Y. A. Kuznetsov and K. N. Lipnikov, *3D Helmholtz wave equation by fictitious domain method*, Russian J. Numer. Anal. Math. Modelling, 13 (1998), pp. 371–387.
10. Y. A. Kuznetsov and A. Matsokin, *Partial solution of systems of linear algebraic equations*, in Numerical methods in linear algebra (Proc. Third Sem. Methods of Numer. Appl. Math., Novosibirsk, 1978, Akad. Nauk SSSR Sibirsk. Otdel., pp. 62–89. Russian.
11. C. L. Nesbitt and J. L. Lopes, *Subcritical detection of an elongated target buried under a rippled interface*, in Proceedings of OCEANS '04, vol. 4, IEEE, 2004, pp. 1945–1952.
12. R.-E. Plessix and W. A. Mulder, *Separation-of-variables as a preconditioner for an iterative Helmholtz solver*, Appl. Numer. Math., 44 (2003), pp. 385–400.

Numerical Simulation of Free Seepage Flow on Non-matching Grids

Bin Jiang[1] and John C. Bruch, Jr.[2]

[1] Department of Mathematics and Statistics, Portland State University, Portland, OR 97207, USA. bjiang@pdx.edu

[2] Department of Mechanical and Environmental Engineering and Department of Mathematics, University of California, Santa Barbara, CA 93106, USA. jcb@engineering.ucsb.edu

Summary. A new domain decomposition technique on non-matching grids for free boundary problems is considered. An iterative DD scheme is used to reduce the original free boundary problem to a sequence of problems on two subdomains, one of which includes the free boundary and is described by a variational inequality and the other includes the remainder of the problem and is described by a second order partial differential equation. In the subdomain which contains the free boundary, a fine grid is utilized in order to capture the free boundary more precisely; while in the other subdomain, a coarse grid is used in order to speed up the computation. At each step of the iteration, two sub-problems are solved by using a Robin boundary condition on the interface.

1 Introduction

Both overlapping and non-overlapping domain decomposition (DD) methods have been intensively studied for partial differential equations, see e.g. [9, 5, 10, 6, 2]. In the last few decades, mathematicians began to apply the overlapping domain decomposition methods to solve variational inequality problems. The basic idea is to split the original domain into several overlapping subdomains and solve the variational inequality on each subdomain via data transfer from the common area between those subdomains. [1, 8, 3] and their references provide many variants of this approach whereas convergence analyses of the algorithms and their application to many problems in different fields are provided.

However, for many practical problems in the engineering and industrial fields, it is much easier and more convenient to split the original domain into two or three non-overlapping subdomains and then take care of the problems in each subdomain where the original problem may show different behavior. Recently, a non-overlapping DD method which utilizes a Robin boundary condition on the common boundary between these subdomains was proposed in [4] for the variational inequality problem and a convergence analysis of the DD method was provided.

In this paper, we consider free seepage flow through a dam with a toe drain that can be considered as a variational inequality. We will apply the non-overlapping DD method to decompose the original problem into two sub-problems where the partial differential equation is treated in one subdomain while the variational inequality is treated in the second subdomain where the free boundary is located. Since our concern is to find the exact location of the free boundary in the second domain, non-matching grids are applied in those subdomains. In the first subdomain, a coarse grid is used in order to speed up the computation; while in the second subdomain which contains the free boundary, a fine grid is utilized in order to capture the free boundary more precisely. At each step of the iteration, two sub-problems are solved simultaneously by using a Robin boundary condition on the common boundary.

This paper is organized as follows. In Section 2, we formulate the seepage problem and apply the non-overlapping DD method to split the original problem into 2 sub-problems. In Section 3, we utilize the non-matching grid technique in those two subdomains for the discretization of the problem and then apply a finite difference method with projection on the non-matching grids. Numerical results are reported to show the advantage of our new algorithm. In Section 4, a summary of the paper and some future considerations are outlined.

2 Formulation of the problem

Many problems involving free boundaries can be reduced to the study of variational inequalities. In [1], the author proposed several domain decomposition methods to split the domain into two or more subdomains. Then by iterating between these subdomains he solved the whole problem and found the free boundary. However, these schemes require the solution of one problem at a time. Herein, a DD scheme will be used simultaneously.

In this paper, we consider a free boundary seepage problem of flow through a porous dam with a toe drain. For simplicity, the soil in the flow field is assumed to be homogeneous and isotropic, capillary and evaporation effects are neglected. In addition, the flow follows Darcy's law:

$$\overrightarrow{q} = -k\nabla[(\frac{p}{\rho g}) + y], \tag{2.1}$$

where $\overrightarrow{q}$ is the velocity vector, p is the pressure, k is the permeability of the soil, ρ is the density of the fluid, g is the gravitational acceleration, and y is the vertical coordinate(positive upward). The seepage velocity has a potential $\phi(x,y) = k[(\frac{p}{\rho g})+y]$. Meanwhile, let $\psi(x,y)$ be the stream function of the flow. In this study, the location of the free surface $\Gamma_0 = \{x, \overline{f}(x)\}$ and the seepage domain Ω need to be found, see Figure 1. The seepage domain is defined as:

$$\Omega = \{(x,y) : 0 < x \leq x_F,\ 0 < y < \alpha(x); x_F < x < x_C,\ 0 < y < \overline{f}(x)\},$$

where x_F and x_C are the distances in the x-direction to points F and C, respectively, and $\alpha(x)$ is the shape function of the dam profile.

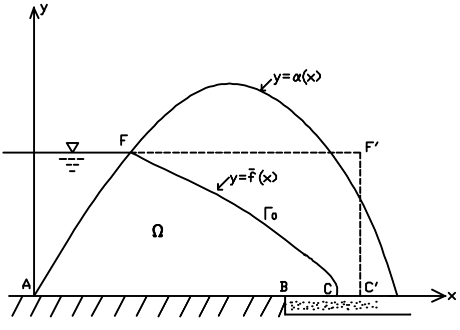


Fig. 1. The seepage problem.

The functions $\phi(x,y)$ and $\psi(x,y)$ are defined on $\overline{\Omega}$ and satisfy the following formulation:

$$\begin{array}{ll}
\Omega = \{(x,y) : 0 < x \le x_F, 0 < y < \alpha(x); x_F < x < x_C, 0 < y < \overline{f}(x)\} & \\
\phi_x - \psi_y = 0 & \text{in } \Omega \\
\phi_y + \psi_x = 0 & \text{in } \Omega \\
\phi = y_F & \text{on } \widehat{AF} \\
\phi = 0 & \text{on } [BC] \\
\psi = q & \text{on } [AB] \\
\psi = 0 & \text{on } \Gamma_0 \\
\phi = y & \text{on } \Gamma_0,
\end{array} \tag{2.2}$$

where y_F is the height at F, and q is the flow rate through the flowfield.

Define $D = \{(x,y) : 0 < x \le x_F, 0 < y < \alpha(x); x_F < x < x_{C'}, 0 \le y < y_F\}$ and extend ϕ and ψ continuously to $\overline{D}$ by setting $\overline{\phi}(x,y) = \phi(x,y)$ in $\overline{\Omega}; = y$ in $\overline{D} - \overline{\Omega}$ and $\overline{\psi}(x,y) = \psi(x,y)$ in $\overline{\Omega}; = 0$ in $\overline{D} - \overline{\Omega}$.

Next, we can define a new dependent variable w using the Baiocchi transformation on $\overline{D}$:

$$w(P) = \int_{\overline{FP}} -\overline{\psi}dx + (y - \overline{\phi})dy, \tag{2.3}$$

where $\overline{FP}$ is a smooth path in D joining F to P in D. The integration is indeed independent of the path. Then $w \in H^2(D) \bigcap C^1(\overline{D})$ satisfies:

$$\begin{array}{ll}
\Delta w = \chi_\Omega & \text{in D} \\
w_y = y - y_F & \text{on } \widehat{AF} \\
w = (\dfrac{q^2}{6}) + q(x_B - x) & \text{on } [AB] \\
w_y = 0 & \text{on } [BC] \\
w = 0 & \text{in } \overline{D} - \overline{\Omega}(\text{ also on } \Gamma_0) \\
w > 0 & \text{in } \Omega \quad (w \ge 0 \text{ in D}),
\end{array} \tag{2.4}$$

where $\chi_\Omega = 1$ in Ω and $\chi_\Omega = 0$ in $D - \Omega$. Hence,

$$w(x,y) \ge 0, \quad 1 - \Delta w(x,y) \ge 0, \quad w(1 - \Delta w) = 0 \text{ in } D. \tag{2.5}$$

If w is found satisfying (2.4), then we can determine $\Omega = \{(x,y) \in D : w(x,y) > 0\}$.

It will be seen shortly that if we can properly split D into two non-overlapping subdomains, the free boundary is only located in one subdomain, which makes the original problem simpler. Therefore, the DD method looks promising for this free boundary problem. [1] applied the non-overlapping D-N algorithm proposed in [6] to the above problem and the numerical results show that this D-N algorithm is better than the traditional one-domain finite difference scheme. However, no convergence property of the D-N algorithm has been proven.

Recently, a convergence analysis was provided for a non-overlapping DD method on uniform meshes with a Robin boundary condition applied to the general free boundary problem represented as a variational inequality [4]. In the following we use that DD scheme from [4] to solve the above seepage problem. First, decompose D into subsets $D_1 = \{(x,y) : 0 < x < x_F, 0 < y < \alpha(x)\}$ and $D_2 = \{(x,y) : x_F < x < x_{C'}, 0 < y < y_F\}$ with the interface between D_1 and D_2 denoted by $\Gamma = \{(x,y) : x = x_F, 0 < y < y_F\}$ in Figure 2. If w_1, w_2 denote the restriction of w in D_1 and D_2, respectively, we can write down the following iterative procedure:
Step 1. Initially set $g_1^1 = g_2^1 = 0$ on Γ.
Step 2. Solve the following two sub-problems for w_1^n and $\{w_2^n, \Omega_2^n\}$, $n = 1, 2, \cdots$, respectively:

Problem 1:

$$\begin{array}{ll} \Delta w_1^n = 1 & \text{in } D_1 \\ w_1^n = (\frac{q^2}{6}) + q(x_B - x) & \text{on } [AF_1'] \\ (w_1^n)_y = y - y_F & \text{on } \widehat{AF} \\ w_1^n + \frac{\partial w_1^n}{\partial n} = g_1^n & \text{on } \Gamma. \end{array} \tag{2.6}$$

Problem 2:

$$\begin{array}{ll} w_2^n(\Delta w_2^n - 1) = 0 & \text{in } D_2 \\ w_2^n = (\frac{q^2}{6}) + q(x_B - x) & \text{on } [F_1'B] \\ w_{2\,y}^n = 0 & \text{on } [BC'] \\ w_2^n \geq 0 & \text{in } \overline{D_2} \\ w_2^n + \frac{\partial w_2^n}{\partial n} = g_2^n & \text{on } \Gamma \\ \Omega_2^n = \{(x,y) : \quad w_2^n(x,y) > 0\}. & \end{array} \tag{2.7}$$

Step 3. Set

$$\begin{array}{ll} g_2^{n+1} = 2w_1^n - g_1^n & \text{on } \Gamma \\ g_1^{n+1} = 2w_2^n - g_2^n & \text{on } \Gamma. \end{array} \tag{2.8}$$

Then repeat Step 2 with n replaced by $n+1$. These iterations are stopped when $\max_\Gamma |w_1^{n+1} - w_1^n| < \varepsilon$ and $\max_\Gamma |w_2^{n+1} - w_2^n| < \varepsilon$, where ε is some fixed error tolerance. Problem 1 and 2 in Step 2 are solved using the Robin boundary condition values g_1^n and g_2^n which are updated iteratively from their previous value and the w value on the common boundary. This avoids the computation of the partial derivatives of w which may reduce the precision.

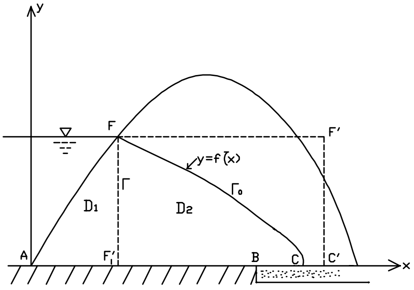


Fig. 2. The domain decomposition.

3 Non-Matching Grid Discretization and Results

In this Section, we will utilize a non-matching grid technique to obtain the numerical scheme for the above seepage problem on non-matching grids. At first, we apply the 2nd-order finite difference scheme to Δw and obtain the discrete formula for the first equation of (2.6) in D_1 as follows:

$$\frac{(w_1^n)_{i+1,j} + (w_1^n)_{i-1,j} + (w_1^n)_{i,j+1} + (w_1^n)_{i,j-1} - 4(w_1^n)_{i,j}}{h_1^2} = 1 \tag{3.1}$$

where D_1 is divided into a rectangular mesh with mesh size $\Delta x = \Delta y = h_1$, and where i , j are the row and column mesh point numbers, respectively. The boundary conditions in (2.6) can be discretized by a forward finite difference scheme.

Meanwhile, we can discretize (2.7) in D_2 in a similar way and obtain

$$(w_2^n)_{i,j}\left(\frac{(w_2^n)_{i+1,j} + (w_2^n)_{i-1,j} + (w_2^n)_{i,j+1} + (w_2^n)_{i,j-1} - 4(w_2^n)_{i,j}}{h_2^2} - 1\right) = 0 \tag{3.2}$$

where D_2 is divided into a rectangular mesh with mesh size $\Delta x = \Delta y = h_2$.

Since our focus is to determine the location of the free boundary in D_2 more precisely, we construct a fine grid in D_2 and at the same time a coarse grid in D_1 to reduce the computation load there. Therefore, we assume $h_2 = \frac{1}{2}h_1$ throughout our computation, i.e., the grid size of D_2 is only half of that of D_1 . Because of the different grid sizes in D_1 and D_2 , the data transfer equations (2.8) between g_1^n and g_2^n cannot be discretized naturally. In order to discretize (2.8), we have to approximate g_1^n with its neighbouring g_2^n values on Γ , and vice versa, as shown in Figure 3. From Figure 3, we notice that $w_{1,j}$ depends on $w_{2,2j}$. Meanwhile, $w_{2,2j}$ is affected by $w_{1,j}$ and $w_{2,2j+1}$ is affected by both $w_{1,j}$ and $w_{1,j+1}$. Therefore, the reasonable data transfer on Γ will be $w_{1,j} = w_{2,2j}$ from D_2 to D_1 and $w_{2,2j} = w_{1,j}$ and $w_{2,2j+1} = \frac{1}{2}(w_{1,j} + w_{1,j+1})$ from D_1 to D_2 . g_1 and g_2 can be taken care of similarly. Then, (2.8) can be discretized as follows:

$$\begin{array}{ll} (g_1^{n+1})_j & = 2(w_2^n)_{2j} - (g_2^n)_{2j} \\ (g_2^{n+1})_{2j} & = 2(w_1^n)_j - (g_1^n)_j \\ (g_2^{n+1})_{2j+1} & = \frac{1}{2}\{[2(w_1^n)_j - (g_1^n)_j] + [2(w_1^n)_{j+1} - (g_1^n)_{j+1}]\} \end{array} \tag{3.3}$$

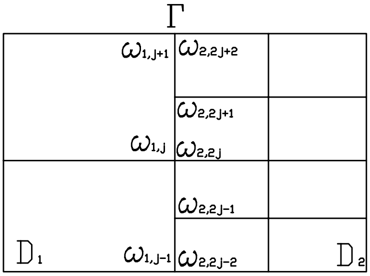


Fig. 3. The nonmatching grids.

The computation is run as follows: we compute w_1^n and w_2^n using (3.1) and (3.2). We then update g_1^n and g_2^n on Γ using (3.3) and repeat the scan through D_1 and D_2 to solve w_1^n and w_2^n . The iteration will stop if the convergence criterion is met.

During the computation, finite difference SOR (Successive over-relaxation) is utilized in D_1 , while in D_2 which contains the free boundary, finite difference SOR (Successive over-relaxation) with a projection is used to make sure the values of w_2 at each point are always non-negative.

Therefore, when applying the SOR in D_1 , (3.1) becomes:

$$\begin{aligned}(w_1^{(n+\frac{1}{2})})_{i,j} &= (\frac{1}{4}((w_1^n)_{i+1,j} + (w_1^n)_{i-1,j} + (w_1^n)_{i,j+1} + (w_1^n)_{i,j-1} - h_1^2) \\ (w_1^{(n+1)})_{i,j} &= (w_1^n)_{i,j} + \overline{\beta}((w_1^{(n+\frac{1}{2})})_{i,j} - (w_1^n)_{i,j})\end{aligned} \tag{3.4}$$

where $\overline{\beta}$ is the relaxation parameter.

Similarly, when applying the SOR with projection in D_2 , (3.2) becomes

$$\begin{aligned}(w_2^{(n+\frac{1}{2})})_{i,j} &= \frac{1}{4}((w_2^n)_{i+1,j} + (w_2^n)_{i-1,j} + (w_2^n)_{i,j+1} + (w_2^n)_{i,j-1} - h_2^2) \\ (w_2^{(n+1)})_{i,j} &= \max(0, (w_2^n)_{i,j} + \overline{\beta}((w_2^{(n+\frac{1}{2})})_{i,j} - (w_2^n)_{i,j}))\end{aligned} \tag{3.5}$$

The flow rate q through the flow field is also unknown *a priori*. Therefore, in addition to the inner iteration to solve for w with a given q , there is also an outer iteration on q to determine the flow rate. The compatibility condition for the outer iteration (see [7]) is $f(q) = (w_2(x_F, y_F - \Delta y)) - \dfrac{\Delta y^2}{2} = 0$. In fact, we can set q_0 and q_1 to arbitrary values. Then we use the secant method to determine q_2 based on q_0 and q_1 from (3.6) for the third outer loop, and so on until we reach some q_n whose $|f(q_n)| < \varepsilon$.

$$q_2 = q_1 - \frac{q_2 - q_1}{f(q_2) - f(q_1)} f(q_1). \tag{3.6}$$

The example uses the following data: $\alpha(x) = x$ where $0 < x < x_F$, $y_F = 30\text{ft}$, $x_F = 30\text{ft}$, $x_B = 60\text{ft}$, $h_1 = 0.5\text{ft}$, $h_2 = 0.25\text{ft}$, $\overline{\beta} = 1.25$, and $\varepsilon = 0.005$.

Figure 4 shows the free boundary obtained by the new DD algorithm. It exactly matches the numerical results from [1]. However, the combination of the new non-overlapping domain decomposition method and the non-matching grid technique generates a better performance than that of [1]. Table 1 shows the required number of iterations for our current algorithm and the algorithm from [1]. We can see that the performance has been improved considerably over [1].

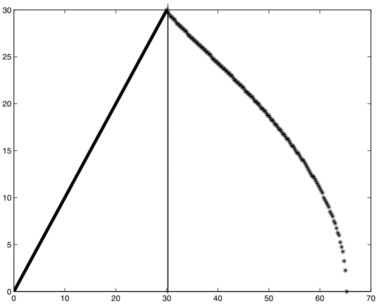


Fig. 4. Free boundary in D_2 .

Outer iteration	Current Algorithm	Algorithm of [1]
1	1867	2374
2	1024	1604
3	1651	2088
4	618	996
5	109	376
6	31	156
7	2	18

Table 1. Comparison of required number of inner iterations between the two algorithms.

4 Conclusion and future directions

In this paper, we studied a free boundary seepage problem of flow through a porous dam with a toe drain. The characteristic of this problem is that the free boundary is unknown in advance. However, we can determine that the free boundary is located in one of the subdomains if we can properly split the domain into two or more subdomains. Then, we can apply the traditional non-overlapping DD method to this problem. Meanwhile, the non-matching grid discretization is utilized on these subdomains in order to obtain higher resolution of the free boundary in the fine-grid subdomain while maintaining computational efficiency in the coarse-grid subdomain.

The promising numerical results motivate us to establish the convergence analysis and error estimates between the numerical solution based on a combination of non-overlapping DD method and non-matching grid discretization and the true solution of the original problem for the general free boundary problem. We will investigate this theoretical issue in the future.

References

1. J. C. Bruch, Jr., *Multi-splitting and domain decomposition techniques applied to free surface flow through porous media*, in Proceedings of the First International Conference on Computational Modeling of Free and Moving Boundary Problems, 1991, pp. 3–20.
2. Q. Deng, *An analysis for a nonoverlapping domain decomposition iterative procedure*, SIAM J. Sci. Comput., 18 (1997), pp. 1517–1525.
3. K. H. Hoffmann and J. Zou, *Parallel algorithms of Schwarz variant for variational inequalities*, Numer. Funct. Anal. Optim., (1992), pp. 449–462.
4. B. Jiang, J. C. Bruch, Jr., and J. M. Sloss, *A non-overlapping domain decomposition method for variational inequalities derived from free boundary problems*, Numerical Methods Partial Differential Eq., (2006).
5. 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 , held in Houston, Texas, March 20-22, 1989, T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, eds., Philadelphia, PA, 1990, SIAM, pp. 202–223.
6. L. D. Marini and A. Quarteroni, *A relaxation procedure for domain decomposition methods using finite elements.*, Numer. Math, 55 (1989), pp. 575–598.
7. J. M. Sloss and J. C. Bruch, Jr., *Free surface seepage problems*, J. Eng. Mech. Div. Proc., 104 (1978), pp. 1099–1111.
8. X.-C. Tai, *Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities*, Numer. Math., 93 (2003), pp. 755–786.
9. O. B. Widlund, *Some Schwarz methods for symmetric and nonsymmetric elliptic problems*, in Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations, D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs, and R. G. Voigt, eds., Philadelphia, PA, 1992, SIAM, pp. 19–36.
10. J. Xu and J. Zou, *Some nonoverlapping domain decomposition methods*, SIAM Review, 40 (1998), pp. 857–914.

Stationary Incompressible Viscous Flow Analysis by a Domain Decomposition Method

Hiroshi Kanayama [1] , Diasuke Tagami [2] , and Masatsugu Chiba [3]

[1] Faculty of Engineering, Kyushu University, Kyuhu, Japan.
kanayama@mech.kyushu-u.ac.jp
[2] Faculty of Engineering, Kyushu University, Kyuhu, Japan.
tagami@mech.kyushu-u.ac.jp
[3] School of Engineering, Kyushu University, Kyuhu, Japan. (former student)

Requirements to compute stationary flow patterns are often encountered. With progress of computer environments and increasing demand of precise analyses, the number of degrees of freedom (DOF) of such a computation has become larger. However, as far as we know, computational codes are rare, which are efficient for large scale, stationary, and nonlinear flow problems. Therefore, we have developed ADVENTURE_sFlow [3], which is one of modules included in the ADVENTURE project [1].

ADVENTURE_sFlow uses the Newton method as the nonlinear iteration, and to compute the problem at each step of the nonlinear iteration a stabilized finite element method is introduced. Moreover, to reduce the computational costs, an iterative domain decomposition method is applied to stabilized finite element approximations of stationary Navier–Stokes equations, for which Generalized Product-type methods based on Bi-CG (GPBiCG) [6] is used as the iterative solver of the reduced linear system in each step of the nonlinear iteration. A parallel computing method using the Hierarchical Domain Decomposition Method (HDDM) is also introduced.

Numerical results show that ADVENTURE_sFLow can analyze a stationary flow problem with 10 million DOF.

1 Formulation

Let Ω be a three-dimensional bounded domain with the Lipschitz continuous boundary Γ . We consider the stationary incompressible Navier–Stokes equations:

$$\begin{cases} -\dfrac{1}{\rho}\nabla\cdot\,\sigma(u,p) + (u\cdot\nabla)\,u = \dfrac{1}{\rho}f & \text{in } \Omega, \quad (1a) \\ \nabla\cdot\,u = 0 & \text{in } \Omega, \quad (1b) \\ u = g & \text{on } \Gamma, \quad (1c) \end{cases}$$

where $u = (u_1, u_2, u_3)^T$ is the velocity $[\mathrm{m/s}]$, p is the pressure $[\mathrm{N/m^2}]$, ρ is the density $[\mathrm{kg/m^3}]$, $f = (f_1, f_2, f_3)^T$ is the body force $[\mathrm{N/m^3}]$, $g = (g_1, g_2, g_3)^T$ is the boundary velocity $[\mathrm{m/s}]$, and $\sigma(u,p)$ is the stress tensor $[\mathrm{N/m^2}]$ defined by

$$\sigma_{ij}(u,p) \equiv -p\delta_{ij} + 2\mu D_{ij}(u), \quad D_{ij}(u) \equiv \frac{1}{2}\Big(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}\Big), \qquad i,j = 1,2,3,$$

with the Kronecker delta δ_{ij} and the viscosity $\mu\,[\mathrm{kg/(ms)}]$.

By application of the Newton method to (1) as the nonlinear iteration method, the k th step linearized equations become the following: find (u^k, p^k) such that

$$\begin{cases} -\dfrac{1}{\rho}\nabla\cdot\sigma(u^k,p^k) + \Big(u^{k-1}\cdot\nabla\Big)\,u^k + \Big(u^k\cdot\nabla\Big)\,u^{k-1} \\ \qquad\qquad = \dfrac{1}{\rho}\,f + \Big(u^{k-1}\cdot\nabla\Big)\,u^{k-1} & \text{in } \Omega, & \text{(2a)} \\ \nabla\cdot u^k = 0 & \text{in } \Omega, & \text{(2b)} \\ u^k = g & \text{on } \Gamma. & \text{(2c)} \end{cases}$$

To avoid some intricate notations, we rewrite the linearized Navier–Stokes equations as follows: find (u,p) such that

$$\begin{cases} -\dfrac{1}{\rho}\nabla\cdot\sigma(u,p) + (w\cdot\nabla)\,u + (u\cdot\nabla)\,w = \widetilde{f} & \text{in } \Omega, & \text{(3a)} \\ \nabla\cdot u = 0 & \text{in } \Omega, & \text{(3b)} \\ u = g & \text{on } \Gamma, & \text{(3c)} \end{cases}$$

where w is a given velocity $[\mathrm{m/s}]$. Obviously, the equations (3) yield (2) by substituting

$$u^{k-1}, \quad u^k, \quad p^k, \quad \text{and} \quad \frac{1}{\rho}f + \big(u^{k-1}\cdot\nabla\big)u^{k-1}$$

for w, u, p, and $\widetilde{f}$, respectively.

Let $\mathscr{T}_h$ be a decomposition of Ω consisting of a union of tetrahedra, and let K be a tetrahedron in $\mathscr{T}_h$. Let u_h and p_h be the velocity and the pressure approximated by $P1/P1$ elements. As in [3], a stabilized finite element method is introduced to (3) as follows: find (u_h , p_h) satisfying (1c) such that

$$\begin{aligned} a_0(u_h,v_h) &+ a_1(w_h,u_h,v_h) + a_1(u_h,w_h,v_h) + b(v_h,p_h) + b(u_h,q_h) \\ &+ \sum_{K\in\mathscr{T}_h}\Big\{\tau_K\Big(\big(w_h\cdot\nabla\big)u_h + \big(u_h\cdot\nabla\big)w_h + \frac{1}{\rho}\nabla p_h, \\ &\qquad \big(w_h\cdot\nabla\big)v_h + \big(v_h\cdot\nabla\big)w_h - \frac{1}{\rho}\nabla q_h\Big)_K + \delta_K(\nabla\cdot u_h, \nabla\cdot v_h)_K\Big\} \\ &= (\widetilde{f},v_h) + \sum_{K\in\mathscr{T}_h}\tau_K\Big(\widetilde{f}, \big(w_h\cdot\nabla\big)v_h + \big(v_h\cdot\nabla\big)w_h - \frac{1}{\rho}\nabla q_h\Big)_K, \end{aligned} \tag{4}$$

where

$$a_0(u,v) \equiv \frac{2\mu}{\rho}\int_\Omega D(u):D(v)\,dx, \qquad a_1(w,u,v) \equiv \int_\Omega \big[(w\cdot\nabla)u\big]v\,dx,$$

$$b(v,q) \equiv -\frac{1}{\rho}\int_\Omega q\,\nabla\cdot v\,dx, \qquad (f,v) \equiv \int_\Omega fv\,dx, \qquad (f,v)_K \equiv \int_K fv\,dx,$$

v_h and q_h are test functions satisfying $v_h = 0$ on Γ, w_h is the convection velocity approximated by $P1$ elements, and the notation " : " denotes the tensor product. The stabilized parameters τ_K and δ_K are defined by

$$\tau_K \equiv \min\left\{ \frac{h_K}{2\,\|w\|_\infty}, \frac{\rho\, h_K^2}{24\mu} \right\}, \quad \delta_K \equiv \min\left\{ \frac{\lambda \rho h_K^2 \|w\|_\infty^2}{12\mu}, \lambda h_K \|w\|_\infty \right\},$$

where λ denotes a positive constant, $\|w\|_\infty$ denotes the maximum norm of w in K, h_K denotes the diameter of K.

Let $\boldsymbol{K}\ \boldsymbol{x} = \boldsymbol{f}$ be the finite element system derived from (4), where $\boldsymbol{K}$ denotes the regular, asymmetric coefficient matrix corresponding to (4), $\boldsymbol{x}$ the vector corresponding to the velocity and the pressure, $\boldsymbol{f}$ the vector corresponding to the body force and the boundary velocity. Let Ω be divided into subdomains. Let $\boldsymbol{x_i}$, $\boldsymbol{x_b}$, and $\boldsymbol{x_t}$ be vectors corresponding to DOF in the interior of Ω, on the interface between subdomains, and on Γ, where $\boldsymbol{x_t}$ is a given vector. Then, the system $\boldsymbol{K}\ \boldsymbol{x} = \boldsymbol{f}$ can be rewritten as follows:

$$\begin{bmatrix} \boldsymbol{K_{ii}} & \boldsymbol{K_{ib}} & \boldsymbol{K_{it}} \\ \boldsymbol{K_{bi}} & \boldsymbol{K_{bb}} & \boldsymbol{K_{bt}} \\ \boldsymbol{0} & \boldsymbol{0} & \boldsymbol{E} \end{bmatrix} \begin{Bmatrix} \boldsymbol{x_i} \\ \boldsymbol{x_b} \\ \boldsymbol{x_t} \end{Bmatrix} = \begin{Bmatrix} \boldsymbol{f_i} \\ \boldsymbol{f_b} \\ \boldsymbol{f_t} \end{Bmatrix}, \tag{5}$$

where $\boldsymbol{E}$ is an identity matrix. Eliminating $\boldsymbol{x_i}$ from (5), we get the linear system on the interface:

$$\boldsymbol{S}\boldsymbol{x_b} = \boldsymbol{\chi}, \tag{6}$$

where

$$\begin{aligned} \boldsymbol{S} &\equiv \boldsymbol{K_{bb}} - \boldsymbol{K_{bi}}\boldsymbol{K_{ii}}^{-1}\boldsymbol{K_{ib}}, \\ \boldsymbol{\chi} &\equiv \boldsymbol{f_b} - \boldsymbol{K_{bi}}\boldsymbol{K_{ii}}^{-1}\boldsymbol{f_i} - (\boldsymbol{K_{bt}} - \boldsymbol{K_{bi}}\boldsymbol{K_{ii}}^{-1}\boldsymbol{K_{it}})\boldsymbol{x_t}. \end{aligned}$$

GPBiCG is apllied to (6), and $\boldsymbol{x_b}$ is obtained. In the implementation, the matrix $\boldsymbol{S}$ is not constructed explicitly. The products of matrices and vectors appearing in GPBiCG can be replaced by solving the Navier–Stokes equations in each subdomain, which implies that the method is fit for parallel computing; see, for example, [2]. The application of the skyline method to a problem in each subdomain yields $\boldsymbol{x_i}$ from $\boldsymbol{x_b}$. The solution in the whole domain at the nth step of the nonlinear iteration is then obtained.

In the actual parallel computing, we adopt HDDM [5] for data and processor management to have the workload balanced among processors. It has already been shown that HDDM is effective for a structural problem where the number of DOF is 100 million [4].

2 Numerical examples

A model of a station is considered as a numerical example; see Fig. 1. The station has one platform on the lower floor, one ticket gate on the upper floor, and three exits from the upper floor to the ground. Two trains are approaching along the red arrows in Fig. 1 with speeds of $1\,[\mathrm{m/s}]$; fixed boundary conditions are imposed on the wall boundaries, and the air flows out from the other sides of the platform and the exits with the stress-free conditions. The body force is set to be 0. The kinematic viscosity μ/ρ is set to be 1.0×10^{-1} [m^2/s].

As in Section 1, Ω is divided into a union of tetrahedra, and the flow field is approximated by $P1/P1$ elements: the number of elements and DOF are $18,873,133$ and $12,943,664$, respectively. The number of subdomains is set to $300,000$. Throughout this section, λ is set to be 1.0.

As in Section 1, the Newton method is used for the nonlinear iteration. The initial value of the nonlinear iteration is the finite element solution of the corresponding Stokes problem. The nonlinear iteration is stopped when the relative rate of changes $\| \boldsymbol{x}^{n+1} - \boldsymbol{x}^{n} \|_\infty / \| \boldsymbol{x}^{n+1} \|_\infty$ becomes smaller than 1.0×10^{-4}, where $\boldsymbol{x}^{n}$ denotes the solution vector at the n th step, and $\| . \|_\infty$ is the maximum norm.

In the Stokes equation to obtain the initial condition of the nonlinear iteration, and in each step of the nonlinear iteration, the resulting linear systems on the interface are solved by GPBiCG with a simplified diagonal scaling preconditioner. The initial vector of the GPBiCG iteration is taken to be zero vector in case of the Stokes equation to obtain the initial condition of the nonlinear iteration, and is taken from the solution vector at the previous step at each step of the nonlinear iteration. The GPBiCG iteration is stopped when the relative residual norm $\| \boldsymbol{\chi} - \boldsymbol{S}\ \boldsymbol{x_b} \|_2 / \| \boldsymbol{\chi} \|_2$ becomes smaller than 1.0×10^{-5}, where $\| . \|_2$ denotes the Euclidean norm. Computation of the model was performed on the Alpha21264 system with 30 CPU at the Computing and Communications Center, Kyushu University. It took about 100 hours to compute.

Fig. 2 shows the residual norm versus the number of GPBiCG iterations at each step of the nonlinear iteration. As the iteration progresses, the convergence of GPBiCG becomes faster. Fig. 3 shows the relative rate of change versus the number of nonlinear iterations. The nonlinear iteration by the Newton method works well. Fig. 4 shows the streamlines in the station. In both cases, the flow comes into the station along the approaches of the trains, and goes out from the other sides of the platform and from the exits.

At the end of this section, we consider the difficulty of computations in cases of high Reynolds numbers and large scale problems. Table 1 shows the computational data on the mesh size and the numbers of DOF. Table 2 shows CPU time [min] for some Reynolds numbers and meshes. In Cases I and II, the problem can be solved for six Reynolds numbers. However, as the scale increases, the problem cannot be solved for higher Reynolds numbers. Finally, in Case VI, the problem can be solved for only $Re = 50$.

3 Conclusion

To analyze the stationary Navier–Stokes equations, ADVENTURE_sFlow has been developed, which is one of the modules produced in the ADVENTURE project [1]. The Newton method has been introduced as the nonlinear iteration, and the stabilized finite element method as the approximation of the linearized equations in every step of the nonlinear iteration. Moreover, for parallel computations, an iterative domain decomposition method and HDDM have been introduced, which are based on GPBiCG.

A station model with about 10 million DOF has been analyzed.

We are going to analyze problems in cases of yet higher Reynolds numbers or coupled problems in the future.

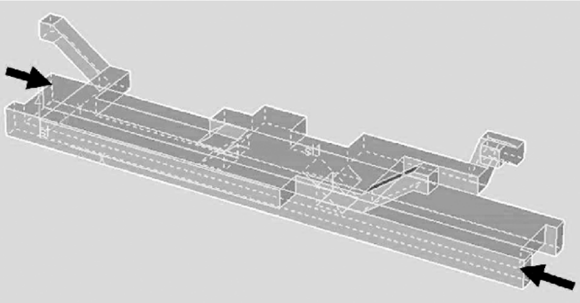


Fig. 1. A station model.

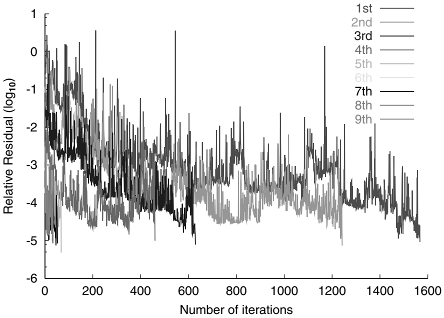


Fig. 2. Relative residuals of GPBiCG at each step of the nonlinear iteration.

References

1. *ADVENTURE project home page.* http://adventure.q.t.u-tokyo.ac.jp/.

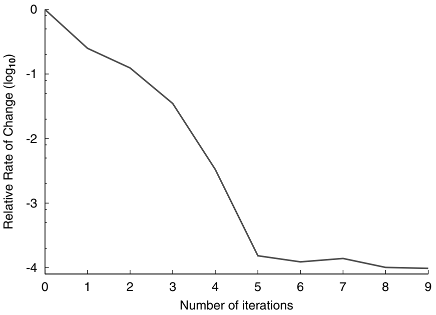


Fig. 3. Relative rates of change in the Newton method.

Table 1. The maximum diameter of mesh and the numbers of DOF.

Case	I	II	III	IV	V	VI
Diameter [m]	1.60	0.90	0.80	0.71	0.59	0.50
DOF [$\times 10^5$]	0.5	2	3	4	7	10

DOF: in round numbers

Table 2. The number of iterations in case of some Reynolds numbers and meshes

Re	I	II	III	IV	V	VI
50	1.67	8.80	15.17	23.07	23.52	48.1
245	1.83	31.60	66.27	120.9	350.5	—
490	1.83	44.45	130.0	343.1	—	—
735	2.00	59.20	152.4	—	—	—
980	2.12	57.77	396.5	—	—	—
1225	2.25	63.91	—	—	—	—

Unit: [min], —: Divergence

2. R. Glowinski, Q. V. Dinh, and J. Pèriaux, *Domain decompostion methods for nonlinear problems in fluid dynamics*, Comp. Meth. Appl. Mech. Engrg, 40 (1983), pp. 27–109.

3. H. Kanayama, D. Tagami, T. Araki, and H. Kume, *A stabilization technique*

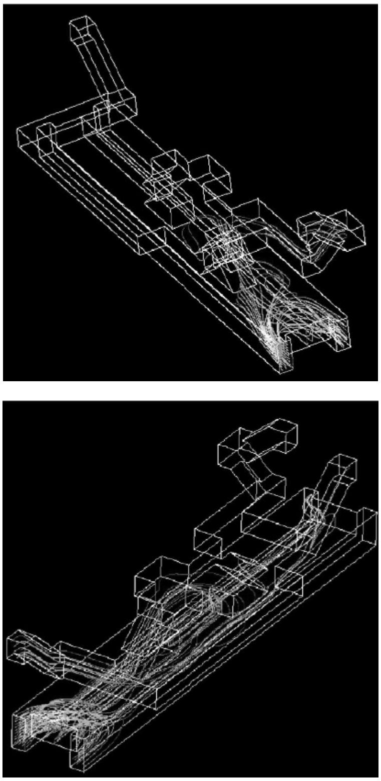


Fig. 4. The streamlines of the station model.

for steady flow problems, Int. J. Comput. Fluid Dyn., 18 (2004), pp. 297–301.

4. R. Shioya and G. Yagawa, *Iterative domain decomposition FEM with preconditioning technique for large scale problem*, in ECM'99, Progress in Experimental and Computational Mechanics in Engineering and Material Behaviour, 1999, pp. 255–260.
5. G. Yagawa and R. Shioya, *Parallel finite elements on a massively parallel computer with domain decomposition*, Comput. Syst. Eng., 4 (1993), pp. 495–503.
6. S.-L. Zhang, *GPBi-CG: Generalized product-type methods based on Bi-CG for solving nonsymmetric linear systems*, SIAM J. Sci. Comput., 18 (1997), pp. 537–551.

New Streamfunction Approach for Magnetohydrodynamics

Kab Seok Kang

Brookhaven National Laboratory, Computational Science Center, Building 463,
Room 255, Upton, NY 11973, USA. kskang@bnl.gov

Summary. We apply the finite element method to two-dimensional, incompressible MHD, using a streamfunction approach to enforce the divergence-free conditions on the magnetic and velocity fields. This problem was considered by Strauss and Longcope [1]. In this paper, we solve the problems with magnetic and velocity fields instead of the velocity stream function, magnetic flux, and their derivatives. Considering the multiscale nature of the tilt instability, we study the effect of domain resolution in the tilt instability problem. We use a finite element discretization on unstructured meshes and an implicit scheme. We use the PETSc library with index sets for parallelization. To solve the nonlinear MHD problem, we compare two nonlinear Gauss-Seidel type methods and Newton's method with several time step sizes. We use GMRES in PETSc with multigrid preconditioning to solve the linear subproblems within the nonlinear solvers. We also study the scalability of this program on a cluster.

1 MHD and streamfunction formulation

Magnetohydrodynamics (MHD) is the fluid dynamics of conducting fluid or plasma, coupled with Maxwell's equations. The fluid motion induces currents, which produce Lorentz body forces on the fluid. Ampere's law relates the currents to the magnetic field. The MHD approximation is that the electric field vanishes in the moving fluid frame, except for possible resistive effects. In this study, we consider finite element methods on an unstructured mesh for two-dimensional, incompressible MHD, using a streamfunction approach to enforce the divergence-free condition on magnetic and velocity fields and an implicit time difference scheme to allow much lager time steps. Strauss and Longcope [1] applied an adaptive finite element method with explicit time difference scheme to this problem.

The incompressible MHD equations are:

$$\begin{aligned} &\frac{\partial}{\partial t}\mathbf{B} = \nabla \times (\mathbf{v} \times \mathbf{B}), && \frac{\partial}{\partial t}\mathbf{v} = -\mathbf{v}\cdot\nabla\mathbf{v} + (\nabla\times\mathbf{B})\times\mathbf{B} + \mu\nabla^2\mathbf{v}, \\ &\nabla\cdot\mathbf{v} = 0, && \nabla\cdot\mathbf{B} = 0, \end{aligned} \tag{1}$$

where $\mathbf{B}$ is the magnetic field, $\mathbf{v}$ is the velocity, and μ is the viscosity. To enforce incompressibility, it is common to introduce stream functions: $\mathbf{v} = \left(\frac{\partial\phi}{\partial y}, -\frac{\partial\phi}{\partial x}\right)$, $\mathbf{B} = \left(\frac{\partial\psi}{\partial y}, -\frac{\partial\psi}{\partial x}\right)$. Formulating for symmetric treatment of the fields, in the sense that the source functions Ω and C are time advanced, and the potentials ϕ and ψ are obtained at each time step by solving Poisson equations, we obtain

$$\begin{aligned} &\frac{\partial}{\partial t}\Omega + [\Omega,\phi] = [C,\psi] + \mu\nabla^2\Omega, \\ &\frac{\partial}{\partial t}C + [C,\phi] = [\Omega,\psi] + 2\left[\frac{\partial\phi}{\partial x}, \frac{\partial\psi}{\partial x}\right] + 2\left[\frac{\partial\phi}{\partial y}, \frac{\partial\psi}{\partial y}\right] \\ &\nabla^2\phi = \Omega, \qquad \nabla^2\psi = C, \end{aligned} \tag{2}$$

where commutator $[a,b] = \frac{\partial a}{\partial x}\frac{\partial b}{\partial y} - \frac{\partial a}{\partial y}\frac{\partial b}{\partial x}$.

To solve (2), we have to compute the partial derivatives of potentials. These partial derivatives can be obtained by solutions of linear problems. To do this, we have to introduce four auxiliary variables. Altogether, this requires the solution of eight equations at each step.

We use the velocity $\mathbf{v}$ and the magnetic field $\mathbf{B}$ to reduce the number of equations to solve (2). To this aim, we put $\mathbf{v} = (v_1, v_2) = \left(\frac{\partial\phi}{\partial y}, -\frac{\partial\phi}{\partial x}\right)$, $\mathbf{B} = (B_1, B_2) = \left(\frac{\partial\psi}{\partial y}, -\frac{\partial\psi}{\partial x}\right)$ in equation (2) and get the following system:

$$\begin{aligned} &\frac{\partial\Omega}{\partial t} + (v_1,v_2)\cdot\nabla\Omega = (B_1,B_2)\cdot\nabla C + \mu\nabla^2\Omega, \\ &\frac{\partial C}{\partial t} + (v_1,v_2)\cdot\nabla C = (B_1,B_2)\cdot\nabla\Omega + 2([v_1,B_1] + [v_2,B_2]), \\ &-\nabla^2 v_1 = -\frac{\partial\Omega}{\partial y}, \quad -\nabla^2 v_2 = \frac{\partial\Omega}{\partial x}, \quad -\nabla^2 B_1 = -\frac{\partial C}{\partial y}, \quad -\nabla^2 B_2 = \frac{\partial C}{\partial x}, \\ &\nabla^2\phi = \Omega, \qquad \nabla^2\psi = C. \end{aligned} \tag{3}$$

In the eight equations in (3), the last two equations for potentials need not be solved to advance the solutions in each time step. If the potentials are needed at a specific time, they are obtained by solving the last two equations in (3). To solve the Poisson's equations for $\mathbf{v}$ and $\mathbf{B}$ in (3), we have to impose boundary conditions which are compatible with boundary conditions of ϕ , ψ , Ω , and C .

2 Finite discretization

To solve (3), we use the first-order backward difference (Euler) derivative scheme leading to an implicit scheme which removes the numerically imposed time-step constraint, allowing much larger time steps. This approach is first order accurate in time and is chosen merely for convenience. Higher order BDF approaches vary only in the weighting of the implicitly discretized and history terms.

Let H^1 denote $H^1(K)$, $H^{1,A}$ denote the subset of $H^1(K)$ whose elements satisfy the boundary condition of A, and $H^{1,A'}$ denote he subspace of $H^1(K)$ whose elements have zero values on the Dirichlet boundary of $A = \Omega, v_1, v_2, B_1, B_2$. Multiplying by test functions and integrating by parts in each equation and using the appropriate boundary conditions, we derive the variational form of (3) as follows: Find $\mathbf{X} = (\Omega, C, v_1, v_2, B_1, B_2) \in H^{1,\Omega} \times H^1 \times H^{1,v_1} \times H^{1,v_2} \times H^{1,B_1} \times H^{1,B_2}$ such that

$$\mathbf{F}^n(\mathbf{X}, \mathbf{Y}) = 0 \tag{4}$$

for all $\mathbf{Y} = (u, w, p_1, p_2, q_1, q_2) \in H^{1,\Omega'} \times H^1 \times H^{1,v_1'} \times H^{1,v_2'} \times H^{1,B_1'} \times H^{1,B_2'}$, where $\mathbf{F}^n = (F_1^n, F_2^n, F_3, F_4, F_5, F_6)^T$,

$$\begin{gathered}
F_1^n(\mathbf{X}, \mathbf{Y}) = M_t^n(\Omega, u) + (\mathbf{v} \cdot \nabla \Omega, u) + \mu a(\Omega, u) - (\mathbf{B} \cdot \nabla C, u), \\
F_2^n(\mathbf{X}, \mathbf{Y}) = M_t^n(C, w) + (\mathbf{v} \cdot \nabla C, w) - (\mathbf{B} \cdot \nabla \Omega^n, w) - 2P(\mathbf{v}, \mathbf{B}, w), \\
F_3(\mathbf{X}, \mathbf{Y}) = a(v_1, p_1) + \left(\frac{\partial \Omega}{\partial y}, p_1\right), \quad F_4(\mathbf{X}, \mathbf{Y}) = a(v_2, p_2) - \left(\frac{\partial \Omega}{\partial x}, p_2\right), \\
F_5(\mathbf{X}, \mathbf{Y}) = a(B_1, q_1) + \left(\frac{\partial C}{\partial y}, q_1\right), \quad F_6(\mathbf{X}, \mathbf{Y}) = a(B_2, q_2) - \left(\frac{\partial C}{\partial x}, q_2\right), \\
(u, w) = \int_K uw dx, \quad M_t^n(u, w) = \frac{1}{\Delta t}(u - u^n, w), \quad a(u, w) = \int_K \nabla u \cdot \nabla w dx, \\
P((u_1, u_2), (v_1, v_2), w) = \int_K [u_1, v_1] w dx + \int_K [u_2, v_2] w dx.
\end{gathered}$$

Let $\mathcal{K}_h$ be a given triangulation of domain K with the maximum diameter h of the element triangles. Let V_h be the continuous piecewise linear finite element space. Let V_h^A, $A = \Omega, v_1, v_2, B_1, B_2$, be the subsets of V_h which satisfy the boundary conditions of A on every boundary point of $\mathcal{K}_h$ and $V_h^{A'}$ be subspaces of V_h and $H^{1,A'}$. Then we can write the discretized MHD problems as follows: For each discrete time step n, find the solution $\mathbf{X}_h^n = (\Omega_h^n, C_h^n, v_{1,h}^n, v_{2,h}^n, B_{1,h}^n, B_{2,h}^n) \in V_h^\Omega \times V_h \times V_h^{v_1} \times V_h^{v_2} \times V_h^{B_1} \times V_h^{B_2}$ that satisfies

$$F^n(\mathbf{X}_h^n, \mathbf{Y}_h) = 0 \tag{5}$$

for all $\mathbf{Y}_h \in V_h^{\Omega'} \times V_h \times V_h^{v_1'} \times V_h^{v_2'} \times V_h^{B_1'} \times V_h^{B_2'}$.

3 Nonlinear and linear solvers

System (5) is a nonlinear problem in the six variables consisting of two time dependent equations and four Poisson equations. However, if we consider the equations separately, each equation is linear with respect to one variable. Specifically, the last four equations are linear equations and Poisson problems. From the above observations, we naturally consider a nonlinear Gauss-Seidel iterative solvers (GS1) which solve linear each equation on one variable in (5) in consecutive order with recent approximate solutions.

Poisson solvers are well developed and the first two equations are time dependent problems. From this observation, we consider another nonlinear Gauss-Seidel iterative solvers (GS2) that solve first two equations of (5) as one equation and then solve four Poisson equations.

Nonlinear Gauss-Seidel iterative method doesn't guarantee convergence, but converges well in many cases, especially for small time step sizes in time dependent problems.

Next, we consider the Newton linearization method. Newton's method has, asymptotically, second-order convergence for nonlinear problems and greater scalability with respect to mesh refinement than the nonlinear Gauss-Seidel method, but requires computation of the Jacobian of nonlinear problem which can be complicated.

In all three nonlinear solvers, we need to solve linear problems. Krylov iterative techniques are well suited because they can be preconditioned for efficiency. Among the various Krylov methods, GMRES (Generalized Minimal RESiduals) is selected because it guarantees convergence with nonsymmetric, indefinite systems. However, GMRES can be memory intensive (storage increases linearly with the number of GMRES iterations per Jacobian solve) and expensive (computational complexity of GMRES increases with the square of the number of GMRES iterations per Jacobian solve). Restarted GMRES can in principle deal with these limitations; however, it lacks a theory of convergence, and stalling is frequently observed in real applications.

Preconditioning consists of operating on the system matrix J_k where

$$J_k \delta x_k = -F(x_k) \tag{6}$$

with an operator P_k^{-1} (preconditioner) such that $J_k P_k^{-1}$ (right preconditioning) or $P_k^{-1} J_k$ (left preconditioning) is well-conditioned. In this study, we use left preconditioning. Consider the equivalent linear system:

$$P_k^{-1} J_k \delta x_k = -P_k^{-1} F(x_k). \tag{7}$$

The system in equation (7) is equivalent to the original system (6) for any nonsingular operator P_k^{-1} . Thus, the choice of P_k^{-1} does not affect the accuracy of the final solution, but crucially determines the rate of convergence of GMRES, and hence the efficiency of the algorithm.

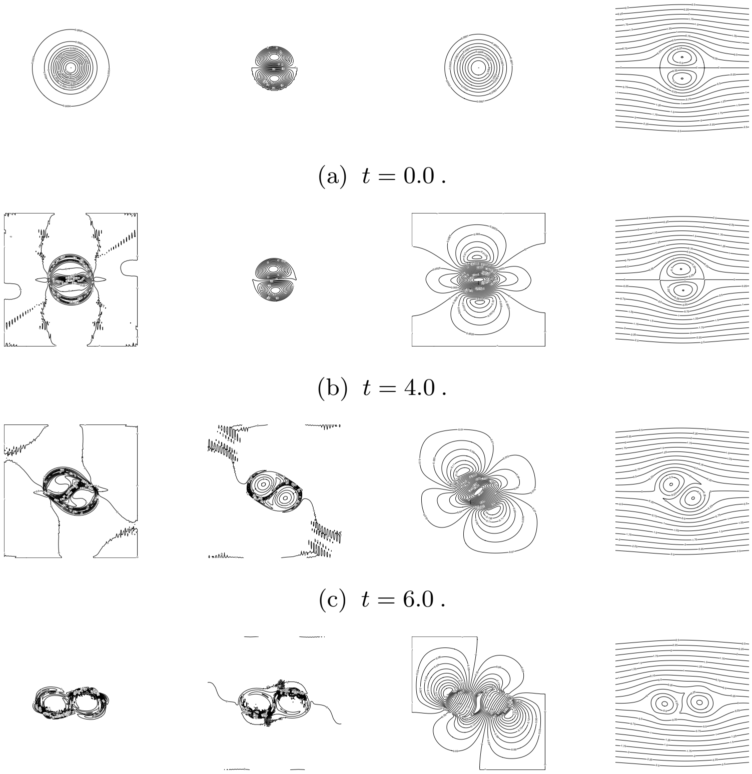
In this study, we use multigrid which is well known as a successful preconditioner, as well as a scalable solver in unaccelerated form, for many problems. We consider the symmetrized diagonal term of Jacobian, i.e.,

$$J_{S,k} = \frac{1}{2}\left(J_{R,k} + J_{R,k}^T\right), \tag{8}$$

where $J_{R,k}$ is a block diagonal matrix. The system $J_{S,k}$ may be less efficient than $J_{R,k}$ as a preconitioner but more numerically stable because it is symmetric. To implement the finite element solver for two-dimensional, incompressible MHD on parallel machines, we use the PETSc library, which is well developed for nonlinear PDE problems and easily implements a multigrid preconditioner with GMRES. We use PETSc's index sets for parallelization of our unstructured finite element discretization.

4 Numerical experiments: Tilt Instability

We consider the initial equilibrium state as $\psi = \begin{cases} [2/kJ_0(k)]J_1(kr)\dfrac{y}{r}, & r < 1, \\ (1/r - r)\dfrac{y}{r}, & r > 1, \end{cases}$ where J_n is the Bessel function of order n, k is any constant that satisfies $J_1(k) = 0$, and $r = \sqrt{x^2 + y^2}$.



(a) $t = 0.0$.

(b) $t = 4.0$.

(c) $t = 6.0$.

(d) $t = 7.0$.

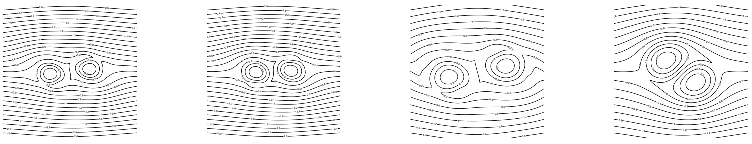
Fig. 1. Contours of Ω, C, ϕ, and ψ at time $t = 0.0, 4.0, 6.0, 7.0$.

In our numerical experiments, we solve on the finite square domain $K = [-R, R] \times [-R, R]$ with the initial condition of the tilt instability problem from the above initial equilibrium and perturbation of ϕ (originating from perturbations of velocity) such that

$$\begin{aligned}
&\Omega(0) = 0.0, && C(0) = \begin{cases} 19.0272743 J_1(kr)y/r & \text{if } r < 1 \\ 0.0 & \text{if } r > 1 \end{cases}, \\
&\phi(0) = 10^{-3} e^{-(x^2+y^2)}, && \psi(0) = \begin{cases} -1.295961618 J_1(kr)y/r & \text{if } r < 1 \\ -(\frac{1}{r} - r)y/r & \text{if } r > 1, \end{cases}
\end{aligned}$$

where $k = 3.831705970$ and with Dirichlet boundary conditions $\Omega(x,y,t) = 0.0$, $\phi(x,y,t) = 0.0$, and $\psi(x,y,t) = y - \dfrac{y}{x^2+y^2}$ and Neumann boundary condition for C, i.e., $\dfrac{\partial C}{\partial n}(x,y,t) = 0.0$. The initial and boundary condition for velocity $\mathbf{v}$ and magnetic field $\mathbf{B}$ are derived from the initial and boundary condition of Ω, C, ϕ, and ψ. Numerical simulation results are illustrated in Fig. 1.

The tilt instability problem is defined on unbounded domain. To investigate the effect of size of domains, we simulate two methods, one uses ϕ, ψ and their derivatives (denoted Strauss-Longcope, or "SL") and the other uses $\mathbf{v}$ and $\mathbf{B}$ (denoted "K") on the square domains with $R = 2$ and $R = 3$. Numerical results are shown in Fig. 2, the contours of ψ at $t = 6.0$. The average growth rate γ of kinetic energy is shown in Table 1. These numerical simulation results show that the solutions of two formulations are closer when the domain is enlarged, with the previous approach converging from above and new approach converging from below.



(a) $R = 3$ (SL). (b) $R = 3$ (K). (c) $R = 2$ (SL). (d) $R = 2$ (K).

Fig. 2. Contours of ψ at $T = 7.0$.

Table 1. Average growth rate γ of kinetic energy from $t = 0.0$ to $t = 6.0$.

previous, $R = 2$	previous, $R = 3$	new, $R = 2$	new, $R = 3$
2.167	2.152	1.744	2.102

From here, we consider the convergence behaviors of several nonlinear and linear solvers as a function of time step sizes. In Table 2, we report the number of nonlinear iterations of nonlinear solvers according to time step sizes for the fixed starting time and fixed mesh level 5. We choose $t = 0.0$ and $t = 6.0$ as the base times because many simulations have trouble at start up and the magnitudes of the velocity (v_1, v_2) and magnetic field (B_1, B_2) increase with time. These numerical results show that GS2 and Newton method are more nonlinearly robust than GS1.

To investigate another convergence behavior, we report the average number of linear iterations in one time step according to preconditioners in Table 3. Numerical

Table 2. The average number of nonlinear iterations of one time step according to time step sizes dt.

dt	GS1		GS2		NM	
	$t = 0$	$t = 6$	$t = 0$	$t = 6$	$t = 0$	$t = 6$
0.0005	3	4	3	4	2	4
0.001	4	4	3	4	3	4
0.002	5	5	3	3	5	4(5)
0.005	12	8	4	6	3	5
0.01	*	*	4	8	4	7
0.02	*	*	6	13	5	11

results show that the multigrid preconditioner applying on symmetrized reduced system (S; see (8)) is robust at $t = 0.0$ and $t = 6.0$, but multigrid applied to the reduced system (R) is robust only at $t = 0.0$, very similar to the symmetrized case, because the values of velocity are small at $t = 0.0$. These results show that we have to use multigrid preconditioner applied to the symmetrized reduced system to get robust convergence.

Table 3. The average number of linear iterations in one time step according to time step sizes.

dt	GS2(R)		GS2(S)		NM(R)		NM(S)	
	$t = 0$	$t = 6$	$t = 0$	$t = 6$	$t = 0$	$t = 6$	$t = 0$	$t = 6$
0.0005	4.3	4	4.3	4	5	5	5	5
0.001	5.3	4.5	5.3	5	6	5	6	5
0.002	6.6	5	6.6	5.2	7	6.8	7	6.25
0.005	11	7.8	11	8.8	12	*	12	10
0.01	18	*	18	15.2	18.5	*	18.5	17
0.02	28.8	*	28.8	27.3	31.6	*	31.6	33.3

In Table 4, we report the average number of nonlinear and linear iterations from $t = 0.0$ to $t = 0.05$ with $dt = 0.005$ according to the levels. These results show that two numerical method GS2(S) and Newton method (S) have very similar behaviors.

Table 4. Average number of iterations according to the number of level.

Solvers	GS2(S)		NM(S)	
á level	nonlinear	linear	nonlinear	linear
4	4	7.9	3	8
5	3.1	11.2	3	11.7
6	3	16.1	3	16.4
7	3.4	19.1	3.4	20.1

In Table 5, we report the solution times of one time step according to level and number of processors on the cluster machine BGC (the Brookhaven Galaxy Cluster) at BNL. We run the program on the same speed (696 MHz) CPU's. This table shows that Newton's method has a better scalabilty than GS2 though neither scales strongly beyond a certain interval.

Table 5. Average solution time of one time step (linear system) according to level and number of processors at $t = 0.0$ and $dt = 0.005$.

level	# CPU	GS2(S)	NM(S)
4	1	13.7 (2.39)	12.3 (2.02)
	2	13.3 (2.74)	7.76 (1.40)
5	2	42.5 (11.3)	33.5 (6.62)
	4	29.4 (7.79)	21.0 (4.23)
	8	38.6 (11.12)	19.9 (4.89)
6	8	120.5 (35.9)	59.9 (14.4)
	16	64.5 (19.3)	39.0 (9.65)
	32	118.1 (36.95)	61.6 (17.8)
7	32	226.3(61.8)	142.9 (36.2)

5 Conclusions

We study a new streamfunction approach method for two-dimensional, incompressible magnetohydrodynamics with finite element discretization on the tilt instability example. We show that nonlinear Gauss-Seidel (GS2) and Newton's method have similar numerical behaviors and multigrid preconditioning on the symmetrized reduced system provides good linear convergence.

Acknowledgement

The author would like to thank D. E. Keyes of Columbia University for his valuable advice in the preparation of this work. This work was supported by the United States Department of Energy under contract No. DE-AC02-98CH10886.

References

1. H. R. Strauss and D. W. Longcope, *An adaptive finite element method for magnetohydrodynamics*, J. Comput. Phys., 147 (1998), pp. 318–336.

Control Volume Finite Difference On Adaptive Meshes

Sanjay K. Khattri, Gunnar E. Fladmark, and Helge K. Dahle

Department of Mathematics, University Bergen, Norway. sanjay@mi.uib.no

Summary. In this work we present a finite volume discretization of an elliptic boundary value problem on adaptively refined meshes. This problem is important in many practical applications, e.g. porous media flow. We propose an error indicator functional which is used to select elements that should be refined. Two numerical examples are provided to demonstrate the potential of the proposed refinement strategy.

1 Introduction

Finite volume [1] and finite element [2, 3, 9] are widely used methods for discretizing partial differential equations. Behaviour of finite element methods on adaptive meshes is well understood and studied, e.g., [2, 3, 9], whereas finite volume method seems to be less studied. In this paper, we will consider a cell centered finite volume method also known as control volume finite difference method (CVFD) [1, 5]. Finite volume methods are popular for example in the porous media community since they are based on conservation principles and honour the continuity of fluxes. There are different ways of expressing the fluxes through the boundaries of a cell which give rise to different formulations like the two point flux approximation methods (TPFA) and the multi point flux approximation methods (MPFA), [7, 1]. In this work, we will use a TPFA method. Consider the numerical solution of the following elliptic boundary value problem using adaptive meshes:

$$-\nabla \cdot (\boldsymbol{K}\, \nabla p) = f(x,y) \qquad \text{in} \quad \Omega, \tag{1}$$

$$p(x,y) = p^{\mathrm{D}} \qquad \text{on} \quad \partial\Omega. \tag{2}$$

Here, Ω is a polyhedral domain in $\mathbb{R}^2$, the source function f is assumed to be in $L^2(\Omega)$, and $\boldsymbol{K}$ is symmetric and uniformly positive definite tensor which may depend on the spatial coordinate. In porous media flow, the unknown function $p = p(x,y)$ represents the pressure of a single fluid, and $\boldsymbol{K}$ is the permeability of the porous medium Ω. The rest of the paper is organised as follows: In Section 2, a simple criterion for adaptive refinement is proposed, and an algorithm for an adaptive meshing strategy is given. In Section 3, we give two numerical examples. In the first example, the permeability $\boldsymbol{K}$ is constant while the source exhibits a huge variability. In the second example, the medium properties represented by the permeability $\boldsymbol{K}$ are discontinuous. In both cases an analytic solution is known and the error for the discrete solutions on adaptively and uniformly refined meshes can be computed. These errors are then compared for meshes that possess the same degree of freedom (DOF). Finally in Section 4, we provide some concluding remarks.

2 Adaptive Criteria and Adaptive Algorithm

Adaptive refinement are feed-back based discretizations (**Solve** → **Estimate** → **Refine/Coarse**). Thus we need criterion for selecting finite volumes/cells in the mesh for further refinement. Ultimately these methods construct a sequence of meshes that may converge to an optimal mesh (the most accurate solution at a fixed cost or lowest computational effort for a given accuracy). Generally most of the error occurs in areas where the solution exhibits large gradients, varying curvature or high source variability [2, 3, 9]. Based on these heuristics we propose the following error indicator for a cell i in the mesh:

$$\eta_i = \alpha \, \|p_h\|_{L_2(\Omega_i)} + \alpha_G \, \eta_G + \alpha_F \, \eta_F + \alpha_S \, \eta_S. \tag{3}$$

Here, α, α_G, α_F and α_S are weights belonging to the interval $[0, 1]$, and η_G, η_F and η_S are given as follows:

$$\eta_G := \|\nabla p_h\|_{L_2(\Omega_i)}, \tag{4}$$

$$\eta_F := \|(\boldsymbol{K} \, \nabla p_h) \cdot \hat{\mathbf{n}}\|_{L_2(\partial\Omega_i)}, \tag{5}$$

$$\eta_S := \|f\|_{L_2(\Omega_i)}. \tag{6}$$

In these formulas, we will use least square fitting to approximate the gradient, ∇p_h, of the discrete pressure p_h. An error error indicator need not to represent the error very accurately [3], they just need to select the elements for further refinement. An element i in the mesh will be refined if $\dfrac{\eta_i}{\max_j \eta_j} \geq \delta$ $(0 \leq \delta \leq 1)$. Thus, $\delta = 0$ means a uniform refinement and $\delta = 1$ means that the algorithm will refine a single element per iteration. None of these end point values may be optimal. A trade off between uniform refinement and refining a single element at a time is obtained by choosing $\delta = 0.5$. This value has also been suggested in the literature, e.g., [9]. In general the choice of an optimal set of parameters δ, α, α_G, α_F and α_S is a difficult task. In this work, we have chosen these parameters based on experience with the specific problems. Optimal choice of these numbers will be investigated in future research. It should be noted that if α_S and α_F are equal to zero then the indicator (3) is similar to the indicator proposed in [9] for an Adaptive Discontinuous Galerikin Method, whereas if α and α_G is equal to zero then the indicator is similar to the one given in [2, 3] for an adaptive finite element method. The overall algorithm we are using is presented in Algorithm 1. This adaptive algorithm works on the principle of equally distributing the adaptivity index over all cells in the mesh. For a cell centered finite volume method the degrees of freedom (DOF) are equal to the number of cells in the mesh. In Algorithm 1, the refinement is stopped at a fixed maximum DOFs. In general an a posteriori error estimator should be added as a stopping criterion, (cf. [9, 8, 4]).

Algorithm 1: Adaptive Algorithm.

```
Mesh the domain;
while $DOF < DOF_{\max}$ do
    Discretize the PDE over the mesh by the CVFD;
    Solve the discrete system;
    forall elements $j$ in the mesh do
        if $\eta_j / \max_i \eta_i \geq \delta$ then
            Refine the element $j$ in the mesh;
        end
    end
    Form a new mesh;
end
```

3 Numerical Examples

Let p^k denotes the exact solution for the pressure at the center of cell k, and p_h^k denotes the discrete pressure obtained by the finite volume approximation for the same location. Then the discrete error e in the L_2 norm for a mesh can be expressed as:

$$\|e\|_{L_2} := \left(\sum_{\text{cells}} \left[p^k(x) - p_h^k(x) \right]^2 \Omega_k \right)^{1/2} . \tag{7}$$

Here, the summation is to be taken over all the cells/finite volumes in the mesh. The CVFD approximation of the equation (1) subject to the boundary condition (2) using a two point flux approximation (TPFA) leads to symmetric positive definite linear systems. To solve these systems, we are using the ILU preconditioned conjugate gradient (CG) solver with a tolerance of 1×10^{-10}.

3.1 Example 1

Let the domain be $\Omega = (0,1) \times (0,1)$, and the permeability be the identity tensor $\boldsymbol{K} = \boldsymbol{I}$. We enforce the source term $f = f(x,y)$ such that the analytical solution to Equation (1) is given by

$$u(x,y) = 0.0005 \; [x \; (x-1) \; y \; (y-1)\,]^2 \, e^{10\,(x^2+y^2)}. \tag{8}$$

Note that this solution is consistent with the zero Dirichlet boundary condition (2). Furtheremore, taking the Laplacian of (8) shows that the source term exhibits a huge variability inside the domain and even within the cells. For this problem we found that $\delta = 0.5$, $\alpha = 0.0$, $\alpha_G = 0.10$, $\alpha_F = 0.90$ and $\alpha_S = 1.0$ was a good choice of parameters for the indicator functional (3). However, other choices may work even better. Figures 3 reports the outcome of a numerical experiment comparing the discrete solutions on an adaptively refined mesh and a uniform mesh. The degrees of freedom (DOF) associated with the meshes depicted in these figures are approximately the same. However, the L_2 errors in the solutions on adaptive and uniform meshes are 8.91×10^{-4} and 3.7×10^{-3} , respectively. Thus, the error of the solution on the adaptively refined mesh is much smaller compared to the solution on the uniform grid. In Figure 3.1, we have plotted the error versus DOF for solutions on adaptively refined meshes and for uniform meshes. From this plot we get that $\|e\| \sim \text{DOF}^{-p/2}$ with $p \approx 2$ on the adaptive meshes, which is quasi optimal in the sense of [8, 4]. Since the solution is smooth, we expect the advantage of adaptive refinement to be largest for coarser grids, while this advantage should be reduced compared to a uniform refinement for finer grids. This is indeed what can be observed in Figure 3.1.

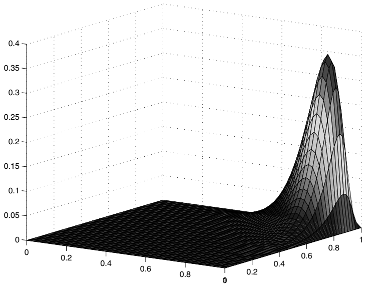


Fig. 1. (Example 3.1) Exact solution.

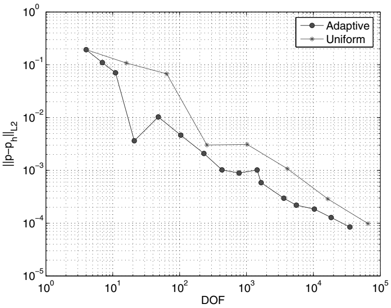


Fig. 2. (Example 3.1) L_2 error vs degrees of freedom for adaptively generated meshes and uniform meshes.

3.2 Example 2

In porous media flow, material properties as given by the permeability is often piecewise constant. The numerical challenges introduced by the discontinuities in the permeability are difficult to handle by standard formulations, see [9, 6, 7, 1, 5]. In this example, we will investigate the behaviour of our refinement strategy for a problem with discontinuous permeability. Let $\Omega = (-1, 1) \times (-1, 1)$. We subdivided Ω into four non overlapping subregions $\Omega_i \quad i = 1 \ldots, 4$ such that $\Omega = \cup_i \Omega_i$ as shown in Figure 4. For each subregion Ω_i we associate a constant permeability $\boldsymbol{K}$, and will assume that

$$\boldsymbol{K_2} = \boldsymbol{K_4} = \boldsymbol{I} \qquad \text{and} \qquad \boldsymbol{K_1} = \boldsymbol{K_3} = R\boldsymbol{I}, \tag{9}$$

where R is a parameter to be determined. An analytic solution can be constructed using the polar representation

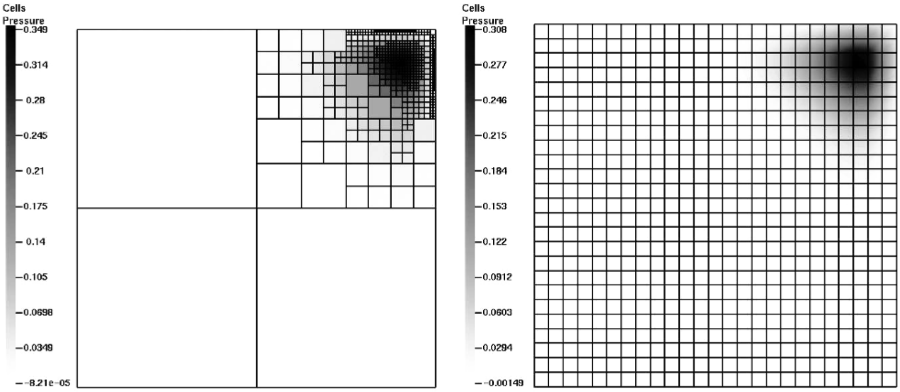


Fig. 3. (Example 3.1) Discrete solution on adaptive and uniform meshes. DOF for the adaptive mesh is 601, and DOF for uniform refinement is 625. L_2 error on adaptive mesh is 8.91×10^{-4} while on uniform mesh it is 3.7×10^{-3} .

$$p(r,\theta) = r^{\gamma}\eta(\theta), \tag{10}$$

see [8, 4]. Let $\eta(\theta)$ be given by

$$\eta(\theta) = \begin{cases} \cos\left[(\pi/2-\sigma)\gamma\right]\,\cos\left[(\theta-\pi/2+\rho)\gamma\right], & \theta \in [0, \frac{\pi}{2}], \\ \cos(\rho\gamma)\,\cos\left[(\theta-\pi+\sigma)\gamma\right], & \theta \in [\frac{\pi}{2}, \pi], \\ \cos(\sigma\gamma)\,\cos\left[(\theta-\pi-\rho)\gamma\right], & \theta \in [\pi, \frac{3\pi}{2}], \\ \cos\left[(\pi/2-\rho)\gamma\right]\,\cos\left[(\theta-3\pi/2-\sigma)\gamma\right], & \theta \in [\frac{3\pi}{2}, 2\pi], \end{cases} \tag{11}$$

and let the numbers R , γ , ρ and σ satisfy the nonlinear relations:

$$\begin{aligned} & R = -\tan\left[(\pi-\sigma)\gamma\right]\,\cot(\rho\gamma), \\ & 1/R = -\tan(\rho\gamma)\,\cot(\sigma\gamma), \\ & R = -\tan(\sigma\gamma)\,\cot\left[(\pi/2-\rho)\gamma\right], \\ & 0 < \gamma < 2, \\ & \max\{0, \pi\gamma-\pi\} < 2\gamma\rho < \min\{\pi\gamma, \pi\}, \\ & \max\{0, \pi-\pi\gamma\} < -2\gamma\rho < \min\{\pi, 2\pi-\pi\gamma\}. \end{aligned} \tag{12}$$

Then it can be shown that (10) satisfies Equation (1) with $\boldsymbol{K}$ given by (9) and $f(x,y) = 0$. Boundary conditions need to be chosen consistently with the form (10). Furthermore, it can be shown that the solution p belongs to the fractional Sobolev space $\mathbf{H}^{1+\xi}(\Omega)$ where $\xi < \gamma$ (cf. [10]). By choosing $\gamma = 0.3$, we can solve the constrained nonlinear relations (12) using Newton's iteration to get $R = 17.3476$, $\sigma = -4.4506$ and $\rho = 0.7853$. We specify the parameters for the indicator functional to be $\delta = 0.6$, $\alpha = 0.0$, $\alpha_G = 0.0$, $\alpha_F = 1.0$, $\alpha_S = 0.0$. In Figure 5, we have plotted the error in the discrete solution against the degrees of freedom for both adaptive and uniform meshes. Again we observe that the convergence on adaptive meshes are much better than for uniform refinement. We also get that $\|e\|_{L_2} \sim \mathrm{DOF}^{-p/2}$ with $p \approx 2.0$ for the solution on adaptive meshes. Because of the regularity of the solution, this convergence is also quasi optimal in the sense of [8, 4]. Finally in Figure 6 we plot the number of CG iterations (without preconditioning) vs. the DOFs for the adaptive and uniformly refined meshes. The plot shows that the uniformly refined meshes require approximately twice as many CG iterations as the adaptive refinement. This suggests that the condition number for the matrix obtained for uniform refinement is four times the condition number for the matrix obtained for adaptive refinement.

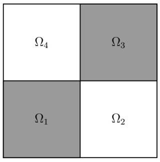


Fig. 4. (Example 3.2) Domain with discontinuous medium properties. The permeability is constant over each sub-domains i.e., $\boldsymbol{K} = \mathbf{K}_i$ in Ω_i .

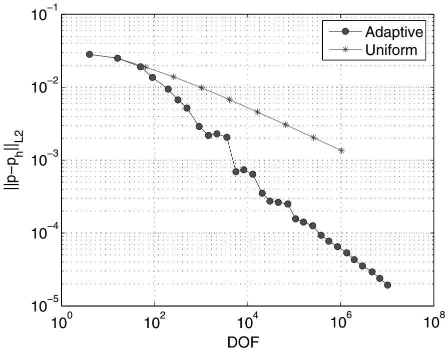


Fig. 5. (Ex. 3.2) Pressure convergence in L_2 norm for adaptive and uniform refinement.

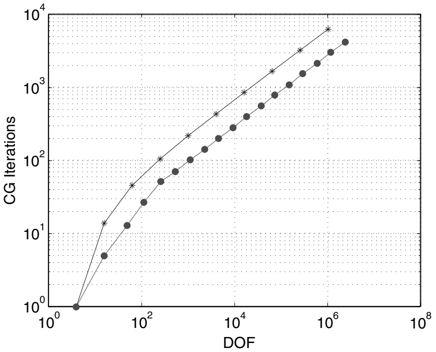


Fig. 6. (Ex. 3.2) Number of CG Iterations (no preconditioner) vs. DOFs. Star is representing uniform refinement while circle is associated with adaptive refinement.

4 Conclusions

In this work we have given a strategy for adaptive refinement in the setting of CVFD discretizations of boundary value problems. The mesh refinement is based on the use of an error indicator functional. We have tested the methods on two test examples. In both cases the solution has a strong local behaviour which is clearly captured by our refinement strategy. We have computed the error in the discrete solution to obtain convergence rates. The numerical experiments suggest that convergence is quasi optimal as the mesh is adaptively refined for both the test examples. Furthermore we have compared CVFD on adaptive and uniform meshes. As expected the solutions obtained for adaptive meshes are significantly more accurate, and the system matrices are better conditioned when we employ adaptive meshes. Even though our preliminary investigations show that the proposed CVFD discretization on adaptive meshes has a great potential, many challenges remain open for further research. Most importantly we need to find better ways of selecting parameters for the indicator functional.

References

1. I. Aavatsmark, *An introduction to multipoint flux approximations for quadrilateral grids*, Comput. Geosci., 6 (2002), pp. 405–432.
2. I. Babuška and A. Miller, *A feedback finite element method with a posteriori error estimation: Part I. The finite element method and some basic properties of the a posteriori error estimators*, Comp. Meth. Appl. Mech. Eng., 61 (1987), pp. 1–40.
3. I. Babuška and W. C. Rheinboldt, *Analysis of optimal finite element meshes in* r^1 , Math. Comp., 33 (1979), pp. 435–463.
4. Z. Chen and S. Dai, *On the efficiency of adaptive finite element methods for elliptic problems with discontinuous coefficients*, SIAM J. Sci. Comput., 24 (2002), pp. 443–462.
5. M. G. Edwards and C. F. Rogers, *Finite volume discretization with imposed flux continuity for the general tensor pressure equation*, Comput. Geosci., 2 (1998), pp. 259–290.
6. G. T. Eigestad and R. A. Klausen, *On the convergence of the multi-point flux approximation O-method: Numerical experiments for discontinuous permeability*, Numer. Methods Partial Differential Equations, 21 (2005), pp. 1079–1098.
7. S. K. Khattri, *Numerical tools for multi-component, multi-phase, reactive transport in porous medium*, PhD thesis, The University of Bergen, Norway, 2006.
8. P. Morin, R. H. Nochetto, and K. G. Siebert, *Data oscillation and convergence of adaptive FEM*, SIAM J. Numer. Anal., 38 (2000), pp. 466–488.
9. B. M. Rivière, *Discontinuous Galerkin Fnite Element Methods for Solving the Miscible Displacement Problem in Porous Media*, PhD thesis, The University of Texas at Austin, 2000.
10. G. Strang and G. J. Fix, *An Analysis of the Finite Element Method*, Prentice-Hall, Englewood Cliffs, N.J., 1973.

Preconditioned Eigensolver LOBPCG in hypre and PETSc

Ilya Lashuk, Merico Argentati, Evgueni Ovtchinnikov, and Andrew Knyazev

Department of Mathematical Sciences, University of Colorado at Denver and
Health Sciences Center, P.O. Box 173364, Campus Box 170, Denver, CO
80217-3364, USA.
{na.ilashuk,na.rargenta,na.eovtchin,na.knyazev}@na-net.ornl.gov

Summary. We present preliminary results of an ongoing project to develop codes of the Locally Optimal Block Preconditioned Conjugate Gradient (LOBPCG) method for symmetric eigenvalue problems for *hypre* and PETSc software packages. *hypre* and PETSc provide high quality domain decomposition and multigrid preconditioning for parallel computers. Our LOBPCG implementation for *hypre* is publicly available in *hypre* 1.8.2b and later releases and in PETSc. We describe the current state of the LOBPCG software for *hypre* and PETSc and demonstrate scalability results on distributed memory parallel clusters using domain decomposition and multigrid preconditioning.

This work is partially supported by the Center for Applied Scientific Computing, Lawrence Livermore National Laboratory and the National Science Foundation DMS 0208773.

1 Introduction

We implement a parallel algorithm, the Locally Optimal Block Preconditioned Conjugate Gradient Method (LOBPCG) [5, 6], for the solution of eigenvalue problems $Ax = \lambda Bx$ for large sparse symmetric matrices A and $B > 0$ on massively parallel computers for the High Performance Preconditioners (*hypre*) [3] and Portable, Extensible Toolkit for Scientific Computation (PETSc) [2] software libraries. Our software package, the Block Locally Optimal Preconditioned Eigenvalue Xolvers (BLOPEX) is available at `http://math.cudenver.edu/~aknyazev/software/BLOPEX/` which contains, in particular, our MATLAB, *hypre* and PETSc codes of LOBPCG. Our native *hypre* LOBPCG version efficiently takes advantage of powerful *hypre* algebraic and geometric multigrid preconditioners. Our native PETSc LOBPCG version gives the PETSc users community an easy access to a customizable code of a high quality modern preconditioned eigensolver.

The LOBPCG method has recently attracted attention as a potential competitor to the Lanczos and Davidson methods due to its simplicity, robustness and fast convergence. Implementations in C++ (by R. Lehoucq, U. Hetmaniuk et al. [1, 4], Anasazi Trilinos), in FORTRAN 77 (by Randolph Bank, a part of PLTMG 9.0 and above) and in FORTRAN 90 (by G. Zèrah, a part of ABINIT v4.5 and above, complex Hermitian matrices) of the LOBPCG are being developed by different groups for such application areas as structural mechanics, mesh partitioning and electronic structure calculations.

2 Abstract LOBPCG implementation for *hypre*/PETSc

For computing only the smallest eigenpair, we take the block size $m = 1$ and then the LOBPCG gets reduced to a local optimization of a 3-term recurrence:

$$x^{(i+1)} = w^{(i)} + \tau^{(i)} x^{(i)} + \gamma^{(i)} x^{(i-1)},$$

$$w^{(i)} = T(Ax^{(i)} - \lambda^{(i)} Bx^{(i)}),\ \lambda^{(i)} = \lambda(x^{(i)}) = (x^{(i)}, Ax^{(i)})/(Bx^{(i)}, x^{(i)})$$

with properly chosen scalar iteration parameters $\tau^{(i)}$ and $\gamma^{(i)}$. The easiest and most efficient choice of parameters is based on an idea of *local optimality* [5, 6], namely, select $\tau^{(i)}$ and $\gamma^{(i)}$ that minimize the Rayleigh quotient $\lambda(x^{(i+1)})$ by using the Rayleigh–Ritz method. For finding the m smallest eigenpairs the Rayleigh–Ritz method on a $3m$–dimensional trial subspace is used during each iteration for the local optimization.

LOBPCG description in [6] skips important details. The complete description of the LOBPCG algorithm as it has been implemented in our MATLAB code rev. 4.10 and the *hypre* code 1.9.0b follows:

```
Input: $m$ starting linearly independent multivectors in $X \in \mathbb{R}^{n \times m}$,
    $l$ linearly independent constraint multivectors in $Y \in \mathbb{R}^{n \times l}$, devices to
    compute $A * X$, $B * X$ and $T * X$.
1.  Allocate memory for ten multivectors
    $W, P, Q, AX, AW, AP, BX, BW, BP, BY \in \mathbb{R}^{n \times m}$.
2.  Apply constraints to $X$:
    $BY = B * Y$; $X = X - Y * \left(Y^T * BY\right)^{-1} * X^T * BY$.
3.  $B$-orthonormalize $X$: $BX = B * X; R = \text{chol}(X^T * BX); X = X * R^{-1}$;
    $BX = BX * R^{-1}$; $AX = A * X$. (”chol” is the Cholesky decomposition)
4.  Compute the initial Ritz vectors: solve the eigenproblem
    $(X^T * AX) * TMP = TMP * \Lambda$;
    and compute $X = X * TMP; AX = AX * TMP$; $BX = BX * TMP$.
5.  Define index set $I$ to be $\{1, \ldots, m\}$
6.  for $k = 0, \ldots, MaxIterations$
7.      Compute residuals: $W_I = AX_I - BX_I * \Lambda_I$.
8.      Exclude from index set $I$ those indices which correspond to residual
        vectors for which the norm has become smaller than the tolerance.
        If $I$ then becomes empty, exit loop.
9.      Apply preconditioner $T$ to the residuals: $W_I = T * W_I$.
```

```
Apply constraints to the preconditioned residuals $W_I$ :
$W_I = W_I - Y * \left(Y^T * BY\right)^{-1} * W_I^T * BY$ .
$B$-orthonormalize $W_I$ : $BW_I = B * W_I$ ; $R = \mathrm{chol}(W_I^T * BW_I)$ ;
$W_I = W_I * R^{-1}$ ; $BW_I = BW_I * R^{-1}$ .
Compute $AW_I$ : $AW_I = A * W_I$ .
if $k > 0$
    $B$-orthonormalize $P_I$ : $R = \mathrm{chol}(P_I^T * BP_I); P_I = P_I * R^{-1}$ ;
    Update $AP_I = AP_I * R^{-1}$ ; $BP_I = BP_I * R^{-1}$ .
end if
Perform the Rayleigh Ritz Procedure:
    Compute symmetric Gram matrices:
    if $k > 0$
        $gramA = \begin{bmatrix} \Lambda & X^T * AW_I & X^T * AP_I \\ \cdot & W_I^T * AW_I & W_I^T * AP_I \\ \cdot & \cdot & P_I^T * AP_I \end{bmatrix}$ .
        $gramB = \begin{bmatrix} I & X^T * BW_I & X^T * BP_I \\ \cdot & I & W_I^T * BP_I \\ \cdot & \cdot & I \end{bmatrix}$ .
    else
        $gramA = \begin{bmatrix} \Lambda & X^T * AW_I \\ \cdot & W_I^T * AW_I \end{bmatrix}$ .
        $gramB = \begin{bmatrix} I & X^T * BW_I \\ \cdot & I \end{bmatrix}$ .
    end if
    Solve the generalized eigenvalue problem:
    $gramA * Y = gramB * Y * \Lambda$ , where the first $m$ eigenvalues in
    increasing order are in the diagonal matrix $\Lambda$ and the
    $gramB$-orthonormalized eigenvectors are the columns of $Y$ .
    Compute Ritz vectors:
    if $k > 0$
        Partition $Y = \begin{bmatrix} Y_X \\ Y_W \\ Y_P \end{bmatrix}$ according to the number of columns in
        $X$, $W_I$ , and $P_I$ , respectively.
        Compute $P = W_I * Y_W + P_I * Y_P$ ;
        $AP = AW_I * Y_W + AP_I * Y_P$ ; $BP = BW_I * Y_W + BP_I * Y_P$ .
        $X = X * Y_X + P; AX = AX * Y_X + AP; BX = BX * Y_X + BP$ .
    else
        Partition $Y = \begin{bmatrix} Y_X \\ Y_W \end{bmatrix}$ according to the number of columns in
        $X$ and $W_I$ respectively.
        $P = W_I * Y_W; AP = AW_I * Y_W; BP = BW_I * Y_W$ .
        $X = X * Y_X + P; AX = AX * Y_X + AP; BX = BX * Y_X + BP$ .
    end if
end for
Output: Eigenvectors $X$ and eigenvalues $\Lambda$ .
```

The LOBPCG eigensolver code is written in C-language and calls a few LAPACK subroutines. The matrix–vector multiply and the preconditioner call are done through user supplied functions. The main LOBPCG code is abstract in the sense that it works only through an interface that determines the particular software environment: *hypre* or PETSc, in order to call parallel (multi)vector manipulation routines.

A block diagram of the high-level software modules is given in Figure 1.

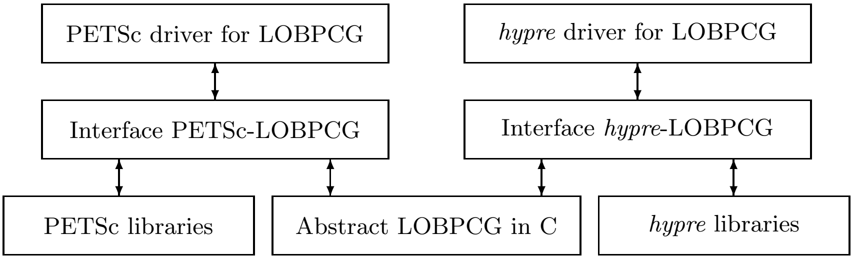


Fig. 1. LOBPCG *hypre*/PETSc software modules.

hypre supports four conceptual interfaces: Struct, SStruct, FEM and IJ. At present, LOBPCG has been tested with all but the FEM interface. *hypre* test drivers for LOBPCG are simple extensions of the *hypre* test drivers for linear system. We anticipate that both types of drives will be merged in the post 1.9.0b *hypre* release.

We do not use shift-and-invert strategy. Preconditioning is implemented directly as well as through calls to the *hypre*/PETSc preconditioned conjugate gradient method (PCG). Specifically, in the latter case the action $x = Tb$ of the preconditioner T on a given vector b is performed by calling a few steps of PCG to solve $Ax = b$.

LOBPCG-*hypre* has been tested with all available *hypre* PCG-capable preconditioners in Struct, SStruct and IJ interfaces, most notably, with IJ AMG–PCG algebraic multigrid, IJ DS–PCG diagonal scaling, IJ additive Schwarz–PCG, and Struct PFMG-PCG geometric multigrid. LOBPCG-PETSc has been tested with PETSc native Additive Schwarz and PETSc linked IJ AMG from *hypre*.

3 *hypre*/PETSc LOBPCG Numerical Results

3.1 Basic Accuracy of Algorithm

In these tests LOBPCG computes the smallest 50 eigenvalues of 3D 7–Point $200 \times 200 \times 200$ and $200 \times 201 \times 202$ Laplacians. In the first case we have eigenvalues with multiplicity and in the second case the eigenvalues are distinct, but clustered. The initial eigenvectors are chosen randomly. We set the stopping tolerance (the norm of the maximum residual) equal to 10^{-6}. The numerical output and exact eigenvalues are compared. In both cases for all eigenvalues the maximum relative error is less than 10^{-8} and the Frobenius norm $\|V^T V - I_{m\times m}\| < 10^{-12}$, where $V \in \mathbb{R}^{n\times m}$

contains the approximate eigenvectors. These tests suggest that LOBPCG is cluster robust, i.e. it does not miss (nearly) multiple eigenvalues.

The LOBPCG code may become unstable because of ill-conditioned Gram matrices in some tests, which is typically a result of bad initial guesses, e.g., generated by a poor quality random number generator. When the ill-conditioning appears restarts are helpful. The simplest restart is to drop the matrix P from the basis of the trial subspace. Such restarts improve the stability of the LOBPCG code as observed in MATLAB tests, and are planned to be implemented in a future *hypre*/PETSc LOBPCG revision.

3.2 Performance Versus the Number of Inner Iterations

Let us remind the reader that we can execute a preconditioner $x = Tb$ directly or by calling PCG to solve $Ax = b$. We do not attempt to use shift-and-invert strategy, but instead simply take T to be a preconditioner for A. Therefore, we can expect that increasing the number of "inner" iterations of the PCG might accelerate the overall convergence, but only if we do not make too many iterations. In other words, for a given matrix A and a particular choice of a preconditioner, there should be an optimal finite number of inner iterations.

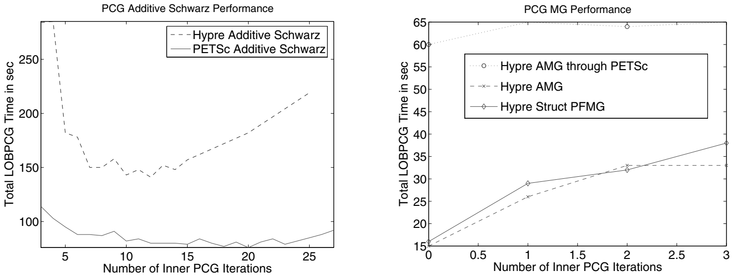


Fig. 2. Performance versus the number of inner iterations. 7–Point 3-D Laplacian, 1,000,000 unknowns. Dual 2.4-GHz Xeon 4GB.

In numerical example illustrated on Figure 2, we try to find this optimal number for the Schwarz–PCG and AMG-PCG preconditioners in *hypre* and PETSc. We measure the execution time as we vary the quality of the preconditioner by changing the maximum number of inner iterations in the corresponding PCG solver. We find that for this problem the optimal number of inner iterations is approximately $10 - 15$ for Schwarz-PCG, but AMG-PCG works best if AMG is applied directly as a preconditioner, without even initializing the AMG-PCG function.

Our explanation of this behavior is based on two facts. First, the Schwarz method is somewhat cheaper, but not of such a good quality, compared to AMG in these tests. Moreover, the costs for matrix vector multiplies and multivector linear algebra in LOBPCG is a relatively small proportion of the AMG application, but comparable to the computational cost of Schwarz here. Second, one PCG iteration is less computationally expensive compared to one LOBPCG iteration because of larger number of linear algebra operations with multivectors in the latter. A single direct application of AMG as the preconditioner in LOBPCG gives enough improvement in convergence to make it the best choice, while Schwarz requires more iterations that are less time consuming if performed using PCG, rather than by direct application in LOBPCG.

3.3 LOBPCG Performance vs. Block Size

We test both *hypre* and PETSc LOBPCG codes on a 7–Point 3-D Laplacian with 2,000,000 unknowns with *hypre* AMG Preconditioner on a Sun Fire 880, 6 CPU 24GB system by increasing the block size m, i.e. the number of computed eigenvectors, from 1 to 16. We observe that the growth of the total CPU time with the increase of the block size is linear, from approximately 100 sec for $m = 1$ to 2,500 sec for $m = 16$. We expect that for larger m the complexity term m^2n will become visible. We note, however, that neither *hypre* nor PETSc currently has efficiently implemented multivectors, e.g., in the current implementation the number of MPI communications in the computation of the Gram matrices grows with m. An efficient implementation of the main multivector functions is crucial in order to significantly reduce the overall costs for large m.

3.4 Scalability with the Schwarz–PCG and Multigrid–PCG preconditioners

We test scalability by varying the problem size so it is proportional to the number of processors. We use a 7–Point 3–D Laplacian and set the block size to $m = 1$.

For the Schwarz–PCG, we set the maximum number of inner iterations of the PCG to 10. The tests are performed on the Beowulf cluster at CU Denver that includes 36 nodes, two PIII 933MHz processors and 2GB memory per node, running Linux RedHat and a 7.2SCI Dolpin interconnect and on MCR cluster (dual Xeon 2.4-GHz, 4 GB nodes) at LLNL. In all these tests, the time per iteration is reasonably scalable, but the number of LOBPCG iterations grows with the problem size i.e., the Schwarz–PCG preconditioner in *hypre* and in PETSc is not optimal in this case.

For the Multigrid–PCG preconditioners, we apply the preconditioners directly, without calling the PCG. We test here *hypre* IJ AMG–PCG algebraic multigrid, *hypre* Struct PFMG-PCG geometric multigrid and PETSc linked IJ AMG from *hypre* on the LLNL MCR cluster, see Figure 3.4 left.

Good LOBPCG scalability can be seen in Figure 3.4, left. The Struct PFMG takes more time compared to AMG here because of the larger convergence factor. To satisfy the reader curiosity, we also provide the scalability data for the preconditioner setup on Figure 3.4 right.

Conclusions

- We present the world's apparently first parallel code for generalized symmetric definite eigenvalue problems, that can apply preconditioning directly. The

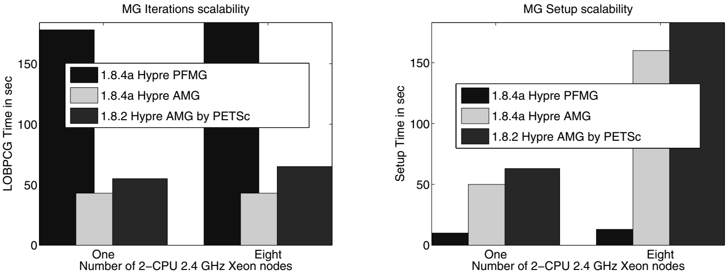


Fig. 3. 7–Point Laplacian, 2,000,000 unknowns per node. Preconditioners: AMG and PFMG. System: LLNL MCR. LOBPCG scalability (left) and preconditioner setup (right).

LOBPCG is our method of choice for preconditioned eigensolver because of its simplicity, robustness and fast convergence.

- Our *hypre*/PETSc LOBPCG code illustrates that the LOBPCG "matrix-free" algorithm can be successfully implemented using parallel libraries that are designed to run on a great variety of multiprocessor platforms.
- In the problems tested with AMG preconditioning, 90%–99% of the computational effort is required for the preconditioner setup and in the applying the preconditioner and thus the LOBPCG scalability is mainly dependent on the scalability of *hypre*/PETSc preconditioning. Initial scalability measurements look promising, but more testing is needed by other users.
- The LOBPCG *hypre* software has been integrated into the *hypre* software at LLNL and has been publicly released in *hypre*–1.8.2b and above. The LOBPCG PETSc software is now available in PETSc at Argonne as a part of our BLOPEX, which is an external PETSc package.

The results of this work have been presented at: 11th and 12th Copper Mountain Conferences on Multigrid Methods, Preconditioning 2003, SIAM Parallel Processing for Scientific Computing 2004, and 16th International Conference on Domain Decomposition Methods. Earlier results and the pecularities of our LOBPCG implementation in *hypre*–1.8.0b can be found in [7].

We are grateful to all members of the Scalable Algorithms Group of the Center for Applied Scientific Computing, Lawrence Livermore National Laboratory and, in particular, to Rob Falgout, Edmond Chow, Charles Tong, and Panayot Vassilevski, for their patient support and help. We thank Matthew Knepley and other members of the PETSc team for their help with incorporating our BLOPEX as an external package into the PETSc library.

References

1. P. Arbenz, U. L. Hetmaniuk, R. B. Lehoucq, and R. S. Tuminaro, *A comparison of eigensolvers for large-scale 3D modal analysis using AMG-preconditioned iterative methods*, Internat. J. Numer. Methods Engrg., 64 (2005), pp. 204–236.
2. 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*, Tech. Rep. ANL-95/11 - Revision 2.1.5, Argonne National Laboratory, 2004.
3. R. D. Falgout, J. E. Jones, and U. M. Yang, *Pursuing scalability for hypre's conceptual interfaces*, ACM Trans. Math. Software, 31 (2005), pp. 326–350.
4. M. A. Heroux, R. A. Bartlett, V. E. Howle, R. J. Hoekstra, J. J. Hu, T. G. Kolda, R. B. Lehoucq, K. R. Long, R. P. Pawlowski, E. T. Phipps, A. G. Salinger, H. K. Thornquist, R. S. Tuminaro, J. M. Willenbring, A. Williams, and K. S. Stanley, *An overview of the Trilinos project*, ACM Trans. Math. Software, 31 (2005), pp. 397–423.
5. A. V. Knyazev, *Preconditioned Eigensolvers: practical algorithms*, SIAM, Philadelphia, 2000, pp. 352–368.
6. ———, *Toward the optimal preconditioned Eigensolver: Locally optimal block preconditioned conjugate gradient method*, SIAM J. Sci. Comput., 23 (2001), pp. 517–541.
7. A. V. Knyazev and M. E. Argentati, *Implementation of a preconditioned Eigensolver using Hypre*, Tech. Rep. UCD-CCM 220, Center for Computational Mathematics, University of Colorado at Denver, April 2005.

A New FETI-based Algorithm for Solving 3D Contact Problems with Coulomb Friction *

Radek Kučera [1] , Jaroslav Haslinger [2] , and Zdeněk Dostál [3]

[1] Department of Mathematics and Descriptive Geometry, VŠB-Technical University Ostrava, Czech Republic. radek.kucera@vsb.cz

[2] Department of Numerical Mathematics, Charles University, Prague, Czech Republic. haslin@met.mff.cuni.cz

[3] Department of Applied Mathematics, VŠB-Technical University Ostrava, Czech Republic. zdenek.dostal@vsb.cz

Summary. The paper deals with solving of contact problems with *Coulomb* friction for a system of 3D elastic bodies. The iterative method of successive approximations is used in order to find a fixed point of certain mapping that defines the solution. In each iterative step, an auxiliary problem with *given* friction is solved that is discretized by the FETI method. Then the duality theory of convex optimization is used in order to obtain the constrained quadratic programming problem that, in contrast to 2D case, is subject to quadratic inequality constraints. The solution is computed (among others) by a novelly developed algorithm of constrained quadratic programming. Numerical experiments demonstrate the performance of the whole computational process.

1 Introduction

The FETI method was proposed by [6] for parallel solution of problems described by elliptic partial differential equations. The key idea is elimination of the primal variables so that the original problem is reduced to a small, relatively well conditioned quadratic programming problem (QPP) in terms of the Lagrange multipliers. Then the iterative solver is used to compute the solution.

In context of 2D contact problems with friction, the FETI procedure leads to the sequence of QPPs constrained by simple inequality bounds (see [3] or [8]) so that the fast algorithm with proportioning and gradient projection (see [4]) can be used. The situation is not so easy in 3D since the QPPs are subject to two types of constraints. The first one, representing nonnegativity of the normal contact stress, are again simple inequality bounds while the second one, representing an effect of

*Supported by grant GAČR 101/04/1145 and 101/05/0423.

isotropic friction, are quadratic inequalities. In our recent papers [9], [12], we have used a linear approximation of quadratic inequalities transforming them to simple inequality bounds so that the fast algorithm mentioned above can be used again. Unfortunately, this procedure increases considerably the size of the QPPs if we require a sufficiently accurate approximation of quadratic inequalities. In order to overcome this drawback, we have developed a new algorithm of quadratic programming that treates directly the quadratic inequalities [11]. In this contribution, we shall show the performance of the whole computational process on model problems.

2 Formulation of the problems

Let us consider a system of elastic bodies that occupy in the reference configuration bounded domains $\Omega^p \subset \mathbb{R}^3$, $p = 1, 2, \ldots, s$, with sufficiently smooth boundaries Γ^p that are split into three disjoint parts Γ^p_u , Γ^p_t and Γ^p_c so that $\Gamma^p = \overline{\Gamma^p_u} \cup \overline{\Gamma^p_t} \cup \overline{\Gamma^p_c}$. Let us suppose that the zero displacements are prescribed on Γ^p_u and that the surface tractions of density $\mathbf{t}^{\,p} \in (L^2(\Gamma^p_t))^3$ act on Γ^p_t . Along Γ^p_c the body Ω^p may get into unilateral contact with some other of the bodies. Finally we suppose that the bodies Ω^p are subject to the volume forces of density $\mathbf{f}^{\,p} \in (L^2(\Omega^p))^3$.

To describe non-penetration of the bodies, we shall use linearized non-penetration condition that is defined by a mapping $\boldsymbol{\chi} : \Gamma_c \longrightarrow \Gamma_c$, $\Gamma_c = \bigcup_{p=1}^{s} \Gamma^p_c$, which assigns to each $\mathbf{x} \in \Gamma^p_c$ some nearby point $\boldsymbol{\chi}(\mathbf{x}) \in \Gamma^q_c$, $p \neq q$. Let $\mathbf{v}^{\,p}(\mathbf{x})$, $\mathbf{v}^{\,q}(\boldsymbol{\chi}(\mathbf{x}))$ denote the displacement vectors at $\mathbf{x}$, $\boldsymbol{\chi}(\mathbf{x})$, respectively. Assuming the small displacements, the *non-penetration condition* reads

$$v^p_n(\mathbf{x}) \equiv (\mathbf{v}^{\,p}(\mathbf{x}) - \mathbf{v}^{\,q}(\boldsymbol{\chi}(\mathbf{x}))) \cdot \mathbf{n}^{\,p}(\mathbf{x}) \leq \delta^p(\mathbf{x}),$$

where $\delta^p(\mathbf{x}) = (\boldsymbol{\chi}(\mathbf{x}) - \mathbf{x}) \cdot \mathbf{n}^{\,p}(\mathbf{x})$ is the initial gap and $\mathbf{n}^{\,p}(\mathbf{x})$ is the critical direction defined by $\mathbf{n}^{\,p}(\mathbf{x}) = (\boldsymbol{\chi}(\mathbf{x}) - \mathbf{x})/\|\boldsymbol{\chi}(\mathbf{x}) - \mathbf{x}\|$ or, if $\boldsymbol{\chi}(\mathbf{x}) = \mathbf{x}$, by the outer unit normal vector to Γ^p_c .

We start with an auxiliary contact problem with given friction. To this end we introduce the space of *virtual displacements* V and its closed convex subset of *kinematically admissible* displacements $\mathcal{K}$ by

$$V = \{\mathbf{v} = (\mathbf{v}^1, \ldots, \mathbf{v}^s) \in \prod_{p=1}^{s} (H^1(\Omega^p))^3 : \mathbf{v}^p = 0 \text{ on } \Gamma^p_u\},$$
$$\mathcal{K} = \{\mathbf{v} \in V : v^p_n(\mathbf{x}) \leq \delta^p(\mathbf{x}) \text{ for } \mathbf{x} \in \Gamma^p_c\}.$$

Let us assume that the normal contact stress $T_n \in L^\infty(\Gamma_c)$, $T_n \geq 0$, is known *apriori* so that one can evaluate the slip bound g on Γ_c by $g = FT_n$, where $F = F^p > 0$ is a coefficient of friction on Γ^p_c . Denote $g^p = g|_{\Gamma^p_c}$.

The variational formulation of the contact problem with *given* friction reads as follows:

$$\min \mathcal{J}(\mathbf{v}) \quad \text{subject to} \quad \mathbf{v} \in \mathcal{K}, \tag{1}$$

where

$$\mathcal{J}(\mathbf{v}) = \frac{1}{2} a(\mathbf{v}, \mathbf{v}) - b(\mathbf{v}) + j(\mathbf{v})$$

is the total potential energy functional with the bilinear form a representing the inner energy of the bodies and with the linear form b representing the work of the applied forces $\mathbf{t}^{p}$ and $\mathbf{f}^{p}$, respectively. The sublinear functional j represents the work of friction forces

$$j(\mathbf{v}) = \sum_{p=1}^{s} \int_{\Gamma_c^p} g^p \| \mathbf{v}_t^p \| \, d\Gamma, \tag{2}$$

where $\mathbf{v}_t^p$ is the projection of the displacement $\mathbf{v}^p$ on the plane tangential to the critical direction $\mathbf{n}^p$. Let us introduce unit tangential vectors $\mathbf{t}_1^p, \mathbf{t}_2^p$ such that the triplet $\mathcal{B} = \{\mathbf{n}^p, \mathbf{t}_1^p, \mathbf{t}_2^p\}$ is an orthonormal basis in $\mathbb{R}^3$ for almost all $\mathbf{x} \in \Gamma_c^p$ and denote $v_{t_1}^p = \mathbf{v}^p \cdot \mathbf{t}_1^p$, $v_{t_2}^p = \mathbf{v}^p \cdot \mathbf{t}_2^p$. Then $\mathbf{v}_t^p = (0, v_{t_1}^p, v_{t_2}^p)$ with respect to the basis $\mathcal{B}$ so that the norm appearing in j reduces to the Euclidean norm in $\mathbb{R}^2$. More details about the formulation of contact problems can be found in [10].

Let us point out that the solution $\mathbf{u} \equiv \mathbf{u}(g)$ of (1) depends on a particular choice of $g \in L^\infty(\Gamma_c)$, $g \geq 0$. We can define a mapping Φ which associates with every g the product $FT_n(\mathbf{u}(g))$, where $T_n(\mathbf{u}(g)) \geq 0$ is the normal contact stress related to $\mathbf{u}(g)$. The classical Coulomb's law of friction corresponds to the fixed point of Φ which is defined by $g = FT_n(\mathbf{u}(g))$. To find it, we can use the *method of successive approximations* which starts from a given $g^{(0)}$ and generates the iterations $g^{(l)}$ by

$$\text{(MSA)} \qquad g^{(l+1)} = \Phi(g^{(l)}), \; l = 1, 2, \ldots .$$

This iterative process converges provided Φ is *contractive*, that is guaranteed for sufficiently small F (see [7]).

3 Domain decomposition and discretization

We divide the bodies Ω^p into tetrahedron finite elements $\mathcal{T}$ with the maximum diameter h and assume that the partitions are regular and consistent with the decompositions of $\partial\Omega^p$ into Γ_u^p, Γ_t^p and Γ_c^p. Moreover, we restrict ourselves to the geometrical conforming situation where the intersection between the boundaries of any two different bodies $\partial\Omega^p \cap \partial\Omega^q$, $p \neq q$, is either empty, a vertex, an entire edge, or an entire face.

Let the domains Ω^p be decomposed into nonoverlaping subdomains $\Omega^{p,i}$, $i = 1, \ldots, n_p$, each of which is the union of finite elements of $\mathcal{T}$. On $\Omega^{p,i}$, we introduce the finite element space $V_h^{p,i}$ by

$$\begin{aligned} V_h^{p,i} = \{ & \mathbf{v}^{p,i} \in (C(\Omega^{p,i}))^3 : \mathbf{v}^{p,i}|_{\mathcal{T}} \in (P_1(\mathcal{T}))^3 \text{ for all } \mathcal{T} \subset \Omega^{p,i}, \\ & \mathbf{v}^{p,i}|_{\partial\Omega^{p,i} \cap \Gamma_u^p} = 0 \}, \end{aligned}$$

where $P_m(\mathcal{T})$ denotes the set of all polynomials on $\mathcal{T}$ of degree $\leq m$. Finally, let us introduce the product space $V_h = \prod_{p=1}^{s} \prod_{i=1}^{n_p} V_h^{p,i}$.

Replacing V by V_h and using the *gluing condition* $\mathbf{v}^{p,i}(\mathbf{x}) = \mathbf{v}^{p,j}(\mathbf{x})$ for any $\mathbf{x}$ in the interface $\partial\Omega^{p,i} \cap \partial\Omega^{p,j}$, we can rewrite the approximative contact problem with given friction (1) into the algebraic form

$$\begin{aligned}&\min \frac{1}{2}\mathbf{u}^\top\mathbf{K}\mathbf{u}-\mathbf{u}^\top\mathbf{f}+\sum_{k=1}^{m} g_k\|((\mathbf{T}_1\mathbf{u})_k,(\mathbf{T}_2\mathbf{u})_k)\|\\&\text{s.t. } \mathbf{N}\mathbf{u}\le\mathbf{d},\ \mathbf{B}_E\mathbf{u}=0.\end{aligned}\tag{3}$$

Here, $\mathbf{K}$ denotes the positive semidefinite block diagonal stiffness matrix, $\mathbf{f}$ is the vector of nodal forces, $\mathbf{N}$, $\mathbf{d}$ describe the discretized non-penetration condition and $\mathbf{B}_E$ describes the gluing condition. The summation term in the minimized functional arises using numerical quadrature in (2), where $\mathbf{T}_1$, $\mathbf{T}_2$ describe projections of displacements at the nodes lying on Γ_c to the tangential planes and g_k are values of slip bound.

Let us point out that the problem (3) is non-differentiable due to $\mathbb{R}^2$-norms appearing in the summation term. Therefore we shall introduce two kinds of Lagrange multipliers $\boldsymbol{\lambda}_t=(\boldsymbol{\lambda}_{t_1}^\top,\boldsymbol{\lambda}_{t_2}^\top)^\top$ and $\boldsymbol{\lambda}_c=(\boldsymbol{\lambda}_I^\top,\boldsymbol{\lambda}_E^\top)^\top$. While the first one removes the non-differentiability, the second one accounts for the constraints in (3). Denote

$$\mathbf{B}_t=\begin{bmatrix}\mathbf{T}_1\\\mathbf{T}_2\end{bmatrix},\quad \mathbf{B}_c=\begin{bmatrix}\mathbf{N}\\\mathbf{B}_E\end{bmatrix},\quad \mathbf{c}=\begin{bmatrix}\mathbf{d}\\\mathbf{o}\end{bmatrix}$$

and introduce the Lagrange multiplier sets

$$\Lambda_t(\mathbf{g})=\{\boldsymbol{\lambda}_t:\|((\boldsymbol{\lambda}_{t1})_k,(\boldsymbol{\lambda}_{t2})_k)\|\le g_k\}\ \text{ and }\ \Lambda_c=\{\boldsymbol{\lambda}_c:(\boldsymbol{\lambda}_I)_k\ge 0\}.$$

It is well-known that (3) is equivalent to the saddle-point problem

$$\text{Find } (\mathbf{u},\boldsymbol{\lambda}_t,\boldsymbol{\lambda}_c) \text{ s.t. } \mathcal{L}(\mathbf{u},\boldsymbol{\lambda}_t,\boldsymbol{\lambda}_c)=\sup_{\substack{\boldsymbol{\mu}_t\in\Lambda_t(\mathbf{g})\\\boldsymbol{\mu}_c\in\Lambda_c}}\inf_{\mathbf{v}}\mathcal{L}(\mathbf{v},\boldsymbol{\mu}_t,\boldsymbol{\mu}_c),\tag{4}$$

where $\mathcal{L}$ is the Lagrangian to (3) defined by

$$\mathcal{L}(\mathbf{u},\boldsymbol{\lambda}_t,\boldsymbol{\lambda}_c)=\frac{1}{2}\mathbf{u}^\top\mathbf{K}\mathbf{u}-\mathbf{u}^\top\mathbf{f}+\boldsymbol{\lambda}_t^\top\mathbf{B}_t\mathbf{u}+\boldsymbol{\lambda}_c^\top(\mathbf{B}_c\mathbf{u}-\mathbf{c}).$$

After eliminating the primal variables $\mathbf{u}$ from (4), we obtain the minimization problem

$$\begin{aligned}&\min \frac{1}{2}\boldsymbol{\lambda}^\top\mathbf{F}\boldsymbol{\lambda}-\boldsymbol{\lambda}^\top\mathbf{h}\\&\text{s.t. } \boldsymbol{\lambda}=\begin{bmatrix}\boldsymbol{\lambda}_t\\\boldsymbol{\lambda}_c\end{bmatrix},\boldsymbol{\lambda}_t\in\Lambda_t(\mathbf{g}),\boldsymbol{\lambda}_c\in\Lambda_c,\ \mathbf{G}\boldsymbol{\lambda}=\mathbf{e}\end{aligned}\tag{5}$$

with

$$\mathbf{F}=\begin{bmatrix}\mathbf{F}_{tt}&\mathbf{F}_{tc}\\\mathbf{F}_{tc}^\top&\mathbf{F}_{cc}\end{bmatrix},\quad \mathbf{h}=\begin{bmatrix}\mathbf{h}_t\\\mathbf{h}_c\end{bmatrix},\quad \mathbf{G}=[\mathbf{G}_t,\mathbf{G}_c],$$

and $\mathbf{F}_{ii}=\mathbf{B}_i\mathbf{K}^\dagger\mathbf{B}_i^\top$, $\mathbf{G}_i=\mathbf{R}^\top\mathbf{B}_i^\top$, $i=t,c$, $\mathbf{F}_{tc}=\mathbf{B}_t\mathbf{K}^\dagger\mathbf{B}_c^\top$, $\mathbf{h}_t=\mathbf{B}_t\mathbf{K}^\dagger\mathbf{f}$, $\mathbf{h}_c=\mathbf{B}_c\mathbf{K}^\dagger\mathbf{f}-\mathbf{c}$, $\mathbf{e}=\mathbf{R}^\top\mathbf{f}$, where $\mathbf{K}^\dagger$ denotes a generalized inverse to $\mathbf{K}$ and $\mathbf{R}$ is the full rank matrix whose columns span the kernel of $\mathbf{K}$.

The problem (5) can be adapted by using the orthogonal projectors as proposed in [5]. To simplify our presentation, we omit description of this modification here.

4 Algorithms

The problem (5) can be solved by using the algorithm based on the augmented Lagrangian

$$L(\boldsymbol{\lambda}, \boldsymbol{\mu}, \rho) = \frac{1}{2}\boldsymbol{\lambda}^\top \mathbf{F}\,\boldsymbol{\lambda} - \boldsymbol{\lambda}^\top \mathbf{h} + \boldsymbol{\mu}^\top(\mathbf{G}\,\boldsymbol{\lambda} - \mathbf{e}) + \frac{\rho}{2}(\mathbf{G}\,\boldsymbol{\lambda} - \mathbf{e})^\top(\mathbf{G}\,\boldsymbol{\lambda} - \mathbf{e}).$$

Algorithm 1. Set $\boldsymbol{\mu}^{(0)}$, $l := 0$.

```
repeat
    $\boldsymbol{\lambda}^{(l+1)} \doteq \mathrm{argmin}\ L(\boldsymbol{\lambda}, \boldsymbol{\mu}^{(l)}, \rho)$, s.t. $\boldsymbol{\lambda} \in \boldsymbol{\Lambda}_t(\mathbf{g}) \times \boldsymbol{\Lambda}_c$
    $\boldsymbol{\mu}^{(l+1)} = \boldsymbol{\mu}^{(l)} + \rho(\mathbf{G}\,\boldsymbol{\lambda}^{(l+1)} - \mathbf{e})$
    Update $\rho$ and increase $l$ by one.
until stopping criterion
```

Algorithms of this type have been intensively studied recently [2], [1] with the inner minimization represented by the QPP with simple inequality bounds of $\boldsymbol{\Lambda}_c$. Here, the quadratic inequality constraints of $\boldsymbol{\Lambda}_t(\mathbf{g})$ are imposed furthermore. In order to separate two types of constraints, we can split the inner minimization by the constrained block Gauss-Seidel method. Then the efficient algorithm using projections and adaptive precision control may be used for the first QPP with simple inequality bounds [4] while the second QPP constrained by quadratic inequalities can be solved by the algorithm proposed in [11]. Let us point out that augmented Lagrangian based algorithms accept an inexact solution of the inner minimizations without loss of the accuracy. Therefore it is natural to reduce the number of Gauss-Seidel iterations even onto one.

The method of successive approximations (MSA) for solving the contact problem with Coulomb friction can be implemented so that the Algorithm 1 is used in each iterative step to evaluate the mapping Φ. We shall present a more efficient version of this method, in which the iterative steps of (MSA) and the loop of the Algorithm 1 are connected in one loop. The resulting algorithm can be viewed as the method of successive approximations with an *inexact* solving of the auxiliary problems with given friction.

Algorithm 2. Set $\boldsymbol{\mu}^{(0)}$, $\boldsymbol{\lambda}_t^{(0)}$, $l := 0$.

```
repeat
    $\boldsymbol{\lambda}_c^{(l+1)} \doteq \mathrm{argmin}\ \{\frac{1}{2}\boldsymbol{\lambda}_c^\top(\mathbf{F}_{cc} + \rho\,\mathbf{G}_c^\top\mathbf{G}_c)\boldsymbol{\lambda}_c - \boldsymbol{\lambda}_c^\top(\mathbf{h}_c + \mathbf{G}_c^\top(\rho\,\mathbf{e} +$
            $\boldsymbol{\mu}^{(l)}) - (\mathbf{F}_{tc}^\top + \rho\,\mathbf{G}_c^\top\mathbf{G}_t)\boldsymbol{\lambda}_t^{(l)})\}$, s.t. $\boldsymbol{\lambda}_c \in \boldsymbol{\Lambda}_c$
    $\boldsymbol{\lambda}_t^{(l+1)} \doteq \mathrm{argmin}\ \{\frac{1}{2}\boldsymbol{\lambda}_t^\top(\mathbf{F}_{tt} + \rho\,\mathbf{G}_t^\top\mathbf{G}_t)\boldsymbol{\lambda}_t - \boldsymbol{\lambda}_t^\top(\mathbf{h}_t + \mathbf{G}_t^\top(\rho\,\mathbf{e} +$
            $\boldsymbol{\mu}^{(l)}) - (\mathbf{F}_{tc} + \rho\,\mathbf{G}_t^\top\mathbf{G}_c)\boldsymbol{\lambda}_c^{(l+1)})\}$, s.t. $\boldsymbol{\lambda}_t \in \boldsymbol{\Lambda}_t(F\boldsymbol{\lambda}_I^{(l+1)})$
    $\boldsymbol{\mu}^{(l+1)} = \boldsymbol{\mu}^{(l)} + \rho(\mathbf{G}\,\boldsymbol{\lambda}^{(l+1)} - \mathbf{e})$
    Update $\rho$ and increse $l$ by one.
until stopping criterion
```

We have used the fact that the Lagrange multiplier $\boldsymbol{\lambda}_I$ represents the normal contact stress so that $\mathbf{g} = F\boldsymbol{\lambda}_I^{(l+1)}$ approximates the slip bound.

5 Numerical experiments and conclusions

Let us consider the model brick $\Omega = \langle 0,3\rangle \times \langle 0,1\rangle \times \langle 0,1\rangle$ made of an elastic isotropic, homogeneous material characterized by Young modulus $E = 21.2 \times 10^{10}$ and Poisson's ratio $\sigma = 0.277$ (steel). The brick is unilaterally supported by the rigid foundation, where the non-penetration condition and the effect of Coulomb friction is considered. The applied surface tractions and the parts of the boundary Γ_u and Γ_c are seen in Figure 1. The volume forces vanish. The brick Ω is artificially decomposed onto three parts as seen in Figure 2 so that the resulting problem has 12 rigid modes.

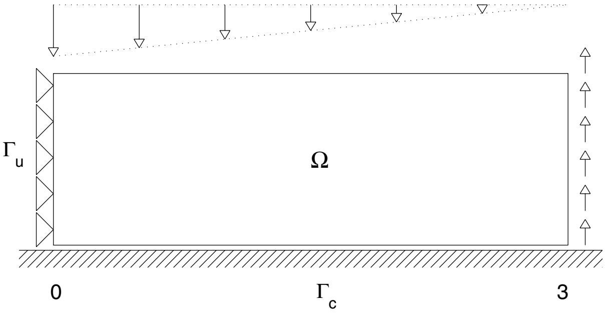


Fig. 1. The cross-section of the brick Ω .

The tables below summarize results of numerical experiments, where F is the coefficient of friction; n denotes the number of primal unknowns (dispalcements); m denotes the number of dual unknowns (stresses); *Time* is CPU time in seconds (in Matlab 7, Pentium(R)4, 3GHz, 512MB); *Iter* is the number of outer iterations; n_A^{QPP} , n_A^{QPQ} is the total number of multiplications by the Hessian in the QPP, QPQ solver, respectively, and $n_A = n_A^{QPP} + n_A^{QPQ}$.

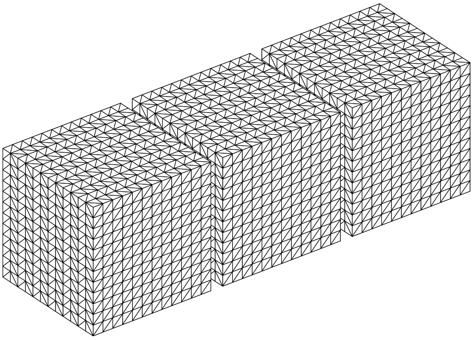


Fig. 2. Discretization and decomposition of the brick Ω .

Table 1. $F = 0.1$

n	m	*Time*	*Iter*	n_A
900	180	1	5	**102**(=63+39)
2646	378	11	5	**180**(=98+82)
5832	648	34	5	**156**(=94+62)
10890	990	67	5	**112**(=50+62)
18252	1404	221	5	**155**(=73+82)

Table 2. $F = 0.3$

n	m	*Time*	*Iter*	n_A
900	180	2	7	**140**(=46+32)
2646	378	12	7	**186**(=54+69)
5832	648	38	7	**169**(=72+50)
10890	990	94	7	**153**(=35+49)
18252	1404	254	7	**176**(=78+54)

Table 1 and Table 2 demonstrate the numerical scalability of the algorithm for various coefficients of friction. Table 3 shows the substantial progress with respect to approximative method used in [9] represented here by *Time2* and *Time4*.

Table 3. $F = 0.3$

n	m	*Time*	*Time2*	*Time4*
900	180	2	15	61
2646	378	12	101	548
5832	648	38	486	2114
10890	990	94	1542	7724
18252	1404	254	5004	20534

References

1. Z. Dostál, *Inexact semimonotonic augmented Lagrangians with optimal feasibility convergence for convex bound and equality constrained quadratic programming*, SIAM J. Num. Anal., 43 (2006), pp. 96–115.
2. Z. Dostál, A. Friedlander, and S. A. Santos, *Augmented Lagrangians with adaptive precision control for quadratic programming with simple bounds and equality constraints*, SIAM J. Opt., 13 (2003), pp. 1120–1140.
3. Z. Dostál, J. Haslinger, and R. Kučera, *Implementation of fixed point method for duality based solution of contact problems with friction*, J. Comput. Appl. Math., 140 (2002), pp. 245–256.

4. Z. Dostál and J. Schöberl, *Minimizing quadratic functions over non-negative cone with the rate of convergence and finite termination*, Comput. Optim. Appl., 30 (2005), pp. 23–43.
5. C. Farhat, J. Mandel, and F.-X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Methods Appl. Mech. Engrg., 115 (1994), pp. 365–385.
6. C. Farhat and F.-X. Roux, *An unconventional domain decomposition method for an efficient parallel solution of large-scale finite element systems*, SIAM J. Sc. Stat. Comput., 13 (1992), pp. 379–396.
7. J. Haslinger, *Approximation of the Signorini problem with friction, obeying Coulomb law*, Math. Methods Appl. Sci., 5 (1983), pp. 422–437.
8. J. Haslinger, Z. Dostál, and R. Kučera, *On splitting type algorithm for the numerical realization of contact problems with Coulomb friction*, Comput. Methods Appl. Mech. Eng., 191 (2002), pp. 2261–2281.
9. J. Haslinger, R. Kučera, and Z. Dostál, *An algorithm for the numerical realization of 3D contact problems with Coulomb friction*, J. Comput. Appl. Math., 164-165 (2004), pp. 387–408.
10. I. Hlaváček, J. Haslinger, J. Nečas, and J. Lovíšek, *Solution of Variational Inequalities in Mechanics*, vol. 66 of Applied Mathematical Sciences, Springer-Verlag New York, 1988.
11. R. Kučera, *Minimizing quadratic functions with separable quadratic constraints*, Opt. Methods Soft., (2007).
12. R. Kučera, J. Haslinger, and Z. Dostál, *The FETI based domain decomposition method for solving 3D-multibody contact problems with Coulomb friction*, Springer-Verlag, Berlin Heidelberg, 2005, pp. 369–376.

A Discontinuous Galerkin Formulation for Solution of Parabolic Equations on Nonconforming Meshes

Deepak V. Kulkarni, Dimitrios V. Rovas, and Daniel A. Tortorelli [*]

1206 West Green St., Mech. & Industr. Engrg. Dept., University of Illinois,
Urbana, IL 61801, USA. {dkulkarn, rovas, dtortore}@uiuc.edu

Summary. Non-conforming meshes are frequently employed in multi-component simulations and adaptive refinement. In this work , we develop a discontinuous Galerkin framework capable of accommodating non-conforming meshes and apply our approach to analyzing the transient heat conduction problem.

1 Introduction

Non-conforming meshes are frequently employed for adaptive solution or simulation of multi-component systems. Even though non-conforming meshes are easy to generate, they require the satisfaction of jump conditions across the non-conforming mesh interface. Several techniques have been developed to enforce these conditions such as mixed methods ([1]), local constraint equation methods ([9], [7]) and mortar methods ([5], [16]).

In this work, we present a Discontinuous Galerkin (DG) framework for accommodating non-conforming meshes. The DG method naturally accommodates jump conditions and has been employed to solve hyperbolic, parabolic and elliptic problems ([6]). For a historical review of DG methods and their applications to elliptic problems refer to [3]. Recently, DG schemes have been applied to enforce jump conditions across non-conforming mesh interfaces such as those encountered in adaptive refinement ([4], [8], [14]). Here, we extend the formulation of [4] to parabolic problems. A benefit of the DG scheme is that it does not introduce constraint equations and their resulting Lagrange multiplier fields, as done in mixed and mortar methods. However, the standard DG formulation leads to a large system of equations due to the presence of “duplicate” nodes.

[*] The first and third author would like to acknowledge the support provided by NSF under grant no. DMR 01-21695

In sections (2) and (3) , we describe our discontinuous Galerkin (DG) formulation and provide an *a priori* analysis. Section (4) presents numerical examples using our formulation. Finally, in section (5) , we draw conclusions and suggest future work.

2 Problem Definition

We consider the following linear heat conduction problem as our representative example of a parabolic equation:

$$\dot{u} - \Delta u = f \quad \text{in } \Omega \times I \tag{1}$$

$$u = 0 \quad \text{on } \partial\Omega_D \times I \tag{2}$$

$$-\nabla u \cdot \mathbf{n} = 0 \quad \text{on } \partial\Omega_N \times I \tag{3}$$

$$u(\cdot, 0) = \tilde{u} \quad \text{in } \Omega \tag{4}$$

where $u(x,t)$ is the scalar temperature field to be computed over the time interval $I = (0,T)$; $\partial\Omega$ defines the boundary of the region Ω which is divided into two complimentary regions, $\partial\Omega_D$, on which homogeneous Dirichlet boundary conditions are specified, and $\partial\Omega_N$, on which homogeneous Neumann boundary conditions are specified; and $\tilde{u}$ is the prescribed initial condition on u . We solve the above partial differential equation by discretizing via a DG finite element method that is based on Nitsche's method ([13]) to weakly enforce Dirichlet boundary conditions. In our DG framework this method enables us to weakly enforce the continuity in u across the non-conforming interface.

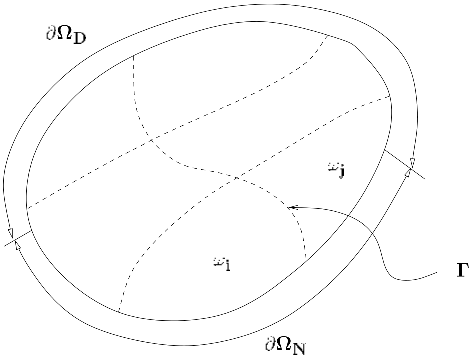


Fig. 1. Domain Partitioning.

The region Ω is divided into n open non-overlapping sub-domains $\omega_1, \ldots, \omega_n$ with boundaries $\partial\omega_1, \ldots, \partial\omega_n$ such that $\Omega = \cup_{i=1}^{n} \omega_i$. Denoting the set of all interior boundaries as Γ , we have:

$$\Gamma = \cup\, e_{ij} \tag{5}$$

where $e_{ij} = \overline{\partial\omega_i} \cap \overline{\partial\omega_j}$ is the interior boundary shared by ω_i and ω_j . On e_{ij} , we define the jump and average operators as

$$[[u]] = u|_{\partial\omega_i} - u|_{\partial\omega_j} \tag{6}$$

$$\langle\langle u\rangle\rangle = \frac{1}{2}(u|_{\partial\omega_i} + u|_{\partial\omega_j}) \tag{7}$$

where $i < j$. In what follows, we describe our DG formulation and relate the final weak statement to the underlying partial differential equations. Standard Galerkin finite element formulations employ test and trial functions that are continuous in Ω . With the DG formulation these functions are no longer continuous across Γ , rather they belong to the following spaces:

$$V_h = \{w_h \in L_2(\Omega) : w_h|_{\omega_i} \in \mathbb{R}^p(\omega_i),\ \text{for } p \geq 1\}$$

with h being the maximal length of the sides of our quasi-uniform triangulation. Assuming sufficiently regular boundary and source data , we require our DG formulation to weakly satisfy the following additional conditions on any interior interface and particularly on Γ , i.e.:

$$[[u_j^{(p_s)}]] = 0 \quad \text{on } \Gamma \times I \tag{8}$$

$$[[\nabla u_j^{(p_s)}]] \cdot \mathbf{n}_i = 0 \quad \text{on } \Gamma \times I \tag{9}$$

Thus, to formulate the weak form of the partial differential equation, we weight equations (1), (3), (8), and (9) by $w_h, w_h, -\langle\langle\nabla w_h\rangle\rangle \cdot \mathbf{n}$, and $\langle\langle w_h\rangle\rangle$ respectively, and integrate over their respective domains to obtain:

$$\begin{aligned}\sum_i \int_{\omega_i} w_h \left(\dot{u}_h - \Delta u_j^{(p_s)} - f\right) \, d\Omega + \int_{\partial\Omega_N} w_h \left(\nabla u_j^{(p_s)} \cdot \mathbf{n}\right) \, ds \\ - \int_\Gamma \left(\langle\langle\nabla w_h\rangle\rangle \cdot \mathbf{n}\right) [[u_j^{(p_s)}]] \, ds + \int_\Gamma \langle\langle w_h\rangle\rangle [[\nabla u_j^{(p_s)}]] \cdot \mathbf{n} \, ds = 0\end{aligned} \tag{10}$$

This weak form could lead to instabilities ([4]), so , we stabilize our formulation by augmenting the above with the following penalty function:

$$P = \int_\Gamma \frac{\eta}{h} [[w_h]][[u_j^{(p_s)}]] \, ds \tag{11}$$

which is related to the jump condition on u^h (cf. equation (8)). Integration by parts, and the identity $[[a\,b]] = [[a]]\langle\langle b\rangle\rangle + \langle\langle a\rangle\rangle[[b]]$ yields the DG problem statement: Find $u_j^{(p_s)} : \Omega \times I \to \mathbb{R}$ where $u_j^{(p_s)}(t) \in V_h$ such that

$$(\dot{u}_h, w_h) + a(u_j^{(p_s)}, w_h) = (f, w_h) \quad \forall\, (w_h, t) \in V_h \times I \tag{12}$$

$$\left(u_j^{(p_s)}(0) - \tilde{u}, w_h\right) = 0 \quad \forall\, w_h \in V_h \tag{13}$$

where $(u_j^{(p_s)}, w_h)$ is the standard L_2 inner product over Ω and

$$\begin{aligned}a(u_j^{(p_s)}, w_h) = &\sum_i \int_{\omega_i} \nabla w_h \cdot \nabla u_j^{(p_s)} \, d\Omega - \int_\Gamma [[w_h]]\langle\langle\nabla u_j^{(p_s)}\rangle\rangle \cdot \mathbf{n} \, ds \\ &- \int_\Gamma [[u_j^{(p_s)}]]\langle\langle\nabla w_h\rangle\rangle \cdot \mathbf{n} \, ds + \int_\Gamma \frac{\eta}{h}[[w_h]][[u_j^{(p_s)}]] \, ds\end{aligned} \tag{14}$$

For the steady-state case, i.e. $\dot{u} = 0$, our weak form is the same as that of [4], and [3], and hence their stability and optimal convergence proofs hold. Our formulation leads to a sparse, symmetric system of equations thereby maintaining the computational efficiency of the regular finite element approach. Though, [2], [15], have formulated two different *non-symmetric* weak forms for solving nonlinear parabolic equations, to our knowledge there is no literature where our proposed DG scheme has been used for parabolic equations.

3 A priori analysis

In this section , we highlight the important results of our *a priori* analysis. One can refer to [12] for a detailed description of the *a priori* analysis. Through our *a priori* analysis , we demonstrate that our methodology is consistent, stable and converges at a rate similar to that of a standard Galerkin scheme. The following lemma, an extension of that in [4] proves consistency i.e. the exact solution to the partial differential equation (1)-(4) satisfies the DG weak form (12).

Lemma 1. *If u is the solution to equations (1)-(4) then it also solves (12)*

Proof. Since u solves equations (1)-(4) $[[u]] = 0$, $\langle\langle u\rangle\rangle = u$, $\langle\langle \nabla u\rangle\rangle \cdot \mathbf{n} = \nabla u \cdot \mathbf{n}$. Thus

$$\begin{aligned}(f, w_h) - a(u, w_h) &= (f, w_h) - (\dot{u}, w_h) - (\nabla u, \nabla w_h) + \int_{\Gamma} [[w_h]] \nabla u \cdot \mathbf{n}\, ds \\ &= (f - \dot{u} + \Delta u, w_h) \\ &= 0 \end{aligned} \tag{15}$$

which proves the lemma. □

♯

For our error analysis , we introduce $\mathcal{H}^r$ the standard Hilbert space with its associated norm $||\cdot||_{\mathcal{H}^r(\Omega)}$. Since the DG formulation allows for a discontinuous field across Γ , we introduce the following norm that accounts for the discontinuity in $u_j^{(p_s)}$:

$$\left|\left|\left|u_j^{(p_s)}\right|\right|\right|^2 = \left|\left|\nabla u_j^{(p_s)}\right|\right|^2_{L_2(\Omega)} + \left|\left|h^{1/2} \langle\langle \nabla u_j^{(p_s)}\rangle\rangle \cdot \mathbf{n}\right|\right|^2_{L_2(\Gamma)} + \left|\left|h^{-1/2} [[u_j^{(p_s)}]]\right|\right|^2_{L_2(\Gamma)} \tag{16}$$

This norm is equivalent to a $\mathcal{H}^1$ norm on a broken space. We now state without proof our *a priori* error estimate:

Theorem 1. *If $u_j^{(p_s)}$ is the solution of (12) and u is the solution of (1)-(4), then*

$$\begin{aligned}\left|\left|\left|u_j^{(p_s)} - u\right|\right|\right| \leq & C \left|\left|\left|u_j^{(p_s)}(0) - \tilde{u}\right|\right|\right| \\ & + Ch^{r-1} \left(||\tilde{u}||_{\mathcal{H}^r(\Omega)} + ||u(t)||_{\mathcal{H}^r(\Omega)} + C_2 \left(\int_0^t ||\dot{u}||^2_{\mathcal{H}^r(\Omega)}\, dz \right)^{1/2} \right) \\ & \text{for} \quad 2 \leq r \leq p+1 \quad u \in \mathcal{H}^r(\Omega) \cap \mathcal{H}^1_0(\Omega) \end{aligned} \tag{17}$$

The first term on the right side of the inequality accounts for the error in projecting the initial condition onto the finite element space; weak satisfaction of the initial condition via equation (13) would ensure that this term converges at an optimal rate of h^{r-1}. The second term shows that our error converges at the rate of h^{r-1} which represents an optimal order of convergence in the $|||\cdot|||$ norm.

The following lemma (also stated without proof), an extension of that in [10] shows the stability of our DG formulation i.e. under suitable assumptions on the smoothness of the initial condition the DG solution remains bounded and decays over time.

Lemma 2 (Stability). *Let $u_j^{(p_s)}$ be the solution to (12) with $f = 0$ then it satisfies the property*

$$\left|\left|u_j^{(p_s)}(t)\right|\right|_{L_2(\Omega)} \leq \left|\left|u_j^{(p_s)}(0)\right|\right|_{L_2(\Omega)} \leq ||\tilde{u}||_{L_2(\Omega)} \quad \forall\, t \in I \tag{18}$$

This lemma which uses coercivity of the bilinear operator $a(u_j^{(p_s)}, w_h)$ proves the stability of the DG formulation for the case when $f = 0$.

4 Numerical Results

To validate the DG formulation and the *a priori* analysis , we consider equation (1) with initial condition $\tilde{u} = \sin(x)$ over a 1-D domain $\Omega = (0, \pi)$. The analytical solution for this problem is according to Kreyzig [11]

$$u(x,t) = e^{-t}\sin(x) \tag{19}$$

The numerical examples employ linear elements and a backward-Euler time stepping scheme. Each element is considered to be a separate sub-domain ω_i and the interface

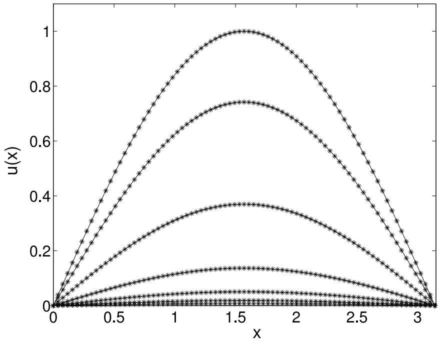


Fig. 2. $u(x)$ at various time steps.

Γ is the collection of all the end points of the elements. In Fig. (2) , we plot $u(x,t)$ at various instants in time. As expected u decays with increasing time. Fig. 3(a), Fig. 3(b) illustrate the error norm (cf. equation (16)) versus the element size h. Plot 3(a) which is obtained using a time step of $\Delta t = 0.0001$, and Fig. 3(b) obtained by varying the time step as $\Delta t \propto h^2$ show the optimal order of error convergence. To further validate the DG formulation , we repeat the above example using the hat function initial condition

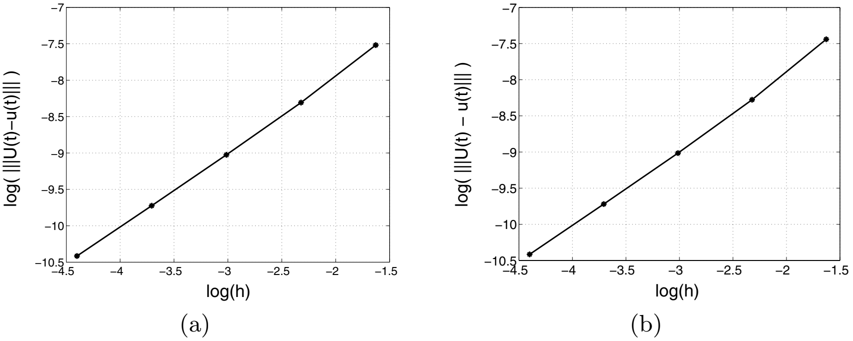


Fig. 3. Error norm with (a) $\Delta t = 0.0001$ and (b) with $\Delta t \propto h^2$.

$$\tilde{u} = \begin{cases} x, & \text{if} \quad 0 < x \leq \pi/2\,; \\ \pi - x & \text{if} \quad \pi/2 < x < \pi\,. \end{cases} \tag{20}$$

for which the analytical solution is [11]

$$u(x,t) = \sum_{n=1}^{\infty} B_n \sin(nx)\, e^{-n^2 t} \tag{21}$$

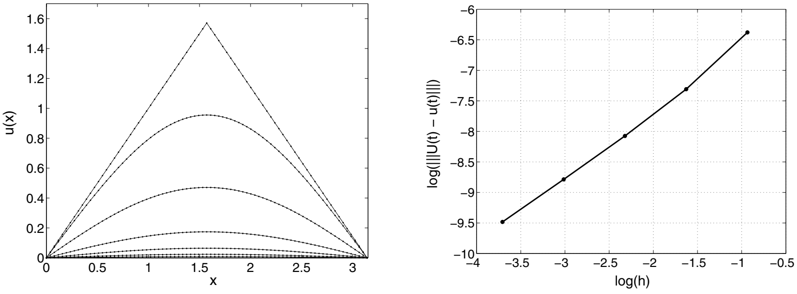
where

$$B_n = \begin{cases} \dfrac{4}{n^2\pi} & \text{for} \quad n = 1, 5, 9, \ldots \\ -\dfrac{4}{n^2\pi} & \text{for} \quad n = 3, 7, 11, \ldots \end{cases} \tag{22}$$

In Fig. 4(a) , we plot the evolution of u over time. Though the initial condition in this example problem is not as smooth as in the previous example, Fig. 4(b) shows that we still obtain optimal convergence rate.

5 Conclusions

A DG formulation for solving parabolic equations on non-conforming meshes has been developed. The formulation leads to a symmetric, sparse system and does not



(a) (b)

Fig. 4. Simulation with hat function as initial condition (a) $u(x)$ at various time steps (b) Error with $\Delta t = 0.001$.

involve constraint equations or Lagrange multiplier fields like the mortar method. The *a priori* analysis of the method shows that the method is consistent, stable and demonstrates optimal order of convergence. Numerical results validate our analysis. We believe the method has applications to efficient multi-component simulation and adaptive refinement. Currently , we are applying this scheme to adaptively refine interface evolution problems ([12]).

References

1. T. Arbogast and I. Yotov, *Non-mortar mixed finite element method for elliptic problems on non-matching multiblock grids*, Comput. Meth. Appl. Mech. Engrg., 149 (1997), pp. 255–265.
2. D. N. Arnold, *An interior penalty finite element method with discontinuous elements*, SIAM J. Numer. Anal., 19 (1982), pp. 742–760.
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 (2002), pp. 1749–1779.
4. R. Becker and P. Hansbo, *A finite element method for domain decomposition with non-matching grids*, Tech. Rep. 3613, INRIA, January 1999.
5. 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 application, H. Brezis and J. L. Lions, eds., Pitman, 1989.
6. B. Cockburn, G. E. Karniadakis, and C.-W. Shu, eds., *Discontinuous Galerkin Methods*, vol. 11 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2000.

7. L. Demkowicz, J. T. Oden, W. Rachowicz, and O. Hardy, *Toward a universal h-p adaptive finite element strategy, part 1. Constrained approximation and data structure*, Comput. Meth. Appl. Mech. Engrg., 77 (1989), pp. 79–112.
8. E. G. D. do Carmo and A. V. C. Duarte, *A discontinuous finite-element based domain decomposition method*, Comput. Meth. Appl. Mech. Engrg., (2000), pp. 825–843.
9. B. Guo and I. Babuška, *The h-p version of the finite element method, part 1, and the h-p version of the finite element method, part 2*, Computational Mechanics, 1 (1986), pp. 21–41 203–220.
10. C. Johnson, *Numerical Solutions of Partial Differential Equations by the Finite Element Method*, Cambridge University Press, Cambridge, 1987.
11. E. Kreyszig, *Advanced engineering mathematics*, John Wiley & Sons (Asia), fifth ed., 1993.
12. D. V. Kulkarni, D. V. Rovas, and D. A. Tortorelli, *Discontinuous Galerkin framework for adaptive solution of interface evolution problems*, Internat. J. Numer. Methods Engrg., (2006). Accepted.
13. J. A. Nitsche, *Über ein variationsprinzip zur lösung von Dirichlet-problemen bei verwendung von teilräumen, die keinen randbedingungen unterworfen sind*, Abh. Math. Univ. Hamburg, 36 (1970), pp. 9–15.
14. I. Perugia and D. Schötzau, *On the coupling of local discontinuous Galerkin and conforming finite element methods*, J. Sci. Comput., (2001), pp. 411–433.
15. B. M. Rivière and M. F. Wheeler, *A discontinuous Galerkin method applied to nonlinear parabolic equations*, in Discontinuous Galerkin methods: Theory, Computation and Applications, B. Cockburn, G. E. Karniadakis, and C.-W. Shu, eds., vol. 11 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, February 2000, pp. 231–244.
16. B. I. Wohlmuth, *A comparison of dual Lagrange multiplier spaces for mortar finite element discretizations*, Math. Model. Numer. Anal., 36 (2002), pp. 995–1012.

On a Parallel Time-domain Method for the Nonlinear Black-Scholes Equation

Choi-Hong Lai [1] , Diane Crane [2] , and Alan Davies [2]

[1] School of Computing and Mathematical Sciences, University of Greenwich, Old Royal Naval College, Greenwich, London SE10 9LS, UK. C.H.Lai@gre.ac.uk
[2] Department of Physics, Astronomy, and Mathematics, University of Hertfordshire, Hatfield Campus, Herts AL10 9AB, UK.
{D.Crann,A.J.Davies}@herts.ac.uk

1 Introduction

A parallel time-domain algorithm is described for the time-dependent nonlinear Black-Scholes equation, which may be used to build financial analysis tools to help traders making rapid and systematic evaluation of buy/sell contracts. The algorithm is particularly suitable for problems that do not require fine details at each intermediate time step, and hence the method applies well for the present problem.

The method relies on a Laplace transform technique applied to the Black-Scholes equation and generates subproblems that can be executed in a parallel/distributed computing environment. These subproblems are thus solved independently without subproblem communication. Early studies of the scalability of the algorithm for linear Black-Scholes equation may be found in [3] and [4]. This paper extends the previous work to nonlinear Black-Scholes equation. Two linearization methods are presented, one based on the updating of nonlinear coefficients within an iterative loop and the other based on a Newton's method. A numerical inverse [6, 7] of the approximate solution is used to retrieve the final solution of the nonlinear Black-Scholes equation. Numerical tests are performed to demonstrate the viability of the algorithm. The efficiency of the algorithm is also studied.

This paper concludes with a discussion on an extension of the present Laplace transform technique to a parallel time-domain algorithm in order to obtain details of physical quantities at intermediate finer time steps.

2 A Nonlinear Black-Scholes Model

Let $v(S,t)$ denote the value of an option, where S is the current value of the underlying asset and t is the time. The value of the option relates to the current value of the underlying asset via the Black-Scholes equation:

$$\frac{\partial v}{\partial t} + \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 v}{\partial S^2} + rS - rv = 0 \in \Omega^+ \times [T,0) \tag{1}$$

where $\Omega^+ = \{S : S \geq 0\}$. The stochastic background of the equation is not discussed in this paper, and readers who are interested should consult [8].

Only European options are considered in this paper. This means that the holder of the option may execute at expiry a prescribed asset, known as the underlying asset, for a prescribed amount, known as the strike price. There are two different types of option, namely the call option and the put option. At expiry, the holder of the call option has the right to buy the underlying asset and the holder of the put option has the right to sell the underlying asset. For a European put option with strike price k and expiry date T , it is sensible to impose the boundary condition $v(0,t) = ke^{-r(T-t)}$, $v(L,t) = 0$, where L is usually a large value. At expiry, if $S < k$ then one should exercise the call option, i.e. handing over an amount k to obtain an asset with S . However, if $S > k$ at expiry, then one should not exercise the option because of the loss $k - S$. Therefore the final condition $v(S,T) = \max\{k - S, 0\}$ needs to be imposed. The solution v for $t < T$ is required.

Since (1) is a backward equation, it needs to be transformed to a forward equation by using $\tau = T - t$, which leads to,

$$\frac{\partial V}{\partial \tau} = \frac{1}{2}\sigma^2 S^2 \frac{\partial^2 V}{\partial S^2} + rS - rV \in \Omega^+ \times (0,T] \tag{2}$$

subject to the initial condition $V(S,0) = \max\{k - S, 0\}$ and boundary conditions $V(0,\tau) = ke^{-r\tau}$, $V(L,\tau) = 0$. A field method, such as a finite volume method, is of interest for two reasons. First, there are many examples of multi-factor models showing that a reduction of the time dependent or nonlinear coefficient to a constant coefficient heat is impossible. Hence analytic form of solutions cannot be found. Second, financial modelling typically requires large number of simulations and solutions at intermediate time steps are usually not of interest. Efficiency of the numerical algorithm is very important in order to make evaluation and decision before the agreement of a contract is reached. Ideally one would like to use an algorithm which can be completely distributed onto a number of processors with only minimal communications between processors.

Very often, over a short period of time the interest rate, r , is fixed while the volatility, σ , is varying. The volatility may be a function of the transaction costs [5], the second derivative of the option value [1], or, in some cases, the solution of a nonlinear initial value problem [5]. In order to develop the nonlinear solver in this section, the volatility $\sigma = \sigma_0\sqrt{1+a}$ proposed in [2] is used, where a is the proportional transaction cost scaled by σ_0 and the transaction time. Very often the transaction cost is related to the option value and follows a Gaussian distribution. In order to demonstrate the time-domain parallel algorithm for nonlinear problems, a sine function is used in the subsequent tests to produce the effect of a pulse-like distribution instead of a Gaussian distribution, i.e. $a = \sin(\frac{V\pi}{k})$ where k is the strike price.

3 Reference Solutions Using a Temporal Integration

The forward Black-Scholes equation given by (2) is written as

$$\frac{\partial V}{\partial \tau} = A(V)\frac{\partial^2 V}{\partial S^2} + rS - rV = 0 \in \Omega^+ \times (0,T] \tag{3}$$

where $A(V) = \frac{1}{2}\sigma(V)^2 S^2$. In order to obtain a reference solution for (3) a linearisation method combined with a temporal integration may be applied. The coefficient A is computed by using an approximation $\bar{V}$, which is updated in every step of a nonlinear iterative update process. Each step of the nonlinear iterative update process involves a numerical solution to the equation

$$\frac{\partial V}{\partial \tau} = A(\bar{V})\frac{\partial^2 V}{\partial S^2} + rS - rV = 0 \in \Omega^+ \times (t_i, t_{i+1}] \tag{4}$$

defined in the time interval $\tau \in (t_i, t_{i+1}]$. Let $V^{(n)}(S, t_{i+1})$ and $V^{(n)}(S, t_i)$ be the numerical solutions of (3) at $\tau = t_{i+1}$ and $\tau = t_i$ respectively. The nonlinear iterative update process to obtain the numerical solution $V^{(n)}(S, t_{i+1})$, using $V^{(n)}(S, t_i)$ as the initial approximation to $\bar{V}$, is described in the algorithm below.

```
Algorithm R: Obtain a reference solution for (3).
    do i = 0,1,2,...
    $t_i = i\delta\tau$ ;
    Initial approximation:- $V^{(0)}(S, t_{i+1}) := V^{(n)}(S, t_i)$ ; $k := 0$ ;
    Iterate
        $k := k + 1$ ;
        $\bar{V} := V^{(k-1)}(S, t_{i+1})$ ;
        Compute $A(\bar{V})$ ;
        $V^{(k)}(S, t_{i+1})$ := Apply Euler's method to (4);
    Until $\|V^{(k)}(S, t_{i+1}) - V^{(k-1)}(S, t_{i+1})\| < \varepsilon$
    $n := k$ ;
    end-do
```

4 The Parallel Time-Domain Method

Let

$$l(V) = \int_0^\infty e^{-\lambda\tau} V(S, \tau) d\tau = U(\lambda; S)$$

be the Laplace transform of the function $V(S.\tau)$. Application of the Laplace transform [7] to (4), now being defined in $\Omega^+ \times (T_i, T_{i+1}]$, leads to

$$A(\bar{V})\frac{d^2 U}{dS^2} + rs\frac{dU}{dS} - (r + \lambda)U = -V(S, T_i) \in \Omega^+ \tag{5}$$

where $U = U(\lambda; S)$ defined in the Laplace space. Here $\lambda \in \{\lambda_j\}$ is a finite set of transformation parameter defined by

$$\lambda_j = j\frac{\ln 2}{T_{i+1} - T_i}.\ j = 1, 2, ..., m \tag{6}$$

where m should be chosen as an even number [6]. Therefore the problem defined in (4) is converted to m independent parametric boundary value problems as described by (5), and these problems may be distributed and solved independently in a distributed environment.

In order to retrieve $V(S, T_{i+1})$, we use the approximate inverse Laplace transform due to Stehfest [6] given by

$$V(S, T_{i+1}) \approx \frac{\ln 2}{T_{i+1} - T_i} \sum_{j=1}^{m} w_j U(\lambda_j; S) \tag{7}$$

where

$$w_j = (-1)^{m/2+j} \sum_{k=(1+j)/2}^{\min(j,m/2)} \frac{k^{m/2}(2k)!}{(m/2-k)!k!(k-1)!(j-k)!(2k-j)!}$$

is known as the weighting factor. We have selected Stehfest's method because of previous experience with the method used for linear problems [3, 4] and a wish to investigate the application of the inverse method to nonlinear problems.

A nonlinear iterative update process is required to update $\bar{V}$ and to obtain the numerical solution $V^{(n)}(S, T_{i+1})$, using $V^{(n)}(S, T_i)$ as the initial approximation to $\bar{V}$, and it is described in Algorithm P1:

```
Algorithm P1: Parallel algorithm 1 for (3).
    do i = 0,1,2,...
    $T_i = i\Delta\tau$ ;
    Initial approximation:- $V^{(0)}(S, T_{i+1}) := V^{(n)}(S, T_i)$ ; $k := 0$ ;
    Iterate
        $k := k + 1$ ; $\bar{V} := V^{(k-1)}(S, T_{i+1})$ ; Compute $A(\bar{V})$ ;
        Parallel for $j := 1$ to $m(i)$
            Solve (5) for $U(\lambda_j; S)$ ;
        End parallel for
        Compute $V^{(k)}(S, T_{i+1})$ using inverse Laplace transform (7);
    Until $\|V^{(k)}(S, T_{i+1}) - V^{(k-1)}(S, T_{i+1})\| < \varepsilon$
    $n := k$ ;
    end-do
```

Here $m(i)$ is the number of transformation parameters and $T_i = \Delta\tau$. In order to solve (5) for $U(\lambda_j; S)$, one can employ the finite volume technique as the one used in Section 3. In essence the actual implementation does not require different values of $m(i)$ for many problems, and the results shown in this paper use the same number of transformation parameters, denoted as $\bar{m}$, for different values of

i during the outer iteration loop. Note that in this case $\Delta\tau$ can be chosen to be much greater than $\delta\tau$ because the fine details of $V(S,\tau)$ at each time step of a temporal integration is not required in the present example.

5 Newton Linearisation

Alternatively, a small perturbation may be applied to (2), defined in the time interval $\tau \in (T_i, T_{i+1}]$, which leads to

$$\{\frac{\partial}{\partial\tau} - (A\prime(V)\frac{\partial^2 V}{\partial S^2} + A(V)\frac{\partial^2}{\partial S^2} + rS\frac{\partial}{\partial S} - r)\}\delta V$$

$$= -\{\frac{\partial V}{\partial\tau} - (A(V)\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV)\} \tag{8}$$

where δV is a small incremental change of V . Application of the Laplace transform to (8), defined in the interval $\tau \in (T_i, T_{i+1}]$, results in

$$l(\delta V) - \delta V(S,T_i) - (A\prime(V)\frac{\partial^2 V}{\partial S^2} + A(V)\frac{\partial^2}{\partial S^2} + rS\frac{\partial}{\partial S} - r)\}l(\delta V)$$

$$= -l(V) - V(S,T_i) - (A(V)\frac{\partial^2 V}{\partial S^2} + rS\frac{\partial V}{\partial S} - rV)\} \tag{9}$$

The method requires the numerical solution $l(\delta V^{(n)}(S,T_{i+1}))$ using $V^{(n)}(S,T_i)$ as the initial approximation to $V^{(0)}(S,T_{i+1})$ and is described in Algorithm P2:

```
Algorithm P2: Parallel algorithm 2 for (3).
    do i = 0,1,2,...
    $T_i = i\Delta\tau$ ;
    Initial approximation:- $V^{(0)}(S,T_{i+1}) := V^{(n)}(S,t_i)$ ; $k := 0$ ;
    Iterate
        $k := k+1$ ; $\bar{V} := V^{(k-1)}(S,T_{i+1})$ ;
        Compute $A(\bar{V})$ ; Compute $A'(\bar{V})$ ; Compute $A'(\bar{V})\frac{\partial^2\bar{V}}{\partial S^2}$ ;
        Compute $-l(\bar{V}) - V(S,T_i) - (A(\bar{V})\frac{\partial^2\bar{V}}{\partial S^2} + rS\frac{\partial\bar{V}}{\partial S} - r\bar{V})\}$ ;
        Parallel for $j := 1$ to $m(i)$
            Solve (9) for $l(\delta V^{(k)}(S,T_{i+1}))$ ;
        End parallel for
        Compute $\delta V^{(k)}(S,T_{i+1})$ using inverse Laplace transform (7);
        $V^{(k)}(S,T_{i+1} := \bar{V} + \delta V^{(k)}(S,T_{i+1})$ ;
    Until $\|\delta V^{(k)}(S,T_{i+1})\| < \varepsilon$
    $n := k$ ;
    end-do
```

6 Numerical Examples

The problem of European put option is solved up to the expiry date $T = 0.25$ at the strike price $k = 100$. The volatility σ is chosen as the function described in Section 2 and the parameters σ_0 and r are chosen to be 0.4 and 0.5 respectively. A second order finite volume method is applied to each parametric equation as given by (5) or (9). The mesh size is chosen to be $h = 320/2^9$.

A sequential computational environment is used in the tests. The approximations to $V(S,T)$ obtained by means of algorithms P1 and P2 are denoted as V_{P1} and V_{P2} respectively. Using $\Delta\tau = \frac{T}{10}, \frac{T}{20}, \frac{T}{30}, \frac{T}{40}$, the number of outer iterations or time steps required for algorithms P1 and P2 are 10, 20, 40, and 80 respectively.

The above two parallel time-domain algorithms are compared with the reference solution obtained by means of algorithm R with $\delta\tau = 1/365$, i.e. 1 day, in conjunction with the second order finite volume scheme applied along the spatial axis S . This discretisation leads to a number of tri-diagonal systems of equations due to the linearisation step at every time step, which may be solved by a direct method. The numerical solution $V(S,T)$ obtained by this temporal integration is denoted as V_R . The stopping criterion used in the linearization step is chosen as $\varepsilon = 10^{-5}$.

In order to examine the efficiency of the parallel time-domain algorithms, the computational work required for solving a tri-diagonal system of equations results from a chosen mesh size is counted as one work unit. The total sequential work unit is obtained by multiplying the total number of work unit to $\bar{m}$, and the total parallel work unit is simply the total work unit plus overhead due to the calculation of inverse Laplace transform and communication.

Discrepancies in the solutions, i.e. $\|V_R - V_{P1}\|$ and $\|V_R - V_{P2}\|$ using various $\Delta\tau$, are presented in Fig. 1 and 2. In general the discrepancy levels off when $m \geq 8$, which suggests that the use of more terms in the inverse Laplace transform at a fixed value of $\Delta\tau$ has no effect on the accuracy. On the other hand smaller $\Delta\tau$ produces smaller discrepancy at the expense of requiring more work unit as recorded in Table 1. Furthermore the number of work units required using algorithm P2 is less than that of algorithm P1, and there is no sudden increase of work when $\bar{m} = 12$.

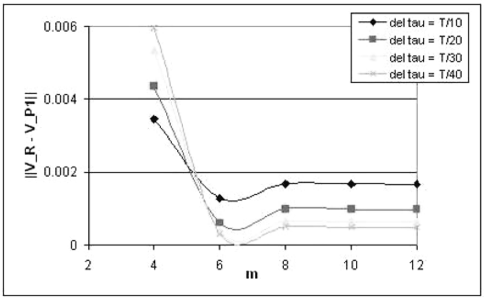


Fig. 1. Discrepancies of solutions: $\|V_R - V_{P1}\|$.

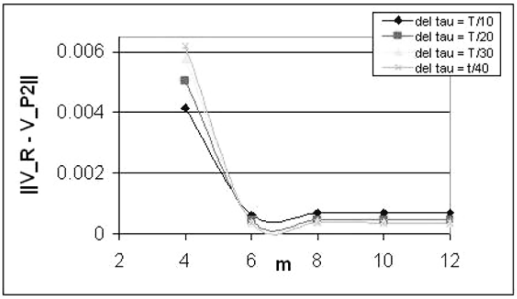


Fig. 2. Discrepancies of solutions: $\|V_R - V_{P2}\|$.

Table 1. Work unit comparison (V_R requires 246 work units).

$\bar{m}$ $\Delta\tau$	4	6	8	10	12
	Algorithm P1				
$9.125\delta\tau$	58	58	58	58	180
$4.5626\delta\tau$	103	103	103	103	123
$2.28125\delta\tau$	177	177	177	177	186
$1.140625\delta\tau$	326	326	326	326	327
	Algorithm P2				
$9.125\delta\tau$	43	43	43	43	43
$4.5626\delta\tau$	83	70	71	71	71
$2.28125\delta\tau$	134	126	126	126	126
$1.140625\delta\tau$	249	245	220	214	213

7 Conclusion

Two linearisation methods were used in conjunction with the Laplace transform method for non-linear Black-Scholes models. Work unit counts of the numerical experiments suggest that the present techniques have advantages in solving nonlinear option pricing problems using parallel or distributed computing environments. One such advantage is the use of a larger time step, i.e. $\Delta\tau$, when the fine details at intermediate time steps of the time interval (T_i, T_{i+1}) are not required. Parallelisation is introduced by solving in parallel a number of parametric problems, each of which defines in the interval (T_i, T_{i+1}) , i =1,2,..., $T/\delta\tau$, in the Laplace space. Note that as $\Delta\tau$ approaches $\delta\tau$ the tranform into Laplace space does not offer any advantages as can be seen from the results in Table 1. Therefore fine details on a fine time step should not be computed by means of Laplace transform method. Instead fine details within the time interval (T_i, T_{i+1}) , for all values of i , may be obtained in parallel using a temporal integration method. Effectively the present algorithm provides initial conditions for every interval (T_i, T_{i+1}) , i =1,2,..., $T/\delta\tau$. As a result fine details of the time interval (T_i, T_{i+1}) are decoupled from other time intervals and may be obtained independently with a smaller time-step, say $\delta\tau$.

References

1. M. Avellaneda and A. Parás, *Dynamic hedging portfolios for derivative securities in the presence of large transaction costs*, Applied Mathematical Finance, (1994), pp. 165–193.
2. P. P. Boyle and T. Vorst, *Option replication in discrete time with transaction costs*, J. Finance, (1992), pp. 271–293.
3. D. Crann, *The Laplace transform: Numerical inversion for computational methods*, Tech. Rep. 21, University of Hertfordshire, UK, 1996.
4. D. Crann, A. J. Davies, C.-H. Lai, and S. H. Leong, *Time domain decomposition for European options in financial modelling*, in Tenth International Conference on Domain Decomposition Methods, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, 1998.
5. H. M. Soner and G. Barles, *Option pricing with transaction costs and a nonlinear Black-Scholes equation*, Finance and Stochastics, 2 (1998), pp. 369–397.
6. H. Stehfest, *Numerical inversion of Laplace transforms*, Comm. ACM, (1970), pp. 47–49.
7. D. V. Widder, *The Laplace Transform*, Princeton University Press, Princeton, NJ, 1946.
8. P. Wilmott, S. Howison, and J. Dewynne, *The Mathematics of Financial Derivatives*, Cambridge University Press, 1995.

Domain-decomposition Based $\mathcal{H}$-LU Preconditioners

Sabine Le Borne [1], Lars Grasedyck [2], and Ronald Kriemann [2]

[1] Tennessee Technological University, Cookeville, TN 38505, USA.
sleborne@tntech.edu. This material is based upon work supported by the US Department of Energy under Grant No. DE-FG02-04ER25649.

[2] Max-Planck-Institute for Mathematics in the Sciences, Leipzig, Germany.
lgr,rok@mis.mpg.de

1 Introduction

Hierarchical matrices (in short: $\mathcal{H}$-matrices) have first been introduced in 1998 [7] and since then have entered into a wide range of applications. They provide a format for the data-sparse representation of fully populated matrices. The key idea is to approximate certain subblocks of a matrix by low rank approximations which are represented by a product of two low rank matrices: Let $A \in R^{n\times n}$ with rank(A)= k and $k \ll n$. Then there exist matrices $B, C \in R^{n\times k}$ such that $A = BC^T$. Whereas A has n^2 entries, B and C together have $2kn$ entries which results in significant savings in storage if $k \ll n$. A new $\mathcal{H}$-matrix arithmetic has been developed which allows (approximate) matrix-vector multiplication and matrix-matrix operations such as addition, multiplication and inversion of matrices in this format in nearly optimal complexity $\mathcal{O}(n \log^\alpha n)$ [5].

In finite element methods, the stiffness matrix is sparse but its inverse is fully populated and can be approximated by an $\mathcal{H}$-matrix. Such an approximate inverse may then be used as a preconditioner in iterative methods [1]. Even though the complexity of the $\mathcal{H}$-matrix inversion is nearly optimal, there are relatively large constants involved in these complexity estimates which in the past have prevented $\mathcal{H}$-matrix based preconditioners to be competitive with other state-of-art methods. The following recent developments have addressed this drawback successfully and allowed $\mathcal{H}$-matrix based preconditioners to be competitive also in the FEM context: 1) a weak admissibility condition [10] yielding coarser block structures and therefore reduced constants in the complexity estimates, 2) the introduction of an $\mathcal{H}$-LU decomposition [13, 2] which is computed significantly faster than an approximate inverse and provides an (in general more accurate) preconditioner, and 3) the parallelization of $\mathcal{H}$-matrix arithmetic [12]. In this paper, we will add another improvement to these three components: We will introduce (recursive) domain decompositions with an interior boundary, also known as *nested dissection*, into the construction of the index cluster tree of an $\mathcal{H}$-matrix. The new clustering algorithm will yield a block structure in which large subblocks are zero and remain zero in a subsequent LU-factorization, As a result, the constants in the (nearly optimal) storage and work complexities will be significantly smaller than for the standard $\mathcal{H}$-matrix setting. Furthermore, the $\mathcal{H}$-LU factorization is parallelizable. We will then construct preconditioners based on such an incomplete $\mathcal{H}$-LU-decomposition to accelerate the iterative solution of linear systems of equations. We will illustrate our new preconditioner with some numerical examples for convection-dominated partial differential equations.

The remainder of this paper is structured as follows: Section 2 is devoted to preliminaries and will provide a review of the nested dissection method as well as a brief introduction to $\mathcal{H}$-matrices. Section 3 introduces the new clustering algorithm. In Section 4, we will conclude with some numerical results for convection-dominated problems.

2 Preliminaries: Nested dissection and $\mathcal{H}$-matrices

2.1 A review of nested dissection

Most direct methods for sparse linear systems perform an LU factorization of the original matrix after some reordering of the indices in order to reduce fill-ins. One such popular reordering method is the so-called *nested dissection* which exploits the concept of separation. The idea of nested dissection has been introduced more than 30 years ago [4] and since then attracted considerable attention (see, e.g., [3, 11] and the references therein). The main idea is to separate a (matrix) graph into three parts, two of which have no coupling between each other. The third one, referred to as an interior boundary or separator, contains couplings with (possibly both of) the other two parts. The nodes of the separator are numbered last. This process is then repeated recursively in each subgraph. An illustration of the resulting sparsity pattern is shown in Figure 1 for the first decomposition step. In domain-decomposition terminology, we recursively subdivide our domain into two disjoint subdomains and an interior boundary.

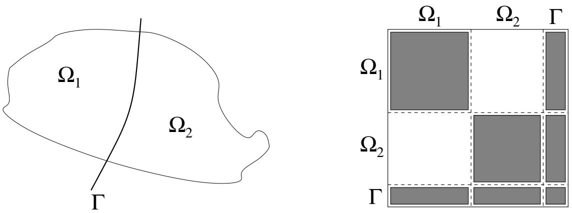


Fig. 1. Nested dissection and resulting matrix sparsity structure.

A favorable property of such an ordering is that a subsequent LU factorization maintains a major part of this sparsity structure, i.e., there occurs no fill-in in the large, off-diagonal zero matrix blocks. In fact, in the case of regular two-dimensional grids, the computational complexity amounts to $\mathcal{O}(n^{1.5})$ for a matrix $A \in R^{n \times n}$. In order to obtain a (nearly) optimal complexity, we approximate all nonzero, off-diagonal blocks in $\mathcal{H}$-matrix representation and compute them using $\mathcal{H}$-matrix arithmetic. The blocks along the diagonal and the corresponding LU factorizations will be stored as full matrices.

2.2 A brief introduction to $\mathcal{H}$-matrices

An $\mathcal{H}$-matrix approximation to a given (full) matrix is obtained by replacing certain blocks of the matrix by matrices of low rank, stored in so-called Rk-format defined in Definition 3. The formal definition of an $\mathcal{H}$-matrix depends on appropriate hierarchical partitionings of the index set which is organized in a cluster tree. Instead of a fixed partitioning, such a tree provides a hierarchy of partitions leading to a more flexible structure.

Definition 1 (Cluster tree). *Let I be a finite index set and let $T_I = (V, E)$ be a tree with vertex set V and edge set E. For a vertex $v \in V$, we define the set of sons of v as $S(v) := \{w \in V \mid (v, w) \in E\}$. Correspondingly, the father of a non-root vertex v is defined as the unique vertex $F(v)$ s.t. $(F(v), v) \in E$. The tree T_I is called a cluster tree of I if its vertices consist of subsets of I and satisfy the following conditions:*

1. *$I \in V$ is the root of T_I and $v \subset I$, $v \neq \emptyset$, for all $v \in V$.*
2. *For all $v \in V$ there either holds $S(v) = \emptyset$ or $v = \dot{\bigcup}_{w \in S(v)} w$.*

In the following we identify V and T_I, i.e., we write $v \in T_I$ instead of $v \in V$. The nodes $v \in V$ are called clusters. The nodes with no successors are called leaves and define the set $\mathcal{L}(T) = \{v \in T \mid S(v) = \emptyset\}$.

Several approaches to construct a cluster tree have been suggested in previous papers [7, 9, 5, 6]. All these constructions considered the cardinalities and/or the geometries of the resulting clusters. These constructions have in common that a cluster is either not subdivided (a *leaf*) or has exactly two sons. In Section 3, we will derive a new clustering algorithm in which clusters may have up to three sons, and thus obtain completely new cluster trees and subsequent partitions.

A hierarchy of block partitionings of the product index set $I \times I$ is based upon a cluster tree T_I and is organized in a block cluster tree $T_{I \times I}$:

Definition 2 (Block cluster tree). *Let T_I be a cluster tree of the index set I. A cluster tree $T_{I \times I}$ is called a block cluster tree (based upon T_I) if for all $v \in T_{I \times I}$ there exist $t, s \in T_I$ such that $v = t \times s$. The nodes $v \in T_{I \times I}$ are called block clusters.*

A block cluster tree may be constructed from a given cluster tree in the following canonical way. Here, the admissibility condition $Adm : T_{I \times I} \to \{True, False\}$ is a boolean function which we will specify in more detail later. Given a cluster tree T_I, we construct the block cluster tree $T_{I \times I}$ by $\text{root}(T_{I \times I}) := I \times I$, and each vertex $s \times t \in T_{I \times I}$ has the set of successors

$$S(s \times t) := \begin{cases} \emptyset & \text{if } Adm(s \times t) = \text{True}; \\ \emptyset & \text{if } \min\{\#t, \#s\} \leq n_{\min}; \\ \{s' \times t' \mid s' \in S(s), t' \in S(t);\} & \text{otherwise.} \end{cases} \quad (1)$$

The parameter n_{min} ensures that blocks do not become so small that the matrix arithmetic of a full matrix is more efficient. It is typically set to $n_{min} = 32$ or even $n_{min} = 64$. The leaves of a block cluster tree obtained through this construction yield a disjoint partition of the product index set $I \times I$. Matrix blocks which correspond to admissible block clusters will be approximated by low rank matrices in the following Rk-matrix representation:

Definition 3 (Rk-matrix representation). *Let $k, n, m \in N_0$. Let $M \in R^{n \times m}$ be a matrix of at most rank k . A representation of M in factorized form*

$$M = AB^T, \qquad A \in R^{n \times k}, B \in R^{m \times k}, \tag{2}$$

with A and B stored in full matrix representation, is called an Rk-matrix representation of M , or, in short, we call M an Rk-matrix.

If the rank k is small compared to the matrix size given by n and m , we obtain considerable savings in the storage and work complexities of an Rk-matrix compared to a full matrix [5].

A *standard* (or *strong*) admissibility condition has been employed in most previous papers [7, 9, 5, 6, 1] and is given by

$$Adm_s(s \times t) = True \quad :\Leftrightarrow \quad min(diam(s), diam(t)) \leq \eta \; dist(s,t) \tag{3}$$

for some $0 < \eta$. Here, “diam” and “dist” denote the Euclidean diameter/distance of the (union of the) supports of the basis functions with indices in s, t , resp. A weaker admissibility condition which yields smaller constants in (storage and work) complexities for $\mathcal{H}$ -matrices has been introduced and analyzed in [10]. It is given by

$$Adm_w(s \times t) = True \quad :\Leftrightarrow \quad s \neq t. \tag{4}$$

The block partition which is provided by the leaves of a block cluster tree is used to define an $\mathcal{H}$ -matrix as follows.

Definition 4 ($\mathcal{H}$ -matrix). *Let $k, n_{\min} \in N_0$. The set of $\mathcal{H}$ -matrices induced by a block cluster tree $T := T_{I \times I}$ with blockwise rank k and minimum block size $n_{\min}$ is defined by $\mathcal{H}(T,k) := \{M \in R^{I \times I} \mid \forall t \times s \in \mathcal{L}(T) : \text{rank}(M|_{t \times s}) \leq k \text{ or } \min\{\#t, \#s\} \leq n_{\min}\}$. Blocks $M|_{t \times s}$ with $\text{rank}(M|_{t \times s}) \leq k$ are stored as Rk-matrices whereas all other blocks are stored as full matrices.*

It is our goal to approximate an LU-factorization of a (stiffness) matrix by $\mathcal{H}$ -matrices $L^{\mathcal{H}}$, $U^{\mathcal{H}}$. The storage and computational complexities and also the accuracy of such an $\mathcal{H}$ -LU factorization depend strongly on the construction of the cluster tree, i.e., the hierarchy of index set partitionings. In the following Section 3 we will derive a new index clustering algorithm which will permit a subsequent $\mathcal{H}$ -LU factorization in which 1) large blocks remain zero, 2) non-zero off-diagonal blocks can be approximated in $\mathcal{H}$ -matrix format, and 3) the factorization process can be parallelized. The actual $\mathcal{H}$ -LU factorization is defined recursively in the block structure and has been derived in [13, 2] for matrices arising in finite element methods.

3 A new domain decomposition clustering algorithm

In [8], a direct domain decomposition method is combined with the hierarchical matrix technique. In particular, a domain Ω is subdivided into p subdomains and an interior boundary Γ which separates the subdomains. Within each subdomain, standard $\mathcal{H}$-matrix techniques are used, i.e., $\mathcal{H}$-matrices are constructed by the standard index clustering with zero or two subsets. Here, we propose to use the canonical block cluster tree construction starting from a different cluster tree which will be derived below. The new idea is *not* to distinguish between the index clustering which the domain decomposition yields and the index clustering needed for the $\mathcal{H}$-matrix construction, but to unify these two clusterings.

In Figure 1, the two subdomains Ω_1, Ω_2 are not admissible w.r.t. (3), but since these blocks remain zero during the LU-factorization, we should admit them (rank zero). Thus, we distinguish between the sets of domain-clusters $\mathcal{C}_{\mathrm{dom}}$ and interface clusters $\mathcal{C}_{\mathrm{int}}$ in order to define the admissibility.

We assume some underlying domain decomposition algorithm (e.g., nested dissection) which divides an index set into three disjoint subsets of indices: $I(\Omega_1)$ consists of indices in the first subdomain Ω_1, $I(\Omega_2)$ consists of indices in the second subdomain Ω_2, and $I(\Gamma)$ consists of indices of an interior boundary and separates the index sets $I(\Omega_1)$ and $I(\Omega_2)$, i.e., matrix entries a_{ij} equal zero if $i \in I(\Omega_k)$ and $j \in I(\Omega_l)$ for $k \neq l$. Any interior boundary Γ is bisected into Γ_1, Γ_2 with corresponding index sets $I(\Gamma_1)$, $I(\Gamma_2)$. Based upon such a domain decomposition, we construct the cluster tree from the root to the leaves as follows: We initialize $root(T_I) := I$, $\mathcal{C}_{\mathrm{dom}} := \{I\}$, $\mathcal{C}_{\mathrm{int}} := \{\}$. Each cluster $v \in T_I \cap \mathcal{C}_{\mathrm{dom}}$ with $\#v > n_{\min}$ is subdivided by the rule

$$S(v) := \{I(\Omega_1), I(\Omega_2), I(\Gamma)\} \tag{5}$$

and we add the sons to the corresponding sets of clusters:

$$\mathcal{C}_{\mathrm{dom}} := \mathcal{C}_{\mathrm{dom}} \cup \{I(\Omega_1), I(\Omega_2)\}, \quad \mathcal{C}_{\mathrm{int}} := \mathcal{C}_{\mathrm{int}} \cup \{I(\Gamma)\}.$$

Each node $v \in T_I \cap \mathcal{C}_{\mathrm{int}}$ is subdivided by the rule

$$S(v) := \{w\},\ S(w) := \{I(\Gamma_1), I(\Gamma_2)\}, \qquad \mathcal{C}_{\mathrm{int}} := \mathcal{C}_{\mathrm{int}} \cup S(w), \tag{6}$$

where the cluster w is given by $w := v$, i.e., the boundary Γ corresponding to v is (eventually) split into two subsets and $v = I(\Gamma_1) \cup I(\Gamma_2)$.

Example 1. Figure 2 gives an illustration of the new clustering applied to the 25 vertices (indices) of a regular triangulation of the unit square.

Remark 1. In the new clustering, an index cluster v is either subdivided into the three clusters (5) corresponding to indices in the two subdomains and the interior boundary, resp., or it is “subdivided” only every second step by a simple bisection (6). The latter only happens for clusters corresponding to interior boundaries. This is motivated by the underlying geometry of two-dimensional subdomains versus one-dimensional interior boundaries. Roughly, two subdivision steps decrease the Euclidean diameters by a factor of two for both subdomains and the interior boundary (which is effectively only subdivided once). This has a favorable impact on the resulting $\mathcal{H}$-matrix structure in terms of its storage requirements and approximation accuracy.

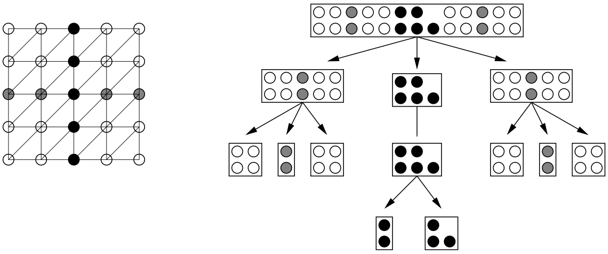


Fig. 2. Example for the new index clustering with $n_{min} = 4$.

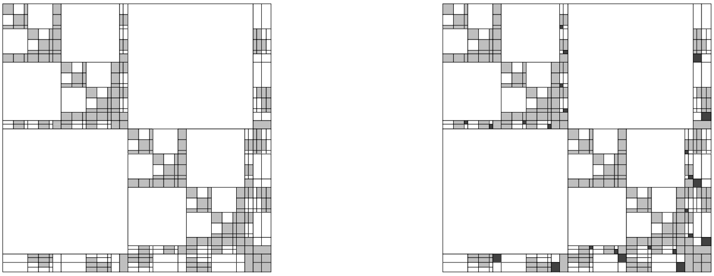


Fig. 3. Example for a domain decomposition $\mathcal{H}$-matrix (left) and a single precision $\mathcal{H}$-LU decomposition (right). Gray blocks correspond to full submatrices and black blocks represent non-zero Rk-matrices.

The block cluster tree $T_{I\times I}$ is build from the new cluster tree T_I by (1), where a pair (t, s) of clusters is admissible, if they are admissible with respect to (3) or if both are domain clusters: $t, s \in \mathcal{C}_{\rm dom}$. A typical structure of the resulting $\mathcal{H}$-matrix and its $\mathcal{H}$-LU decomposition is plotted in Figure 3. The approximation of non-zero, off-diagonal blocks by Rk-matrices will yield the order reduction from $\mathcal{O}(n^{1.5})$ (exact LU based on nested dissection) down to $\mathcal{O}(n \log n)$ (approximate $\mathcal{H}$-LU). Savings for three-dimensional problems (to which the new clustering generalizes easily) are even more significant and will be illustrated in a forthcoming paper.

4 Numerical results

We will present numerical results of the domain-decomposition based $\mathcal{H}$-LU preconditioner applied to the convection-diffusion equation

$$-\varepsilon \Delta u + b(x,y) \cdot \nabla u = f \qquad \text{in } \Omega = [-1,1] \times [-1,1]$$

with recirculating flow $b(x,y) = (4x(x-1)(1-2y), -4y(y-1)(1-2x))$ and $\varepsilon = 10^{-8}$. In all the following numerical experiments, we set $\eta = 4.0$ in the admissibility condition (3), and we choose adaptive ranks to enforce certain accuracies of the local Rk-blocks. In particular, we choose the rank k of a given Rk-block such that $\sigma(k) \leq a \cdot \sigma(1)$ where $\sigma(j)$ denotes the j'th singular value, and we show numerical results for relative accuracies $a \in \{0.1, 0.25, 0.126, 0.0625\}$. The following examples have been computed on a DELL Precision 530 workstation (2.4 GHz, 4GB memory).

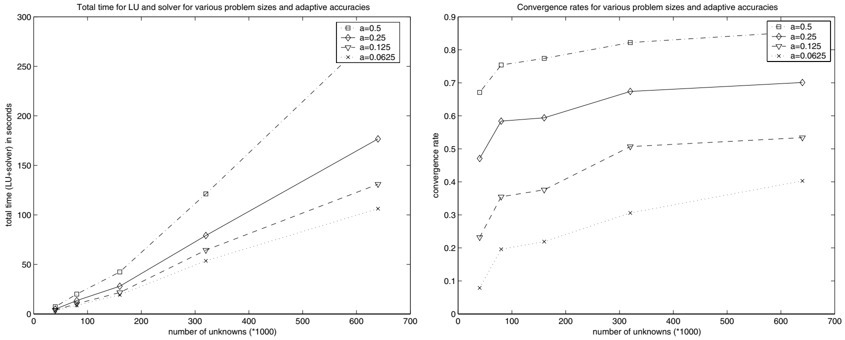


Fig. 4. Total time for the $\mathcal{H}$-LU decomposition and iterative solver (left) and corresponding convergence rates (right).

In Figure 4, on the left we show the time (in seconds) to compute the $\mathcal{H}$-LU decomposition and the subsequent iterative solution depending on the problem size n (starting from $n = 40000$ up to $n = 640000$) for various adaptive accuracies a. Here, we have used the $\mathcal{H}$-LU preconditioner in a bicg-stab iteration, and we stopped the iteration when the residual had been reduced by 10^{-6}. We note that the $\mathcal{H}$-LU preconditioner with higher accuracy a leads to significantly faster convergence, especially for larger problem sizes. The highest accuracy $a = 0.0625$ yields the overall fastest method for the larger problem sizes. The convergence rates improve significantly with higher accuracy a, indicating that for a given problem size, we are able to construct very efficient $\mathcal{H}$-LU preconditioners by increasing the relative accuracy a.

References

1. S. L. Borne, *Hierarchical matrices for convection-dominated problems*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., vol. 40 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 631–638.
2. S. L. Borne and L. Grasedyck, *$\mathcal{H}$-matrix preconditioners in convection-dominated problems*, SIAM J. Mat. Anal., 27 (2006), pp. 1172–1183.
3. I. Brainman and S. Toledo, *Nested-dissection orderings for sparse LU with partial pivoting*, SIAM J. Mat. Anal. Appl., 23 (2002), pp. 998–1012.
4. A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.
5. L. Grasedyck and W. Hackbusch, *Construction and arithmetics of calH-matrices*, Computing, 70 (2003), pp. 295–334.

6. L. Grasedyck, W. Hackbusch, and S. L. Borne, *Adaptive geometrically balanced clustering of $\mathcal{H}$ - matrices*, Computing, 73 (2004), pp. 1–23.
7. W. Hackbusch, *A sparse matrix arithmetic based on $\mathcal{H}$ -matrices. Part I: Introduction to $\mathcal{H}$ -matrices*, Computing, 62 (1999), pp. 89–108.
8. ———, *Direct domain decomposition using the hierarchical matrix technique*, in Fourteenth International Conference on Domain Decomposition Methods, I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, eds., ddm.org, 2003, pp. 39–50.
9. W. Hackbusch and B. Khoromskij, *A sparse $\mathcal{H}$ -matrix arithmetic. Part II: Application to multi-dimensional problems*, Computing, 64 (2000), pp. 21–47.
10. W. Hackbusch, B. N. Khoromskij, and R. Kriemann, *Hierarchical matrices based on a weak admissibility criterion*, Computing, 73 (2004), pp. 207–243.
11. B. Hendrickson and E. Rothberg, *Improving the run time and quality of nested dissection ordering*, SIAM J. Sci. Comput., 20 (1998), pp. 468–489.
12. R. Kriemann, *Parallel $\mathcal{H}$ -matrix arithmetics on shared memory systems*, Computing, 74 (2005), pp. 273–297.
13. M. Lintner, *The eigenvalue problem for the 2D laplacian in $\mathcal{H}$ -matrix arithmetic and application to the heat and wave equation*, Computing, 72 (2004), pp. 293–323.

Condition Number Estimates for C^0 Interior Penalty Methods

Shuang Li [1] and Kening Wang [2]

[1] Department of Mathematics, University of South Carolina, Columbia, SC 29208, USA. sli@math.sc.edu

[2] Department of Mathematics and Statistics, University of North Florida, Jacksonville, FL 32224, USA. kening.wang@unf.edu

Summary. In this paper we study the condition number of the system resulting from C^0 interior penalty methods for fourth order elliptic boundary value problems. We show that the condition number can be bounded by Ch^{-4} and that this bound is sharp, where h is the mesh size of the triangulation and C is a positive constant independent of the mesh size.

1 Introduction

C^0 interior penalty methods provide a new approach for the solution of fourth order elliptic problems [10, 5]. These methods combine the ideas of continuous Galerkin methods, discontinuous Galerkin methods and stabilization techniques, and can be illustrated by the following model problem on a bounded polygonal domain Ω in $\mathbb{R}^2$:
Find $u \in H_0^2(\Omega)$ such that

$$\sum_{i,j=1}^{2} \int_{\Omega} \frac{\partial^2 u}{\partial x_i \partial x_j} \frac{\partial^2 v}{\partial x_i \partial x_j} dx = \int_{\Omega} fv\,dx \qquad \forall\, v \in H_0^2(\Omega), \tag{1}$$

where $f \in L_2(\Omega)$.

Let $\mathcal{T}_h$ be a simplicial or convex quadrilateral triangulation of Ω. In C^0 interior penalty methods, we choose the discrete space $V_h \subset H_0^1(\Omega)$ to be either a $\mathcal{P}_\ell$ $(\ell \geq 2)$ triangular Lagrange finite element space or a $\mathcal{Q}_\ell$ $(\ell \geq 2)$ tensor product finite element space associated with $\mathcal{T}_h$. By an integration by parts argument [5], it can be shown that the solution u of (1), which, by elliptic regularity [11, 9, 13, 2], belongs to $H^{2+\alpha}(\Omega)$ for some $\alpha > 1/2$, satisfies

$$\mathcal{A}_h(u,v) = \int_{\Omega} fv\,dx \qquad \forall\, v \in V_h, \tag{2}$$

where

$$\mathcal{A}_h(w,v) = \sum_{D\in\mathcal{T}_h}\sum_{i,j=1}^{2}\int_D \frac{\partial^2 w}{\partial x_i\partial x_j}\frac{\partial^2 v}{\partial x_i\partial x_j}dx + \sum_{e\in\mathcal{E}_h}\frac{\eta}{|e|}\int_e \left[\!\left[\frac{\partial w}{\partial n}\right]\!\right]\left[\!\left[\frac{\partial v}{\partial n}\right]\!\right]ds$$
$$+\sum_{e\in\mathcal{E}_h}\int_e\left(\left\{\!\!\left\{\frac{\partial^2 w}{\partial n^2}\right\}\!\!\right\}\left[\!\left[\frac{\partial v}{\partial n}\right]\!\right] + \left\{\!\!\left\{\frac{\partial^2 v}{\partial n^2}\right\}\!\!\right\}\left[\!\left[\frac{\partial w}{\partial n}\right]\!\right]\right)ds. \tag{3}$$

In (3), $\mathcal{E}_h$ is the set of all the edges of $\mathcal{T}_h$, and η is a penalty parameter. The jumps $[\![\cdot]\!]$ and averages $\{\!\{\cdot\}\!\}$ are defined as follows.

Let e be an interior edge of $\mathcal{T}_h$ shared by two elements D_- and D_+ and n_e be the unit normal vector of e pointing from D_- to D_+. We define on e, for any function v that is piecewise H^s with respect to the triangulation $\mathcal{T}_h$ and for some $s > \frac{5}{2}$,

$$\left[\!\left[\frac{\partial v}{\partial n}\right]\!\right] = \frac{\partial v_+}{\partial n_e} - \frac{\partial v_-}{\partial n_e} \quad \text{and} \quad \left\{\!\!\left\{\frac{\partial^2 v}{\partial n^2}\right\}\!\!\right\} = \frac{1}{2}\left[\frac{\partial^2 v_+}{\partial n_e^2} + \frac{\partial^2 v_-}{\partial n_e^2}\right], \tag{4}$$

where $v_\pm = v\big|_{D_\pm}$. For an edge e that is a subset of $\partial\Omega$, we take n_e to be the outward pointing unit normal vector and define

$$\left[\!\left[\frac{\partial v}{\partial n}\right]\!\right] = -\frac{\partial v}{\partial n_e} \quad \text{and} \quad \left\{\!\!\left\{\frac{\partial^2 v}{\partial n^2}\right\}\!\!\right\} = \frac{\partial^2 v}{\partial n_e^2}. \tag{5}$$

Note that $[\![\partial v/\partial n]\!]$ and $\{\!\{\partial^2 v/\partial n^2\}\!\}$ are independent of the choice of n_e.

The discrete problem for (1) is then given by:
Find $u_h \in V_h$ such that

$$\mathcal{A}_h(u_h, v) = \int_\Omega f v\, dx \qquad \forall\, v \in V_h. \tag{6}$$

In view of (2), the C^0 interior penalty method defined by (6) is consistent and for a sufficiently large η, it is also stable. Therefore the discretization error $u - u_h$ is quasi-optimal with respect to appropriate norms [10, 5].

In this paper, we show that the condition number of the system of (6) is of order h^{-4}, where h is the mesh size of the triangulation. This result implies that the system of the discrete problem resulting from C^0 interior penalty methods is very ill-conditioned for small h, in which case the convergence rates of classical iterative methods are very slow. Therefore it is necessary to use modern fast solvers such as multigrid methods [4] and domain decomposition methods [6] to improve the efficiency.

The rest of the paper is organized as follows. We introduce the finite element space and some preliminaries in section 2. In section 3, we derive the upper bound for the condition number of the system. We obtain the lower bound for the condition number in the last section.

2 Preliminaries

In this section, we define the finite element space and derive some preliminary estimates that can help us to obtain the estimates for the condition number. For simplicity we will focus on the case that $\mathcal{T}_h$ is a quasi-uniform rectangular mesh in this paper. The results we will show are still true for general convex quadrilateral meshes and triangular elements.

To avoid the proliferation of constants, we henceforth use the notation $A \lesssim B$ to represent the statement $A \leq C \times B$, where C is a constant which depends only on the aspect ratios of $\mathcal{T}_h$. The notation $A \approx B$ is equivalent to $A \lesssim B$ and $B \lesssim A$.

Let $V_h \subset H_0^1(\Omega)$ be the Q_2 finite element space associated with $\mathcal{T}_h$. For η sufficiently large (which is assumed to be the case), the following relation [5] holds:

$$\mathcal{A}_h(v,v) \approx |v|^2_{H^2(\Omega,\mathcal{T}_h)} \qquad \forall\, v \in V_h, \tag{7}$$

where

$$|v|^2_{H^2(\Omega,\mathcal{T}_h)} = \sum_{D\in\mathcal{T}_h} |v|^2_{H^2(D)} + \sum_{e\in\mathcal{E}_h} \frac{1}{|e|} \|[\![\partial v/\partial n]\!]\|^2_{L_2(e)}. \tag{8}$$

Here and throughout this paper we follow the standard notation for L_2 -based Sobolev spaces [1, 3, 8].

Let

$$\mathbf{A}_h = (\mathcal{A}_h(\varphi_1,\varphi_2))_{1\leq i,j\leq n} \tag{9}$$

be the stiffness matrix, where n is the dimension of V_h and $\varphi_1,\cdots,\varphi_n$ are the nodal basis functions for V_h . We want to estimate the condition number of $\mathbf{A}_h$ given by

$$\kappa(\mathbf{A}_h) = \frac{\lambda_{\max}(\mathbf{A}_h)}{\lambda_{\min}(\mathbf{A}_h)}. \tag{10}$$

Note that

$$\lambda_{\max}(\mathbf{A}_h) = \max_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \frac{x^T\mathbf{A}_h x}{x^T x} \approx \max_{\substack{v\in V_h \\ v\neq 0}} \frac{\mathcal{A}_h(v,v)}{h^{-2}\|v\|^2_{L_2(\Omega)}}, \tag{11}$$

$$\lambda_{\min}(\mathbf{A}_h) = \min_{\substack{x\in\mathbb{R}^n \\ x\neq 0}} \frac{x^T\mathbf{A}_h x}{x^T x} \approx \min_{\substack{v\in V_h \\ v\neq 0}} \frac{\mathcal{A}_h(v,v)}{h^{-2}\|v\|^2_{L_2(\Omega)}}. \tag{12}$$

3 Upper bound for the condition number

In this section, we obtain an upper bound for the condition number of $\mathbf{A}_h$. From (11) and (12), it is sufficient to find an upper bound for the maximum eigenvalue of $\mathbf{A}_h$ and a lower bound for the minimum eigenvalue of $\mathbf{A}_h$.

Lemma 1. *For all $v \in V_h$, it holds that*

$$\lambda_{\max}(\mathbf{A}_h) \lesssim h^{-2}. \tag{13}$$

Proof. Let $v \in V_h$ be arbitrary, using (7), (8), inverse estimates [3], (4) and the trace theorem (with scaling), we obtain that

$$\begin{aligned}\mathcal{A}_h(v,v) &\approx |v|^2_{H^2(\Omega,\mathcal{T}_h)}\\ &= \sum_{D\in\mathcal{T}_h}|v|^2_{H^2(D)} + \sum_{e\in\mathcal{E}_h}\frac{1}{|e|}\|\,[\![\partial v/\partial n]\!]\,\|^2_{L_2(e)}\\ &\lesssim \sum_{D\in\mathcal{T}_h}(\operatorname{diam} D)^{-4}\|v\|^2_{L_2(D)} + \sum_{e\in\mathcal{E}_h}\frac{1}{|e|}\sum_{D\in\mathcal{T}_e}\|\partial v_{_D}/\partial n\|^2_{L_2(e)} \qquad (14)\\ &\lesssim \sum_{D\in\mathcal{T}_h}(\operatorname{diam} D)^{-4}\|v\|^2_{L_2(D)}\\ &\quad + \sum_{e\in\mathcal{E}_h}\sum_{D\in\mathcal{T}_e}\Big[(\operatorname{diam} D)^{-2}|v|^2_{H^1(D)} + |v|^2_{H^2(D)}\Big]\\ &\lesssim \sum_{D\in\mathcal{T}_h}(\operatorname{diam} D)^{-4}\|v\|^2_{L_2(D)} + \sum_{e\in\mathcal{E}_h}\sum_{D\in\mathcal{T}_e}(\operatorname{diam} D)^{-4}\|v\|^2_{L_2(D)}\\ &\lesssim \sum_{D\in\mathcal{T}_h}(\operatorname{diam} D)^{-4}\|v\|^2_{L_2(D)}\\ &\lesssim h^{-4}\|v\|^2_{L_2(\Omega)}.\end{aligned}$$

where $\mathcal{T}_e$ is the set of all rectangles sharing e as a common edge.

Here we have used the fact that

$$h \approx \operatorname{diam} D \qquad \forall\ D\in\mathcal{T}_h.$$

Therefore, the estimate (13) follows from (11) and (14).

♯

Next we derive a lower bound for the minimum eigenvalue of $\mathbf{A}_h$.

Lemma 2. *It holds that*

$$\lambda_{\min}(\mathbf{A}_h) \gtrsim h^2 \qquad \forall\, v\in V_h. \tag{15}$$

Proof. For general piecewise H^2 functions v, we have the following Poincar é-Friedrichs inequality [7]:

$$\begin{aligned}\|v\|^2_{L_2(\Omega)} + |v|^2_{H^1(\Omega,\mathcal{T}_h)} \lesssim \Big[&|v|^2_{H^2(\Omega,\mathcal{T}_h)} + [\Phi(v)]^2\\ &+ \sum_{e\in\mathcal{E}_h}\Big(\frac{1}{|e|^3}\|\pi_{e,1}[\![v]\!]_e\|^2_{L_2(e)} + \frac{1}{|e|}\|\pi_{e,0}[\![\partial v/\partial n]\!]_e\|^2_{L_2(e)}\Big)\Big],\end{aligned} \tag{16}$$

where $\Phi : H^2(\Omega,\mathcal{T}_h) \longrightarrow \mathbb{R}$ is a seminorm that satisfies certain properties (cf. (I.2), (I.3), (II.15) and (III.3) of [7]) and the operator $\pi_{e,0}$ (resp. $\pi_{e,1}$) is the orthogonal projection operator from $L_2(e)$ onto $\mathcal{P}_0(e)$ (resp. $\mathcal{P}_1(e)$).

In (16), taking $\Phi(v) = \|\pi_{\partial\Omega,1}\, v\|_{L_2(\Omega)}$ and applying it to $v\in V_h$, we have

$$\begin{aligned}\|v\|^2_{L_2(\Omega)} + |v|^2_{H^1(\Omega,\mathcal{T}_h)} &\lesssim \sum_{D\in\mathcal{T}_h}|v|^2_{H^2(D)} + \sum_{e\in\mathcal{E}_h}\frac{1}{|e|}\|\,[\![\partial v/\partial n]\!]\,\|^2_{L_2(e)}\\ &= |v|^2_{H^2(\Omega,\mathcal{T}_h)},\end{aligned}$$

which implies for all $v\in V_h$

$$\|v\|^2_{L_2(\Omega)} \lesssim |v|^2_{H^2(\Omega,\mathcal{T}_h)}. \tag{17}$$

Therefore, by (12), (7) and (17), we obtain

$$\lambda_{\min}(\mathbf{A}_h) \approx \min_{\substack{v\in V_h\\ v\neq 0}} \frac{\mathcal{A}_h(v,v)}{h^{-2}\|v\|^2_{L_2(\Omega)}} \gtrsim h^2.$$

♯

From Lemma 1 and Lemma 2 we have the following condition number estimate.

Theorem 1. *The condition number of* $\mathbf{A}_h$ *satisfies the estimate*

$$\kappa(\mathbf{A}_h) = \frac{\lambda_{\max}(\mathbf{A}_h)}{\lambda_{\min}(\mathbf{A}_h)} \lesssim h^{-4}. \tag{18}$$

4 Lower bound for the condition number

In this section we will show that the bound for the condition number obtained in the last section is sharp. We begin with an easy lower bound for $\lambda_{\max}(\mathbf{A}_h)$.

Lemma 3. *It holds that*

$$\lambda_{\max}(\mathbf{A}_h) \gtrsim h^{-2}. \tag{19}$$

Proof. In view of (12) and (7), it suffices to construct a function $v_* \in V_h$ such that

$$|v_*|^2_{H^2(\Omega,\mathcal{T}_h)} \gtrsim h^{-4}\|v_*\|^2_{L_2(\Omega)}. \tag{20}$$

Let D_* be an arbitrary element in $\mathcal{T}_h$. Take $v_* \in V_h$ to be a nodal basis function which is defined by

$$v_*(p) = \begin{cases} 1, & \text{if } p \text{ is the central node of } D_*,\\ 0, & \text{otherwise.}\end{cases} \tag{21}$$

Then it is not difficult to obtain that

$$\begin{aligned} v_*(x_1,x_2) = (\operatorname{diam} D_*)^{-4}\big((\operatorname{diam} D_*)^2 16x_1x_2 - (\operatorname{diam} D_*)16x_1^2x_2 \\ -(\operatorname{diam} D_*)16x_1x_2^2 + 16x_1^2x_2^2\big)\,. \end{aligned} \tag{22}$$

So (21) and (22) imply that

$$\|v_*\|^2_{L_2(\Omega)} = \|v_*\|^2_{L_2(D_*)} = \frac{64}{225}(\operatorname{diam} D_*)^2, \tag{23}$$

and

$$|v_*|^2_{H^2(\Omega,\mathcal{T}_h)} = |v_*|^2_{H^2(D_*)} + \sum_{\substack{e\in\mathcal{E}_h\\ e\subset \bar{D}_*}} \frac{1}{|e|}\|\partial v_*/\partial n\|^2_{L_2(e)} = \frac{5312}{45}(\operatorname{diam} D_*)^{-2}. \tag{24}$$

Therefore, combining (23) and (24), we obtain

$$|v_*|^2_{H^2(\Omega,\mathcal{T}_h)} \geq h^{-4}\|v_*\|^2_{L_2(\Omega)}.$$

♯

We now derive an upper bound for the minimum eigenvalue of $\mathbf{A}_h$.

Lemma 4. *The following estimate for the minimum eigenvalue of* $\mathbf{A}_h$ *holds:*

$$\lambda_{\min}(\mathbf{A}_h) \lesssim h^2. \tag{25}$$

Proof. From the theory of partial differential equations [12], there exist $0 < \lambda_1 \leq \lambda_2 \leq \cdots$ and $u_1, u_2, \cdots \in H_0^2(\Omega)$ such that

$$\triangle^2 u_i = \lambda_i u_i \quad \text{and} \quad \int_\Omega u_i u_j \, dx = \delta_{ij}.$$

We now consider the following system:

$$\begin{cases} \triangle^2 u_1 = \lambda_1 u_1 & \text{in } \Omega, \\ u_1|_{\partial\Omega} = 0 \quad \text{and} & \dfrac{\partial u_1}{\partial n}|_{\partial\Omega} = 0. \end{cases} \tag{26}$$

Let $\hat{u}_1$ be the Q_2 interpolant of u_1 . Then standard interpolation error estimates [3] imply

$$\|u_1 - \hat{u}_1\|^2_{L_2(\Omega)} \lesssim h^4 |u_1|^2_{H^2(\Omega)} \lesssim h^4 \|u_1\|^2_{L_2(\Omega)}, \tag{27}$$

$$|u_1 - \hat{u}_1|^2_{H^1(\Omega)} \lesssim h^2 |u_1|^2_{H^2(\Omega)}, \tag{28}$$

$$\sum_{D \in \mathcal{T}_h} |u_1 - \hat{u}_1|^2_{H^2(D)} \lesssim |u_1|^2_{H^2(\Omega)}. \tag{29}$$

For h small enough, it follows from (27) that

$$\begin{aligned} \|\hat{u}_1\|^2_{L_2(\Omega)} &\gtrsim \|u_1\|^2_{L_2(\Omega)} - \|u_1 - \hat{u}_1\|^2_{L_2(\Omega)} \\ &\gtrsim \|u_1\|^2_{L_2(\Omega)} - h^4 \|u_1\|^2_{L_2(\Omega)} \\ &\gtrsim \|u_1\|^2_{L_2(\Omega)}. \end{aligned} \tag{30}$$

On the other hand, since $u_1 \in H_0^2(\Omega)$, by (8), the triangle inequality, (29), the trace theorem with scaling and (28), we obtain that

$$\begin{aligned} &|\hat{u}_1|^2_{H^2(\Omega, \mathcal{T}_h)} \\ &= \sum_{D \in \mathcal{T}_h} |\hat{u}_1|^2_{H^2(D)} + \sum_{e \in \mathcal{E}_h} \frac{1}{|e|} \| [\![\partial \hat{u}_1 / \partial n]\!] \|^2_{L_2(e)} \\ &\lesssim \sum_{D \in \mathcal{T}_h} |\hat{u}_1 - u_1|^2_{H^2(D)} + \sum_{D \in \mathcal{T}_h} |u_1|^2_{H^2(D)} + \sum_{e \in \mathcal{E}_h} \frac{1}{|e|} \| [\![\partial (\hat{u}_1 - u_1) / \partial n]\!] \|^2_{L_2(e)} \\ &\lesssim \sum_{D \in \mathcal{T}_h} |u_1|^2_{H^2(D)} + \sum_{D \in \mathcal{T}_h} \left[(\operatorname{diam} D)^{-2} |\hat{u}_1 - u_1|^2_{H^1(D)} + |\hat{u}_1 - u_1|^2_{H^2(D)} \right] \\ &\lesssim \sum_{D \in \mathcal{T}_h} |u_1|^2_{H^2(D)} \\ &= |u_1|^2_{H^2(\Omega)}. \end{aligned} \tag{31}$$

Therefore, the estimate (25) follows from (12), (7), (30) and (31):

$$\begin{aligned}\lambda_{\min}(\mathbf{A}_h) &\approx \min_{\substack{v\in V_h\\ v\neq 0}} \frac{\mathcal{A}_h(v,v)}{h^{-2}\|v\|^2_{L_2(\Omega)}}\\ &\lesssim \frac{\mathcal{A}_h(\hat{u}_1,\hat{u}_1)}{h^{-2}\|\hat{u}_1\|^2_{L_2(\Omega)}}\\ &\approx \frac{|\hat{u}_1|^2_{H^2(\Omega,\mathcal{T}_h)}}{h^{-2}\|\hat{u}_1\|^2_{L_2(\Omega)}}\\ &\lesssim \frac{|u_1|^2_{H^2(\Omega)}}{h^{-2}\|u_1\|^2_{L_2(\Omega)}}\\ &\lesssim h^2.\end{aligned} \tag{32}$$

♯

Combining Lemmas 3 and 4, we have the following theorem.

Theorem 2. *The following estimate holds for our model problem:*

$$\kappa(\mathbf{A}_h) = \frac{\lambda_{\max}(\mathbf{A}_h)}{\lambda_{\min}(\mathbf{A}_h)} \gtrsim h^{-4}. \tag{33}$$

Acknowledgement. We give special thanks to Dr. Susanne C. Brenner and Dr. Li-yeng Sung for helpful discussions. The work of the second author was supported in part by the National Science Fundation under Grant No. DMS-03-11790.

References

1. R. A. Adams and J. J. F. Fournier, *Sobolev Spaces*, Academic Press, second ed., 2003.
2. C. Bacuta, J. H. Bramble, and J. E. Pasciak, *Shift theorems for the biharmonic Dirichlet problem*, Kluwer/Plenum, New York, 2002, pp. 1–26.
3. S. C. Brenner and L. R. Scott, *The Mathematical Theory of Finite Element Methods*, Springer-Verlag, New York, second ed., 2002.
4. S. C. Brenner and L.-Y. Sung, *Multigrid algorithms for C^0 interior penalty methods*, Tech. Rep. 2004:11, Industrial Mathematics Institute, Department of Mathematics, University of South Carolina, 2004.
5. ———, *C^0 interior penalty methods for fourth order elliptic boundary value problems on polygonal domains*, J. Sci. Comput., 22-23 (2005), pp. 83–118.
6. S. C. Brenner and K. Wang, *Two-level additive Schwarz preconditioners for C^0 interior penalty methods*, Numer. Math., 102 (2005), pp. 231–255.
7. S. C. Brenner, K. Wang, and J. Zhao, *Poincar é -Friedrichs inequalities for piecewise h^2 functions*, Numer. Funct. Anal. Optim., 25 (2004), pp. 463–478.

8. P. G. Ciarlet, *The Finite Element Method for Elliptic Problems*, North-Holland, Amsterdam, 1978.
9. M. Dauge, *Elliptic boundary value problems on corner domains*, Lecture Notes in Mathematics, Springer Verlag, Berlin, 1988.
10. G. Engel, K. Garikipati, T. J. R. Hughes, M. G. Larson, L. Mazzei, and R. L. Taylor, *Continuous/discontinuous finite element approximations of fourth order elliptic problems in structural and continuum mechanics with applications to thin beams and plates, and strain gradient elasticity*, Comput. Methods Appl. Mech. Engrg., 191 (2002), pp. 3669–3750.
11. P. Grisvard, *Elliptic problems in nonsmooth domains*, Pitman Publishing, Boston, 1985.
12. J. Jost, *Partial Differential Equations*, Springer-Verlag, 2002.
13. S. A. Nazarov and B. A. Plamenevsky, *Elliptic Problems in Domains with Piecewise Smooth Boundaries*, W. de Gruyter, Berlin–New York, 1994.

An Iterative Substructuring Method for Mortar Nonconforming Discretization of a Fourth-Order Elliptic Problem in Two Dimensions

Leszek Marcinkowski*

Department of Mathematics, Warsaw University, Banacha 2, 02-097 Warszawa, Poland. lmarcin@mimuw.edu.pl

Summary. In this paper we consider an iterative substructuring method for solving system of equations arising from mortar Morley finite element discretization of a model fourth order elliptic problem in 2D. A parallel preconditioner for the interface problem is introduced using Additive Schwarz Method framework. The method is quasi-optimal i.e. the number of CG iterations for the preconditioned problem grows polylogarithmically as the sizes of the meshes decrease and it is independent of the jumps of the coefficients.

1 Introduction

The discretization methods for partial differential equations are usually built on a mesh in a uniform way, however sometimes it is necessary to develop discretization methods which allow us to apply different type of discretization techniques in subdomains. The mortar method introduced in [4] is a domain decomposition method which enable us to introduce independent meshes or discretization methods in non-overlapping subdomains. A general presentation of mortar method in two and three dimensions for elliptic boundary value problems of second order can be found e.g. in [4, 2, 11], see also references therein. Mortar approach for discretizations of fourth order elliptic problems was studied in [3] where locally spectral discretizations were utilized, in [6] for DKT local discretizations, and in [9] for HCT and Morley finite element discretizations. Many parallel algorithms for solving a discrete problem were also developed, see e.g. [1, 8, 11] and the references therein.

In this paper we consider a mortar nonconforming Morley discretization of the fourth order elliptic problems. This discretization method was first proposed in [9],

*This work was partially supported by Polish Scientific Grant 2/P03A/005/24.

a paper that includes error bounds. A multigrid algorithm for mortar Morley discretization of plate bending problem was discussed in [12] (in a bit different mortar setting). To our knowledge no domain decomposition methods for solving the discrete problems obtained by this type of discretization was discussed in literature.

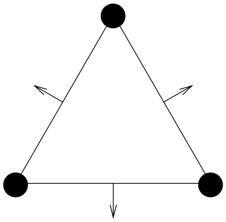
Our method is a substructuring one i.e. we first eliminate the unknowns related to degrees of freedom interior to subdomains (interior in a special sense) and then propose a parallel preconditioner based on an Additive Schwarz abstract scheme (ASM) for the derived system of equations, cf. e.g. [10]. We introduce local subspaces which form a decomposition of the discrete space. Then the ASM abstract theory allows us to construct a parallel preconditioner and prove condition number estimates of the preconditioned problem.

In our case we introduce a subdomain based coarse space and edge base spaces. The condition number of the arising preconditioner is proportional to $(1+\log(H/h))^2$ where h is the minimum of the local mesh sizes and H is the maximum of the diameters of the subdomain and is independent of the jumps of the coefficients.

2 Discrete space

We first assume that we have a polygonal domain Ω in the plane which is divided into non-overlapping subdomains Ω_k that form a coarse decomposition i.e. $\overline{\Omega} = \bigcup_{k=1}^{N} \overline{\Omega}_k$ and $\overline{\Omega}_l \cap \overline{\Omega}_k$ is an empty set, a common edge or vertex. We assume shape regularity of that decomposition in the sense of Section 4 in [5] and let $H = \max_k H_k$ for $H_k = \text{diam } \Omega_k$.

Fig. 1. Morley element.



The model differential problem is to find $u^* \in H_0^2(\Omega)$ such that

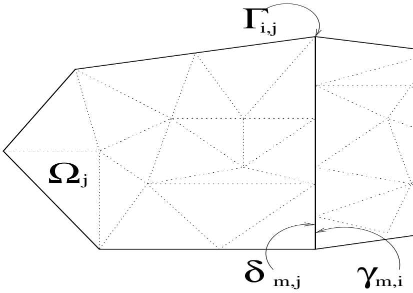
$$a(u^*, v) = f(v) \quad \forall v \in H_0^2(\Omega), \tag{1}$$

where $a(u,v) = \sum_{k=1}^{N} a_k(u,v)$ for $a_k(u,v) = \rho_k \int_{\Omega_k} \sum_{|\alpha|=2} \partial^\alpha u \partial^\alpha v \, dx$. Here $\rho_k > 0$ is a constant, $\alpha = (\alpha_1, \alpha_2), (\alpha_k \geq 0)$ is a multi-index and $|\alpha| = \alpha_1 + \alpha_2$ is the order of the multi-index. Of course we have that $a_k(u,u)$ is equivalent to $|u|^2_{H^2(\Omega_k)}$. In a subdomain Ω_k we introduce an independent quasiuniform triangulation $T_h(\Omega_k)$

made of triangles with a parameter $h_k = \max_{\tau \in T_h(\Omega_k)} \operatorname{diam}(\tau)$. Note that each interface (a common edge of two substructures) $\overline{\Gamma}_{ij} = \partial\Omega_i \cap \partial\Omega_j$ inherits 1D triangulations $T^i_{h_i}(\Gamma_{ij})$ and $T^j_{h_j}(\Gamma_{ij})$ from the respective triangulations of Ω_i and Ω_j , cf. Figure 2.

In each Ω_k we introduce a nonconforming local Morley finite element space $X_h(\Omega_k)$ formed by piecewise quadratic functions which are continuous at all vertices of all triangles from $T_h(\Omega_k)$, have continuous normal derivatives at the midpoints of all edges of elements from $T_h(\Omega_k)$, and have all respective degrees of freedom related to vertices and midpoints on $\partial\Omega_k \cap \partial\Omega$ equal to zero, cf. Figure 1. We introduce a global space $X_h(\Omega) = \prod_{k=1}^{N} X_h(\Omega_k)$. We now have to choose one side of

Fig. 2. Master and slave sides of of an interface Γ_{ij} .

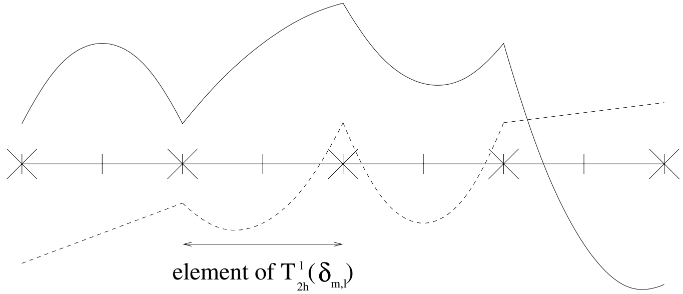


Γ_{ij} as the master (mortar) one denoted by $\gamma_{m,i}$ associated with Ω_i and the other one as the slave one (nonmortar) denoted by $\delta_{m,j}$ (associated with Ω_j) according to the rule $\rho_i \geq \rho_j$, cf. Figure 2. An interface $\Gamma = \bigcup_{k=1}^{N} \overline{\partial\Omega_k \setminus \partial\Omega}$ will play an important role. We also have to add a technical assumption that $h_i \leq Ch_j$, where C is a positive constant, due to the proof technique. This assumption is necessary for the proofs of some technical results and is due to the fact that any local Morley finite element function is not sufficiently regular. Because we assume that $h_i \leq Ch_j$ and both triangulations are quasiuniform, we can also assume that the two side elements of the slave triangulation $T^j_{h_j}(\delta_{m,j})$, i.e. the ones that touch the ends of $\delta_{m,j}$, are longer than the respective elements of the master (mortar) triangulation $T^i_{h_i}(\gamma_{m,i})$. Let $\gamma_{m,i,h}$ (or $\delta_{m,j,h}$) denotes the set of all midpoints and vertices of $T^i_{h_i}(\gamma_{m,i})$ (or $T^j_{h_j}(\delta_{m,j})$, respectively).

For the simplicity of presentation we also assume that the both 1D triangulations of the interface Γ_{kl} : $T^k_{h_k}(\gamma_{m,k})$ and $T^l_{h_l}(\delta_{m,l})$, have even numbers of the elements.

For $\delta_{m,l}$, we then introduce a coarser $2h_l$ triangulation by joining together two neighboring elements and get $T^l_{2h_l}(\delta_{m,l})$ - $2h_l$ 1D triangulation of $\delta_{m,l}$ formed by elements which are the union of two neighboring elements of $T^l_{h_l}(\delta_{m,l})$, cf. Fig. 3. Note that the midpoints of elements of $T^l_{2h_l}(\delta_{m,l})$ are also vertices of $T_h(\Omega_l)$. Then let $I_{2h_l,2} : C(\Gamma_{kl}) \to C(\Gamma_{kl})$ be a continuous piecewise quadratic interpolant defined on $T^l_{2h_l}(\delta_{m,l})$ and let $M^{2h_l}_t(\delta_{m,l})$ be the space of continuous piecewise quadratic

Fig. 3. Tangential test space and $2h_l$ interpolant on $T^l_{h_l}(\delta_{m,l})$. Broken line - $v \in M^{2h_l}_t(\delta_{m,l})$, solid line - $I_{2h_l,2}u$, $|$ - the endpoints of elements in $T^l_{h_l}(\delta_{m,l})$, X - the endpoints of elements in $T^l_{2h_l}(\delta_{m,l})$.



function on $T^l_{2h_l}(\delta_{m,l})$ which are linear in two end elements of $T^l_{2h_l}(\delta_{m,l})$. We also need another test space related to the trace of normal derivative of finite element functions: $M^{h_l}_n(\delta_{m,l})$ formed by functions piecewise constant on $T^l_{h_l}(\delta_{m,l})$.

The $2h_k$ triangulation of the master $\gamma_{m,k}$: $T^k_{2h_k}(\gamma_{m,k})$, and the operator $I_{2h_l,2}$ - piecewise quadratic interpolant on $T^k_{2h_k}(\gamma_{m,k})$ defined analogously based on elements of $T^k_{h_k}(\gamma_{m,k})$. Then for each interface $\Gamma_{kl} = \gamma_{m,k} = \delta_{m,l} \subset \Gamma$ we say that $u_k \in X_h(\Omega_k)$ and $u_l \in X_h(\Omega_l)$ satisfy the mortar conditions if

$$\begin{aligned} \int_{\delta_m} (I_{2h_k,2}u_k - I_{2h_l,2}u_l)\phi \, ds = 0 \qquad \forall \phi \in M^{2h_l}_t(\delta_{m,l}) \\ \int_{\delta_m} (\partial_n u_k - \partial_n u_l)\psi \, ds = 0 \qquad \forall \psi \in M^{h_l}_n(\delta_{m,l}). \end{aligned} \tag{2}$$

Here ∂_n is an outer unit normal derivative to Γ_{mk} . We now introduce a discrete space V^h as the space formed by all functions from $X_h(\Omega)$ which are continuous at the crosspoints (vertices of the subdomains) and satisfy the mortar conditions (2). Our discrete problem is to find $u^*_h \in V^h$ such that

$$a_H(u^*_h, v) = \sum_{k=1}^{N} a_{h,k}(u, v) = f(v) \quad \forall v \in V^h, \tag{3}$$

where $a_{h,k}(u,v) = \rho_k \sum_{\tau \in T_k(\Omega_k)} \int_\tau \sum_{|\alpha|=2} \partial^\alpha u \, \partial^\alpha v \, dx$. The problem has a unique solution, cf. [9].

3 An interface problem

We first eliminate some unknowns in the interiors of subdomains. Because the Morley element is nonconforming, there are some functions which have all degrees freedom corresponding to respective vertices or midpoints on $\partial\Omega_k$ equal to zero and still the traces onto a master $\gamma_{m,k}$ may be nonzero. Therefore we introduce the set Δ_k which consists of all vertices and midpoints that either are on $\partial\Omega_k$, or are interior to Ω_k such that at least one of its edges is on $\gamma_{m,k}$ but is not an end element of $T_h^k(\gamma_{m,k})$ for any master $\gamma_{m,k} \subset \partial\Omega_k$. We see that Δ_k is a set of nodal points either on $\partial\Omega_k$ or interior to Ω_k and such that a nodal basis function corresponding to a degree of freedom of this nodal point may have nonzero traces onto any master $\gamma_{m,k} \subset \partial\Omega_k$. Then let $X_{h,0}(\Omega_k) = \{v \in X_h(\Omega_k) : v(p) = \partial_n v(m) = 0$ for all vertices p and midpoints m in $\Delta_k\}$. We excluded the end elements of $T_h^k(\Gamma_{m,k})$ in the definition of Δ_k because of our condition on the length of the end elements on the interface, see above. The situation is analogous to the case of mortar Crouzeix-Raviart element, cf. [8] where the similar set was introduced.

Each $u \in X_h(\Omega_k)$ is split into two $a_{h,k}$ orthogonal parts: $P_k u$ and discrete biharmonic part of u : $H_k u = u - P_k u$ defined by

$$\begin{cases} a_{h,k}(H_k u, v) = 0 & \text{for all } v \in X_{h,0}(\Omega_k) \\ H_k u(x) = u(x) & \text{for all vertices } x \in \Delta_k \\ (\partial_n H_k u)(m) = \partial_n u(m) & \text{for all midpoints } m \in \Delta_k. \end{cases} \tag{4}$$

We then define $Pu = (P_1 u, \ldots, P_N u)$ and $Hu = u - Pu$ the discrete biharmonic in all subdomains part of u . We also set

$$\tilde{V}_h = HV_h = \{u \in V_h : u \text{ is discrete biharmonic in all } \Omega_k\} \tag{5}$$

Note that each function in $\tilde{V}_h$ is uniquely defined by the values of all degree of freedoms associated with nodal points of $\bigcup_{k=1}^{N} \Delta_k \setminus (\bigcup_{\delta_{m,j} \subset \Gamma} \delta_{m,j,h})$ since the values of the degrees of freedom corresponding to the nodes on the nonmortar (slave) are set by the mortar conditions (2) and that the values of the degrees of freedom of the nodes interior to the subdomains (i.e. not in Δ_k) are set by (4). Note that all $P_k u_h^*$ can be precomputed in parallel and there remains to calculate $\tilde{u}_h^* = Hu_h^* \in \tilde{V}_h$ such that

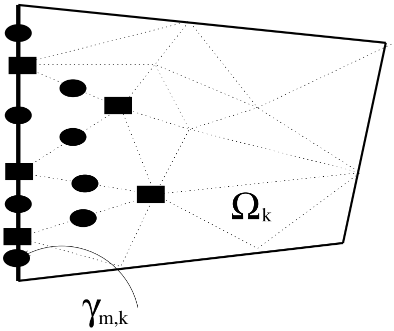
$$a_H(\tilde{u}_h^*, v) = f(v) \quad \forall v \in \tilde{V}_h. \tag{6}$$

4 An additive Schwarz method

Here we describe our Additive Schwarz method for solving (6). We use an abstract ASM scheme, cf. [10], i.e. give the method in terms of the decomposition of $\tilde{V}_h$ into subspaces, we also need bilinear forms defined on these subspaces. We first introduce $\Delta_{\gamma_{m,k}} \subset \Delta_k$, cf.[8], where a similar set was introduced, a set of these vertices and midpoints that are either in $\gamma_{m,k,h}$ or are interior to Ω_k and are on the boundary of the elements $e \in T_h(\Omega_k)$ such that at least one edge of this triangle e is contained in $\overline{\gamma}_{m,k} \subset \partial\Omega_k$ and this edge is neither of the two end elements of $T_h^k(\Gamma_{kl})$, cf. Figure 4.

Then we introduce $V_{\gamma_{m,k}}$ as the subspace of $\tilde{V}_h$ formed by functions such that the respective degrees of freedom related to the crosspoints (the ends of all edges in Γ) and all the vertices and the midpoints in $\bigcup_{\gamma_s \subset \Gamma} \Delta_{\gamma_s} \setminus \Delta_{\gamma_{m,k}}$ are equal to zero. In nodal points on slaves and interior to subdomains (not in Δ_k) the values of the respective degrees of freedom are determined by (2) and (4), respectively. Next we define a coarse space V_0 . It is sufficient to define the values of normal

Fig. 4. The set $\varDelta_{\gamma_{m,k}}$. The midpoints are denoted by circles and the vertices by squares.



derivatives of $u \in V_0$ at the midpoints and the values of u at the crosspoints and in $\bigcup_{\gamma_{m,k} \subset \partial\varOmega_k} \varDelta_{\gamma_{m,k}}$. Note that $\varDelta_{\gamma_{m,k}} \cap \varDelta_{\gamma_{s,k}} = \emptyset, m \neq s$. Let $V_0 \subset \tilde{V}_h$ be formed by all functions $u \in \tilde{V}_h$ such that for any master (mortar) $\gamma_{s,k} \subset \partial\varOmega_k$ there exists linear polynomial $p_s(x,y) = ax + by + c$, i.e. defined over $\varOmega_k$, for which it holds that

$$\begin{array}{ll} u(x) = p_s(x) & \text{for a vertex } x \in \varDelta_{\gamma_{m,k}} \cup \partial\gamma_{m,k} \\ (\partial_n u)(m) = \partial_n p_s(m) & \text{for a midpoint } m \in \varDelta_{\gamma_{m,k}}. \end{array} \tag{7}$$

Here $\partial\gamma_{m,k}$ denotes the two element set containing the end vertices of this mortar. Because $u \in V_0$ is continuous at the crosspoints, it is easy to see that the dimension of V_0 is equal to the number of crosspoints (vertices of subdomain not on $\partial\varOmega$) and the number of masters $\gamma_m \subset \varGamma$.

Again for simplicity of presentation we assume that the bilinear forms for all subspaces equal to $a_H(u,u)$.

Then we can define orthogonal projections: $P_0 : V_0 \rightarrow \tilde{V}_h$ and $P_m : V_{\gamma_m} \rightarrow \tilde{V}_h$ as

$$a_H(P_0 u, v) = a_H(u,v) \quad \forall v \in V_0,$$
$$a_H(P_m u, v) = a_H(u,v) \quad \forall v \in V_{\gamma_m}.$$

Let $P = P_0 + \sum_{k=1}^{N} P_m$. Next we replace problem (6) by

$$P\tilde{u}_h^* = g, \tag{8}$$

where $g = g_0 + \sum_{\gamma_m \subset \varGamma} g_m$ for $g_0 = P_0 \tilde{u}_h^*$ and $g_m = P_m \tilde{u}_h^*$.

We should point out that g_0, g_m can be computed without knowing $\tilde{u}_h^*$. Then we have the following result:

Theorem 1. *For any* $u \in \tilde{V}_h$ *it holds that*

$$c(1 + \log(H/\underline{h}))^{-2} a_H(u,u) \leq a_H(Pu,u) \leq C a_H(u,u),$$

where C, c *are positive constant independent of* H *and any* h_k *and* $H = \max_k H_k$ *and* $\underline{h} = \min_k h_k$ *.*

Sketch of the proof.

The proof of this theorem is based on the abstract ASM scheme, cf. e.g. [10]. We will give only a brief sketch of the proof here. It is enough to check three key assumptions, cf. Th. 2.7, p. 43 in [10]. In our case the assumption II (Strengthened Cauchy-Schwarz Inequalities), cf. Ass. 2.3, p.40 in [10], is satisfied with a constant independent of the number of subdomains by the coloring argument and the constant ω in the assumption III (Local Stability), cf. Ass. 2.4, p.40 in [10], is equal to one as P_0 and P_m are orthogonal projections. It remains to prove assumption I (Stable Decomposition), cf. Ass. 2.2, p.40 in [10], i.e. we have to prove that there exists a positive constant such that for any $u \in \tilde{V}_h$ there are $u_0 \in V_0$ and $u_m \in V_m$ for $\gamma_m \subset \Gamma$ such that $u = u_0 + \sum_{\gamma_m \subset \Gamma} u_m$ and

$$a_H(u_0, u_0) + \sum_{\gamma_m \subset \Gamma} a_H(u_m, u_m) \le C(1 + \log(H/\underline{h}))^2 a_H(u, u). \tag{9}$$

We first define this decomposition. Let $u \in \tilde{V}_h$ and let us define $u_0 \in V_0$. It is sufficient to define the values of the respective degrees of freedom at each $\Delta_{\gamma_{s,k}}$ associated with each mortar $\gamma_{s,k} \subset \Gamma$. Let a, b be the ends of $\gamma_{s,k} \subset \partial\Omega_k$ and $\overline{u}_{\gamma_{s,k}} = \frac{1}{N_k} \sum_{m \in \gamma_{s,k,h}} \partial_n u(m)$, where the sum is taken over all midpoints on $\gamma_{s,k}$ and N_k is the number of those midpoints on $\gamma_{s,k}$. Then for any mortar $\gamma_{s,k} \subset \partial\Omega_k$ we introduce a linear polynomial p_s such that

$$p_s(a) = u(a) \qquad p_s(b) = u(b) \qquad \partial_n p_s = \overline{u}_{\gamma_{s,k}}.$$

Note that the linear polynomial p_s is properly defined by these three conditions. Then we define $u_0 \in V_0$ by setting the values of the respective degrees of freedom associated with the vertices and the midpoints in $\Delta_{\gamma_{s,k}}$ as

$$\begin{aligned} u_0(x) &= p_s(x) \quad \text{for } x \text{ a vertex in } \Delta_{\gamma_{s,k}} \\ \partial_n u_0(m) &= \partial_n p_s(m) \quad \text{for } m \text{ a midpoint in } \Delta_{\gamma_{s,k}} \end{aligned}$$

Thus u_0 is properly defined. Next we define $u_s \in V_{\gamma_{s,k}}$. Again it is sufficient to determine the values of the respective degrees of freedom at the nodal points in $\Delta_{\gamma_{s,k}}$ for all masters $\gamma_{s,k} \subset \Gamma$. Let $w = u - u_0$ and let:

$$\begin{aligned} u_s(x) &= w(x) \quad \text{for } x \text{ a vertex in } \Delta_{\gamma_{s,k}} \\ \partial_n u_s(m) &= \partial_n w(m) \quad \text{for } m \text{ a midpoint in } \Delta_{\gamma_{s,k}} \end{aligned}$$

and $u_s(x) = \partial_n u_s(m) = 0$ for all the vertices x and the midpoints m in $\bigcup_{k=1}^{N} \Delta_n \setminus \Delta_{\gamma_{s,k}}$. It is obvious that we have $u = u_0 + \sum_{\gamma_{m,k}} u_m = u_0 + w = u.$

Then, using a local equivalence operator introduced in [5], some technical tools (modified) from [8] and following the lines of proofs of [7] we can prove (9).

References

1. Y. Achdou, Y. Maday, and O. B. Widlund, *Iterative substructuring preconditioners for mortar element methods in two dimensions*, SIAM J. Numer. Anal., 36 (1999), pp. 551–580.
2. F. B. Belgacem and Y. Maday, *The mortar element method for three dimensional finite elements*, RAIRO Mathematical Modelling and Numerical Analysis, 31 (1997), pp. 289–302.
3. Z. Belhachmi, *Nonconforming mortar element methods for the spectral discretization of two-dimensional fourth-order problems*, SIAM J. Numer. Anal., 34 (1997), pp. 1545–1573.
4. C. Bernardi, Y. Maday, and A. T. Patera, *A New Non Conforming Approach to Domain Decomposition: The Mortar Element Method*, vol. 299 of Pitman Res. Notes Math. Ser., Pitman, 1994, pp. 13–51.
5. S. C. Brenner and L. yeng Sung, *Balancing domain decomposition for nonconforming plate elements*, Numerische Mathematik, 83 (1999), pp. 25–52.
6. C. Lacour, *Non-conforming domain decomposition method for plate and shell problems*, in Tenth International Conference on Domain Decomposition Methods, J. Mandel, C. Farhat, and X.-C. Cai, eds., AMS, 1998, pp. 304–310.
7. L. Marcinkowski, *An additive Schwarz method for mortar Morley finite element problem for 4th order elliptic problem in 2d*. In preparation.
8. ———, *The mortar element method with locally nonconforming elements*, BIT Numerical Mathematics, 39 (1999), pp. 716–739.
9. ———, *A mortar element method for some discretizations of a plate problem*, Numer. Math., 93 (2002), pp. 361–386.
10. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.
11. B. I. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, vol. 17 of Lecture Notes in Computational Science and Engineering, Springer, Berlin, 2001.
12. X. Xu, L. Li, and W. Chen, *A multigrid method for the mortar-type Morley element approximation of a plate bending problem*, SIAM J. Numer. Anal., (2002), pp. 1712–1731.

Local Defect Correction for Time-Dependent Partial Differential Equations

Remo Minero, Martijn J.H. Anthonissen, and Robert M.M. Mattheij

Eindhoven University of Technology, Department of Mathematics and Computer Science, Den Dolech 2, P.O. Box 513, 5600MB Eindhoven, the Netherlands.
{r.minero,m.j.h.anthonissen,r.m.m.mattheij}@tue.nl

Summary. A Local Defect Correction (LDC) method for solving time-dependent partial differential equations whose solutions have highly localized properties is discussed. We present some properties of the technique. Results of numerical experiments illustrate the accuracy and the efficiency of the method.

1 Introduction

Solutions of partial differential equations (PDEs) are often characterized by highly localized properties. Examples are frequently encountered in the area of shock hydrodynamics, transport in turbulent flow fields, combustion, etc. An efficient solution of this kind of problems requires the use of adaptive grid techniques, where a fine grid spacing and, possibly, a small time step are adopted only where high activity occurs. Among other techniques, the Local Defect Correction (LDC) method for time-dependent problems described in [6] has the advantage that only uniform grid and uniform grid solvers need to be used. At each time step, LDC is an iterative process in which a global coarse grid solution and a local fine grid solution are iteratively improved. In particular, the local approximation improves the solution globally through a defect correction.

The LDC method was introduced in [5] for solving elliptic boundary value problems. LDC is a domain decomposition technique in which the local domain fully overlaps the global one. An analysis of LDC in combination with finite differences is presented in [4]. In [1] the method is extended to include adaptivity, multilevel refinement, domain decomposition and regridding. In this paper we present the LDC technique for solving time-dependent PDEs (Section 2) and we discuss some properties of the method (Section 3). Results of numerical experiments illustrate the accuracy and the efficiency of the method (Section 4).

2 The LDC method

We consider the following two-dimensional problem

$$\begin{cases} \dfrac{\partial u(\mathbf{x},t)}{\partial t} = Lu(\mathbf{x},t) + f(\mathbf{x},t), & \text{in } \Omega \times \Theta, \\ u(\mathbf{x},t) = \psi(\mathbf{x},t), & \text{on } \partial\Omega \times \Theta, \\ u(\mathbf{x},0) = \varphi_0(\mathbf{x}), & \text{in } \Omega \cup \partial\Omega, \end{cases} \tag{1}$$

where Ω is a spatial domain, $\partial\Omega$ its boundary and Θ the time interval $(0, t_{\text{end}}]$. Moreover, L is a linear differential operator, f a source term, ψ a Dirichlet boundary condition and φ_0 a given initial condition.

Problem (1) has to be discretized in space and time in order to be solved numerically. For this reason, we introduce a global uniform coarse grid (grid size H), which we denote by Ω^H. We also introduce the time step Δt. We assume that u has, at each time level, a region of high activity that covers a small part of Ω. At time $t_n := n\Delta t$ a coarse grid approximation computed with a time step Δt might not be adequate to represent $u(\mathbf{x}, t_n)$. In order to better capture the local high activity, we introduce a local uniform fine grid (grid size $h < H$), which we denote by Ω_l^h. On Ω_l^h the time integration is performed using a time step $\delta t = \Delta t/\tau$, with τ an integer ≥ 1. In LDC the local solution is used to improve the global approximation through a defect correction.

In the remainder of this section we will assume that a solution $u^{H,h,n-1}$ is known at time t_{n-1} on the *composite grid* $\Omega^{H,h} := \Omega^H \cup \Omega_l^h$, see Fig. 1. It is given by

$$u^{H,h,n-1} := \begin{cases} u_l^{h,n-1}, & \text{in } \Omega_l^h, \\ u^{H,n-1}, & \text{in } \Omega^H \setminus \Omega_l^h, \end{cases} \tag{2}$$

where $u_l^{h,n-1}$ and $u^{H,n-1}$ are a local and a global approximation of $u(\mathbf{x}, t_{n-1})$ respectively. We want to compute an approximation of the solution at the new time level t_n on the composite grid.

The coarse grid problem

A first coarse grid approximation at t_n, $u_0^{H,n}$, can be computed applying the backward Euler method to the PDE in (1). The use of explicit time integrators on the global grid is not of interest in LDC; this is discussed in [6]. We obatin

$$(I - \Delta t\, L^H) u_0^{H,n} = u^{H,h,n-1}|_{\Omega^H} + f^{H,n}\,\Delta t, \tag{3}$$

where L^H is some spatial discretization of L. In (3), $f^{H,n}$ also includes the Dirichlet boundary conditions. We rewrite (3) as

$$M^H u_0^{H,n} = u^{H,h,n-1}|_{\Omega^H} + f^{H,n}\,\Delta t. \tag{4}$$

We assume M^H to be invertible. We denote by Γ the interface between Ω_l and $\Omega \setminus \Omega_l$. For convenience, we partition the coarse grid points as follows

$$\Omega^H = \Omega_l^H \cup \Gamma^H \cup \Omega_c^H, \tag{5}$$

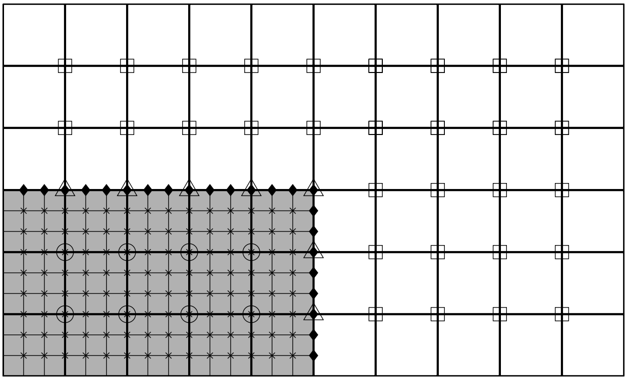


Fig. 1. Example of composite grid $\Omega^{H,h}$.

where $\Omega_l^H := \Omega^H \cap \Omega_l$, $\Gamma^H := \Omega^H \cap \Gamma$ and $\Omega_c^H := \Omega^H \setminus (\Omega_l^H \cup \Gamma^H)$. In Fig. 1 the coarse grid points Ω_l^H are marked with circles, while the points Γ^H and Ω_c^H are denoted by triangles and squares respectively. Assuming that the spatial discretization on the coarse grid is such that the stencil at grid point (x, y) involves at most function values at $(x + iH, y + jH)$, with $i, j \in \{-1, 0, 1\}$, we can rewrite (4) as

$$\begin{pmatrix} M_l^H & B_{l,\Gamma}^H & 0 \\ B_{\Gamma,l}^H & M_\Gamma^H & B_{\Gamma,c}^H \\ 0 & B_{c,\Gamma}^H & M_c^H \end{pmatrix} \begin{pmatrix} u_{l,0}^{H,n} \\ u_{\Gamma,0}^{H,n} \\ u_{c,0}^{H,n} \end{pmatrix} = \begin{pmatrix} u^{H,h,n-1}|_{\Omega_l^H} \\ u^{H,h,n-1}|_{\Gamma^H} \\ u^{H,h,n-1}|_{\Omega_c^H} \end{pmatrix} + \begin{pmatrix} f_l^{H,n}\,\Delta t \\ f_\Gamma^{H,n}\,\Delta t \\ f_c^{H,n}\,\Delta t \end{pmatrix} . \qquad (6)$$

The fine grid problem

In order to formulate a discrete problem on Ω_l^h , we have to define *artificial boundary conditions* on Γ . We can prescribe artificial Dirichlet boundary conditions at time t_n by applying an *interpolation operator in space*, $P^{h,H}$. The operator $P^{h,H}$ maps function values in Γ^H to function values at grid points of the fine grid that lie on the interface, denoted by Γ^h . In Fig. 1 the points Γ^h are marked with small diamonds. If we want to perform time integration with a time step $\delta t = \Delta t/\tau$, we also need to provide boundary conditions on Γ^h at all the intermediate time levels $t_{n-1+k/\tau}$, with $k = 1, 2, \ldots, \tau - 1$. Therefore we perform linear *time interpolation* between $u^{H,h,n-1}|_{\Gamma^h}$ and $P^{h,H} u_0^{H,n}$. A fine grid approximation at time t_n can thus be computed by solving

$$\begin{aligned} M_l^h u_{l,0}^{h,n-1+k/\tau} &= u_{l,0}^{h,n-1+(k-1)/\tau} + f_l^{h,n-1+k/\tau}\,\delta t \\ &- B_{l,\Gamma}^h \left(\frac{k}{\tau} P^{h,H} u_{\Gamma,0}^{H,n} + \frac{\tau - k}{\tau} u^{H,h,n-1}|_{\Gamma^h} \right), \qquad \text{for } k = 1, 2, \ldots, \tau. \quad (7) \end{aligned}$$

The procedure (7) is initialized using

$$u_{l,0}^{h,n-1} = u^{H,h,n-1}|_{\Omega_l^h} . \qquad (8)$$

We can combine all the equations in (7) to express $u_{l,w}^{h,n}$, with $w = 0$, directly in terms of $u^{H,h,n-1}|_{\Omega_l^h}$. We obtain

$$\begin{aligned}\left(M_l^h\right)^{\tau} u_{l,w}^{h,n} &= u^{H,h,n-1}|_{\Omega_l^h} + \sum_{k=1}^{\tau} \left(M_l^h\right)^{k-1} f_l^{h,n-1+k/\tau}\, \delta t \\ &\quad - \sum_{k=1}^{\tau} \left(M_l^h\right)^{k-1} B_{l,\Gamma}^h \left(\frac{k}{\tau} P^{h,H} u_{\Gamma,w}^{H,n} + \frac{\tau-k}{\tau} u^{H,h,n-1}|_{\Gamma^h}\right), \qquad (9)\end{aligned}$$

or

$$\left(M_l^h\right)^{\tau} u_{l,w}^{h,n} = u^{H,h,n-1}|_{\Omega_l^h} + F_l^{h,n}\, \delta t - W_{l,\Gamma}^n\, u_{\Gamma,}^{H,n} + Z_{l,\Gamma}^n\, u^{H,h,n-1}|_{\Gamma^h}. \qquad (10)$$

In (10) $F_l^{h,n}$ depends only on the source term and on the fine grid operator M_l^h, while $W_{l,\Gamma}^n$ and $Z_{l,\Gamma}^n$ only depend on M_l^h and $B_{l,\Gamma}^h$.

Defect correction and LDC iteration

The fine grid approximation is now used to *overall improve* the coarse grid solution at t_n. The fine grid solution is regarded as more accurate than the coarse grid approximation because it is computed with a grid size $h < H$ and a time step $\delta t \leq \Delta t$. The fine grid solution can therefore be used to approximate the local discretization error or *defect* in Ω_l^H. For $w = 1$, the approximated defect is given by (cf. the first equation in (6))

$$\tilde{d}_{l,w-1}^{H,n} := M_l^H R^{H,h} u_{l,w-1}^{h,n} + B_{l,\Gamma}^H u_{\Gamma,w-1}^{H,n} - u^{H,h,n-1}|_{\Omega_l^H} - f_l^{H,n}\, \Delta t, \qquad (11)$$

where $R^{H,h}$ is a restriction operator from the fine to the coarse grid, such that $(R^{H,h} u_{l,w-1}^{h,n})(x,y) = u_{l,w-1}^{h,n}(x,y)$ for every $(x,y) \in \Omega_l^H$. The defect $\tilde{d}_{l,w-1}^{H,n}$ is now added on the right hand side of (6). A more accurate coarse grid approximation can be computed by solving

$$\begin{aligned} M^H u_w^{H,n} &= \begin{pmatrix} u^{H,h,n-1}|_{\Omega_l^H} \\ u^{H,h,n-1}|_{\Gamma^H} \\ u^{H,h,n-1}|_{\Omega_c^H} \end{pmatrix} + \begin{pmatrix} f_l^{H,n}\, \Delta t + \tilde{d}_{l,w-1}^{H,n} \\ f_\Gamma^{H,n}\, \Delta t \\ f_c^{H,n}\, \Delta t \end{pmatrix} \\ &= \begin{pmatrix} 0 \\ u^{H,h,n-1}|_{\Gamma^H} \\ u^{H,h,n-1}|_{\Omega_c^H} \end{pmatrix} + \begin{pmatrix} M_l^H R^{H,h} u_{l,w-1}^{h,n} + B_{l,\Gamma}^H u_{\Gamma,w-1}^{H,n} \\ f_\Gamma^{H,n}\, \Delta t \\ f_c^{H,n}\, \Delta t \end{pmatrix}. \end{aligned} \qquad (12)$$

The new coarse grid solution can be used to update the boundary conditions for a new local problem on Ω_l^h, which in turn will correct the coarse grid approximation. At each time step the LDC method is thus an iterative procedure and, as established in [2] for stationary cases, its convergence is very fast.

Adaptivity

In a time dependent problem it is likely that the high activity moves as time proceeds. As a consequence, the local region Ω_l might be located in different positions and a have a different size or shape at the various time levels in Θ. At each time, in order to perform the next time step Δt, we have to determine a suitable Ω_l. This can be done, for example, by measuring some characteristics of the solution (e.g. slope, gradients, etc.). Many methods have proposed in the literature, see for example [3]. If the composite grid changes in time, we interpolate the solution found at t_{n-1} to the new grid to construct the initial solution $u^{H,h,n-1}$.

3 Properties of the LDC method

In this section we will discuss some properties of the LDC method for time-dependent PDEs. The following lemma shows that once the coarse grid approximations do not change on the interface Γ, the LDC algorithm converges and a fixed point of the iteration has been reached.

Lemma 1. *If $u_{\Gamma,w}^{H,n} = u_{\Gamma,w-1}^{H,n}$ for a certain index w, then the LDC iteration converges and*

$$u_q^{H,n} = u_w^{H,n}, \qquad u_q^{h,n} = u_w^{h,n}, \tag{13}$$

for all $q = w, w+1, \ldots$.

Proof. Assume that $u_{\Gamma,w}^{H,n} = u_{\Gamma,w-1}^{H,n}$ for a certain index w. From (10), we have that $u_w^{h,n} = u_{w-1}^{h,n}$, and hence, from (12),

$$M^H u_{w+1}^{H,n} = \begin{pmatrix} 0 \\ u^{H,h,n-1}|_{\Gamma^H} \\ u^{H,h,n-1}|_{\Omega_c^H} \end{pmatrix} + \begin{pmatrix} M_l^H R^{H,h} u_{l,w}^{h,n} + B_{l,\Gamma}^H u_{\Gamma,w}^{H,n} \\ f_\Gamma^{H,n}\,\Delta t \\ f_c^{H,n}\,\Delta t \end{pmatrix}$$

$$= \begin{pmatrix} 0 \\ u^{H,h,n-1}|_{\Gamma^H} \\ u^{H,h,n-1}|_{\Omega_c^H} \end{pmatrix} + \begin{pmatrix} M_l^H R^{H,h} u_{l,w-1}^{h,n} + B_{l,\Gamma}^H u_{\Gamma,w-1}^{H,n} \\ f_\Gamma^{H,n}\,\Delta t \\ f_c^{H,n}\,\Delta t \end{pmatrix} = M^H u_w^{H,n}$$

Because we have assumed that M^H is invertible, we have $u_{w+1}^{H,n} = u_w^{H,n}$, for all grid points in Ω_H. Since $\Gamma_H \subset \Omega_H$, we have $u_{\Gamma,w+1}^{H,n} = u_{\Gamma,w}^{H,n}$. By induction, we find $u_q^{H,n} = u_w^{H,n}$ and $u_q^{h,n} = u_w^{h,n}$, for all $q = w, w+1, \ldots$

♯

We can combine (12), (11) and (10), and express the LDC iteration as

$$\begin{pmatrix} \left(M_l^h\right)^\tau & 0 & W_{l,\Gamma}^n & 0 \\ 0 & M_l^H & B_{l,\Gamma}^H & 0 \\ 0 & B_{\Gamma,l}^H & M_\Gamma^H & B_{\Gamma,c}^H \\ 0 & 0 & B_{c,\Gamma}^H & M_c^H \end{pmatrix} \begin{pmatrix} u_{l,w}^{h,n} \\ u_{l,w}^{H,n} \\ u_{\Gamma,w}^{H,n} \\ u_{c,w}^{H,n} \end{pmatrix} = \begin{pmatrix} 0 & 0 & 0 & 0 \\ M_l^H R^{H,h} & 0 & B_{l,\Gamma}^H & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} u_{l,w-1}^{h,n} \\ u_{l,w-1}^{H,n} \\ u_{\Gamma,w-1}^{H,n} \\ u_{c,w-1}^{H,n} \end{pmatrix}$$

$$+ \begin{pmatrix} u^{H,h,n-1}|_{\Omega_l^h} \\ 0 \\ u^{H,h,n-1}|_{\Gamma_H} \\ u^{H,h,n-1}|_{\Omega_c^H} \end{pmatrix} + \begin{pmatrix} F_l^{h,n}\,\Delta t \\ 0 \\ f_\Gamma^{H,n}\,\delta t \\ f_c^{H,n}\,\delta t \end{pmatrix} + \begin{pmatrix} Z_{l,\Gamma}^n\, u^{H,h,n-1}|_{\Gamma_H} \\ 0 \\ 0 \\ 0 \end{pmatrix}. \qquad (14)$$

We rewrite (14) using the short notation

$$M^{H,h} u_w^{H,h,n} = S^{H,h} u_{w-1}^{H,h,n} + \tilde{u}^{H,h,n-1} + \tilde{f}^{H,h,n} + \tilde{z}^{H,h,n-1}. \qquad (15)$$

If the LDC algorithm converges, then (15) has a fixed point, which we denote by $u^{H,h,n}$ (we remove the subscript that numbers the LDC iterations). The fixed point $u^{H,h,n}$ satisfies by definition

$$M^{H,h} u^{H,h,n} = S^{H,h} u^{H,h,n} + \tilde{u}^{H,h,n-1} + \tilde{f}^{H,h,n} + \tilde{z}^{H,h,n-1}. \qquad (16)$$

The following theorem states that, if the LDC iteration converges, the fine and the coarse grid approximation coincide in points common to the fine and coarse grids.

Theorem 1. *Assume that the LDC iteration converges. Then* $u^{H,h,n}$ *is such that the projection of* $u_l^{h,n}$ *on the local coarse grid equals* $u_l^{H,n}$ *, viz.*

$$R^{H,h} u_l^{h,n} = u_l^H \qquad (17)$$

Proof. Combining (16) and (14) yields

$$\begin{pmatrix} \left(M_l^h\right)^\tau & 0 & W_{l,\Gamma}^n & 0 \\ -M_l^H R^{H,h} & M_l^H & 0 & 0 \\ 0 & B_{\Gamma,l}^H & M_\Gamma^H & B_{\Gamma,c}^H \\ 0 & 0 & B_{c,\Gamma}^H & M_c^H \end{pmatrix} \begin{pmatrix} u_l^{h,n} \\ u_l^{H,n} \\ u_\Gamma^{H,n} \\ u_c^{H,n} \end{pmatrix} = \tilde{u}^{H,h,n-1} + \tilde{f}^{H,h,n} + \tilde{z}^{H,h,n-1}. \qquad (18)$$

The second equation of the system reads

$$M_l^H R^{H,h} u_l^{h,n} + M_l^H u_l^{H,n} = 0, \qquad (19)$$

which gives (17), since we supposed M^H (and hence M_l^H) to be invertible.

♯

We finally write the system of equations that the limit of the LDC iteration satisfies at time t_n .

Theorem 2. *Assume that the LDC iteration converges. Then* $u_l^{h,n}$, $u_\Gamma^{H,n}$ *and* $u_c^{H,n}$ *satisfy the following system of equations*

$$\begin{pmatrix} \left(M_l^h\right)^\tau & W_{l,\Gamma}^n & 0 \\ B_{\Gamma,l}^H R^{H,h} & M_\Gamma^H & B_{\Gamma,c}^H \\ 0 & B_{c,\Gamma}^H & M_c^H \end{pmatrix} \begin{pmatrix} u_{l,w}^{h,n} \\ u_{\Gamma,w}^{H,n} \\ u_{c,w}^{H,n} \end{pmatrix}$$
$$= \begin{pmatrix} u^{H,h,n-1}|_{\Omega_l^h} \\ u^{H,h,n-1}|_{\Gamma_H} \\ u^{H,h,n-1}|_{\Omega_c^H} \end{pmatrix} + \begin{pmatrix} F_l^{h,n}\,\Delta t \\ f_\Gamma^{H,n}\,\delta t \\ f_c^{H,n}\,\delta t \end{pmatrix} + \begin{pmatrix} Z_{l,\Gamma}^n\, u^{H,h,n-1}|_{\Gamma_H} \\ 0 \\ 0 \end{pmatrix}. \quad (20)$$

Proof. Elimination of $u_l^{H,n}$ from (18) gives (20).

♯

We notice that (20) implies a discretization on the composite grid, while, for solving that system, we have only used uniform grids and uniform grid solvers.

4 Numerical experiments

In this section we present the results of a $2D$ numerical experiment. We solve the following time-dependent convection-diffusion equation

$$\frac{\partial u}{\partial t} + \nabla u = \nabla^2 u + f, \quad (21)$$

in $\Omega = (0,2) \times (0,1)$. The initial condition, the boundary condition and the source term f are chosen is such a way that the exact solution of the problem is

$$u = 3 - \tanh\left(25\left(x-t\right) + 5\left(y-1\right)\right). \quad (22)$$

At all times, the exact solution (22) has a region of high activity that covers a small part of Ω . The problem is solved by means of LDC with different values of H , h , Δt and δt . The spatial discretization is by finite differences both globally and locally. The backward Euler scheme is used for the time discretization both on the global and the local grid. The local region is chosen in such a way that at time level t_n

$$\Omega_l = \big((t_n - 0.2, t_n + 0.4) \times (0,1)\big) \cap \Omega. \quad (23)$$

In our tests we perform only one LDC iteration per time step. As a comparison, we also solve problem (21) using a single uniform global grid with grid size $h_{\text{unif}} = h$ and time step $\delta t_{\text{unif}} = \delta t$. At time $t = 0.6$ we measure the maximum error $\varepsilon_{\max}$ of the numerical approximations with respect to the exact solution (22). Table 1 shows that LDC can achieve practically the same accuracy as the uniform grid solver. Of course LDC is computanionally less expensive than the uniform grid solver since the fine grid spacing and the small time step are adopted only in a limited portion of the domain.

Table 1. Results of the numerical experiment.

Grid size		Time step		$\varepsilon_{\max}$	
H	$h = h_{\mathrm{unif}}$	Δt	$\delta t = \delta t_{\mathrm{unif}}$	LDC	Unif. grid
1/10	$H/3$	$1.0{\cdot}10^{-1}$	$\Delta t/3$	$4.36{\cdot}10^{-2}$	$4.33{\cdot}10^{-2}$
1/10	$H/5$	$1.0{\cdot}10^{-1}$	$\Delta t/5$	$1.21{\cdot}10^{-2}$	$1.18{\cdot}10^{-2}$
1/20	$H/3$	$2.5{\cdot}10^{-2}$	$\Delta t/3$	$9.50{\cdot}10^{-3}$	$9.50{\cdot}10^{-3}$
1/20	$H/5$	$2.5{\cdot}10^{-2}$	$\Delta t/5$	$3.02{\cdot}10^{-3}$	$3.02{\cdot}10^{-3}$

References

1. M. J. Anthonissen, *Local defect correction techniques applied to a combustion problem*, in Proceedings of the 15th international conference on Domain Decomposition Methods, R. Kornhuber, R. H. W. Hoppe, J. Péeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., vol. 40 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2004, pp. 185–192.
2. M. J. Anthonissen, R. M. Mattheij, and J. H. ten Thije Boonkkamp, *Convergence analysis of the local defect correction method for diffusion equations*, Numer. Math., 95 (2003), pp. 401–425.
3. B. A. V. Bennett and M. D. Smooke, *Local rectangular refinement with application to nonreacting and reacting fluid flow problems*, J. Comput. Phys., 151 (1999), pp. 684–727.
4. P. J. Ferket and A. A. Reusken, *Further analysis of the local defect correction method*, Computing, 56 (1996), pp. 117–139.
5. W. Hackbusch, *Local defect correction and domain decomposition techniques*, in Defect Correction Methods: Theory and Applications., K. Böhmer and H. J. Stetter, eds., vol. 5 of Computing Supplementa, Springer Wien, 1984, pp. 89–113.
6. R. Minero, M. J. Anthonissen, and R. M. Mattheij, *A local defect correction technique for time-dependent problems*, Numer. Methods Partial Differential Equations, 22 (2006), pp. 128–144.

Extending the p-Version of Finite Elements by an Octree-Based Hierarchy

R.-P. Mundani[1], H.-J. Bungartz[1], E. Rank[2], A. Niggl[2], and R. Romberg[2]

[1] Fakultät für Informatik, Technische Universität München, Germany.
[2] Lehrstuhl für Bauinformatik, Technische Universität München, Germany.

Summary. In structural mechanics, a large variety of finite element approaches are used, some of them – especially of p-type – without an inherent hierarchical substructuring. This often turns out to be a drawback. By embedding the finite element decomposition into an octree structure, the elements can be arranged in a hierarchical way, which does not only open the door to efficient iterative solvers based on the classical nested dissection algorithm, but also allows to speed up the solution process in case only parts of the underlying geometric model are changed, as only those parts and their region of direct influence have to be recomputed.

In this paper, we present an efficient method to map an octree-based hierarchy onto an arbitrary finite element mesh, to use this octree structure for implementing a fast iterative solver of nested dissection type, and to set up a framework for completely embedded simulation processes as they, for example, appear in many civil engineering applications.

1 Motivation

The cooperation of different simulation tasks often suffers from proprietary data representations – surface-oriented or volume-oriented, for instance – and insufficient interfaces between single processes. In [6] we presented a framework for process integration for applications from the field of structural engineering, where global consistency among all participants as well as a common data model for all kind of simulation tasks is achieved by octree-based methods.

In this paper, we present an octree-based approach to arrange the elements resulting from the p-version of a finite element (FE) discretisation in a hierarchical way. Thus, we can apply efficient iterative solvers based on the classical nested dissection algorithm. Furthermore, this hierarchical substructuring can also be exploited for a faster computation of the solution in case some geometric modification occurs. As only those parts of the octree influenced directly by a geometric alteration have to be recomputed, the time for obtaining the solution can be significantly reduced. Hence, even computations in real time are possible.

By embedding this hierarchical substructuring approach into the framework mentioned above, different simulation tasks can be handled in a more efficient way. Thus, for any kind of problem the framework can provide – like a construction kit – a specific and unique solution, a so called problem solving environment.

2 Octrees

Many simulation tasks, nowadays, are based on hierarchical data structures or octrees, in particular, as they have turned out to be advantageous for a huge amount of different tasks. Octrees, that is recursively halving a cube containing the entire geometry in each direction, as long as the resulting cells – aka voxels – are lying completely inside or outside the geometry. Thus, the overall amount of necessary cells is reduced from $\mathcal{O}(n^3)$ for an equidistant discretisation to $\mathcal{O}(n^2)$. By a new technique first presented in [7], we are able to create these octrees in real time and even *on-the-fly* also for larger (greater than 12) levels of recursion. Especially in the field of numerical simulation, octrees provide a big potential for mostly all kinds of problems due to their fast and easy access of the underlying geometry.

To address each cell by some uniqe identifier, the so called Morton index is used. By naming a node's eight sons from '0' to '7' in some specific order[3], one can obtain the Morton index of a cell by accumulating all node's numbers on its way down from the root to the desired cell. One main advantage of these identifiers is the possibility of easily determining neighbouring cells, an important aspect when degrees of freedom resulting from an FE discretisation have to be assigned to their corresponding nodes in an octree.

3 *p*-Version of Finite Elements

The *p*-version of the finite element method has turned out to be an efficient discretisation strategy for solving finite element problems arising in structural engineering. In contrast to the classical *h*-version approach, the *p*-version leaves the mesh unchanged and increases the polynomial degree of the shape functions in order to reduce the error of approximation. Our *p*-version implementation uses hierarchical shape functions for the displacement Ansatz, following Szabó and Babuška [8]. Contrary to the classical approach for higher order modes, the hierarchical bases are constructed such that all lower order shape functions are completely contained in the higher order bases. Thus, the finite element basis can be easily extended up to any desired polynomial degree without changing the complete set of shape functions for each different polynomial degree.

In the work shown here, the finite element computation is based on a fully three-dimensional approach using hexahedral elements. The shape functions of the three-dimensional hexahedral Ansatz spaces are constructed by forming the tensor product of the one-dimensional bases. One important property is that the hexahedral *p*-version elements are very robust w. r. t. element distortions – aspect ratios up to a factor of 1000 are possible. This makes it possible to use them equally for solid "thick"

[3] The order itself is not relevant, but it has to be consistent among all nodes.

structures as well as for thin walled, shell-like structures [3]. The computational effort can further be decreased by using different polynomial degrees in different directions. For shell like structures, for example, this *anisotropic* Ansatz space allows us to reduce the polynomial degree in the thickness direction while leaving the polynomial degree in in-plane direction unchanged [4].

With this approach, large structures can be computed using the same element formulation consistently for the whole domain. Figure 1 shows the computation results of the structural model of an office tower under vertical load on all plates. The

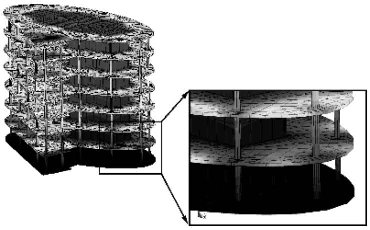


Fig. 1. Structural model of an office tower consisting of 11762 hexahedral p-version elements; displacement field of structure with zoomed view.

example was computed with a moderately high polynomial degree of $p = 5$, which reflects the global behaviour of the system accurately enough. By using the hierarchical organisation in octrees, it is possible to zoom into the structure to a certain level in order to locally refine the computation simply by increasing the polynomial degree. Or, after indentifying critical areas on the global level, it is possible to perform design studies locally in order to explore different design alternatives. Thus, using the hierarchical approach presented in this paper, all these local computations can be done without recomputing the whole domain or without losing global consistency.

4 Hierarchical Approach

Before any hierarchical solver can be applied to a finite element discretisation, the corresponding data – stiffness matrices and load vectors – have to be set up in a hierarchical way. Starting with an octree generation for the elements itself results in a hierarchical sorting according to some criteria such as the elements' centre, for instance. In a second step, all degrees of freedom (DOF) can be assigned to their corresponding nodes by evaluating the elements' Morton indices. Once finished with this initial setup, the system can be processed with a solver of nested dissection type, e. g., consisting of a bottom-up assembly and top-down solution step.

4.1 Building a Finite Element Hierarchy

To sort the elements of a FE discretisation in a hierarchical way, each element has to be separately assigned to one of the octree's cells. To reduce the computational effort while generating the corresponding octree all elements are represented by their centre only. Without loss of generality this could be any arbitrary point of an element, such as a corner, as long as there's no other element with the same representative. For n elements this conforms to a set P of n point coordinates x, y, and z.

Under the assumption of storing exactly one point $p \in P$ in each cell, an octree representation for set P can be easily derived (see Fig. 2). All other cells stay empty and are not relevant as long as the initial finite element mesh isn't altered. For all non-empty cells the corresponding element data – stiffness matrix and load vector – can already be stored at the same location as well as needed for the later computations.

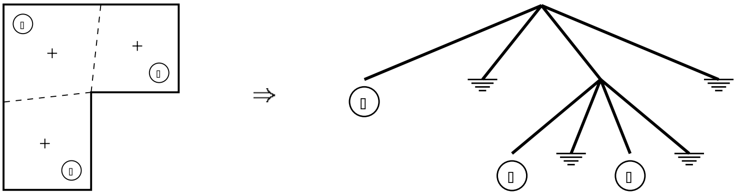


Fig. 2. A sample FE discretisation in 2D with three elements (left-hand side, '+' indicates an element's centre) and the corresponding quadtree – an octree's 2-dimensional counterpart – on the right-hand side.

4.2 Assigning Degrees of Freedom

To finish the setup step all DOFs have to be assigned to the octree, too. As a DOF might belong to more than one element, the lowest common father node (LCF) of all involved elements has to be found. Lowest means the last node visited within a top-down descent, starting at the root node, from which all corresponding elements can still be reached; the octree's root obviously forms a common father node for any arbitrary element. The lower one DOF can be assigned to the octree, the better for the later computations, because it can be eliminated earlier during the nested dissection's assembly.

Finding the LCF of some elements is achieved by comparing the respective Morton indices. They are read number by number from the left-hand side as long as they match. The resulting Morton index then indicates the LCF where the corresponding DOF has to be stored. In the worst case the result is empty, thus, the LCF is the root node. Assume, all of the quadtree's sons in the right part of Fig. 3 are labelled from '0' to '3' from left to right. The LCF of element 2 (Morton index '20') and element 3 (Morton index '22') is '2', the LCF of element 2 and element 1 (Morton index '0') is ' ' (empty) and, thus, the root node.

Assigning all DOFs to the octree finishes the setup and preparatory work. The original finite element mesh is no longer necessary as all further computations are directly processed on the tree structure. Exploiting this hierarchical ordering of elements/DOFs by a nested dissection algorithm is discussed in the next section. Figure 3 shows the DOF distribution for the small example from above.

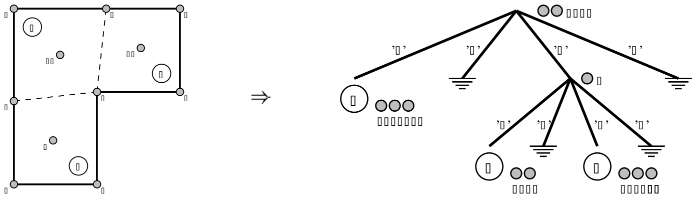


Fig. 3. Assuming the following DOFs (light circles) for the sample FE discretisation on the left-hand side, the final DOF assignment according to the elements' Morton indices is shown on the right-hand side.

4.3 Nested Dissection

Applying a nested dissection algorithm on finite element meshes was done very early by J.A. George [5]. The main idea behind this technique is to decompose the system of linear equations (SLE) into some smaller parts and to eliminate in a bottom-up step local unknowns, i. e. unknowns only partially describing the SLE at this point, before in a final top-down step the solution can be computed. Some more information about nested dissection, especially for solving the convection-diffusion-equation, can be found in [1].

In our case, the decomposition step can be skipped, because it was implicitly done when generating the finite element mesh for some geometric model. Hence, creating an octree for all elements and assigning the corresponding DOFs to the octree's nodes is all the work that is necessary. For precise time measurements related to this setup step, see the results given in Sect. 5.

Once all preparatory work is finished, a bottom-up assembly is initiated. Therefore, each stiffness matrix is first rearranged that way. Thus, all unknowns are separated into blocks of inner (I) – a corresponding DOF is stored in that node – and outer (O) ones, leading to four blocks II, IO, OI, and OO. If one node does not contain any DOFs at all, all of the stiffness matrix's unknowns are treated as outer ones, hence, only a OO block results.

Thus, the SLE $K \cdot u = d$ with stiffness matrix K, solution vector u, and load vector d can be written as

$$\left(\begin{array}{c|c} K_{II} & K_{IO} \\ \hline K_{OI} & K_{OO} \end{array}\right) \cdot \left(\frac{u_I}{u_O}\right) = \left(\frac{d_I}{d_O}\right) . \qquad (1)$$

Evaluating (1) leads to

$$K_{II} \cdot u_I + K_{IO} \cdot u_O = d_I \qquad \text{and} \qquad K_{OI} \cdot u_I + K_{OO} \cdot u_O = d_O \,,$$

which can be rewritten as

$$\big(K_{OO} - K_{OI} \cdot K_{II}^{-1} \cdot K_{IO}\big) \cdot u_O = d_O - K_{OI} \cdot K_{II}^{-1} \cdot d_I \,. \tag{2}$$

In (2) any influence of the inner unknowns u_I has been eliminated, thus, the resulting SLE only depends on the outer unknowns u_O that are stored somewhere higher in the tree. There exist several methods of computing the Schur complement $\widetilde{K}_{OO} := K_{OO} - K_{OI} \cdot K_{II}^{-1} \cdot K_{IO}$. In our approach, we've chosen a direct method by applying a Gaussian elimination.

The Schur complement is then passed to the node's father, that assembles it with all other Schur complements of its sons. For the newly formed SLE the same steps are applied until the root node is reached. Here, all resulting unknowns are only inner ones, hence, the SLE can be solved. This solution is passed to the root's sons that now can modify their right side and solve the SLE for their inner unknows. Successively passing the solution to all of a node's sons until a leaf is reached results in the entire solution vector u.

4.4 Exploiting the Hierarchy

A huge advantage of this hierarchical approach lies in the reduction of computations whenever the underlying geometric model changes. As described above, the p-version of the finite element method allows the alteration of single elements without the necessity of a complete FE mesh generation from scratch. Thus, only parts of the tree have to be reassembled for the new stiffness matrices before the new solution vector can be computed.

Assume, the stiffness of one element changes. Starting from the root node all *Schur* complements of nodes visited on the way down to the node representing this element are obsolete and can be discarded. In fact, the number of necessary assembly steps is directly related to the node's depth in the tree. Whenever a new assembly is initiated, an effort has to be invested only for those nodes without Schur complement, since for all others the Schur complements still exist from the last pass. This obviously diminishes the overall amount of computing time as you can see in the results presented in the next section.

Embedding this approach into the framework presented in [6] allows participating experts to study different alternative models in a more efficient way due to shorter computing times in case of geometry alterations and local refinements, resp.

5 Some Examples and Results

To show the potential behind this hierarchical approach a sample prototype was implemented. Based on two different scenarios with different polynomial degree times were meassured for the setup step, the assembly step, and the solution step. Afterwards some elements were exchanged by newer versions[4] and the times were meassured for the reassembly as well as for the new solution step. All computations ~~were done o~~n an Intel Pentium 4 with 3.4 GHz under Linux.

[4] The time ($\ll 0.01\,s$) for exchanging some stiffness matrices can be neglected.

5.1 Example 1: A Simple Cube

This artificial example shows a cube with an equidistant discretisation in each direction, consisting of 144 elements in total. It has 2625 DOFs ($p = 2$) and 7935 DOFs ($p = 4$), resp. The octree necessary to store all data has a depth of four, counting the root level as zero. For simulating a geometry alteration an element on the lowest level was updated with a new stiffness matrix and load vector. Our results are given in the following table.

Name	DOFs R	DOFs T	Setup	Assembly	Solution	Reassembly	Solution
cube_p2	897	2625	0.42 s	0.21 s	0.11 s	0.03 s	0.10 s
cube_p4	2319	7935	2.66 s	6.74 s	1.03 s	0.50 s	1.02 s

As we can see, the times for a reassembly – possible because of our approach – are significant smaller than those for the initial assembly. The more complex the problem becomes (total amout of DOFs) the more benefit can be achieved. One thing that also could be observed is a declining percentage of DOFs stored on the root level (in the following table labelled as 'DOFs R' for the root level and as 'DOFs T' for the overall amount). Nevertheless, this example with nearly 30 % is really the worst.

5.2 Example 2: An Office Tower

The second and more realistic example consists of two floors from an office tower that can be visited in Vienna[5] (also see Fig. 1). It consists of 4171 elements and was computed for polynomial degrees $p = 1$ (23856 DOFs) and $p = 2$ (84660 DOFs). The necessary octree for storage has a depth of eight. Compared to the example above, a much better percentage of DOFs on the root level could be observed. For $p = 2$ only 8 % of all DOFs are stored there.

Name	DOFs R	DOFs T	Setup	Assembly	Solution	Reassembly	Solution
uniqa_p1	2544	23856	1.89 s	19.27 s	12.64 s	5.14 s	12.09 s
uniqa_p2	6414	84660	10.94 s	343.21 s	77.28 s	14.77 s	76.36 s

Here, also the times for a reassembly are much smaller than for the initial assembly – around 15 s instead of 343 s for $p = 2$. If we take into account that a solution

[5] http://tower.uniqa.at

of the entire system with all 84660 DOFs takes nearly 200 s (cg algorithm) and in case of a geometry alteration everything has to be computed from scratch, the 90 s (re-assembly plus solution) are only half that time.

Currently, most of the time (approx. 99 %) ist spent at the root level because of our simple cg solver; a hierarchical multilevel preconditioner will reduce this effect and, thus, emphasize the advantages of our approach even more. Nevertheless, the efficiency of this approach has been shown. The next steps will comprise testing different solver strategies for the root level and a parallelisation. Thus, even larger problems with higher polynomial degree can be computed.

6 Conclusion

In this paper, we have presented an octree-based approach to set up a hierarchy for the p-version of the finite element method. It has been shown that this approach reduces the necessary computations for geometric alterations, as only parts directly influenced have to be recomputed. Thus, studies of alternative models and local refinements become very attractive due to the reduced computing times compared to the standard approaches that always have to start from scratch. Finally, this is the first step for developing completely embedded simulation processes as they appear in many technical applications.

References

1. M. Bader, *Robuste, parallele Mehrgitterverfahren für die Konvektions-Diffusions-Gleichung*, PhD thesis, Fakultät für Informatik, Technische Universität München, 2001.
2. A. Düster, *High Order Finite Elements for Three-Dimensional, Thin-Walled Nonlinear Continua*, PhD thesis, Lehrstuhl für Bauinformatik, Technische Universität München, 2001.
3. A. Düster, H. Bröker, and E. Rank, *The* p*-version of the finite element method for three-dimensional curved thin-walled structures*, Internat. J. Numer. Methods Engrg., 52 (2001), pp. 673–703.
4. A. Düster, D. Scholz, and E. Rank, pq-*Adaptive solid finite elements for three-dimensional plates and shells*, Comput. Methods Appl. Mech. Engrg., (2005). Submitted.
5. A. George, *Nested dissection of a regular finite element mesh*, SIAM J. Numer. Anal., 10 (1973), pp. 345–363.
6. R.-P. Mundani and H.-J. Bungartz, *An octree-based framework for process integration in structural engineering*, in Proc. of the 8th World Multi-Conf. on Systemics, Cybernetics and Informatics, N. C. et al., ed., 2004.
7. R.-P. Mundani, H.-J. Bungartz, E. Rank, R. Romberg, and A. Niggl, *Efficient algorithms for octree-based geometric modelling*, in Proc. of the Ninth Int. Conf. on Civil and Structural Engineering Computing, B. Topping, ed., Civil-Comp Press, 2003.
8. B. Szabó and I. Babuška, *Finite Element Analysis*, John Wiley & Sons, New York, 1991.

The Multigrid/τ-extrapolation Technique Applied to the Immersed Boundary Method

Francois Pacull [1] and Marc Garbey [2]

[1] Mathematics Department, University of Houston, Houston, TX 77004, USA. fpacull@math.uh.edu

[2] Computer Science Department, University of Houston, Houston, TX 77004, USA. garbey@cs.uh.edu

Summary. The Immersed Boundary Method (IBM), originally developed by Peskin [5], is a very practical method of simulating fluid-structure interactions. It combines Eulerian and Lagrangian descriptions of flow and moving elastic boundaries using Dirac delta functions. Incompressible Navier-Stokes (NS) and elasticity theory can be unified by the same set of equations to get a combined model of the interaction.

There are numerous applications of the IBM in bio-engineering and in more general computational fluid dynamics applications.

We present a numerical study of the accuracy and computational cost of the method, in a framework of finite differences, based on the implementation of several mathematical tools such as multigrid solvers, τ-extrapolation technique, multilevel discretization and more generally numerical methods for differential equations with singular source terms. These implementations are being made on test cases that are relevant for the IBM applications, keeping in mind that we want to keep the simplicity of the method.

1 The IBM

While we are using a more sophisticated time stepping scheme [6], let us start with the basic projection scheme introduced by Chorin [2] for the incompressible NS equations:

1- *Prediction step*

$$\rho \left[\frac{V^* - V^n}{\Delta t} + (V^n . \nabla) V^n \right] - \mu \Delta V^* = F^n; \qquad (1)$$

2- *Pressure evaluation step*

$$\varDelta P^{n+1} = \frac{\rho}{\varDelta t}\nabla . V^*; \tag{2}$$

3- *Correction step*

$$\rho\left[\frac{V^{n+1}-V^*}{\varDelta t}\right] + \nabla P^{n+1} = 0. \tag{3}$$

The notations are as follow: V , P , ρ and μ are respectively the velocity, pressure, uniform density and viscosity coefficient of the fluid. F is the force term, $\varDelta t$, the time step, $\varOmega$, the domain. In this scheme we have a non-conservative convection term, an explicit force term and a semi-implicit diffusion term.
Let $f(s,t)$ be the elastic force density along $\varGamma$. The boundary immersed in the fluid is represented in the Cartesian mesh by $X(s,t)$, where $0 \leq s \leq 1$ is the curvilinear coordinate and $0 \leq t \leq T$ is time. The force term F in Eq. (1) is obtained as follows:

$$F(x,t) = \int_{\varGamma} f(s,t)\delta(x - X(s,t))ds, \;\; (x,t) \in \varOmega \times [0,T]. \tag{4}$$

It is ideally zero everywhere except along $\varGamma$. In the computations, the δ function is regularized by a discrete Dirac delta function of compact support. Let us describe the force term in the two-dimensional case after discretization of the immersed boundary, without considering the time dependency:

$$F_h(x) = h_{\varGamma}\sum_{j=1}^{M} f(s_j)\delta_h(x - X(s_j)), \;\; x \in \varOmega. \tag{5}$$

The immersed boundary is then a one dimensional curve with this discrete mesh: $s_j = \dfrac{j-1}{M-1} = (j-1)h_{\varGamma}, \;\; 1 \leq j \leq M.$
The NS equations implemented with a finite-difference method is of order two in space because of the discretization error, but the order is reduced in the IBM by the discretization of the force term.
If we look at the prediction step (Eq. (1)) of the projection scheme for the NS equations, we have:

$$(I - \nu \varDelta t \varDelta)V^* = RHS^n, \tag{6}$$

where the right-hand side contains singular components, essentially due to the discrete force term that is a sum of discrete Dirac delta functions.
If we look at the pressure correction step (Eq. (2)), we have:

$$\varDelta(\delta P)^n = RHS^*, \tag{7}$$

where the right-hand side also contains singular components, but in the form of dipoles. Consequently, we will focus our study on elliptic equations with singular source terms and more specifically on these two operators $I - k^2\varDelta \; (k \in \mathbb{R})$ and $\varDelta$. The standard IBM is first order in space. Our main goal is to get an order of accuracy larger than one and fast solvers for problems (6) and (7).

2 The discrete Dirac delta function δ_h

Let us introduce the discrete Dirac delta functions. They are usually written in this form in 1D : $\delta_h(x) = \frac{1}{h}\phi(\frac{x}{h})$. The function ϕ needs to satisfy several compatibility conditions:

(a) $\phi \in C^0(\mathbb{R})$.

(b) ϕ has to be of finite support, since the computational cost of the method is proportional to its width.

(c) If we are using the staggered mesh, as introduced by Harlow and Welch [4], which requires a regular, rather than a wide, stencil for the Laplace operator in the pressure equation, we just need to have:

$$\sum_{i\in\mathbb{Z}} \phi(r-i) = 1 \quad \forall r \in \mathbb{R},$$

which guarantees that constant functions are interpolated exactly by δ_h . If we are not using the staggered mesh, we have the condition:

$$\sum_{i(even)} \phi(r-i) = \sum_{i(odd)} \phi(r-i) = \frac{1}{2} \quad \forall r \in \mathbb{R}.$$

(d) $\sum_{i\in\mathbb{Z}} [\phi(r-i)]^2 = C \quad \forall r \in \mathbb{R}$, where C is a constant. That ensures that

$$\sum_{i\in\mathbb{Z}} \phi(r_1-i)\phi(r_2-i) \leq C \quad \forall (r_1,r_2) \in \mathbb{R}^2.$$

J. M. Stockie wrote [9] that it “is analogous to the physically reasonable requirement that when two fiber points interact, the effect of one boundary point on the other is maximized when the points coincide”.

(e) $\sum_{i\in\mathbb{Z}} (r-i)\phi(r-i) = 0 \quad \forall r \in \mathbb{R}$, which ensures along with property 3 that linear functions are interpolated exactly by δ_h .

The minimal width support of a function satisfying these requirements on a traditional mesh is $2h$. It is then defined uniquely, as presented by Peskin [6]. For the staggered mesh, a function with support $\frac{3}{2}h$ is uniquely determined too [7]:

$$\phi(r) = \begin{cases} \frac{1}{6}\left(5 - 3|r| - \sqrt{-3(1-|r|)^2+1}\right), & 0.5 \leq |r| \leq 1.5; \\ \frac{1}{3}\left(1 + \sqrt{-3r^2+1}\right), & |r| \leq 0.5; \\ 0, & \text{otherwise.} \end{cases} \tag{8}$$

This function described in Eq. (8) gives an IBM that is somewhat faster computationally due to the fact that the support is $\frac{3}{2}h$ instead of $2h$. Engquist and Tornberg showed [3] that the discretization error is proportional to the number of moment conditions satisfied by the function. But adding some moment conditions requires increasing the support. If we keep a discrete delta function that has a $2h$

support with the staggered mesh instead of a $\frac{3}{2}h$ [7], we can then increase the accuracy of the method using the following piecewise cubic function [3]:

$$\phi(r) = \begin{cases} 1 - \frac{1}{2}|r| - |r|^2 + \frac{1}{2}|r|^3, & 0 \leq |r| \leq 1; \\ 1 - \frac{11}{6}|r| + |r|^2 - \frac{1}{6}|r|^3, & 1 < |r| \leq 2; \\ 0, & \text{otherwise.} \end{cases} \tag{9}$$

This function satisfies the properties 1,2,3,5 above, as well as these extra moment properties:

$$\sum_{i \in \mathbb{Z}} (r-i)^2 \phi(r-i) = 0 \quad \forall r \in \mathrm{I\!R}, \quad \sum_{i \in \mathbb{Z}} (r-i)^3 \phi(r-i) = 0 \quad \forall r \in \mathrm{I\!R}. \tag{10}$$

Our numerical experiment will use extensively this new discretization of the δ function, adapted to the staggered meshes.

3 The Multigrid τ–extrapolation

The τ–extrapolation [1, 8] is a modified multigrid method that improves the convergence order of a discrete problem. It is based on the Richardson extrapolation technique. It combines two solutions obtained on different grids in order to correct the fine grid solution, but requires the knowledge of the order of the first asymptotic expansion term, which can be evaluated experimentally. If this order is α, we combine the fine solution u_h and the coarse solution u_H with the following linear combination (u^* is the analytic solution):

$$\hat{u}_h = \left(\frac{2^\alpha}{2^\alpha - 1}\right) u_h + \left(1 - \frac{2^\alpha}{2^\alpha - 1}\right) u_H = u^* + o(h^\alpha). \tag{11}$$

Here is the τ–extrapolation multigrid algorithm for the problem $Au = f$:

- pre-smoothing step : $u_h = S^{\nu 1}(A_h, u_h, f_h)$,
- $u_h = u_h + I_H^h A_H^{-1} \left(\left(\frac{2^\alpha}{2^\alpha - 1}\right) \hat{I}_h^H (f_h - A_h u_h) + \left(1 - \frac{2^\alpha}{2^\alpha - 1}\right) (f_H - A_H I_h^H u_h) \right)$,
- post-smoothing step : $u_h = S^{\nu 2}(A_h, u_h, f_h)$,

with the following choices in most cases:
- I_H^h is a trilinear interpolation prolongation operator,
- $\hat{I}_h^H$ is a full weighting restriction operator,
- I_h^H is a full injection prolongation operator,
- (ν_1 , ν_2), the number of smoothing steps per iteration, are small (≤ 2).

The good convergence order property of the multigrid methods is due to the fact that the smoothing iterations improve the high frequency modes of the discrete solution, while the coarse grid correction improves its low frequency modes. This is especially true for the stiff elliptic problems solved in the IBM.

In the τ –extrapolation technique, the linear combination of the Richardson extrapolation significantly improves the discretization order of the coarse grid correction. This is the idea of the double discretization. A high order discretization scheme is used on the coarse grid, different from the scheme used for calculating the residuals transferred to the coarse grid. The smoothing process uses the low order discretization scheme too, which implies that two discrete problems with slightly different fixed points are solved. So the τ –extrapolation is a special case of the double discretization method, where we use the Richardson extrapolation technique to change the discretization order of the coarse grid. The analytic solution needs to be smooth enough and the restrictions operator needs to be chosen carefully, for the τ –extrapolation to improve the regular multigrid method.

A special feature of the τ –extrapolation applied to problems with singular source points is that we use f_H instead of $\hat{I}_h^H f_h$ at the coarse grid correction step. f_H is the discretization of the right-hand side using the discrete Dirac delta functions that have a $2H = 4h$ support, while f_h is evaluated using the same kind of delta function but with a $2h$ support. This is easy to implement and saves an interpolation process per multigrid iteration.

4 Numerical results

4.1 The 1D Helmholtz operator

Let us compare the different behaviors of both elliptic operators introduced in section 1 with a singular source point at the right-hand side. We solve at first the 1D problem [10]:

$$\frac{d^2u}{dx^2}(x) - \alpha^2 u(x) = -2\alpha\delta(x - x_0), \quad x \in [-0.5, 0.5], \ \ \alpha \in \mathbb{R}_+^*; \tag{12}$$

$$x_0 \in [-0.5, 0.5]; \ \ u(-0.5) = e^{-\alpha|-0.5-x_0|} \ \text{ and } \ u(0.5) = e^{-\alpha|0.5-x_0|}.$$

The domain is divided in N equidistant intervals. Finite differences and a classic stencil for the second order derivative are implemented in all of our computations. The computed solution is compared to the exact solution: $u_{ex}(x) = e^{-\alpha|x-x_0|}$, taking $x_0 = 0$ and $\alpha = 60$.

The number of operations represents the number of times the values at the nodes are updated but does not take into account the extrapolation and interpolation operations made in the multigrid algorithms in order to switch from one grid to another. The multigrid algorithm implemented is a classical V-cycle algorithm, with only two levels. We can see in Fig. (2) that the τ –extrapolation significantly improves the convergence order for this 1D problem with Dirac point load.

Since the point loads in the IBM can be located anywhere in a cell, it is relevant to study the behavior of the error depending on the distance between the point load and the nodes of the mesh. In the following graph, the error relative to the exact solution is plotted as a function of d, the minimum distance between x_0 and the nodes of the mesh, from 0 to $\frac{h}{2}$:

$$d(x_0) = \min_{i=1,..,N+1} |(-0.5 + (i-1)h) - x_0| . \tag{13}$$

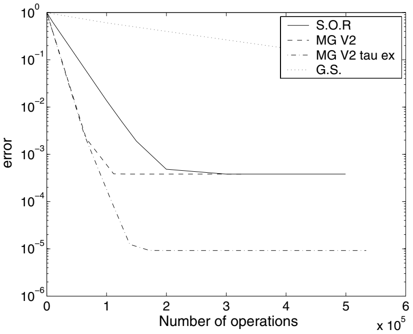


Fig. 1. *Error in L2-norm with respect to the number of operations with the 4 solvers S.O.R., Multigrid V2, Multigrid V2/* τ *ex. and Gauss-Seidel.* $N = 1000$ *,* $x_0 = 0$ *and using the piec. cub. delta func.*

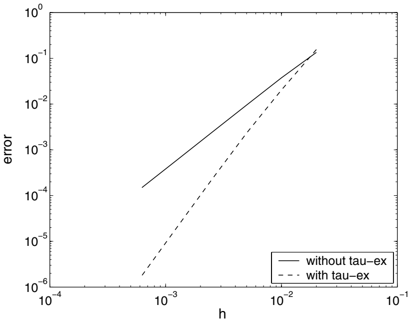


Fig. 2. *Error in L2-norm of the method for the multigrid algo. with or without the* τ *–extrapolation. and using the piec. cub. delta func. The order is improved from 2.0 to 3.4.*

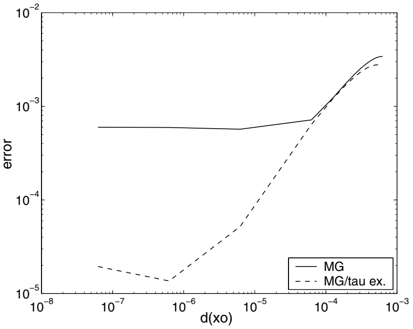


Fig. 3. *Error in L2-norm with respect to* $d(x_0)$ *, the min. distance between* x_0 *and the nodes of the mesh, using the piec. cub. delta func. and the multigrid solver with or without the* τ *–extrapolation.* $N = 800$ *,* $x_0 = 0$ *.*

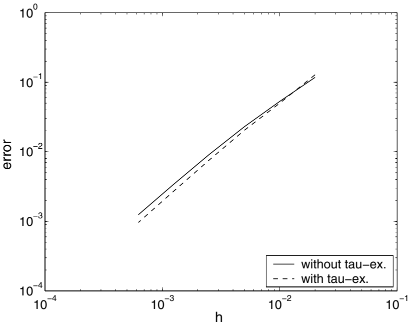


Fig. 4. *Error in L2-norm of the method using the multigrid algorithm with or without the* τ *–extrapolation and using the piec. cub. delta func. centered in the middle of a cells. The order is then 1.4.*

We can see in Fig. (3) that the accuracy depends strongly on the distance $d(x_0)$. If we measure the convergence order of the method when $d(x_0) = \frac{h}{2}$, using the L2-norm, we get only 1.4 (fig. (4)).

4.2 The 2D Laplace operator

Let us study the following benchmark problem [3]:

$$-\Delta u(x,y) = \delta(x,y,\Gamma), \quad (x,y) \in \Omega = [-1,1]^2; \tag{14}$$

$$\Gamma = \left\{(x,y) \in \Omega / x^2 + y^2 = r^2\right\}, \quad r < 1, \quad u_{|\partial\Omega} = u_{ex|\partial\Omega}$$

$$u_{ex}(x,y) = \begin{cases} 1 - \frac{1}{2} ln \left(\frac{1}{r}\sqrt{x^2+y^2}\right), & \text{if } x^2+y^2 > r^2; \\ \qquad 1, & \text{if } x^2+y^2 \leq r^2. \end{cases}$$

In this case the source term is distributed along a circle centered at the origin and with radius $r < 1$, which makes this problem close to those in the IBM. This time we need to use a discrete collection of M Dirac delta functions along the curve Γ . M is usually a large number so that the discretization error of the delta functions along Γ is minimized:

$$\delta_h(x,y,\Gamma) = \frac{2\pi r}{M} \sum_{i=1}^{M} \delta_h \left(x - r\, cos \left(\frac{2(i-1)\pi}{M}\right)\right) \delta_h \left(y - r\, sin \left(\frac{2(i-1)\pi}{M}\right)\right) \tag{15}$$

The error between the computed and analytic solutions is measured along the x-axis in the L2-norm. We find that the convergence order in the L2-norm is 2.0 without the τ –extrapolation and 2.8 with. Since the discrete Dirac delta functions are located along a circle, the distance between them and the nodes of the mesh varies between 0 and $\frac{h}{\sqrt{2}}$. The error is an average of the errors we would get with the delta functions centered at the nodes or at the mesh cell center.

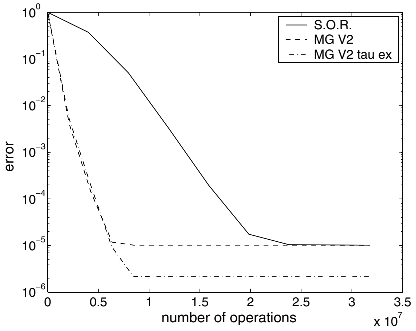


Fig. 5. *Error in L2-norm with respect to the number of operations with the 3 solvers S.O.R., Multigrid V2, Multigrid V2/τ-ex. $N = 200$, $r = 0.5$ and using the piec. cub. delta func.*

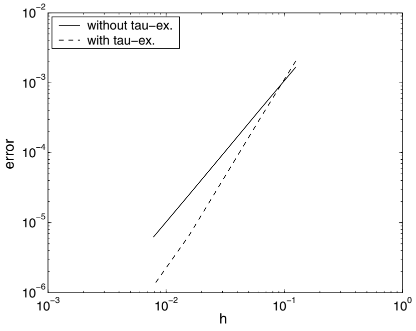


Fig. 6. *Error in L2-norm of the method for the multigrid algorithm with or without the τ –extrapolation and using the piece. cub. delta func. The order is improved from 2.0 to 2.8.*

5 Conclusion

We have shown that one can improve the accuracy of the IBM solvers by combining the τ –extrapolation technique with the piecewise cubic discrete Dirac delta function presented by Engquist and Tornberg [3]. Our current experiments with fluid-structure interactions extends these preliminary results using the IBM on staggered grid meshes.

References

1. K. Bernert, *τ-extrapolation—theoretical foundation, numerical experiment, and application to Navier–Stokes equations*, SIAM J. Sci. Comput., 18 (1997), pp. 460–478.
2. A. J. Chorin, *Numerical solution of Navier-Stokes equations*, Math. Comp., 22 (1968), pp. 745–762.
3. B. Engquist and A.-K. Tornberg, *Numerical approximations of singular source terms in differential equations*, J. Comp. Phys., 200 (2004), pp. 462–488.
4. F. H. Harlow and J. E. Welch, *Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface*, Physics of Fluids, 8 (1965), pp. 2182–2189.
5. C. S. Peskin, *Flow patterns around heart valves: A numerical method*, J. Comp. Phys., 10 (1972), pp. 252–271.
6. ———, *The immersed boundary method*, Acta Numerica, 11 (2002), pp. 479–517.
7. A. M. Roma, *A Multilevel Self Adaptive Version of the Immersed Boundary Method*, PhD thesis, Courant Institute of Mathematical Sciences, New York University, New York, 1996.
8. U. Rüde and H. Köstler, *Accurate techniques for computing singular solutions of elliptic problems*, Tech. Rep. 04-3, Friedrich-Alexander-Universität Erlangen-Nürnberg, Institut für Informatik, Erlangen, Germany, 2004.
9. J. M. Stockie, *Analysis and Computation of Immersed Boundaries, with Applications to Pulp Fibres*, PhD thesis, University of British Columbia, 1997.
10. J. Waldén, *On the approximation of singular source terms in differential equations*, Numer. Methods Partial Differential Equations, 15 (1999), pp. 503–520.

Overlapping Schwarz Preconditioners for Fekete Spectral Elements

Richard Pasquetti [1], Luca F. Pavarino [2], Francesca Rapetti [1], and Elena Zampieri [2]

[1] Laboratoire J.-A. Dieudonné, CNRS & Université de Nice et Sophia-Antipolis, Parc Valrose, 06108 Nice Cedex 02, France. frapetti, rpas@math.unice.fr

[2] Department of Mathematics, Universitá di Milano, Via Saldini 50, 20133 Milano, Italy. Luca.Pavarino, Elena.Zampieri@mat.unimi.it

Summary. We construct and study overlapping Schwarz preconditioners for the iterative solution of elliptic problems discretized with spectral elements based on Fekete nodes (TSEM). These are a generalization to non-tensorial elements of the classical Gauss-Lobatto-Legendre hexahedral spectral elements (QSEM). Even if the resulting discrete problem is more ill-conditioned than in the classical QSEM case, the resulting preconditioned algorithm using generous overlap is optimal and scalable, since its convergence rate is bounded by a constant independent of the number of elements, subdomains and polynomial degree employed.

1 The model problem and SEM formulation

The recent trend toward highly parallel and high-order numerical solvers has led to increasing interest in domain decomposition preconditioners for spectral element methods; see [10, 17, 12, 4, 7, 5, 6]. While very successful algorithms have been constructed and analyzed for classical Gauss-Lobatto-Legendre hexahedral spectral elements (QSEM), many open problems remain for non-tensorial spectral elements. In this paper, we consider Fekete nodal spectral elements (TSEM) and propose an Overlapping Schwarz preconditioner that using generous overlap turns out to be optimal and scalable.

Let $\Omega \in \mathbb{R}^d, d = 2, 3,$ be a bounded Lipschitz domain with piecewise smooth boundary. For simplicity, we consider a model elliptic problem in the plane ($d = 2$) and with homogeneous Dirichlet boundary data, but the techniques presented in this papers apply equally well to more general elliptic problems in three dimensions: Find $u \in V := H_0^1(\Omega)$ such that

$$a(u, v) := \int_\Omega (\alpha \, \mathbf{grad}\, u \cdot \mathbf{grad}\, v + \beta \, u \, v) \, \mathrm{d}\mathbf{x} = \int_\Omega f \, v \, \mathrm{d}\mathbf{x} \qquad \forall v \in V, \tag{1}$$

where α, $\beta > 0$ are piecewise constant in Ω and $f \in L^2(\Omega)$.

The variational problem (1) is discretized by the conforming spectral element method, either quadrilateral based (QSEM) or triangle based (TSEM), which is a Galerkin method that employs a discrete space consisting of continuous piecewise polynomials of degree N; see [1, 4, 6] for a general introduction. Let $T_{\rm ref} = \{(r,s) : -1 \le r,s \le +1,\ r+s \le 0\}$ be the reference triangle and $\mathcal{P}_N(T_{\rm ref})$ the set of polynomials on $T_{\rm ref}$ of total degree $\le N$. Let $Q_{\rm ref}$ be the reference square $[-1,1]^2$ and $\mathbb{P}_N(Q_{\rm ref})$ the set of polynomials on $Q_{\rm ref}$ of degree $\le N$ in each variable. We assume that Ω is decomposed into K nonoverlapping triangular or quadrilateral finite elements Ω_k, $\overline{\Omega} = \bigcup_{k=1}^{K} \overline{\Omega}_k$, each of which is the image of $T_{\rm ref}$ or $Q_{\rm ref}$ by means of a suitable mapping, i.e., $\Omega_k = g_k(T_{\rm ref})$ or $\Omega_k = g_k(Q_{\rm ref})$. The intersection between two distinct Ω_k is either the empty set or a common vertex or a common side. We denote by H the maximum diameter of the subdomains $\Omega_k' s$. The space V is discretized by continuous, piecewise polynomials of total degree $\le N$,

$$V^T_{K,N} = \{v \in V : v|_{\Omega_k} \circ g_k \in \mathcal{P}_N(T_{\rm ref}),\ 1 \le k \le K\},$$

or of degree $\le N$ in each variable,

$$V^Q_{K,N} = \{v \in V : v|_{\Omega_k} \circ g_k \in \mathbb{P}_N(Q_{\rm ref}),\ 1 \le k \le K\}.$$

QSEM and Gauss-Lobatto-Legendre points. We recall here the conforming quadrilateral spectral elements QSEM based on Gauss-Lobatto-Legendre (GLL) quadrature points, which also allows the construction of a very convenient tensor-product basis for $V^Q_{K,N}$. We denote by $\{\xi_i\}_{i=0}^N$ the set of GLL points of $[-1,1]$, and by σ_i the associated quadrature weights. Let $l_i(x)$ be the Lagrange interpolating polynomial of degree $\le N$ which vanishes at all the GLL nodes except ξ_i, where it equals one. The basis functions on the reference square $Q_{\rm ref}$ are defined by a tensor product as $l_i(x)l_j(y),\ 0 \le i,j \le N$. Each function of $\mathbb{P}_N(Q_{\rm ref})$ is expanded in this nodal GLL basis through its values at GLL nodes $u(\xi_i,\xi_j),\ 0 \le i,j \le N$. We replace each integral of (1) by GLL quadrature:

$$(u,v)^Q_{K,N} = \sum_{k=1}^{K} \sum_{i,j=0}^{N} (u \circ g_k)(\xi_i,\xi_j)(v \circ g_k)(\xi_i,\xi_j)|J^Q_k|\sigma_i\sigma_j, \tag{2}$$

where $|J^Q_k|$ is the Jacobian of g_k. This inner product is uniformly equivalent to the standard one on $\mathbb{P}_N(\Omega)$. We then obtain the discrete problem: Find $u \in V^Q_{K,N}$ such that

$$a^Q_{K,N}(u,v) = (f,v)^Q_{K,N}, \quad \forall v \in V^Q_{K,N}, \tag{3}$$

where $a^Q_{K,N}(\cdot,\cdot)$ is obtained from $a(\cdot,\cdot)$ by replacing each integral with the GLL quadrature rule described in (2). The matrix form of (3) is a linear system $A_Q\mathbf{u} = \mathbf{b}$, where A_Q is here the assembled QSEM matrix (positive definite and symmetric), $\mathbf{b}$ is the load vector and $\mathbf{u}$ is the vector of nodal values of the unknown function u.

TSEM and Fekete points. On triangular elements it is no longer possible to define spectral elements by tensor product as in QSEM. Let $\{\psi_j\}_{j=1}^n$ be an orthonormal basis of $\mathcal{P}_N(T_{ref})$ for the usual $L^2(T_{ref})$ inner product (for example, the Koornwinder-Dubiner polynomials may be used to constitute such a basis, see [7]).

The Fekete points on T_{ref} are defined as the points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ that maximize the determinant of the Vandermonde matrix V with elements $V_{ij} = \psi_j(\hat{\mathbf{x}}_i)$, $1 \le i, j \le n$, where $n = (N+1)(N+2)/2$. For the TSEM introduced in [14], the Fekete points are used as approximation points and the Lagrange polynomials $\{\phi_i\}_{i=1}^n$ built on these points are used as basis functions. Among the main properties of Fekete points proved in [15], we recall that Fekete points are Gauss-Lobatto points for the cube, thus providing a strong link with the usual QSEM. Unlikely GLL points, a quadrature formula based on Fekete points is only exact for integrands in $\mathcal{P}_N(T_{ref})$. This fact has suggested for the TSEM to separate the sets of approximation and quadrature points, using the Fekete points $\{\hat{\mathbf{x}}_i\}_{i=1}^n$ for the first set and other points $\{\hat{\mathbf{y}}_i\}_{i=1}^m$ for the second set, imposing an exact integration of polynomials, e.g., in $\mathcal{P}_{2N}(T_{ref})$; see [8]. Given the values at the approximation points of a polynomial $u_N \in \mathcal{P}_N(T_{ref})$, one can set up interpolation and differentiation matrices to compute, at the quadrature points, the values of u_N and of its derivatives, respectively. For instance, denoting by $\underline{u}$ the vector of the $u_i = u_N(\hat{\mathbf{x}}_i)$, $1 \le i \le n$, and by $\underline{u}'$ that of the $u_N(\hat{\mathbf{y}}_i)$, $1 \le i \le m$, we have $\underline{u}' = V'V^{-1}\underline{u},$ where $V'_{ij} = \psi_j(\hat{\mathbf{y}}_i)$. On a generic triangle $\Omega_k = g_k(T_{ref})$, the same relation between $\underline{u}'$ and $\underline{u}$ holds true, provided that $u_i = (u_N \circ g_k)(\hat{\mathbf{x}}_i)$ and $u'_j = (u_N \circ g_k)(\hat{\mathbf{y}}_j)$. The TSEM requires of course the use of highly accurate integration rules based on Gauss points. Unfortunately, in practice such integration rules are difficult to define for large values of N (recent publications show that this is still an open subject of research). In the present case, we can use integration rules based on Gauss points for quadrilaterals and then map them to T_{ref} ; see [6]. On a generic triangle $\Omega_k = g_k(T_{ref})$:

$$(u,v)_{\Omega_k,N} = \sum_{j=1}^{m} u'_j\, v'_j\, |J_k^T(\hat{\mathbf{y}}_j)|\, \omega_j,$$

where $\omega_j > 0$, $1 \le j \le m$, are the quadrature weights and $|J_k^T|$ the Jacobian of the mapping g_k between T_{ref} and Ω_k . As for (3), we obtain a discrete problem

$$a_{K,N}^T(u,v) = (f,v)_{K,N}^T, \quad \forall v \in V_{K,N}^T, \tag{4}$$

that can be written in matrix form as a linear system $A_T\mathbf{u} = \mathbf{b}$. The TSEM matrix A_T is less sparse than the QSEM matrix A_Q and more ill-conditioned, since its condition number grows as $O(N^{2(d-1)})$ (see Sec. 3).

2 Overlapping Schwarz Preconditioners

We now consider the iterative solution of the discrete systems $A\mathbf{u} = \mathbf{f}$ by the conjugate gradient method with an Overlapping Schwarz preconditioner; see e.g. [16, 13, 11] for a general introduction.

Let τ_0 be the coarse finite element triangulation of the domain Ω determined by the elements Ω_k , $k = 1, ..., K,$ of characteristic diameter H . Let τ_N be the fine triangulation determined by either the Fekete or the GLL nodes introduced in each element Ω_k in Sections 2.1 and 2.2. Thus we can define two different coarse and fine triangulations and related overlapping partitions of Ω , according to the spectral element method at issue.

QSEM. The coarse triangulation τ_0 is given by quadrilaterals Ω_k providing a coarse problem with bilinear finite element ($N = 1$ in each direction). Then the local fine discretization τ_N is determined by the GLL nodes in each quadrilateral Ω_k . We define the overlapping partition of Ω by extending each subdomain Ω_k to a larger subdomain Ω'_k , consisting of all elements of τ_N within a certain distance from Ω_k ; we measure this distance by the number δ of GLL points extending Ω_k in each direction. See Figure 1 (left) for a two-dimensional example.

TSEM. The coarse triangulation τ_0 is given by triangles Ω_k providing a coarse problem with linear finite element ($N = 1$). Then the local fine discretization τ_N is determined by Fekete nodes within each Ω_k . The overlapping partition of Ω is generated by extending each triangle Ω_k to a large subdomain Ω'_k consisting of all triangles sharing with Ω_k either a vertex or an edge. See Figure 1 (right) for a two-dimensional example. Overlapping techniques involving a smaller number of subdomains (e.g., sharing edges of Ω_k only) proved unsuccessful, whereas less generous overlapping partitions considering a few nodes around Ω_k can not be designed straightforwardly since the internal Fekete nodes are not distributed regularly as in tensorial elements.

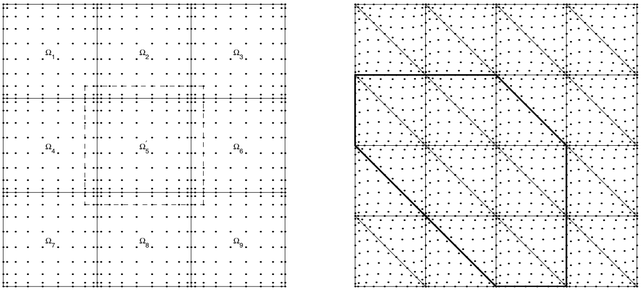


Fig. 1. Example of Ω'_k subdomains for QSEM with small overlap ($\delta = 2$, left) and TSEM with generous overlap (right).

The overlapping Schwarz preconditioner B^{-1} for A is based on solving a) a coarse problem with linear or bilinear elements on the coarse mesh τ_0 ; b) local problems on the overlapping subdomains Ω'_k .

For the coarse solve, we need to define:

a1) a restriction matrix R_0 ; its transpose R_0^T interpolates coarse linear (resp. bilinear) functions on τ_0 to spectral elements functions on the fine Fekete (resp. GLL) mesh τ_N ;

a2) a coarse stiffness matrix $A_0 = R_0 A R_0^T$ needed for the solution of the coarse problem with $N = 1$ on τ_0 .

For the local solves, we need to define:

b1) restriction matrices R_k (with 0,1 entries) returning only the degrees of freedom inside each subdomain Ω'_k ;

b2) local stiffness matrices $A_k = R_k A R_k^T$ needed for the solution of the k th local problem on Ω'_k with zero Dirichlet boundary conditions on $\partial\Omega'_k$.

These are the building blocks of the proposed preconditioners. The additive form of the overlapping Schwarz preconditioner is

$$B_{add}^{-1} = R_0^T A_0^{-1} R_0 + \sum_{k=1}^{K} R_k^T A_k^{-1} R_k, \tag{5}$$

Multiplicative and hybrid variants can be considered too, see [13, 16].

These preconditioners are associated with the space decomposition $V_{K,N} = V_0 + \sum_{k=1}^{K} V_k$, where either $V_{K,N} \equiv V_{K,N}^T$ or $V_{K,N} \equiv V_{K,N}^Q$. V_0 is the subspace of $V_{K,N}$ consisting of piecewise linear or bilinear functions on the coarse mesh τ_0 and

$$V_k = \{\, v \in V_{K,N}^T : v = 0 \text{ at all the Fekete nodes outside } \Omega_k' \text{ and on } \partial\Omega_k' \,\}$$

in the case of triangles, and

$$V_k = \{\, v \in V_{K,N}^Q : v = 0 \text{ at all the GLL nodes outside } \Omega_k' \text{ and on } \partial\Omega_k' \,\}$$

in the case of quadrilaterals. Defining the operators $T_k : V_{K,N} \longrightarrow V_k$ by $a_{K,N}(T_k u, v) = a_{K,N}(u, v)\ \forall v \in V_k,\ 0 \le k \le K$ where $a_{K,N} \equiv a_{K,N}^T$ for TSEM and $a_{K,N} \equiv a_{K,N}^Q$ for QSEM, then (5) is exactly the matrix form of the additive Schwarz operator $T_{add} = T_0 + T_1 + \cdots + T_K$. The theory developed by Casarin [3] for QSEM and scalar symmetric positive definite problems allows us to transfer the main domain decomposition results from the finite elements to QSEM (see e.g. Toselli and Widlund [16, Ch. 7]).

Theorem 1. *The condition number of the overlapping Schwarz QSEM operator is bounded by*

$$cond(T_a) \le C(1 + \frac{H}{\tilde{\delta}}),$$

with $\tilde{\delta} = \min_k \{\mathrm{dist}(\partial\Omega_k, \partial\Omega_k')\}$ *and the constant* C *is independent of* $N, H, \tilde{\delta}$.

In case of generous overlap $\tilde{\delta} = CH$, we have a constant upper bound for both $cond(T_a)$ and the number of iterations; this was already proved in Pavarino [9] for more general hp finite elements. The analyses in [3] and [9] are no longer valid for unstructured hp elements on nontensorial elements, and preconditioners with small overlap are not known; the only theory available is for nonoverlapping methods in Bica's doctoral thesis [2]. Nevertheless, we can build preconditioners with generous overlap as shown before and the numerical results of the next section show that they are optimal and scalable, hence we conjecture that a bound as in Theorem 1 also holds for TSEM.

3 Numerical results

In this section, we report the results of numerical experiments for the overlapping Schwarz preconditioner applied to the model problem (1) discretized with triangular spectral elements using Fekete nodes. We consider an homogeneous material with $\alpha = \beta = 1$. The computational domain is $\Omega = [-1, 1]^2$ and the body force f is consistent with $u(x) = \sin(\pi x)\sin(\pi y)$ as the exact solution of (1). The mesh is obtained by first dividing Ω into $K = k^2$ identical squares and then by dividing similarly each of them into two triangles. The grid-size parameter H is chosen equal to $2/k$. The resulting discrete problem is solved by the preconditioned conjugate gradient (PCG) method without or with Schwarz preconditioner (5), the latter with or without the coarse solver $R_0^T A_0^{-1} R_0$. The initial guess is zero and the stopping criterion is $|\mathbf{r}^{(\nu)}|/|\mathbf{b}^{(\nu)}| \leq 10^{-7}$, where $\mathbf{r}^{(\nu)}$ is the νth residual. In Table 1, we report the iteration counts (It.), spectral condition number ($\kappa_2(A)$) and extreme eigenvalues (λ_{max}, λ_{min}), fixing $H = 1/2$ (32 subdomains) and varying the degree N from 3 to 15. Columns 2-3 refers to CG, columns 4-7 refer to PCG without a coarse solver, and columns 8-11 refer to PCG with a coarse solver. The same quantities are reported in Tables 2 fixing now $N = 3$ and varying $1/H$ from 2 to 10. These results are also plotted in Fig. 2 and 3, that clearly show that while the very ill-conditioned original TSEM matrix has a condition number that grows as $O(N^4 H^{-2})$, the overlapping Schwarz preconditioned operator is optimal and scalable (i.e. independent of N and H).

References

1. C. Bernardi and Y. Maday, *Spectral methods*, in Handbook of numerical anal-

Table 1. CG and PCG preconditioners for the model problem (1) with $\alpha = \beta = 1$ and $\Omega = [-1, 1]^2$. Iteration counts, condition number and extreme eigenvalues with fixed $1/H = 2$ and varying N.

	CG		PCG				PCG + coarse			
N	It.	$\kappa_2(A)$	It.	λ_{max}	λ_{min}	$\kappa_2(\tilde{A})$	It.	λ_{max}	λ_{min}	$\kappa_2(\tilde{A})$
3	28	84.34	12	12.99	2.66	4.87	13	13.00	3.34	3.88
6	85	729.37	13	12.99	2.67	4.85	13	12.99	3.35	3.87
9	206	4819.90	13	12.99	2.67	4.85	13	12.99	3.35	3.87
12	299	8899.07	13	12.99	2.67	4.85	13	12.99	3.35	3.87
15	456	21738.04	13	12.99	2.67	4.85	13	12.99	3.35	3.87

Table 2. CG and PCG preconditioners for the model problem (1) with $\alpha = \beta = 1$ and $\Omega = [-1, 1]^2$. Iteration counts, condition number and extreme eigenvalues with fixed $N = 3$ and varying $1/H$.

	CG		PCG				PCG + coarse			
1/H	It.	$\kappa_2(A)$	It.	λ_{max}	λ_{min}	$\kappa_2(\tilde{A})$	It.	λ_{max}	λ_{min}	$\kappa_2(\tilde{A})$
2	28	84.34	12	12.99	2.66	4.87	13	12.99	3.34	3.88
3	39	190.08	14	12.99	1.41	9.16	13	12.99	2.34	5.53
4	48	337.99	16	12.99	0.84	15.37	14	13.00	1.81	7.17
5	56	527.00	18	12.99	0.55	23.40	15	13.00	1.53	8.46
6	63	753.65	20	12.99	0.39	33.23	17	13.00	1.37	9.43
7	69	1007.16	22	12.99	0.28	44.85	18	13.00	1.27	10.15
8	90	1352.64	25	13.00	0.22	58.27	19	13.00	1.21	10.69
9	98	1710.50	27	13.00	0.17	73.48	20	13.00	1.17	11.10
10	104	2104.60	30	13.00	0.14	90.48	20	13.00	1.13	11.41

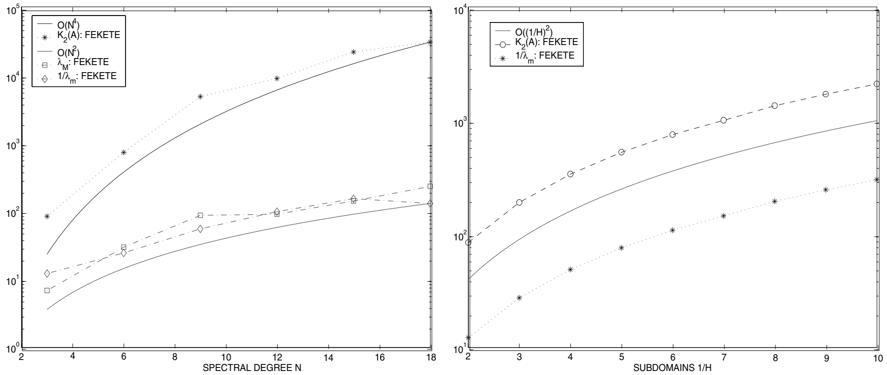


Fig. 2. Condition number and extreme eigenvalues of the unpreconditioned stiffness matrix as a function of N (left) and of $1/H$ (right).

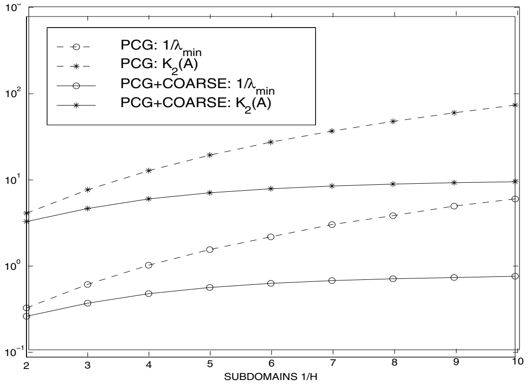


Fig. 3. Condition number ($*$) and extreme eigenvalues ($\circ$) of the overlapping Schwarz preconditioned matrix with (solid lines) and without (dashed lines) coarse solver as a function of $1/H$.

ysis, Vol. V, P. G. Ciarlet and J. L. Lions, eds., North-Holland, 1997, pp. 209–485.

2. I. Bica, *Iterative substructuring methods for the p-version finite element method for elliptic problems*, PhD thesis, Courant Institute of Mathematical Sciences, New York University, New York, September 1997.
3. M. A. Casarin, Jr., *Quasi-optimal Schwarz methods for the conforming spectral element discretization*, SIAM J. Numer. Anal., 34 (1997), pp. 2482–2502.
4. M. O. Deville, P. F. Fischer, and E. H. Mund, *High-Order Methods for Incompressible Fluid Flow*, Cambridge University Press, 2002.
5. F. X. Giraldo and T. Warburton, *A nodal triangle-based spectral element method for the shallow water equations on the sphere*, J. Comp. Phys., 207 (2005), pp. 129–150.
6. G. E. Karniadakis and S. J. Sherwin, *Spectral/hp Element Methods for CFD*, Oxford University Press, second ed., 2005.
7. R. Pasquetti and F. Rapetti, *Spectral element methods on triangles and quadrilaterals: comparisons and applications*, J. Comput. Phys., 198 (2004), pp. 349–362.
8. ———, *Spectral element methods on unstructured meshes: comparisons and recent advances*, J. Sci. Comp., 27 (2005), pp. 377–387.
9. L. F. Pavarino, *Additive Schwarz methods for the p-version finite element method*, Numer. Math., 66 (1994), pp. 493–515.
10. L. F. Pavarino and T. Warburton, *Overlapping Schwarz methods for unstructured spectral elements*, J. Comput. Phys., 160 (2000), pp. 298–317.
11. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford Science Publications, 1999.
12. S. J. Sherwin and M. A. Casarin, *Low-energy basis preconditioning for elliptic substructured solvers based on unstructured spectral/hp element discretization*, J. Comput. Phys., 171 (2001), pp. 394–417.
13. B. F. Smith, P. E. Bjørstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*, Cambridge University Press, 1996.
14. M. A. Taylor and B. A. Wingate, *A generalized diagonal mass matrix spectral element method for non-quadrilateral elements*, Appl. Num. Math., (2000), pp. 259–265.
15. M. A. Taylor, B. A. Wingate, and R. E. Vincent, *An algorithm for computing Fekete points in the triangle*, SIAM J. Numer. Anal., 38 (2000), pp. 1707–1720.
16. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.
17. T. Warburton, L. F. Pavarino, and J. S. Hesthaven, *A pseudo-spectral scheme for the incompressible Navier-Stokes equation using unstructured nodal elements*, J. Comput. Phys., (2000), pp. 1–21.

Solution of Reduced Resistive Magnetohydrodynamics using Implicit Adaptive Mesh Refinement

Bobby Philip [1] , Michael Pernice [1] , and Luis Chacón [2]

[1] Computer and Computational Sciences Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. {bphilip,pernice}@lanl.gov

[2] Theoretical Division, Los Alamos National Laboratory, Los Alamos, NM 87545, USA. chacon@lanl.gov

Summary. Computational study of the macroscopic stability of plasmas is a challenging multi-scale problem. Implicit time integration can be used to relieve stability constraints due to fast Alfvén waves, and adaptive mesh refinement (AMR) can be used to resolve highly localized solution features. The strong nonlinearities and numerical stiffness of magnetohydrodynamics (MHD) models present further challenges that must be solved to make implicit AMR practical. We present initial results on the application of implicit AMR to a reduced resistive MHD model.

1 Introduction

Magnetohydrodynamics (MHD) models are useful for studying the macroscopic behavior of plasmas. Plasmas exhibit a wide range of complex behavior, including magnetic reconnection, where the magnetic field undergoes a rapid reconfiguration accompanied by conversion of energy stored in the magnetic field to kinetic energy. Reconnection is associated with the formation of thin, localized current sheets, which profoundly influence macroscopic behavior. Thus, it is natural to consider adaptive mesh refinement (AMR) to locally resolve these features.

MHD models also display behaviors that occur over a wide range of time scales, many of which are much faster than the time scale over which reconnection occurs. Time integration methods that are subject to stability constraints that arise from the fastest time scales are inappropriate for simulation-based study of reconnection phenomena. We therefore employ implicit time integration methods where time steps are constrained only by accuracy, not stability.

Implicit time integration requires the solution of large-scale systems of nonlinear equations at each time step, and fast, robust solution methods are necessary for implicit methods to be practical. Fortunately, Newton-Krylov methods [2] have met this requirement in a variety of contexts [6], provided effective preconditioning is used. In particular, we have demonstrated excellent preconditioner performance for reduced MHD models [4, 3]. The key to this success is the design of physics-based preconditioners that preserve important couplings between variables and allow the effective use of multigrid methods. On AMR grids, fast multilevel methods that exploit the structure of the mesh are needed.

2 Adaptive Mesh Refinement

We use *structured* AMR (SAMR), in which the grid is organized as a collection of refinement levels. Each refinement level is the union of rectangular patches having fixed resolution, and is fully nested in the next coarser level (except at physical boundaries); see Figure 2 for some examples. This hierarchical structure naturally lends itself to domain decomposition, by treating each refinement level as a separate subdomain and exploiting the natural partition of each level by its patches. The fully overlapping nature of the domains enables use of coarse grid points that are covered by a finer level to accelerate the solution process. In particular, we use the Fast Adpative Composite grid (FAC) method [7], which is a multiplicative approach that treats levels sequentially, analogous to a multigrid V-cycle. FAC can employ simple smoothing on refinement levels, and an approximate solve on the coarsest, global level.

SAMR requires special discretization procedures to enforce smoothness of the solution at the interfaces between coarse and fine regions. Continuity of the solution at coarse/fine interfaces is ensured by providing each level with Dirichlet boundary conditions that are determined from piecewise quadratic interpolation of data from the next coarser level. Flux continuity at coarse/fine interfaces is enforced by using this Dirichlet data to compute $\mathcal{O}(h^3)$ -accurate gradients normal to the level boundary. Nevertheless, careful selection of the MHD model is required to successfully use SAMR in this context.

3 Current-Vorticity Formulation of Reduced MHD

The two-dimensional reduced MHD formalism assumes that the plasma is strongly magnetized by a large magnetic field in the (ignorable) z -direction [5]. It follows that the dynamics is restricted to the $x - y$ plane, where the plasma is incompressible. This allows descriptions of the plasma velocity $\mathbf{v} = (u, v)^\mathsf{T}$ in terms of a streamfunction $\varPhi$ (with $u = -\varPhi_y$ and $v = \varPhi_x$) and the magnetic field $\mathbf{B} = (B_1, B_2)^\mathsf{T}$ in terms of a poloidal flux function $\varPsi$ (with $B_1 = -\varPsi_y$ and $B_2 = \varPsi_x$). This leads to the streamfunction-vorticity formulation

$$\begin{aligned} \partial_t \omega + \mathbf{v} \cdot \nabla \omega - \nu \nabla^2 \omega &= \mathbf{B} \cdot \nabla J \\ \partial_t \varPsi + \mathbf{v} \cdot \nabla \varPsi - \eta \nabla^2 \varPsi &= 0 \\ \nabla^2 \varPhi &= \omega \end{aligned} \tag{1}$$

where ω is the vorticity, $J = \nabla^2 \varPsi$ is the electric current, ν is the viscosity, and η is the resistivity. While this formulation was successfully treated in [4], it is not well-suited to SAMR, because of the need to compute $J = \nabla^2 \varPsi$ at coarse/fine interfaces, even if much higher-order interpolation is used. The fact that J is determined by differentiation leads to small instabilities along the coarse/fine interface that grow as the simulation proceeds. Similar difficulties were reported in [9] for ideal MHD on unstructured grids. Instead, and following [9], we use a current-vorticity formulation

$$\begin{aligned} \partial_t J + \mathbf{v}\cdot\nabla J - \eta\nabla^2 J - \mathbf{B}\cdot\nabla\omega &= \{\Phi,\Psi\} \\ \partial_t \omega + \mathbf{v}\cdot\nabla \omega - \nu\nabla^2 \omega - \mathbf{B}\cdot\nabla J &= 0 \\ \omega - \nabla^2\Phi &= 0 \\ J - \nabla^2\Psi &= 0 \end{aligned} \tag{2}$$

where $\{\Phi,\Psi\} = 2[\Phi_{xy}(\Psi_{xx}-\Psi_{yy}) - \Psi_{xy}(\Phi_{xx}-\Phi_{yy})]$.

Our main task here is to extend the physics-based preconditioner developed in [4] to handle (2). For the sake of brevity, we assume the reader is familiar with the derivation in [4], and do not repeat it here. Discretizing (2) in time with a theta difference scheme yields

$$\begin{aligned} (J^{n+1}-J^n)/\Delta t + [\mathbf{v}\cdot\nabla J]^{n+\theta} - \eta\nabla^2 J^{n+\theta} - [\mathbf{B}\cdot\nabla\omega]^{n+\theta} &= \{\Phi,\Psi\}^{n+\theta} \\ (\omega^{n+1}-\omega^n)/\Delta t + [\mathbf{v}\cdot\nabla \omega]^{n+\theta} - \nu\nabla^2 \omega^{n+\theta} - [\mathbf{B}\cdot\nabla J]^{n+\theta} &= 0 \\ \omega^{n+\theta} - \nabla^2\Phi^{n+\theta} &= 0 \\ J^{n+\theta} - \nabla^2\Psi^{n+\theta} &= 0 \end{aligned} \tag{3}$$

where $n+\theta$ -level quantities are calculated as $\xi^{n+\theta} = (1-\theta)\xi^n + \theta\xi^{n+1}$. Backward Euler time discretization is obtained by $\theta = 1$ and Crank-Nicolson time discretization corresponds to $\theta = 1/2$. We represent (3) generically by $\mathbf{G}(\mathbf{x}^{n+1}) = 0$ and compute the time-advanced solution with a Jacobian-free Newton-Krylov (JFNK) method.

Each iteration of JFNK requires solution of the linearized system

$$\mathcal{L}_\eta \delta J + \tag{4}$$

$$\theta(\delta\mathbf{v}\cdot\nabla J_0 - \mathbf{B_0}\cdot\nabla\delta\omega - \delta\mathbf{B}\cdot\nabla\omega_0 - \{\delta\Phi,\Psi_0\} - \{\Phi_0,\delta\Psi\}) = -G_J \tag{5}$$

$$\mathcal{L}_\nu \delta\omega + \theta(\delta\mathbf{v}\cdot\nabla\omega_0 - \mathbf{B_0}\cdot\nabla\delta J - \delta\mathbf{B}\cdot\nabla J_0) = -G_\omega \tag{6}$$

$$\delta J - \nabla^2\delta\Psi = -G_\Psi \tag{7}$$

$$\delta\omega - \nabla^2\delta\Phi = -G_\Phi, \tag{8}$$

where $\mathcal{L}_\alpha = \dfrac{1}{\Delta t} + \theta(\mathbf{v}_0\cdot\nabla - \alpha\nabla^2)$, $\alpha = \eta, \nu$. Quantities with subscript 0 refer to solution quantities at the previous Newton iterate and $(G_J, G_\omega, G_\psi, G_\phi)^t$ refers to the nonlinear residual.

We extend the semi-implicit preconditioner for (1) to handle (2) by first substituting (7) and (8) in (5) and (6), respectively, and approximating as in [4], to obtain

$$\mathcal{P}\begin{pmatrix}\delta\Psi\\ \delta\Phi\end{pmatrix} \approx -(\nabla^\mathbf{2})^{-1}\left[\begin{pmatrix}G_J\\ G_\omega\end{pmatrix} - \mathcal{P}\begin{pmatrix}G_\Psi\\ G_\Phi\end{pmatrix}\right] \tag{9}$$

where

$$\mathcal{P} \equiv \begin{pmatrix} \mathcal{L}_\eta & -\theta\mathbf{B}_0\cdot\nabla \\ -\theta\mathbf{B}_0\cdot\nabla & \mathcal{L}_\nu \end{pmatrix}.$$

The system in (9) is only approximate, and is treated here as a predictor step for $\delta\Psi$ and $\delta\Phi$. Solution of (9) is done as described in [4], namely, by a few sweeps of the stationary method obtained by the splitting of $\mathcal{P}$ that is induced by separating $\mathcal{L}_\nu$ into its diagonal and off-diagonal parts, and forming the Schur complement of the split block matrix for inversion. After this, the system

$$\mathcal{P}\begin{pmatrix}\delta J\\ \delta\omega\end{pmatrix} = -\begin{pmatrix}G_J + \theta(\delta\mathbf{v}\cdot\nabla J_0 - \delta\mathbf{B}\cdot\nabla\omega_0 - \{\delta\Phi, \Psi_0\} - \{\Phi_0, \delta\Psi\})\\ G_\omega + \theta(\delta\mathbf{v}\cdot\nabla\omega_0 - \delta\mathbf{B}\cdot\nabla J_0)\end{pmatrix} \tag{10}$$

is solved in the same manner for δJ and $\delta\omega$ using the predicted $\delta\Psi$ and $\delta\Phi$ in the right hand side.

4 Computational Results

We present initial results of applying implicit AMR to the classical tearing resistive instability problem described in [4]. The current-vorticity formulation (2) is used on the physical domain $\Omega = [0,4]\times[0,1]$, with periodic boundary conditions in x and homogeneous Dirichlet boundary conditions in y. Initial conditions are given by $\omega_0 = 0$ and a Harris current sheet $J_0 = \operatorname{sech}^2\left((y-0.5)/\lambda\right)/\lambda$ with $\lambda = 0.2$. We use a fixed time step $\Delta t = 1$ and integrate to $t = 250$.

The software infrastructure described in [8] is used. In particular, we use the implementation of JFNK from PETSc's Scalable Nonlinear Equation Solver (SNES) package [1] with a constant forcing term, with both absolute and relative stopping tolerances of 10^{-4}. We developed implementations of FAC for solving the Poisson and convection-tensor diffusion sub-problems that are needed to implement the preconditioner [(9) and (10)].

Our criteria for dynamic mesh refinement are based on detecting solution features. Cells are selected for subdivision when $|J|$ exceeds 85% of its maximum value (following [9]) or when the curvature in ω exceeds 0.40. Regridding is done every fourth time step. These choices were determined experimentally to produce acceptable results. While a systematic study of the accuracy of the adaptive simulation is needed, Figure 1 shows good agreement of the growth of the magnetic perturbation calculated on different grid configurations with the same finest resolution. All these calculations predicted a growth rate of 0.046.

Figure 2 depicts evolution of the solution and grid for a 32×32 base grid with 3 refinement levels. Initially, refinement is concentrated in a strip surrounding $y = 0.5$ in order to resolve the current sheet. By $t = 120$ the magnetic island has opened up and the flow has organized itself into four distinct vortices of alternating sign. The mesh tracks the evolution of the solution, with refinement level 1 expanding to capture the magnetic island, de-refinement in the center of the island, and the remaining refinement levels focused on the vorticity. By $t = 200$, J has increased in the center of the magnetic island, and re-refinement has occurred to capture this behavior.

Finally, Table 1 shows the average number of nonlinear and linear iterations per time step as finer base grids and increasing numbers of refinement levels are used. The entries marked "–" are cases that were not run. The number of nonlinear iterations per time step is roughly constant for all cases. Reading horizontally, we note an increase in the number of linear iterations per time step as resolution is increased locally. This is consistent with the trend observed by reading vertically, where resolution is increased globally by using increasingly finer base grids. These

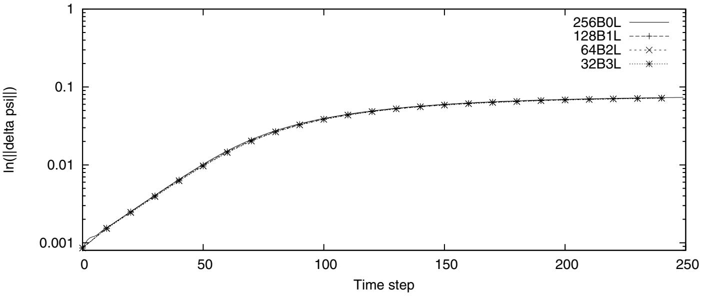
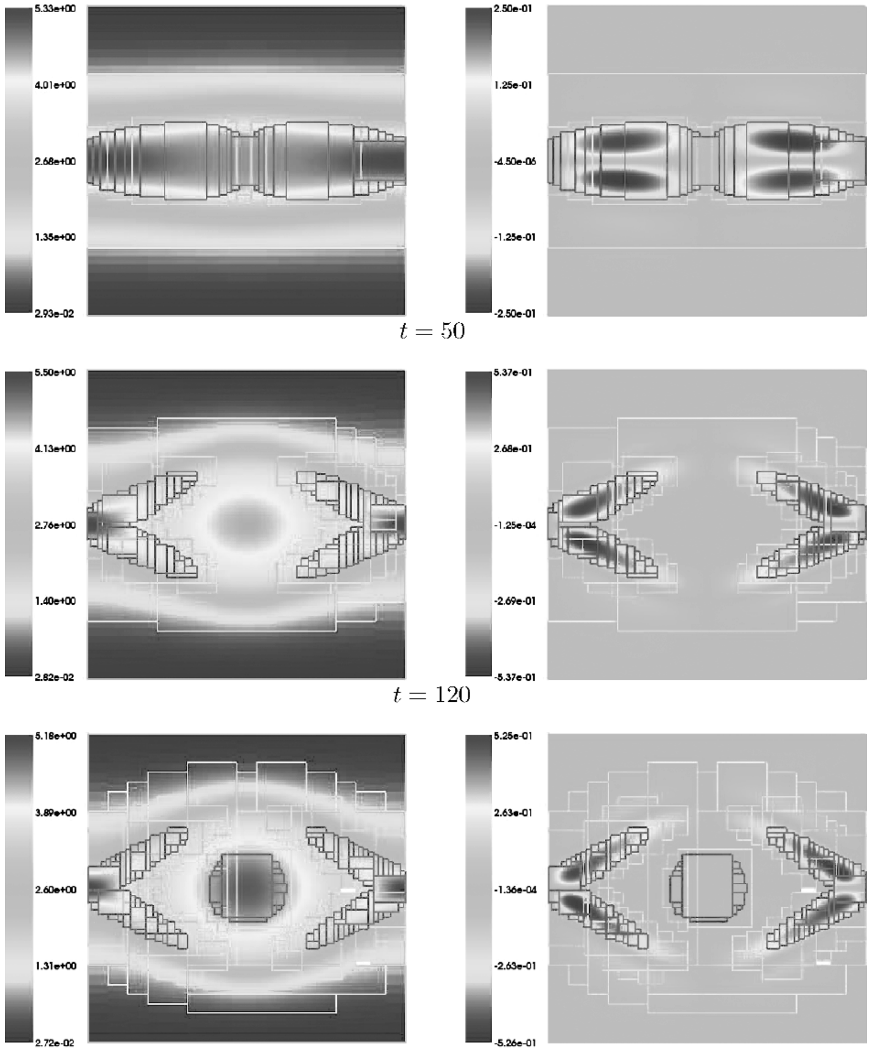


Fig. 1. Comparison of growth in $\delta\Psi$ for different grid configurations with the same finest resolution. The curves are labeled as "mBℓL", which indicates an $m \times m$ base grid and ℓ refinement levels.

trends are consistent with results found in [4], and are expected, because by increasing spatial resolution while running at a fixed time step, we are effectively running at larger multiples of the shear Alfvén wave explicit CFL limit. More importantly, reading diagonally (from lower left to upper right), we see that the number of linear iterations per time step is nearly constant for different grid configurations with a fixed finest resolution.

Table 1. Number of nonlinear iterations (NNI) and linear iterations (NLI), for different base grids and different numbers of refinement levels.

	NNI					NLI				
Levels	1	2	3	4	5	1	2	3	4	5
32 × 32	1.5	2.0	2.0	2.1	2.5	3.4	7.9	12.0	19.3	33.7
64 × 64	1.8	2.0	2.0	2.4	–	6.5	11.7	19.1	33.2	–
128 × 128	1.8	2.0	2.4	–	–	12.5	20.1	27.2	–	–
256 × 256	1.9	2.0	–	–	–	19.9	27.5	–	–	–
512 × 512	1.9	–	–	–	–	26.3	–	–	–	–



$t = 50$

$t = 120$

$t = 200$

Fig. 2. Evolution of solution and grid over time . The y -axis is scaled by a factor of 4. The current J is on the left and the vorticity ω is on the right. Nested refinement levels of AMR grid are outlined over the solution.

5 Conclusions

We have successfully demonstrated the use of implicit AMR for a reduced resistive model of MHD. The conventional streamfunction-vorticity formulation was found to be inappropriate for SAMR, but the current-vorticity formulation was shown to be amenable to this approach. Although a more formal accuracy study remains to be undertaken, we have demonstrated good agreement among predictions of the growth rate of the magnetic perturbation obtained from a variety of grid configurations having the finest resolution. By using FAC to implement our physics-based preconditioner, we have shown that the number of linear iterations per time step at a given resolution is nearly constant, a property that is necessary for implicit AMR to achieve performance gains that are commensurate with the reduction in problem size made possible by local mesh refinement.

6 Disclaimer

This work was performed under the auspices of the U.S. Department of Energy by Los Alamos National Laboratory, which is operated by the University of California under contract W-7405-ENG-36. Los Alamos National Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness. LA-UR 05-2645

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*, Tech. Rep. ANL-95/11 - Revision 2.1.6, Argonne National Laboratory, 2004.
2. P. N. Brown and Y. Saad, *Hybrid Krylov methods for nonlinear systems of equations*, SIAM J. Sci. Stat. Comput., 11 (1990), pp. 450–481.
3. L. Chacón and D. A. Knoll, *A 2D high-β Hall MHD implicit nonlinear solver*, J. Comput. Phys., 188 (2003), pp. 573–592.
4. L. Chacón, D. A. Knoll, and J. M. Finn, *An implicit, nonlinear reduced resistive MHD solver*, J. Comput. Phys., 178 (2002), pp. 15–36.
5. R. D. Hazeltine, M. Kotschenreuther, and P. J. Morrison, *A four-field model for tokamak plasma dynamics*, Phys. Fluids, 28 (1985), pp. 2466–2477.
6. D. A. Knoll and D. E. Keyes, *Jacobian-free Newton-Krylov methods: a survey of approaches and applications*, J. Comput. Phys., 193 (2004), pp. 357–397.
7. S. F. McCormick, *Multilevel Adaptive Methods for Partial Differential Equations*, SIAM, Philadelphia, PA, 1989.
8. M. Pernice and R. D. Hornung, *Newton-Krylov-FAC methods for problems discretized on locally refined grids*, Comput. Vis. Sci., 8 (2005), pp. 107–118.
9. H. R. Strauss and D. W. Longcope, *An adaptive finite element method for magnetohydrodynamics*, J. Comput. Phys., 147 (1998), pp. 318–336.

Embedded Pairs of Fractional Step Runge-Kutta Methods and Improved Domain Decomposition Techniques for Parabolic Problems

Laura Portero and Juan Carlos Jorge

Dpto. Matemática e Informática, Universidad Pública de Navarra[†], Campus Arrosadía s/n, 31.006, Pamplona, Spain. laura.portero@unavarra.es, jcjorge@unavarra.es

Summary. In this paper we design and apply new embedded pairs of Fractional Step Runge-Kutta methods to the efficient solution of multidimensional parabolic problems. These time integrators are combined with a suitable splitting of the elliptic operator subordinated to a decomposition of the spatial domain and a standard spatial discretization. With this technique we obtain parallel algorithms which have the main advantages of classical domain decomposition methods and, besides, avoid iterative processes like Schwarz iterations, typical of them. The use of these embedded methods permits a fast variable step time integration process.

1 Introduction

Let us consider a linear multidimensional parabolic problem with time dependent coefficients which we formulate in the following operational form: find $u : [t_0, T] \to \mathcal{H}$ such that

$$\begin{cases} \dfrac{\partial u}{\partial t} = A(t)u + f(t) \;\; \forall\, t \in (t_0, T], \\ u(t_0) = u_0 \;\in \mathcal{H}, \qquad Bu(t) = g(t) \;\in \mathcal{H}^b, \end{cases} \tag{1}$$

where $(\mathcal{H}, \|.\|)$ and $(\mathcal{H}^b, \|.\|^b)$ are two Hilbert spaces of functions defined on a bounded open subset $\Omega \subseteq \mathbb{R}^d$ and on its boundary Γ, respectively. $A(t) : \mathcal{D} \subseteq \mathcal{H} \to \mathcal{H}$ is an unbounded elliptic differential operator which contains the derivatives of the unknown u with respect to the spatial variables and $B : \mathcal{D} \subseteq \mathcal{H} \to \mathcal{H}^b$ is an abstract trace operator which determines the type of boundary conditions

[†] This research is partially supported by the MEC research project num. MTM2004-05221.

considered. We assume that the source term f, the initial condition u_0 and the boundary data g are sufficiently smooth and mutually compatible.

Numerical algorithms for the approximate solution of (1) can be designed and analyzed by combining a standard spatial discretization (using, for example, finite differences or finite elements) with an ODE solver as a time integrator. It is well known that if we choose fine grids for the spatial discretization and classical ODE solvers like Runge-Kutta (RK) or multistep methods, a large computational cost is required to obtain the numerical solution. Thus, the task of developing faster algorithms has been of great interest during the last decades and many different ideas have arisen in order to reduce somehow the computation time.

One alternative to obtain fast and robust algorithms is to discretize problem (1) first in time using an implicit Runge-Kutta scheme and then to use domain decomposition techniques (see [5]) to solve numerically the elliptic boundary value problems which arise in each internal stage. In this framework, where we consider the spatial domain Ω decomposed as the union of certain subdomains, the solution of a large linear system per internal stage is reduced to the solution of several sets of smaller linear systems. The main advantage of this technique is that the linear systems of every set can be solved in parallel. Nevertheless, the cost of an additional iterative process (e.g. A Schwarz iteration) is required to adjust the boundary conditions on the interior boundaries of the subdomains.

An interesting alternative to a classical ODE solver is to use a Fractional Step Runge-Kutta (FSRK) method as time integrator. The key to the efficiency of these schemes lies in splitting the original elliptic operator as the sum of certain "simpler" operators ($A = \sum_{i=1}^{m} A_i$). This decomposition combined with a FSRK method permits that only a part A_i of the elliptic differential operator A acts implicitly at each internal stage of the method in such a way that the derived elliptic boundary value problems are easier to solve. In this work we propose to decompose operator A into parts of the form $A_i = \psi_i A$, where $\{\psi_i\}_{i=1}^{m}$ is a smooth partition of unity subordinated to a decomposition of the spatial domain in m suitable overlapped subdomains. Similarly to what happens when classical domain decomposition techniques are used, in this case the numerical solution of each fractional step consists of solving a set of smaller linear systems whose solution can be parallelized. Besides, these schemes have an advantage over the classical domain decomposition schemes since they do not need any kind of Schwarz iterative processes to get the numerical solution. This technique was first introduced by Mathew et als. in [3], where they analyze this kind of splitting for certain low-order classical fractional step methods applied to solving parabolic equations with constant coefficients. The generalization of such a technique to the class of FSRK schemes used to approximate the solution of parabolic equations with time dependent coefficients is developed in [4].

The aim of the current paper is to follow these ideas but to decrease the computational cost even more by performing a variable time step integration. This will permit us to adapt the step sizes to the local behaviour of the solution as long as we have an estimate of the local error. In order to obtain a cheap estimate of such error we have developed some embedded pairs of FSRK methods of different orders. As with other classical one-step methods, the use of embedded formulas provides estimates of the local errors at a lower computational cost than if we choose other classical options like extrapolation methods or the use of two methods with different orders which do not share the internal stages.

2 Time semidiscretization

Let us consider for A and f partitions of the form: $A(t) = \sum_{i=1}^{m} A_i(t)$, $f(t) = \sum_{i=1}^{m} f_i(t)$, with $A_i(t) = \psi_i A(t), f_i(t) = \psi_i f(t)$, where $\psi_i(\bar{x})$ are sufficiently smooth functions such that $\sum_{i=1}^{m} \psi_i(\bar{x}) = 1, \forall \bar{x} \in \Omega$. To settle the definition of ψ_i, $i = 1, \ldots, m$, we decompose Ω as the union of m overlapping subdomains $\Omega = \bigcup_{i=1}^{m} \Omega_i$, each of them consisting of the union of a certain number of connected components $\Omega_i = \bigcup_{j=1}^{m_i} \Omega_{ij}$ such that $\Omega_{ij} \cap \Omega_{ik} = \emptyset$ for all $j, k \in \{1, \ldots, m_i\}$ with $j \neq k$. Then the partition of unity $\{\psi_i\}_{i=1}^{m}$ subordinated to the previous domain decomposition is constructed in such a way that, for each $i = 1, \ldots, m$, the function ψ_i vanishes outside subdomain Ω_i, takes the value 1 in every point which belongs only to Ω_i and some values between 0 and 1 in the overlaps of Ω_i with the remaining subdomains. For domain decompositions which have internal boundaries with simple geometries, $\psi_i(x)$, $i = 1, \ldots, m$, can be easily constructed as products of dilations, translations, etc., of the following $\mathbb{C}^\infty$ function (see section 5)

$$h(x) = 1 \text{ if } x < 0, \quad h(x) = e^{\frac{1}{2} e^{2} \log(2) \frac{e^{-\frac{1}{x}}}{x-1}} \text{ if } 0 \leq x \leq 1, \quad h(x) = 0 \text{ if } x > 1. \tag{2}$$

Let us establish now the formulation of a variable time step integration using an embedded pair of FSRK methods with m levels as follows

$$\left\{ \begin{array}{l} \left\{ \begin{array}{rl} U^{n,j} &= u_n + \tau_n \displaystyle\sum_{k=1}^{j} a_{jk}^{i_k} \Big(A_{i_k}(t_{n,k}) U^{n,k} + f_{i_k}(t_{n,k}) \Big), \\ B_{i_j} U^{n,j} &= g_{i_j}(t_{n,j}), \quad \text{for } j = 1, \ldots, s, \end{array} \right. \\ \widetilde{u}_{n+1} = u_n + \tau_n \displaystyle\sum_{j=1}^{s} \widetilde{b}_j^{i_j} \Big(A_{i_j}(t_{n,j}) U^{n,j} + f_{i_j}(t_{n,j}) \Big), \\ u_{n+1} = u_n + \tau_n \displaystyle\sum_{j=1}^{s} b_j^{i_j} \Big(A_{i_j}(t_{n,j}) U^{n,j} + f_{i_j}(t_{n,j}) \Big), \end{array} \right. \tag{3}$$

where $i_\bullet \in \{1, \ldots, m\}$, τ_n is the variable time step, $t_n = t_{n-1} + \tau_n$ and $t_{n,j} = t_n + c_j \tau_{n+1}$. $B_i : \mathcal{D}_i \to \mathcal{H}_i^b$, $i = 1, \ldots, m$, are the abstract trace operators which establish the type of boundary conditions required to calculate each internal stage and g_i are the boundary data; in this case, $B_i = \psi_i B$, $g_i = \psi_i g$, $\forall i = 1, \ldots, m$.

We assume that $\widetilde{u}_{n+1}$ approximates $u(t_{n+1})$ with order $\widetilde{p}$ and that u_{n+1} approximates the same semidiscrete solution also at t_{n+1} but with a higher order of approximation $p > \widetilde{p}$. Consequently, $est_{n+1} = \|u_{n+1} - \widetilde{u}_{n+1}\|$ estimates the local error for the lower order method at t_{n+1}. Notice that the most expensive calculations done to obtain $\widetilde{u}_{n+1}$ (i.e., the internal stages $U^{n,j}$, $j = 1, \ldots, s$) are also used in obtaining u_{n+1}.

In order to come to a more compact notation for FSRK schemes, (3) can be formulated as an embedded pair of Additive RK schemes

$$\left\{\begin{array}{l}\left\{\begin{array}{rl} U^{n,j} &= u_n + \tau_n \displaystyle\sum_{i=1}^{m}\sum_{k=1}^{s} a^i_{jk}\left(A_i(t_{n,k})U^{n,k} + f_i(t_{n,k})\right), \\ B_{i_j}U^{n,j} &= g_{i_j}(t_{n,j}), \quad \text{for } j = 1,\dots,s, \end{array}\right. \\ \widetilde{u}_{n+1} = u_n + \tau_n \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{s} \widetilde{b}^i_j\left(A_i(t_{n,j})U^{n,j} + f_i(t_{n,j})\right), \\ u_{n+1} = u_n + \tau_n \displaystyle\sum_{i=1}^{m}\sum_{j=1}^{s} b^i_j\left(A_i(t_{n,j})U^{n,j} + f_i(t_{n,j})\right), \end{array}\right. \tag{4}$$

if we extend the sums which appear in (3) by considering many additional zero coefficients: $a^i_{jk} = 0$ for $k > j$ and $a^i_{jk} = b^i_k = \widetilde{b}^i_k = 0$ for $i \neq i_k$.

Grouping the coefficients of the method into the following vectors and matrices $c = (c_i) \in \mathbb{R}^s$, $\widetilde{b}_i = (\widetilde{b}^i_j) \in \mathbb{R}^s$, $b_i = (b^i_j) \in \mathbb{R}^s$, $\mathcal{A}_i = (a^i_{jk}) \in \mathbb{R}^{s\times s}$ we can organize the coefficients of (4) in a table

$$\begin{array}{c|c|c|c|c} c & \mathcal{A}_1 & \mathcal{A}_2 & \dots & \mathcal{A}_m \\ \hline \text{order } \widetilde{p} & \widetilde{b}^T_1 & \widetilde{b}^T_2 & \dots & \widetilde{b}^T_m \\ \hline \text{order } p & b^T_1 & b^T_2 & \dots & b^T_m \end{array} ,$$

which is an extension of the Butcher's notation for a classical RK scheme. From now on, we will denote with $(c, (\mathcal{A}_i)^m_{i=1}, (\widetilde{b}_i)^m_{i=1})$ and $(c, (\mathcal{A}_i)^m_{i=1}, (b_i)^m_{i=1})$ the FSRK schemes involved in the embedded pair (3).

3 Spatial discretization and convergence results

We have to complete the previous time semidiscretization with a suitable spatial discretization to obtain a totally discrete scheme. Thus, we introduce a spatial discretization parameter h which tends to zero and we consider Ω_h meshes of $\overline{\Omega}$ which have been constructed taking into account the interior boundaries of the m subdomains. Next we denote with $(\mathcal{H}_h, \|.\|_h)$ and $(\mathcal{H}^b_{i,h}, \|.\|^b_{i,h})$ some finite dimensional Hilbert spaces of functions whose dimensions grow to infinity as h tends to zero; e.g. $\mathcal{H}_h$ consists of discrete functions on Ω_h if we use finite differences or piecewise polynomial functions associated to the mesh Ω_h if we use finite elements. In this framework we define operators $A_{i,h} : \mathcal{H}_h \to \mathcal{H}_h$ and $B_{i,h} : \mathcal{H}_h \to \mathcal{H}^b_{i,h}$ as certain consistent approximations of the operators A_i and B_i and we define $r_{i,h}(t) : \mathcal{D}_i \subseteq \mathcal{H} \to \mathcal{H}_h$, $\pi_h : \mathcal{H} \to \mathcal{H}_h$ and $\pi^b_{i,h} : \mathcal{H} \to \mathcal{H}^b_{i,h}$ as certain restriction or projection operators depending on whether we consider a spatial discretization using finite differences or finite elements, respectively. Using the previous notation, the totally discrete scheme can be expressed as follows

$$\begin{cases} \begin{cases} U_h^{n,j} = u_{n,h} + \tau_n \sum\limits_{k=1}^{j} a_{jk}^{i_k} \Big(A_{i_k,h}(t_{n,k}) U_h^{n,k} + \pi_h f_{i_k}(t_{n,k}) \Big) , \\ B_{i_j,h} U_h^{n,j} = \pi_{i_j,h}^b g_{i_j}(t_{n,j}), \qquad \text{for } j = 1, \ldots, s, \end{cases} \\ \widetilde{u}_{n+1,h} = u_{n,h} + \tau_n \sum\limits_{j=1}^{s} \widetilde{b}_j^{i_j} \Big(A_{i_j,h}(t_{n,j}) U_h^{n,j} + \pi_h f_{i_j}(t_{n,j}) \Big) , \\ u_{n+1,h} = u_{n,h} + \tau_n \sum\limits_{j=1}^{s} b_j^{i_j} \Big(A_{i_j,h}(t_{n,j}) U_h^{n,j} + \pi_h f_{i_j}(t_{n,j}) \Big) . \end{cases} \tag{5}$$

We can now take $est_{n,h} = \|u_{n,h} - \widetilde{u}_{n,h}\|_h$ as an approximation of est_n and use the same ideas of time step adaptation as for classical variable step ODE solver codes in order to keep $est_{n,h}$ below the value of a tolerance but close to it.

The solution of each internal stage in (5) consists of solving a linear system of the form $(\mathcal{I}_h - \tau_n\, a_{jj}^k A_{kh}(t_{n,j}) U_h^{n,j}) = F_h^{n,j}$, $(k = i_j)$, which can be decomposed into m_k independent linear subsystems that can be solved in parallel. Each one of these subsystems has a number of unknowns proportional to the number of mesh points on each component Ω_{ki} of Ω_k. It is also important to notice that no Schwarz iterations are required to obtain $u_{h,n+1}$.

Let us now give a brief review of the hypotheses assumed in order to guarantee an unconditional convergence result for the totally discrete scheme (5). The local errors of the time semidiscretization are $\rho_{n+1} = \|u(t_{n+1}) - u_{n+1}[t_n, u(t_n)]\|$ and $\widetilde{\rho}_{n+1} = \|u(t_{n+1}) - \widetilde{u}_{n+1}[t_n, u(t_n)]\|$, where $u_{n+1}[t_n, u(t_n)]$ and $\widetilde{u}_{n+1}[t_n, u(t_n)]$ are the approximations to $u(t_{n+1})$ obtained after one step of scheme (3) starting from $u_n = u(t_n)$. We assume that the embedded pair of FSRK methods (3) has orders $\widetilde{p}(p)$, i.e., $\widetilde{\rho}_{n+1} \le C\tau^{\widetilde{p}+1}$, $\rho_{n+1} \le C\tau^{p+1}$, where $\tau \equiv \max\limits_n \tau_n$ and C is a constant independent of τ. With the aim of obtaining a convergence result for the semidiscrete scheme (3), we combine the consistency with a suitable stability property. We say that the FSRK method $(c, (\mathcal{A}_i)_{i=1}^m, (b_i)_{i=1}^m)$ is A-stable iff $|R(z_1, \ldots, z_m)| \le 1$, $\forall\, z_1, \ldots, z_m \in \mathbb{C}^- \equiv \{z \in \mathbb{C} : \mathrm{Re}(z) \le 0\}$, where $R(z_1, \ldots, z_m) = 1 + \sum_{i=1}^m z_i\, b_i^T (\mathcal{I} - \sum_{j=1}^m z_j\, \mathcal{A}_j)^{-1} e$ is the amplification function associated to the FSRK method. In [1] it is proven that, under suitable hypotheses on operators $A_i(t)$ the use of an FSRK scheme which is consistent and A-stable guarantees the convergence of the time discretization process. Regarding the spatial discretization, we must assume typical order r properties of consistency as well as suitable stability properties.

Combining all these properties, the following unconditional convergence results are obtained for the totally discrete scheme (5) $\|r_h(t_n)u(t_n) - \widetilde{u}_{h,n}\|_h \le C(h^r + \tau^{\widetilde{p}})$, $\|r_h(t_n)u(t_n) - u_{h,n}\|_h \le C(h^r + \tau^p)$, where C is a constant independent of τ and h (see [4]).

4 Design of two embedded pairs of FSRK methods

We start with the design of a simple pair of orders 1(2). Let us consider the Fractionary Implicit Euler scheme with two levels

$$\begin{array}{c|cc|cc} 1 & 1 & & 0 & \\ 1 & 1 & 0 & 0 & 1 \\ \hline & 1 & 0 & 0 & 1 \end{array} \tag{6}$$

as the lower order method of the pair; it is first order consistent and A-stable. Now we want to construct a second order scheme whose two first stages coincide with the two first stages of (6). The sufficient and necessary conditions which a FSRK scheme should satisfy to have order p are shown in [2]; in this case ($p = m = 2$) such order conditions are $b_i^T e = 1,\ b_i^T c = \frac{1}{2},\ b_i^T \mathcal{A}_j\, e = \frac{1}{2}\ \forall\, i, j \in \{1, 2\},$ where $e = (1, \ldots, 1) \in \mathbb{R}^s$.

We need to add two implicit stages to (6) in order to obtain a second order method; in such a case we come to a system of 8 non linear equations which depend on 13 unknowns. After solving it we obtain a family of embedded pairs of FSRK methods of orders 1(2) with 5 free parameters ($b_3^1,\ b_4^2,\ a_{33}^1,\ a_{43}^1,\ a_{44}^2$).

Next we impose the property of A-stability. To simplify the study, we assume that $a_{33}^1 = a_{44}^2 = a$ and then we impose that $a_{43}^1 = \dfrac{2ab_3^1}{b_4^2}$ to permit a nearly L-stable behaviour (i.e., $R(\infty, \infty) \simeq 0$). By means of a numerical swept we obtain that $a \geq 2.35$ is a necessary requirement in order to have an A-stable FSRK scheme of order 2. We still have three parameters: a, b_3^1, b_4^2 , which we fix in such a way that the method has simple rational coefficients and also that the main term of the local error of the second order FSRK method is almost minimized. Using these ideas we have chosen the values $b_4 = \frac{3}{4}$, $b_3 = \frac{9}{10}$, $a = \frac{12}{5}$ and the resulting pair is

$$
\begin{array}{c|cccc|cccc}
1 & 1 & & & & 0 & & & \\
1 & 1 & 0 & & & 0 & 1 & & \\
\frac{4}{9} & -\frac{88}{45} & 0 & \frac{12}{5} & & 0 & \frac{5}{9} & 0 & \\
\frac{1}{3} & -\frac{407}{75} & 0 & \frac{144}{25} & 0 & 0 & -\frac{31}{15} & 0 & \frac{12}{5} \\
\hline
\text{order } 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
\hline
\text{order } 2 & \frac{1}{10} & 0 & \frac{9}{10} & 0 & 0 & \frac{1}{4} & 0 & \frac{3}{4}
\end{array}
$$

Following a similar technique, we have designed an embedded pair of FSRK schemes of orders 2(3). In this case we have chosen as the second order method the time integrator involved in the classical Peaceman & Rachford scheme and, by adding 4 suitable implicit stages, we have obtained the following pair

$$
\begin{array}{r|ccccccc|ccccccc}
0 & 0 & & & & & & & 0 & & & & & & \\
\frac{1}{2} & 0 & \frac{1}{2} & & & & & & \frac{1}{2} & 0 & & & & & \\
1 & 0 & 1 & 0 & & & & & \frac{1}{2} & 0 & \frac{1}{2} & & & & \\
\frac{7}{17} & 0 & -\frac{3}{34} & 0 & \frac{1}{2} & & & & \frac{7}{17} & 0 & 0 & 0 & & & \\
\frac{1}{2} & 0 & -\frac{11}{12} & 0 & \frac{17}{12} & 0 & & & \frac{1}{8} & 0 & -\frac{1}{8} & 0 & \frac{1}{2} & & \\
\frac{13}{17} & 0 & -\frac{27}{34} & 0 & \frac{18}{17} & 0 & \frac{1}{2} & & \frac{113}{289} & 0 & 0 & 0 & \frac{108}{289} & 0 & \\
1 & 0 & -\frac{208}{81} & 0 & \frac{289}{108} & 0 & \frac{289}{324} & 0 & \frac{1}{6} & 0 & -\frac{1}{3} & 0 & \frac{2}{3} & 0 & \frac{1}{2} \\
\hline
\text{order } 2 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 & 0 & 0 & 0 \\
\hline
\text{order } 3 & 0 & -\frac{208}{81} & 0 & \frac{289}{108} & 0 & \frac{289}{324} & 0 & \frac{1}{6} & 0 & -\frac{1}{3} & 0 & \frac{2}{3} & 0 & \frac{1}{2}
\end{array}
$$

5 Numerical examples

We consider the following diffusion-reaction problem

$$
\begin{cases}
\dfrac{\partial u}{\partial t} = (1+e^{-t})xy\Delta u - u + f(t,x,y), \quad (t,x,y) \in (0,500] \times \Omega, \\
u(0,x,y) = u_0(x,y), \;\; (x,y) \in \overline{\Omega}, \\
u(t,x,y) = 0, \;\; (t,x,y) \in (0,500] \times \Gamma,
\end{cases}
$$

where $\Omega = (0,1) \times (0,1)$ and data f and u_0 are chosen in such a way that $u(t,\bar{x}) = 3te^{-3t+1}\sin(\pi x)\sin(\pi y)$ is its exact solution.

We have decomposed domain Ω as the union of two overlapped subdomains $\Omega_1 = ((0,\frac{5}{16}) \cup (\frac{7}{16},\frac{13}{16})) \times (0,1)$, $\Omega_2 = ((\frac{3}{16},\frac{9}{16}) \cup (\frac{11}{16},1)) \times (0,1)$; each subdomain has two disjoint components. The partition of unity chosen subordinated to this decomposition is: $\psi_1(x,y) = h(8x - \frac{3}{2})$, if $x \in (0,\frac{3}{8})$, $\psi_1(x,y) = h(8x - \frac{7}{2})$, if $x \in [\frac{3}{8},\frac{5}{8})$, $\psi_1(x,y) = h(8x - \frac{11}{2})$, if $x \in [\frac{5}{8},1)$, where $h(x)$ is given in (2), and $\psi_2(x,y) = 1 - \psi_1(x,y)$. Finally, we decompose the elliptic operator and the source term into two parts as follows: $A_i(t,x,y) \equiv \psi_i(x,y)\Big((1+e^{-t})xy\Delta - \mathcal{I}\Big)$, $f_i(t,x,y) = \psi_i(x,y)f(t,x,y)$, $i = 1,2$.

We show in the following table the results obtained with the designed embedded pairs of orders 1(2) and 2(3), respectively. The spatial discretization chosen in both cases is central differences on a uniform rectangular mesh of $N \times N$ points which is convergent of second order; that is the reason why we have chosen a tolerance equal to $\dfrac{1}{N^2}$ to control the sizes of the time steps with the aim of having errors of the same size in space and time.

For different values of N, we show in the table the total number of steps (including the accepted and rejected ones), the efficacy, which is the percentage of accepted steps compared with the total number of steps, the average size of the accepted time steps and the maximum global errors committed along the whole integration interval. Note that the efficacy is very high and it improves for smaller tolerances

1(2) 2(3)	n_{tot}		efficacy %		$\overline{\tau}$		global error	
$N = 16$	71	34	91.55	88.24	7.6923	16.6667	3.4636E-2	3.8192E-2
$N = 32$	234	52	95.30	90.38	2.2422	10.6383	1.3269E-2	1.3289E-2
$N = 64$	630	81	97.46	93.83	0.8143	6.5789	4.5409E-3	4.5367E-3
$N = 128$	1532	128	98.63	96.09	0.3309	4.0650	1.4294E-3	1.4173E-3
$N = 256$	2017	211	99.90	95.73	0.2481	2.4752	4.3804E-4	4.7210E-4

and that the global errors obtained show a reduction according to the reduction of the tolerance ($\frac{1}{4}$) chosen when N doubles. As the exact solutions of these problems decay exponentially (in t) to the stationary state (0 in this case), the sizes of the time steps τ_n tend to grow along the integration in time from a certain point which provides a time integration which requires much fewer steps than when using constant time step integrators. Notice also that the same tolerance ($\frac{1}{N^2}$) has been used in both pairs for every value of N and that for these tolerances the embedded pair 1(2) needs many more time steps than the pair 2(3) to realize the integration. This implies that, although the pair 2(3) has two internal implicit stages more than the pair 1(2), the total computational cost of the integration for the same tolerance is much smaller for the embedded pair of orders 2(3), as expected. On the basis of this comparison, we think that the design of embedded pairs of FSRK schemes of higher orders is a very interesting task which we plan to pursue in the near future.

References

1. B. Bujanda and J. C. Jorge, *Stability results for fractional step discretizations of time dependent coefficient evolutionary problem*, Appl. Numer. Math., 38 (2001), pp. 69–86.
2. ———, *Fractional step Runge-Kutta methods for time dependent coefficient parabolic problems*, Appl. Numer. Math., 45 (2003), pp. 99–122.
3. T. P. Mathew, P. L. Polyakov, G. Russo, and J. Wang, *Domain decomposition operator splittings for the solution of parabolic equations*, SIAM J. Sci. Comput., 19 (1998), pp. 912–932.
4. L. Portero, B. Bujanda, and J. C. Jorge, *A combined fractional step domain decomposition method for the numerical integration of parabolic problems*, in Parallel Processing and Applied Mathematics: 5th International Conference, PPAM 2003, R. Wyrzykowski, J. Dongarra, M. Paprzycki, and J. Wasniewski, eds., vol. 3019 of Lecture Notes in Computuer Science, 2004, pp. 1034–1041.
5. A. Quarteroni and A. Valli, *Domain Decomposition Methods for Partial Differential Equations*, Oxford University Press, 1999.

Algebraic Multilevel Preconditioners for Nonsymmetric PDEs on Stretched Grids

Marzio Sala, Paul T. Lin, John N. Shadid, and Ray S. Tuminaro

Sandia National Laboratories, Albuquerque, NM 87185, USA. [†]

Summary. We report on algebraic multilevel preconditioners for the parallel solution of linear systems arising from a Newton procedure applied to the finite-element (FE) discretization of the incompressible Navier-Stokes equations. We focus on the issue of how to coarsen FE operators produced from high aspect ratio elements. The method uses on each level ℓ an auxiliary matrix B_ℓ, which contains inter-nodal distance information of the underlying initial FE grid. Then, a standard coarsening procedure is performed on B_ℓ and non-smoothed transfer operators are defined. Preliminary numerical results obtained on distributed memory parallel computers show that the use of the auxiliary matrix can greatly improve the convergence rate of the resulting multilevel preconditioner.

1 Introduction

We consider linear systems of type

$$Ax = b, \tag{1}$$

where $A \in \mathbb{R}^{n \times n}$ is a real square (sparse) matrix, arising from a stabilized FE discretization of the incompressible Navier-Stokes equations, possibly with heat and mass transfer, and $x, b \in \mathbb{R}^n$ are the solution vector and the right-hand side, respectively. The elements of A are defined by a Newton procedure (see for instance [11]), since the original problem is nonlinear.

The linear problem (1) is usually solved using a Krylov accelerator; therefore a preconditioner is mandatory. Several solution strategies have been presented in the literature; in this paper we will focus on multilevel methods.

[†] This work was partially supported by the ASC program and the DOE Office of Science MICS program at Sandia National Laboratory. 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.

The basic idea of multilevel methods is to capture errors by utilizing multiple resolutions in the iterative scheme. Oscillatory components are effectively reduced through a simple relaxation procedure. In these methods the smooth components are handled using an auxiliary, lower-resolution version of the problem. The idea is applied recursively on the next coarser level.

The first and best known example of a multilevel preconditioner is multigrid (see, for example, [5]). Although extremely successful for certain classes of problems, multigrid methods have the notable disadvantage of requiring the generation of a set of coarser grids, which can be difficult to generate for problems defined on complex geometries and unstructured grids. For this reason, we consider algebraic methods of the aggregation type; see [12].

Aggregation provides an automatic way of generating coarse levels and transfer functions to move solutions between the levels. The method has been thoroughly developed for symmetric systems and relies on the idea of generating low energy (or smooth) basis functions that capture the kernel (or near kernel) of the discrete system being solved. In [6] we have shown that aggregation methods can deliver convergence rates comparable to that of geometric multigrid, while being more flexible, for problems defined on structured non-stretched grids.

For problems with anisotropies, a procedure equivalent to the so-called semi-coarsening is required; see [7]. The basic idea of semi-coarsening is that the mesh is only coarsened in directions where smoothing is easily accomplished. Thus, for a problem which has weak coupling in the x direction, coarsening would only be performed in the y and z directions. Algebraic methods can mimic this approach by ignoring connections which are “weak” in the graph coarsening phase. That is, if the coupling between unknowns i and j is ignored, they will not be agglomerated together to define a coarse unknown. However, this strategy fails to deliver the required semi-coarsening if applied to matrices arising from bilinear FE on stretched grids, since all the entries in the computational stencil have comparable value. Without a proper semi-coarsening, the resulting multilevel preconditioner performs poorly on anisotropic problems.

In order to recover semi-coarsening, we proceed as follows. On each level ℓ, we introduce an auxiliary matrix, B_ℓ, defined using some information about the grid, so that the magnitude of the elements of B_ℓ reflects weak and strong connections in the FE problem. Anisotropic aggregates can be constructed using B_ℓ, adopting a conventional dropping technique. B_ℓ is defined using additional information that is usually available in standard finite-element codes. The use of an auxiliary matrix is certainly not new in the geometric multigrid community, although to the best of our knowledge no paper reports on its use with aggregation-based preconditioners.

Several other approaches have been proposed in the literature to improve the coarsening of algebraic multilevel methods. Chow [3] suggested to compute algebraically smoothed vectors, Broker *et al.* [2] proposed to take advantage of SPAI smoothers, and Brezina *et al.* [1] introduced the adaptive smoothed aggregation technique. Although promising, these techniques all rely on the computation of either a set of slowly converging vectors or SPAI smoothers, which are usually expensive operations.

This paper is organized as follows. Section 2 introduces the multilevel preconditioning algorithm we have adopted. Section 3 describes the proposed procedure to obtain semi-coarsening. Section 4 presents the numerical results, obtained on a distributed parallel computer. Finally, Section 5 outlines the conclusions.

2 Aggregation Multilevel Preconditioner

In this paper we focus on non-smoothed aggregation only, since no general theory is available to define a proper prolongator smoother for non-symmetric equations. The construction of the multilevel hierarchy in non-smoothed aggregation is performed by the following five steps. For each level ℓ (except the coarsest), do:

(a) Extract from A_ℓ the graph $\mathcal{G}_\ell$ to coarsen.
(b) Coarsen $\mathcal{G}_\ell$ to define a set of aggregates. Each aggregate defines a "grid point" on the coarser level.
(c) Define the prolongator P_ℓ and restriction R_ℓ .
(d) Compute the next-level matrix $A_{\ell+1}$ as $R_\ell \, A_\ell \, P_\ell$.

We now focus in more details on steps 1 and 2. For systems of equations, we define $\mathcal{G}_\ell$ by "condensing" all the physical unknowns corresponding to the same grid point, thus defining the "block" structure of A_ℓ . Each block has size $m \times m$, m being the number of physical unknowns. The graph coarsening is defined as follows:

$$e_{ij} \text{ is an edge of } \mathcal{G}_\ell \text{ iff } \quad |a_{ij}| \geq \theta \sqrt{|a_{ii}| \cdot |a_{jj}|}. \tag{2}$$

θ is the *threshold*, and $| \cdot |$ is an appropriate matrix norm. The $m \times m$ block elements $a_{i,j}$ that do not fulfill (2) are *dropped* in the construction of $\mathcal{G}_\ell$. A graph decomposition algorithm (such as those in METIS) is then applied to $\mathcal{G}_\ell$. The goal of this algorithm is to define groups of vertices (aggregates) such that each aggregate contains a tightly connected subgraph and so that each vertex is included in just one subgraph. Each aggregate will effectively become an unknown on the coarse mesh. Once the aggregates are defined, the prolongator matrix P_ℓ is constructed such that each row corresponds to a grid point and each column corresponds to an aggregate.

Once the multilevel hierarchy has been establish, an iteration (V-cycle) of the recursive algorithm is as follows. Starting from $\ell = 0$, on each level do:

(a) If on the coarsest level, solve with a direct solver and return.
(b) Do ν_1 iterations of pre-smoothing S_ℓ^{pre} .
(c) Compute the restricted residual $r_{\ell+1} = R_\ell \, r_\ell$.
(d) Recursively solve $A_{\ell+1} \, e_{\ell+1} = r_{\ell+1}$.
(e) Interpolate error, $e_\ell = P_\ell \, e_{\ell+1}$.
(f) Add the correction e_ℓ to the current iterate.
(g) Do ν_2 iterations of post-smoothing S_ℓ^{post} .

3 Definition of the Auxiliary Matrix

For problems defined on stretched grids, the distribution of nodes in the stretched direction will correctly represent the low frequencies, whereas, in the direction perpendicular to the stretching, it will represent the high frequencies. The closer two nodes are, the better they will represent the high frequency components of the error. For the problems considered in this paper, the matrix coefficients do not properly reflect the strength of connection between points, while the geometric information does (i.e., points that are geometrically distant from each other have a weak connection between them compared to points that are close to each other). Therefore, we want to form a matrix which captures this geometric information that can be used in the coarsening stage of the algorithm. The basic idea is to create a discrete Laplacian matrix where the size of the off-diagonal entries is related to the distance between points. In particular, we define

$$b_{i,j} = -\frac{1}{\|\mathbf{x}_i - \mathbf{x}_j\|^2}, \quad i \neq j, \quad b_{i,i} = \sum_{i \neq j} -b_{i,j},$$

where $\mathbf{x}_k$ represents the coordinates of node k .

B represents the finite-element mesh in the following sense: if a grid node i is "far" from j , then $b_{i,j}$ is "small", while if i is "close" to j , then $b_{i,j}$ is "large". The dropping technique (2) can now be straightforwardly applied to B to produce anisotropic aggregates.

The resulting algorithm for the definition of the multilevel preconditioner reads as follows:

```
(a) Build the auxiliary matrix $B_0$ using the nodal coordinates $\mathbf{x}$
(b) For each level $\ell$, do
(c)       Define the dropping value
(d)       Build graph $\mathcal{G}_\ell$ based on $B_\ell$
(e)       Create the aggregates using $\mathcal{G}_\ell$
(f)       Create the tentative prolongators $P_\ell$ and $R_\ell$
(g)       $B_{\ell+1} = R_\ell B_\ell P_\ell$
(h)       Destroy $B_\ell$
(i) EndFor
(j) Build a new hierarchy using $A$ and the $P_\ell$, $R_\ell$ previously computed
```

4 Numerical Results

We apply the algorithm described in Section 3 to the solution of the linear system arising from a stabilized FE discretization of the incompressible Navier-Stokes equations with energy and mass transport; see for instance [10]. The equations in residual form are:

$$R_P = \frac{\delta \rho}{\delta t} + \nabla \cdot (\rho \mathbf{u}) \tag{3}$$

$$\mathbf{R_m} = \rho \frac{\delta \mathbf{u}}{\delta t} + \rho(\mathbf{u} \cdot \nabla \mathbf{u}) - \nabla \cdot \mathbf{T} - \rho \mathbf{g} \tag{4}$$

$$R_T = \rho C_p \left[\frac{\delta T}{\delta t} + \mathbf{u} \cdot \nabla T \right] + \nabla \cdot \mathbf{q} - \phi - \sum_{k=1}^{N_s} h_k \nabla \cdot \mathbf{j}_k \tag{5}$$

$$R_{Y_k} = \rho \left[\frac{\delta Y_k}{\delta t} + \mathbf{u} \cdot \nabla Y_k \right] + \nabla \cdot \mathbf{j}_k \qquad k = 1, 2, ..., N_s - 1 \tag{6}$$

The FE code used for this work is MPSalsa [10], which uses a parallel Newton-Krylov solver on unstructured meshes. The calculations were performed on the Sandia Cplant machine, composed of nodes with one 500-MHz Dec Alpha processor and 1 GB of RAM, connected together by Myrinet. A classical aggregation procedure has been used to define the aggregates [9]. The smoother is one sweep of Gauss-Seidel (with damping parameter of 0.67) for either the first level or the first two levels,

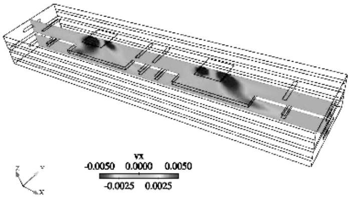


Fig. 1. Steady-state x-component of velocity for model 3D building.

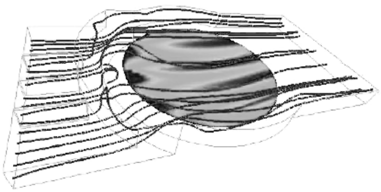


Fig. 2. 3D horizontal CVD reactor with a rotating disk.

while Aztec's incomplete factorization were adopted for the other levels. The KLU solver of the Amesos [8] library was used to solve the coarse problem. The threshold used in (2) was 0.05.

The first example involves the calculation of fluid flow, without thermal effects, in a simple prototype model of a building. This model represents a two-story building with the floors separated by two atria. Figure 1 shows a typical laminar steady-state solution. The centerline cutting plane shows the x-component of velocity. The worst aspect ratio hexahedral elements have the largest dimension that is five times larger than the smallest dimension. We consider laminar steady-state calculations to allow direct-to-steady-state solutions. Seven nonlinear iterations were required to reach convergence.

Table 1 shows an algorithmic scaling study for the steady-state calculations on hexahedral meshes and shows the reduction in iteration count provided by the auxiliary matrix as compared to without it. The larger meshes are generated by uniform refinement of previous meshes, with the number of processors being increased to maintain a roughly constant number of unknowns per processor. After each level of uniform refinement of the building geometry, the fine mesh is load-balanced using the ParMETIS graph partitioner through Zoltan [4]. The first three columns present the number of processors and unknowns and nonzeros in the fine level matrix. For both the case with and without an auxiliary matrix, the table presents the complexity of the hierarchy (sum of nonzeros of matrices on all levels divided by those of the finest matrix), the setup time in seconds, the average linear iterations per Newton step, and the average time per Newton step in seconds.

In the results, the number of unknowns per processor is kept roughly constant, and therefore a perfectly scalable preconditioner would converge in the same number of iterations as the number of processors used in the computation is increased. From Table 1, one can note that using isotropic aggregates with 16 and 128 processors the iterations increase from 57 to 90 (an increment of about 57%), while using anisotropic aggregates the difference is modest (about 25%). This makes the preconditioner based on the auxiliary matrix nearly scalable in terms of iterations to convergence, but still unsatisfactory from the point of view of CPU time. The large CPU times are due to one of the drawbacks of semi-coarsening: higher complexity. For this example, while isotropic aggregates reduces grid complexity between two consecutive levels by a factor of 27 in 3D, semi-coarsening only achieves a grid complexity reduction of 9 in 3D. This increases the setup and application cost of the resulting multilevel cycle, as well as the time required to compute the ILU factorizations.

proc	fine		5-level (GS/GS/ILU/ILU/KLU)							
	unks	nonzero	no auxiliary matrix				auxiliary matrix			
			complex	setup time (sec)	avg its/ Newt	time/ Newt (sec)	complex	setup time (sec)	avg ts/ Newt	time/ Newt (sec)
2	227K	22.4M	1.02	3.5	41	120	1.16	6.4	25	92
16	1.70M	175M	1.02	4.3	57	164	1.18	13.7	27	112
128	13.1M	1390M	1.02	8.0	90	434	1.21	30.4	34	264

Table 1. Comparison of five-level preconditioner (GS/GS/ILU/ILU/KLU) with and without auxiliary matrix for 3D model building; uncoupled aggregation; Cplant machine.

The second example involves the deposition of poly-Silicon in a rotating disk chemical vapor deposition (CVD) reactor. A mixture of trichlorosilane ($SiCl_3H$), HCl , and H_2 enters from the four inlets on the left, flows over a forward facing step, and over an inset rotating disk, depositing silicon on the wafer. Chemical reactions occur on the surface of the disk only and not in the flow. Figure 2 shows a schematic of the CVD reactor. A contour plot of poly-silicon deposition rate on the disk is shown, along with representative streamlines of the flow through the reactor.

Table 2 shows a scaling study of a simple continuation step where the thermodynamic pressure was increased from 0.6 to 0.85 atmospheres and the inlet flow velocity from 30 cm/sec to 35 cm/sec. The worst aspect ratio hexahedral element has largest dimension that is about a factor of ten larger than the smallest dimension. This table shows a comparison of the 1-level DD ILU preconditioner with the 5-level preconditioners with and without the auxiliary matrix. The smoothers for the 5-level preconditioner were one sweep of Gauss-Seidel on the finest level with damping parameter of 0.67 and ILU on the next three levels. Non-restarted GMRES was used with a linear solve convergence criterion of 3×10^{-4} . From the table, one can see that the auxiliary matrix has improved the iteration count, while the CPU time is only marginally reduced. This situation might be improved by using GS on the second level as in the previous example.

5 Conclusions

In this paper we have presented the application of a multilevel preconditioner for the parallel solution of large, sparse linear systems for FE discretizations on stretched grids. We have concentrated on the coarsening process. In order to improve the performance of our preconditioner, we introduced an auxiliary matrix, which contains information about the underlying finite-element grid. The coarsening is performed on an auxiliary matrix, then the final hierarchy is rebuilt on the linear system matrix.

proc	fine unks	1-level		5-level (GS/ILU/ILU/ILU/KLU)					
				no auxiliary matrix			auxiliary matrix		
		avg its/ Newt step	time (sec)	complex	avg its/ Newt step	time/ Newt (sec)	complex	avg its/ Newt step	time/ Newt (sec)
2	87400	49	125	1.01	75	118	1.06	48	103
16	636K	95	183	1.02	107	168	1.12	66	141
128	4.85M	221	409	1.02	164	319	1.19	97	313

Table 2. Comparison of five-level preconditioner (GS/ILU/ILU/ILU/KLU) with and without auxiliary matrix for CVD reactor; uncoupled aggregation; Cplant machine.

By resorting to an auxiliary matrix, anisotropic aggregation can be constructed at a negligible computational cost. The reported preliminary numerical results, obtained on a distributed parallel computer, show that the proposed approach can significantly improve the performance of the algebraic multilevel preconditioner in terms of iterations to convergence. Although more effective, the preconditioner, of higher complexity, is more expensive to construct and to apply.

Using anisotropic aggregates, the CPU time is significantly reduced for linear systems arising from the discretization of the incompressible Navier-Stokes equations, while for chemically reacting flows the results are less satisfactory. These preliminary results are encouraging although much more work on a broader range of numerical tests is required.

6 Acknowledgment

The authors wish to thank Andrew Salinger for the use of the CVD reactor mesh and problem definition, as well as supplying the figure of the reactor.

References

1. M. Brezina, R. Falgout, S. MacLachlan, T. Manteuffel, S. McCormick, and J. Ruge, *Adaptive smoothed aggregation (α SA)*, SIAM J. Sci. Comput., 25 (2004), pp. 1896–1920.
2. O. Bröker, M. J. Grote, C. Mayer, and A. Reusken, *Robust parallel smoothing for multigrid via sparse approximate inverses*, SIAM J. Sci. Comput., 23 (2001), pp. 1396–1417.
3. E. Chow, *An unstructured multigrid method based on geometric smoothness*, Numer. Linear Algebra Appl., 10 (2003), pp. 401–421.

4. K. Devine, E. Boman, R. Heaphy, B. Hendrickson, and C. Vaughan, *Zoltan data management services for parallel dynamic applications*, Computing in Science and Engineering, 4 (2002), pp. 90–97.
5. W. Hackbusch, *Multi-Grid Methods and Applications*, vol. 4 of Series in Computational Mathematics, Springer-Verlag, 1985.
6. P. T. Lin, M. Sala, J. N. Shadid, and R. S. Tuminaro, *Performance of fully coupled algebraic multilevel domain decomposition preconditioners for incompressible flow and transport*, Internat. J. Numer. Methods Engrg., (2006).
7. E. Morano, D. J. Mavriplis, and V. Venkatakrishnan, *Coarsening strategies for unstructured multigrid techniques with applications to anisotropic problems*, SIAM J. Sci. Comput., 20 (1998), pp. 393–415.
8. M. Sala, *Amesos 2.0 reference guide*, Tech. Rep. SAND2004-4820, Sandia National Laboratories, September 2004.
9. M. Sala, J. J. Hu, and R. S. Tuminaro, *ML 3.1 smoothed aggregation user's guide*, Tech. Rep. SAND2004-4819, Sandia National Laboratories, September 2004.
10. J. Shadid, S. Hutchinson, G. Hennigan, H. Moffet, K. Devine, and A. G. Salinger, *Efficient parallel computation of unstructured finite element reacting flow solutions*, Parallel Comput., 23 (1997), pp. 1307–1325.
11. J. N. Shadid, *A fully-coupled Newton-Krylov solution method for parallel unstructured finite element fluid flow, heat and mass transfer simulations*, Int. J. CFD, 12 (1999), pp. 199–211.
12. P. Vaněk, J. Mandel, and M. Brezina, *Convergence of algebraic multigrid based on smoothed aggregation*, Numer. Math., 88 (2001), pp. 559–579.

A Balancing Algorithm for Mortar Methods

Dan Stefanica

Baruch College, City University of New York, NY 10010, USA.
Dan_Stefanica@baruch.cuny.edu

Summary. The balancing methods are hybrid nonoverlapping Schwarz domain decomposition methods from the Neumann-Neumann family. They are efficient and easy to implement. We present a new balancing algorithm for mortar finite element methods. We prove a condition number estimate which depends polylogarithmically on the number of nodes on each subregion edge and does not depend on the number of subregions of the partition of the computational domain, just as in the conforming finite element case.

1 Introduction

The balancing method of Mandel [7] is a hybrid nonoverlapping Schwarz domain decomposition method from the Neumann-Neumann family. It is easy to implement and uses a natural coarse space of minimal dimension which allows for an unstructured partition of the computational domain. The condition numbers of the resulting algorithms depend polylogarithmically on the number of degrees of freedom in each subregion. There is a close connection between the balancing method and the FETI [5] and FETI–DP [4] methods; cf. [6]. A new version of the balancing method, also related to FETI–type algorithms, was recently proposed by Dohrmann et al. [9, 10].

Mortar finite elements were first introduced by Bernardi et al. [2] and are actively used in practice because of their advantages over the conforming finite elements, e.g., flexible mesh generation and straightforward local refinement. In this paper, we propose an extension of the balancing method to mortar finite elements. As in conforming cases, every local space is associated with a subregion from the partition of the computational domain. The values of the mortar function on a nonmortar side depend on, but are not equal to, its values on the mortar sides opposite the nonmortar. To account for this dependence, the local spaces are defined on extended subregions, instead of using local spaces and local solvers defined on each subregion. In this regard, our algorithm is different from classical Neumann-Neumann methods.

We establish a polylogarithmic upper bound for the condition number of our algorithm. The same bound has been obtained for the balancing algorithm of Dryja [3], as well as for other mortar algorithms, e.g., the iterative substructuring method of Achdou et al. [1], in the geometrically nonconforming case.

While the algorithm proposed here is based on a similar philosophy as the method suggested in [3], since the Schwarz framework is used to study the convergence properties of both algorithms, major differences exist between the two algorithms. For example, in the algorithm of [3], the local spaces are associated with pairs of opposite nonmortar and mortar sides.

2 Abstract Schwarz Theory

We use this elegant framework of the abstract Schwarz theory [12] to study the convergence properties of the balancing algorithm proposed in this paper.

Let V be a finite dimensional space, with a coercive inner product $a : V \times V \to \mathtt{R}$, and let $f : V \to \mathtt{R}$ be a continuous operator. We want to find the unique solution $u \in V$ of

$$a(u,v) \;=\; f(v), \quad \forall \; v \in V. \tag{1}$$

Assume that V can be written as $V = V_0 + V_1 + \ldots + V_N$, where the sum is not necessarily direct nor do we necessarily have $V_i \subset V$, $i = 0 : N$. Let $I_i : V_i \to V$ be embedding operators and let $\tilde{a}_i : V_i \times V_i \to \mathtt{R}$ be bilinear forms which are symmetric, continuous, and coercive. The corresponding projection-like operators $\widetilde{T}_i : V \to V_i$ are defined by

$$\tilde{a}_i(\widetilde{T}_i v, v_i) \;=\; a(v, I_i v_i), \quad \forall \; v_i \in V_i, \; v \in V. \tag{2}$$

Using the operators $T_i : V \to V$, $T_i = I_i \widetilde{T}_i$, the additive and multiplicative Schwarz methods for solving (1) can be introduced.

The balancing method is a hybrid method, combining the potential for parallelization of the additive methods and the fast convergence of the multiplicative methods. Choose the bilinear form $\tilde{a}_0$ to be exact, i.e., $\tilde{a}_0(\cdot,\cdot) = a(\cdot,\cdot)$. The coarse space solver T_0 is therefore a projection, subsequently denoted by P_0. The balancing method consists of solving $T_{bal} u = g_{bal}$, where

$$T_{bal} \;=\; P_0 + (I - P_0)(T_1 + \cdots + T_N)(I - P_0). \tag{3}$$

Here, g_{bal} is obtained by solving N local problems of the same form as (2) that do not require any knowledge of u. The equation $T_{bal} u = g_{bal}$ is a preconditioned version of (1) and can be solved without further preconditioning using CG or GMRES algorithms.

3 A Mortar Discretization of an Elliptic Problem

As a model problem for two dimensional self–adjoint elliptic PDEs with homogeneous coefficients, we choose the Poisson problem with mixed boundary conditions on Ω: given $f \in L^2(\Omega)$, find $u \in H^1(\Omega)$ such that

$$-\Delta u = f \text{ on } \Omega, \quad \text{with } u = 0 \text{ on } \partial\Omega_D \text{ and } \partial u/\partial n = 0 \text{ on } \partial\Omega_N, \tag{4}$$

where $\partial\Omega_N$ and $\partial\Omega_D$ are the parts of $\partial\Omega = \partial\Omega_N \cup \partial\Omega_D$ where Neumann and Dirichlet boundary conditions are imposed, respectively, and $\partial\Omega_D$ has positive Lebesgue measure.

To keep the presentation concise, we only discuss geometrically conforming mortar elements. Let $\{\Omega_i\}_{i=1:N}$ be a geometrically conforming mortar decomposition of a polygonal domain Ω of diameter 1 into rectangles of diameter of order H. (This notation is not coincidental: for the balancing method proposed here, each local space V_i will correspond to one subregion Ω_i.) The restriction of the mortar finite element space V^h to any rectangle Ω_i is a Q_1 finite element function on a mesh of diameter h. Weak continuity is required across Γ, the interface between the subregions $\{\Omega_i\}_{i=1:N}$. We choose a set of edges of $\{\Omega_i\}_{i=1:N}$, called nonmortars, which form a disjoint partition of Γ. For each nonmortar side γ there exists exactly one side opposite to it, which is called a mortar side. The jump $[w]$ of a mortar function $w \in V$ across any nonmortar γ must be orthogonal to a space of test functions $\Psi(\gamma)$, i.e.,

$$\int_\gamma [w] \; \psi \; ds \;=\; 0, \quad \forall \; \psi \in \Psi(\gamma). \tag{5}$$

In [2], $\Psi(\gamma)$ consists of continuous, piecewise linear functions on γ that are constant in the first and last mesh intervals of γ. Note that the end points of the nonmortar sides are associated with genuine degrees of freedom.

We discretize the Poisson problem (4) by using the mortar finite element space V^h and obtain the discrete problem:

$$\text{find } u_h \in V^h \text{ such that } a^\Gamma(u_h, v_h) = f(v_h), \quad \forall \; v_h \in V^h, \tag{6}$$

where the bilinear form $a^\Gamma(\cdot,\cdot)$ is defined as the sum of contributions from the individual subregions, and $f(\cdot)$ is the L^2-inner product by the function f:

$$a^\Gamma(v_h, w_h) \;=\; \sum_{i=1}^N \int_{\Omega_i} \nabla v_h \cdot \nabla w_h \; dx \quad \text{and} \quad f(v) = \int_\Omega f v \; dx.$$

Let $V^h(\Gamma)$ be the restriction of V^h to the interface Γ, and let V be the space of discrete piecewise harmonic functions defined as follows: if $v_\Gamma \in V^h(\Gamma)$, then its harmonic extension $\mathcal{H}(v_\Gamma) \in V$ is the only function in V^h which, on every subregion Ω_i, is equal to the harmonic extension of $v_\Gamma \mid_{\partial\Omega_i}$ with respect to the H^1-seminorm.

As in other substructuring methods, we eliminate the unknowns corresponding to the interior of the subregions. Problem (6) becomes a Schur complement problem on $V^h(\Gamma)$:

$$\text{find } u_\Gamma \in V^h(\Gamma) \quad \text{s.t.} \quad a^\Gamma(\mathcal{H}(u_\Gamma), \mathcal{H}(v_\Gamma)) = f(\mathcal{H}(v_\Gamma)), \; \forall \; v_\Gamma \in V^h(\Gamma). \tag{7}$$

For simplicity, we denote $V^h(\Gamma)$ by V and let $a(\cdot,\cdot) = a^\Gamma(\mathcal{H}(\cdot), \mathcal{H}(\cdot))$ be the inner product on V. Problem (7) can be formulated on V as follows:

$$\text{find } u \in V \text{ s.t. } a(u, v) = f(v), \quad \forall \; v \in V. \tag{8}$$

4 A Balancing Algorithm for Mortars

In this section, we introduce a new balancing algorithm for mortar finite elements. Our results can be extended to second order self-adjoint elliptic problems with mixed boundary conditions discretized by geometrically nonconforming mortars, and to three dimensional problems.

We solve (8) using the technique outlined in Section 2. To do so, we need to introduce a coarse space V_0 and local spaces V_i , $i = 1 : N$. The major difference between the classical balancing method and our algorithm for mortars is related to the extended subregions $\widetilde{\Omega}_i$, which replace the individual subregions in the definition of the local bilinear forms $\tilde{a}_i(\cdot,\cdot)$. An important role in the balancing algorithm is played by the counting functions associated with the interface nodes of each extended subregion. In [11], we have shown that defining $\tilde{a}_i(\cdot,\cdot)$ only on Ω_i does not lead to a convergent algorithm.

Extended Subregions: The extended subregion $\widetilde{\Omega}_i$ is defined as the union of Ω_i and all its neighbors that have a mortar side opposite $\partial\Omega_i$. Let $\mathtt{N}_i$ be the set of corner nodes of Ω_i , all the nodes on the mortar sides of Ω_i , and all the nodes on the mortar sides opposite the nonmortar sides of Ω_i .

The counting function $\nu_i : \Gamma \rightarrow \mathtt{R}$ corresponding to Ω_i is a mortar function taking the following values at the genuine degrees of freedom:

$$\nu_i(x) = \begin{cases} \text{number of sets } \mathtt{N}_j \text{ with } x \in \mathtt{N}_j, & \text{if } x \in \mathtt{N}_i; \\ 0, & \text{if } x \notin \mathtt{N}_i; \\ 1, & \text{if } x \in \partial\Omega_i \cap \partial\Omega_N. \end{cases}$$

In the geometrically conforming case, the value of ν_i at every interior node of the mortar sides where ν_i does not vanish is equal to 2 , and $\mathrm{Range}(\nu_i) \subseteq \{0, 1, \ldots, 4\}$. Let $\nu_i^\dagger$ be the mortar function with nodal values $\nu_i^\dagger(x) = 1/\nu_i(x)$ if $\nu_i(x) \neq 0$ and $\nu_i^\dagger(x) = 0$ otherwise. As in the continuous finite element case, $\nu_i^\dagger$ form a partition of unity, i.e., $\sum_{i=1}^{N} \nu_i^\dagger = 1$.

Coarse Space V_0 : The coarse space V_0 has one basis function, $\mathcal{H}(\nu_i^\dagger)$, the harmonic extension of $\nu_i^\dagger$, per subregion Ω_i . The bilinear form a_0 is exact, i.e., $\tilde{a}_0(\cdot,\cdot) = a(\cdot,\cdot)$. Therefore, $a(P_0 u, \mathcal{H}(\nu_i^\dagger)) \ = \ a(u, \mathcal{H}(\nu_i^\dagger))$, and

$$a((I - P_0)u, \mathcal{H}(\nu_i^\dagger)) \ = \ 0, \quad \forall \ u \in V. \tag{9}$$

Local Spaces: The local space V_i is associated with the subregion Ω_i , is embedded in V , i.e., $V_i \subset V$, and consists of piecewise harmonic functions which vanish at all the genuine degrees of freedom of $\Gamma \setminus \mathtt{N}_i$. The bilinear form $\tilde{a}_i(\cdot,\cdot) : V_i \times V_i \rightarrow R$ is defined using the extended subregion $\widetilde{\Omega}_i$:

$$\tilde{a}_i(v_i, w_i) \ = \ \sum_{\Omega_j \subset \widetilde{\Omega}_i} \int_{\Omega_j} \nabla\mathcal{H}(I_h(\nu_i v_i)) \cdot \nabla\mathcal{H}(I_h(\nu_i w_i)) \, dx, \tag{10}$$

where $I_h : L^2(\Omega) \rightarrow V$ is the nodal basis interpolation onto the mortar space V . The projection-like operator T_i is given by $T_i = I_i \widetilde{T}_i$, where

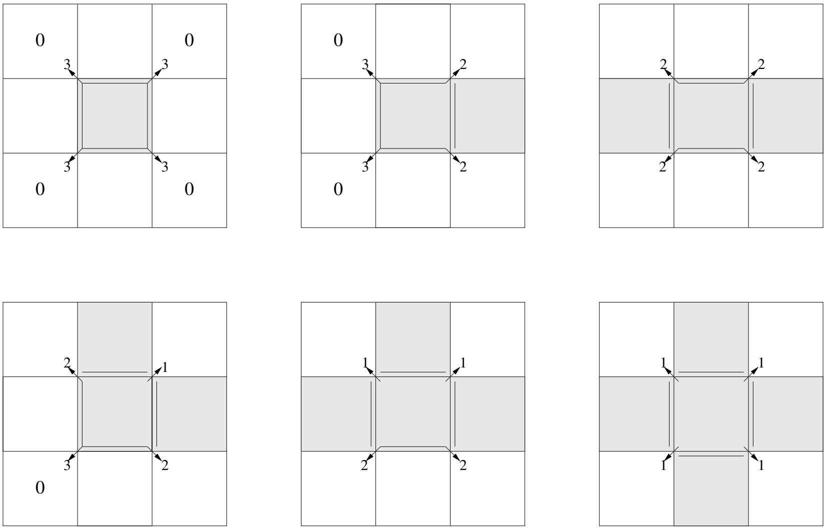


Fig. 1. All possible instances of extended subregions $\widetilde{\Omega}_i$ (shaded) corresponding to one subregion Ω_i (center in each picture). The values of the counting function ν_i at the corners of Ω_i are recorded. Mortar sides are marked with an additional solid line.

$$\tilde{a}_i(\widetilde{T}_i u, v_i) \; = \; a(u, v_i), \quad \forall \; v_i \in V_i. \tag{11}$$

If $\widetilde{\Omega}_i \neq \Omega_i$, i.e., if $\widetilde{\Omega}_i$ contains more than one subregion, then any $v_i \in V_i$ vanishes on $\partial\widetilde{\Omega}_i \backslash \partial\Omega_i$. The problem (11) is well–posed since it is a Poisson problem on $\widetilde{\Omega}_i$ with zero Dirichlet boundary conditions on $\partial\widetilde{\Omega}_i \setminus \partial\Omega_i$.

If $\widetilde{\Omega}_i = \Omega_i$, then all the sides of Ω_i are mortars; cf. Figure 1, upper left picture. This corresponds to the case of a floating subregion in the classical balancing algorithm, and requires using *balanced functions.* Note that $\mathcal{H}(\nu_i\nu_i^\dagger)$ is equal to 1 on Ω_i, and therefore

$$\tilde{a}_i(\widetilde{T}_i u, \mathcal{H}(\nu_i^\dagger)) \; = \; \int_{\Omega_i} \nabla\mathcal{H}(I_h(\nu_i \; \widetilde{T}_i u)) \cdot \nabla\mathcal{H}(\nu_i\nu_i^\dagger) \; dx \; = \; 0.$$

For the local problem (11) to be solvable, u must satisfy

$$a(u, \mathcal{H}(\nu_i^\dagger)) = 0, \tag{12}$$

for every floating subregion Ω_i. Such functions are called *balanced functions.* From (9), we conclude that any function in $\mathrm{Range}(I - P_0)$ is balanced.

Moreover, if $\widetilde{\Omega}_i = \Omega_i$, the local problem (11) corresponds to a pure Neumann problem. We make the solution unique by requiring $\widetilde{T}_i u$ to satisfy

$$\int_{\Omega_i} \mathcal{H}(I_h(\nu_i \; \widetilde{T}_i u)) \; dx \; = \; 0. \tag{13}$$

The preconditioned operator of our balancing algorithm for mortars is $T_{bal} = P_0 + (I - P_0)(T_1 + \cdots + T_N)(I - P_0)$. The convergence analysis of T_{bal} relies on that of the Neumann-Neumann operator $T_{N-N} = P_0 + T_1 + \cdots + T_N$, since $\kappa(T_{bal}) \leq \kappa(T_{N-N})$; cf., e.g., [8]. However, Neumann-Neumann algorithms with the spaces and approximate solvers considered in this paper would not converge.

5 Condition number estimate

The condition number estimate for our algorithm is based on abstract Schwarz theory; see, e.g., [12]. A technical results has to be proven first, and the techniques are somewhat different for floating and non-floating regions:

Lemma 1. *Let $u \in V$ and let $\overline{u}_i = \mathcal{H}(I_h(\nu_i^\dagger(u - \alpha_i))) \in V_i$, where α_i is the weighted averages of u over Ω_i , i.e.,*

$$\alpha_i = \frac{1}{\mu(\Omega_i)} \int_{\Omega_i} u \; dx. \tag{14}$$

Then,

$$a(u_i, u_i) \leq C\big(1 + log(H/h)\big)^2 \tilde{a}_i(u_i, u_i), \quad \forall \; u_i \in Range(T_i) \tag{15}$$

$$a(\overline{u}_i, \overline{u}_i) \leq C\big(1 + log(H/h)\big)^2 |u|^2_{H^1(\widetilde{\Omega}_i)}. \tag{16}$$

Also, if Ω_i is a floating subregion, i.e., if $\widetilde{\Omega}_i = \Omega_i$, then,

$$\tilde{a}_i(\overline{u}_i, \overline{u}_i) = |u|^2_{H^1(\widetilde{\Omega}_i)}.$$

If Ω_i is a nonfloating subregion, i.e., if $\widetilde{\Omega}_i \neq \Omega_i$, then $\tilde{a}_i(\overline{u}_i, \overline{u}_i) \leq C\big(1 + log(H/h)\big)^2 |u|^2_{H^1(\widetilde{\Omega}_i)}$.

Using the results of Lemma 1, we can show that $\tilde{a}_i(\cdot, \cdot)$ is bounded from below by $a(\cdot, \cdot)$, and prove that for any function in V there exists a stable splitting into local functions; see [11] for detailed proofs.

Lemma 2. *There exists a constant C , not depending on the local spaces V_i , such that*

$$a(u_i, u_i) \; \leq \; C\big(1 + log(H/h)\big)^2 \tilde{a}_i(u_i, u_i), \quad \forall \; u_i \in Range(T_i), \quad \forall \; i = 1 : N.$$

Lemma 3. *Let $u \in V$ and let α_i be the weighted averages (14) of u over Ω_i . Define $u_0 \in V_0$ as $u_0 = \sum_{i=1}^N \alpha_i \mathcal{H}(\nu_i^\dagger)$ and let $\overline{u}_i \in V_i$ be given by $\overline{u}_i = \mathcal{H}(I_h(\nu_i^\dagger(u - \alpha_i)))$. Then $u = u_0 + \sum_{i=1}^N \overline{u}_i$ and*

$$a(u_0, u_0) + \sum_{i=1}^N \tilde{a}_i(\overline{u}_i, \overline{u}_i) \; \leq \; C\big(1 + log(H/h)\big)^2 a(u, u).$$

Based on the results of Lemmas 2 and 3, a bound on $\kappa(T_{N-N})$, and therefore on $\kappa(T_{bal})$, can be established using the abstract Schwarz theory.

Theorem 1. *The condition number of the balancing algorithm is independent of the number of subregions and grows at most polylogarithmically with the number of nodes in each subregion, i.e.,*

$$\kappa(T_{bal}) \ \leq\ C\big(1 + log(H/h)\big)^4,$$

where C is a constant that does not depend on the properties of the partition.

6 Numerical Results

We have tested the convergence properties of our balancing algorithm for a two dimensional problem discretized by geometrically nonconforming mortar finite elements. The model problem was the Poisson equation on the unit square Ω with mixed boundary conditions. We partitioned Ω into $N = 16$, 32 , 64 , and 128 *geometrically nonconforming* rectangles, and Q_1 elements were used in each rectangle. For each partition, the number of nodes on each edge, H/h , was taken to be, on average, 4 , 8 , 16 , and 32 , respectively, for different sets of experiments. The preconditioned conjugate gradient iteration was stopped when the residual norm had decreased by a factor of 10^{-6} . The experiments were carried out in MATLAB. We report iteration counts, condition number estimates, and flop counts of our algorithm in Table 1.

Table 1. Convergence results, geometrically nonconforming mortars.

N	H/h	Iter	Cond	Mflops	N	H/h	Iter	Cond	Mflops
16	4	11	9.2	4.7e-1	64	4	14	9.9	4.0e+0
16	8	13	10.8	2.6e+0	64	8	15	12.1	1.6e+1
16	16	14	12.1	1.6e+1	64	16	17	13.4	9.4e+1
16	32	15	13.3	1.3e+2	64	32	19	13.9	7.2e+2
32	4	12	9.6	1.5e+0	128	4	14	10.3	1.0e+1
32	8	14	11.3	7.2e+0	128	8	15	12.0	3.6e+1
32	16	15	12.9	4.5e+1	128	16	18	13.7	2.1e+2
32	32	16	13.6	3.3e+2	128	32	19	13.9	1.5e+3

Our balancing algorithm has similar scalability properties as those of the classical balancing algorithm. When the number of nodes on each subregion edge, H/h , was fixed and the number of subregions, N , was increased, the iteration count showed only a slight growth. When H/h was increased, while the partition was kept unchanged, the small increase in the number of iterations was satisfactory. The condition number estimates exhibited a similar dependence, or lack thereof, on N and H/h .

References

1. Y. Achdou, Y. Maday, and O. B. Widlund, *Iterative substructuring preconditioners for mortar element methods in two dimensions*, SIAM J. Numer. Anal., 36 (1999), pp. 551–580.
2. C. Bernardi, Y. Maday, and A. T. Patera, *A new non conforming approach to domain decomposition: The mortar element method*, in Collège de France Seminar, H. Brezis and J.-L. Lions, eds., Pitman, 1994.
3. M. Dryja, *An iterative substructuring method for elliptic mortar finite element problems with a new coarse space*, East-West J. Numer. Math., 5 (1997), pp. 79–98.
4. C. Farhat, M. Lesoinne, P. LeTallec, 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 (2001), pp. 1523–1544.
5. C. Farhat and F.-X. Roux, *A method of Finite Element Tearing and Interconnecting and its parallel solution algorithm*, Int. J. Numer. Meth. Engrg., 32 (1991), pp. 1205–1227.
6. A. Klawonn and O. B. Widlund, *FETI and Neumann–Neumann iterative substructuring methods: Connections and new results*, Comm. Pure Appl. Math., 54 (2001), pp. 57–90.
7. J. Mandel, *Balancing domain decomposition*, Comm. Numer. Meth. Engrg., 9 (1993), pp. 233–241.
8. J. Mandel and M. Bresina, *Balancing domain decomposition for problems with large jumps in coefficients*, Math.Comp., 65 (1996), pp. 1387–1401.
9. J. Mandel and C. R. Dohrmann, *Convergence of a balancing domain decomposition by constraints and energy minimization*, Numer. Linear Algebra Appl., 10 (2003), pp. 639–659.
10. J. Mandel, C. R. Dohrmann, and R. Tezaur, *An algebraic theory for primal and dual substructuring methods by constraints*, Appl. Numer. Math., 54 (2005), pp. 167–193.
11. D. Stefanica, *A balancing algorithm for mortar finite elements.* preprint, 2005.
12. A. Toselli and O. B. Widlund, *Domain Decomposition Methods – Algorithms and Theory*, vol. 34 of Series in Computational Mathematics, Springer, 2005.

A Hybrid Parallel Preconditioner Using Incomplete Cholesky Factorization and Sparse Approximate Inversion

Keita Teranishi [1] and Padma Raghavan [1]

Department of Computer Science and Engineering, The Pennsylvania State University, 111 IST Bldg., University Park, PA 16802, USA.
teranish,raghavan@cse.psu.edu.

Summary. We have recently developed a preconditioning scheme that can be viewed as a hybrid of incomplete factorization and sparse approximate inversion methods. This hybrid scheme attempts to deliver the strengths of both types of preconditioning schemes to accelerate the convergence of Conjugate Gradients (CG) on multiprocessors. We provide an overview of our algorithm and report on initial results for some large sparse linear systems.

1 Introduction

Consider the solution of a sparse linear system $Ax = b$ on a distributed memory multiprocessor. When A is symmetric positive definite, preconditioned Conjugate Gradients (PCG) [9, 17] can be used to solve the system. In such a scheme, an effective preconditioner can accelerate the convergence of CG. Traditionally, incomplete Cholesky factorization with a drop threshold (ICT) [20] scheme can be used to construct a preconditioner $\hat{L}$ as an approximation to L, the sparse Cholesky factor of A ($A = LL^T$). Such an ICT preconditioner is often the method of choice on uniprocessors, but its scalable parallel implementation poses many challenges.

A parallel ICT scheme should ideally allow (i) efficient preconditioner construction, and (ii) latency-tolerant application at each CG iteration. Applying an ICT preconditioner requires distributed triangular solution which is typically inefficient due to the relatively large latencies of interprocessor communication on multiprocessors. We had earlier addressed this issue by developing a parallel ICT preconditioner with a feature called 'Selective Inversion' (SI) [14, 16, 18]. In our preconditioning scheme (ICT-SI), certain triangular submatrices were explicitly inverted to replace distributed substitution schemes by latency-tolerant matrix-vector multiplication.

Additionally, ordering, partitioning and blocking techniques from parallel sparse direct solvers were used to construct the ICT preconditioner efficiently. Our ICT-SI scheme enabled the scalable application of the preconditioner at each CG step while effectively accelerating the convergence of CG [16]. However, preconditioner construction using ICT-SI was still relatively expensive. In this paper, we attempt to address this issue by using sparse approximate inversion techniques [2, 5, 4, 8] instead of the explicit inversion required in ICT-SI. We call our new scheme ICT-SSAI, i.e., ICT with 'selective sparse approximate inversion' [18].

We provide a brief overview of sparse approximate inverse preconditioning and our incomplete Cholesky preconditioner with SI (ICT-SI) in Section 2. In Section 3, we describe our new ICT-SSAI scheme where sparse approximate inversion is used on selected submatrices in the incomplete factor $\hat{L}$ as an alternative to the SI scheme. We also provide some empirical results on the performance of our schemes and other preconditioners for three large sparse linear systems. We end with some concluding remarks in Section 4.

2 Background

Incomplete Cholesky factorization is a popular preconditioning scheme on uniprocessors. However, on multiprocessors with large latencies of interprocessor communication, the application of such preconditioners using parallel substitution does not scale well. This gave rise to a new class of preconditioning schemes that attempted to approximate an inverse of A which could then be applied using efficient parallel matrix-vector multiplication. However, these preconditioners may not be as effective as those from incomplete Cholesky [3] when systems from a wide range of applications are considered. Earlier, we had developed the ICT-SI scheme to enable latency tolerant application of ICT preconditioners on parallel multiprocessors. In this section, we provide a brief overview of sparse approximate inverse preconditioners and our ICT-SI. Our new ICT-SSAI preconditioner is in effect a hybrid of these two schemes.

2.1 Sparse Approximate Inverse Preconditioners

One sparse approximate inverse technique is based on the Frobenius norm minimization [5, 4, 8] of $\|I - AM\|_F$, where M is the preconditioner. This problem can be formulated as multiple least squares problems of the form: $\|I - AM\|_F^2 = \sum_{j=1}^{n} \|e_j - Am_j\|_2^2$. In this expression, e_j is a canonical vector and m_j is a j th column of M. The least-squares solution is computed using either iterative methods such as minimum-residual [5] or dense direct methods such as QR factorization [4, 8]. We use the freely available implementations of the latter named 'SPAI' [4, 8].

In the SPAI algorithm, the sparsity pattern of M is selected a priori. If A is symmetric and positive definite (SPD), the method tries to minimize $\|I - LG\|_F$, where L is a Cholesky factor of A and G is a lower triangular preconditioner matrix. Since each least-square solution can be computed independently, the method is highly suitable for parallel implementation as shown by Grote and Huckle [8], and by Chow [4]. However, preconditioning quality may lag that of ICT preconditioners.

2.2 Incomplete Cholesky with Selective Inversion

Our parallel incomplete Cholesky with SI uses many of the ideas from parallel sparse direct multifrontal solution. We start with a good fill-reducing strategy such as minimum-degree and nested dissection [7]; the latter also helps provide a natural data partitioning for the parallel implementations. We then compute an approximate $\hat{L}$ corresponding to the true factor L for the given ordering.

The parallel factorization and triangular solution of $\hat{L}$ is guided by the traversal of the elimination tree [12], and data is structured using supernodes [11, 13]. The elimination tree represents the data dependency between columns during factorization, and a supernode comprises a set of consecutive columns with nested sparsity structure to enable the use of cache-efficient techniques. A compact tree can be obtained from the elimination tree in terms of ancestor-descendant relationships between supernodes. We use such a supernodal tree in our implementations. The relationship between the separators and their supernodes in the tree is illustrated in Figure 1; the separators recursively partition the domain to form supernodes in a tree structure.

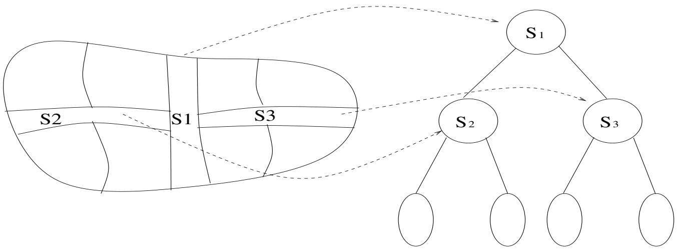


Fig. 1. Two levels of separators applied to a domain (left) and corresponding nodes in the supernodal tree (right).

During our ICT factorization some nonzero elements are dropped based on the drop threshold condition. Consequently, the column dependencies and the structure for supernodes derived from the coefficient matrix are not exact. However, these structures are used to manage the implementation of flexible dropping schemes to compute $\hat{L}$ for a range of fill to meet a variety of preconditioning needs. In addition, we utilize efficient dense matrix kernels [6] to perform the factorization at a supernode before applying drop threshold conditions.

The parallel implementation is based on the supernodal tree. Individual subtrees rooted at vertices located at approximately $\log P$ levels below the root of the entire tree are computed independently on each processor; these correspond to local sub-domains. At levels above the local subtrees, each supernode corresponds to a distributed separator and is processed by multiple processors using data-parallel dense/blocked operations. The incomplete factorization proceeds bottom-up on the tree. The forward solution proceeds bottom-up on the tree followed by a top-down backward solution. Figure 2 illustrates the computational scheme using four processors; computations for processor 0 are highlighted.

The performance of parallel triangular solution using substitution is typically degraded by the high latencies of interprocessor communication at each distributed

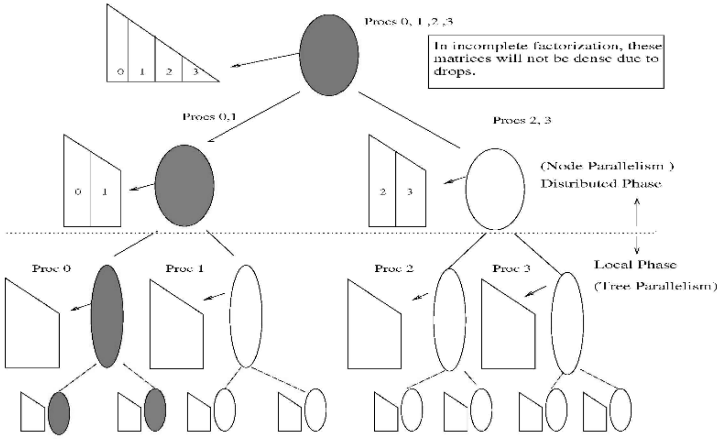


Fig. 2. The structure of the supernodal tree and submatrices associated with each node; a 4-processor computation is shown.

supernode. The triangular solution at a supernode a involves:

$$\begin{bmatrix} {L_{11}}^{a} \\ {L_{21}}^{a} \end{bmatrix} \begin{bmatrix} {x_1}^{a} \end{bmatrix} = \begin{bmatrix} {b_1}^{a} \\ {b_2}^{a} \end{bmatrix}.$$

The submatrices in the expression above are incomplete forms of the factor submatrix at supernode a. Parallel substitution is performed to obtain ${x_1}^{a}$ using ${L_{11}}^{a}$ and ${b_1}^{a}$; next ${b_2}^{a}$ is updated as $b_2 - {L_{21}}^{a}{x_1}^{a}$ and used in computations at the ancestor supernode of a.

The SI scheme [14, 16] includes a parallel matrix inversion of ${L_{11}}^{a}$ for each distributed supernode a. Subsequently, parallel substitution is replaced by sparse matrix vector multiplication ${x_1}^{a} \leftarrow {L_{11}^{a}}^{-1}{b_1}^{a}$. The scheme incurs the extra computational cost of inversion, but the improvements in applying the preconditioner are substantial [16].

3 ICT with Selective Sparse Approximate Inversion

Although the ICT-SI scheme described earlier achieves scalable application of the preconditioner, the construction of the preconditioner is relatively expensive. One of the reasons is that explicit inversion of the diagonal portion of sparse supernodal matrix causes fill-in, i.e., nonzeros which must be discarded by a drop threshold scheme. Consequently, even if a very sparse preconditioner is required, the cost of the construction is close to that for a true sparse factorization in a sparse direct solver. To alleviate this problem at a distributed supernode a, we use sparse approximate inversion to compute an approximation of ${L_{11}}^{a-1}$. For model sparse matrices from finite difference five-point 2-dimensional grids and seven-point 3-dimensional grids, we can show analytically that the arithmetic and communication costs for constructing the preconditioner using ICT-SSAI is lower in the order of magnitude sense than for ICT-SI [18].

We now provide some preliminary results on the performance of parallel ICT-SI and ICT-SSAI. We include comparisons with Block SSOR, Block Jacobi [1], level-0 incomplete Cholesky (IC(0)) [10], and the sparse approximate inverse preconditioner (ParaSails) [4]. We also report on the performance of DSCPACK [15], a parallel sparse direct solver as another point of comparison.

Our experiments were performed on a cluster with Intel Xeon processors and a Myrinet interconnect using the CG implementation in the PETSc package [1]. We terminate the CG iteration when the relative residual of a given unpreconditioned system is smaller than 10^{-8} . We set default parameters for ParaSails and use the same parameters for the selective sparse approximate inversion of the ICT-SSAI preconditioner construction. We report the performance of ICT-SI and ICT-SSAI with a drop-threshold value of 0.01 and a diagonal shift of 0.01 .

We use three large sparse matrices from finite-element and finite-difference applications described in Table 1. We used 1–16 processors for the first two smaller matrices and 4–64 processors to solve the largest matrix, **augustus7**.

Matrix	N	$\|A\|$	Description
cfd2	123,440	3,087,898	CFD: pressure matrix
engine	143,571	2,424,822	Engine head, linear tetrahedral elements
augustus7	1,060,864	9,313,876	Diffusion equation from 3D mesh

Table 1. Description of sparse matrices. N is the matrix dimension, $|A|$ is the number of nonzeros in the matrix.

The performance of all methods for the first two matrices (**cfd2** and **engine**) is summarized in Table 2. The best value for each measure, i.e., time for solution, and number of iterations, is shown in bold. The parallel performance for **augustus7** is shown in Figure 3 when the number of processors is increased from 4–64. In this figure the performance of DSCPACK on 64 processors is shown by a dotted line, indicating that all PCG schemes on 16 or more processors outperform the direct solver on 64 processors.

In Table 2 and Figure 3, our experiments indicate that ICT-SI leads to the least number of iterations and ICT-SSAI requires a slightly larger number of iterations. The increase in the number of iterations is a consequence of selectively using sparse approximate inversion. Observe that ParaSails leads to a higher number of iterations than ICT-SI and ICT-SSAI but fewer iterations than IC(0) and simpler preconditioners like SSOR and Jacobi. Preconditioner construction is less expensive in ICT-SSAI than in ICT-SI and this difference results in reduced total time on the larger number of processors. ICT-SSAI becomes the fastest method on 16 processors. We expect that the benefits of ICT-SSAI is more significant in applications where the preconditioner construction costs can be amortized over solutions for a sequence of right-hand-side vectors.

Method	Number of Processors										Mem
	1		2		4		8		16		
	Time	Its	Time	Its	Time	Its	Time	Its	Time	Its	
cfd2	Matrix size: 123,440 Nonzeros: 3,087,898										
SSOR		NC		NC		NC		NC		NC	1.0
Jacobi		NC		NC		NC		NC		NC	1.7
IC(0)		NC		NC		NC		NC		NC	2.0
ParaSails	192.8	782	115.7	747	61.4	776	31.7	777	17.5	776	3.9
ICT-SI	**88.7**	**451**	**45.2**	**461**	39.1	**463**	24.2	**464**	21.8	**471**	4.2
ICT-SSAI	**88.7**	**451**	50.8	498	**29.9**	554	**18.8**	569	**12.2**	583	4.0
engine	Matrix size: 143,571 Nonzeros: 2,424,822										
SSOR	170.4	1153		NC		NC		NC		NC	1.0
Jacobi	97.5	994	76.5	1436		NC		NC		NC	1.8
IC(0)		NC		NC		NC		NC		NC	2.0
ParaSails	130.4	760	103.3	760	77.4	761	62.3	762	46.1	761	3.9
ICT-SI	**48.1**	**282**	**35.2**	**252**	36.1	**287**	47.2	**306**	37.8	**308**	2.6
ICT-SSAI	**48.1**	**282**	36.7	336	**24.9**	356	**20.0**	356	**16.6**	387	2.4

Table 2. Performance of parallel preconditioners on two sparse matrix problems using 1 – 16 processors with the best instances shown in bold. The column labeled 'Time' is the total time (in seconds). The column labeled 'Its' is number of CG iterations; NC indicates that convergence was not achieved after 1,500 iterations. The column labeled 'MEM' contains the memory usage as a multiple of the space for the coefficient matrix.

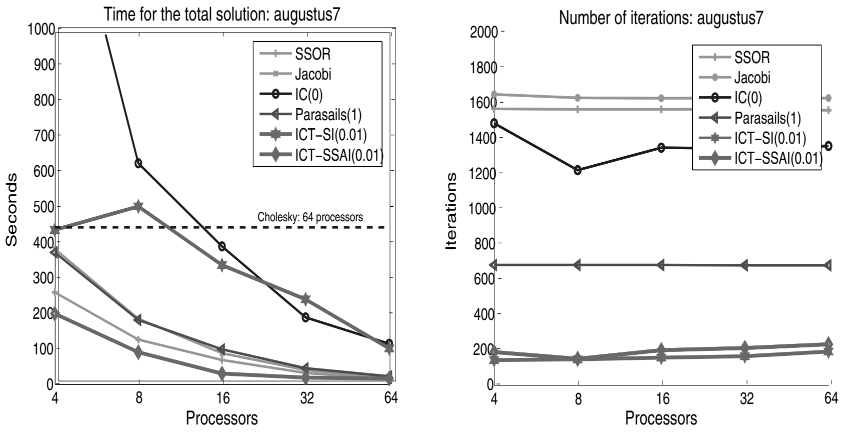


Fig. 3. Time to solve **augustus7** (left) and the number of iterations (right).

4 Conclusions

We have developed a parallel hybrid ICT-SSAI scheme which can potentially meet the preconditioning needs of sparse systems from complex applications. Initial empirical results are indeed encouraging and we are currently collaborating with Barry Smith to further test and refine our methods [19]. Our results indicate that ICT-SSAI successfully trades a slight decrease in the quality of the preconditioner for faster and more scalable preconditioner construction. We expect that our method can serve as a scalable limited memory solution scheme for applications that have traditionally relied on a direct solver for robust sparse linear system solution.

Acknowledgments

This research was funded in part by the National Science Foundation through grants ACI-0102537, EIA-0221916, and DMR-0205232.

We gratefully acknowledge several useful discussions with Edmond Chow and Barry Smith. We thank Michele Benzi for providing the sparse matrices from his test collection. We also thank the Mathematics and Computer Sciences Division at the Argonne National Laboratory for allowing us to use their network of workstations.

References

1. S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith, *PETSc users manual*, Tech. Rep. ANL-95/11 - Revision 2.1.1, Argonne National Laboratory, 2002.
2. M. Benzi, C. D. Meyer, and M. Tuma, *A sparse approximate inverse preconditioner for the conjugate gradient method*, SIAM J. Sci. Comput., 17 (1996), pp. 1135–1149.
3. M. Benzi and M. Tuma, *A comparative study of sparse approximate inverse preconditioners*, Appl. Numer. Math., 30 (1999), pp. 305–340.
4. E. Chow, *Parallel implementation and practical use of sparse approximate inverse preconditioners with a priori sparsity patterns*, Int. J. High Perf. Comput. Apps., 15 (2001), pp. 56–74.
5. E. Chow and Y. Saad, *Approximate inverse preconditioners via sparse-sparse iterations*, SIAM J. Sci. Comput., 19 (1998), pp. 995–1023.
6. J. J. Dongarra, J. Du Croz, S. Hammarling, and I. Duff, *A set of level 3 basic linear algebra subprograms*, ACM Trans. Math. Software, 16 (1990), pp. 1–17.
7. A. George and J. Liu, *Computer Solution of Large Sparse Positive Definite Systems*, Prentice-Hall, Englewood Cliffs, NJ, 1981.
8. M. J. Grote and T. Huckle, *Parallel preconditioning with sparse approximate inverses*, SIAM J. Sci. Comput., 18 (1997), pp. 838–853.
9. M. R. Hestenes and E. Stiefel, *Methods of conjugate gradients for solving linear systems*, J. Res. Nat. Bur. Standards, 49 (1952), pp. 409–436.
10. M. T. Jones and P. E. Plassmann, *The efficient parallel iterative solution of large sparse linear systems*, in Graph Theory and Sparse Matrix Computation, A. George, J. R. Gilbert, and J. W. Liu, eds., vol. 56 of IMA Volumes in Mathematics and Its Applications, Springer-Verlag, 1993, pp. 229–245.
11. J. Liu, E. Ng, and B. Peyton, *On finding supernodes for sparse matrix computations*, SIAM J. Matrix Anal. Appl., 14 (1993), pp. 242–252.

12. J. W. Liu, *The role of elimination trees in sparse factorization*, SIAM J. Matrix Anal. Appl., 11 (1990), pp. 134–172.
13. E. G. Ng and B. W. Peyton, *Block sparse Cholesky algorithms on advanced uniprocessor computers*, SIAM J. Sci. Stat. Comput., 14 (1993), pp. 1034–1056.
14. P. Raghavan, *Efficient parallel triangular solution with selective inversion*, Parallel Processing Letters, 8 (1998), pp. 29–40.
15. ———, *DSCPACK: Domain-separator codes for the parallel solution of sparse linear systems*, Tech. Rep. CSE-02-004, Department of Computer Science and Engineering, The Pennsylvania State University, 2002.
16. P. Raghavan, K. Teranishi, and E. G. Ng, *A latency tolerant hybrid sparse solver using incomplete Cholesky factorization*, Numer. Linear Algebra Appl., 10 (2003), pp. 541–560.
17. Y. Saad, *Iterative Methods for Sparse Linear Systems*, SIAM, Philadelphia, second ed., 2003.
18. K. Teranishi, *Scalable hybrid sparse linear solvers*, PhD thesis, Department of Computer Science and Engineering, The Pennsylvania State University, December 2004.
19. K. Teranishi, P. Raghavan, and B. F. Smith, *Tree-based parallel drop threshold incomplete Cholesky preconditioning using matrix inversion heuristics.* Presented at the 2005 International Conference on Preconditioning Techniques for Large Sparse Matrix Problems in Scientific and Industrial Applications, May 2005.
20. Z. Zlatev, *Use of iterative refinement in the solution of sparse linear systems*, SIAM J. Numer. Anal., 19 (1982), pp. 381–399.

A Three-Scale Finite Element Method for Elliptic Equations with Rapidly Oscillating Periodic Coefficients

Henrique Versieux [1] and Marcus Sarkis [2]

[1] Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, Brazil. versieux@impa.br [‡]

[2] Instituto Nacional de Matemática Pura e Aplicada, Rio de Janeiro, Brazil, and Worcester Polytechnic Institute, Worcester, MA 01609, USA. msarkis@impa.br [§]

1 Introduction

This paper presents a numerical scheme to approximate $u_\varepsilon \in H_0^1(\Omega)$, the weak solution of the problem

$$L_\varepsilon u_\varepsilon = -\frac{\partial}{\partial x_i}(a_{ij}(x/\varepsilon)\frac{\partial}{\partial x_j}u_\varepsilon) = f \text{ in } \Omega, \quad u_\varepsilon = 0 \text{ on } \partial\Omega, \tag{1}$$

where $\varepsilon \in (0,1)$ is the periodicity parameter, $a(y) = (a_{ij}(y))$ is a symmetric matrix with $a_{ij} \in C^{1,\beta}_{\text{per}}(Y)$, $\beta > 0$, i.e. $a_{ij} \in C^{1,\beta}_{\text{loc}}(\mathbb{R}^2)$ and is Y periodic. We assume that there exists $\gamma_a > 0$ such that $a_{ij}(y)\xi_i\xi_j \geq \gamma_a \|\xi\|^2$, $\forall\ \xi \in \mathbb{R}^2$ and $y \in Y$.

In several real world problems the scale ε is so much smaller than Ω that even with very heavy computer effort it is impossible to take $h < \varepsilon$, h being the mesh-size of the discrete method used to approximate u_ε . Recently new numerical methods have been proposed for capturing the oscillations of the scale ε presented in u_ε , and working with meshes sizes $h > \varepsilon$ (or $h >> \varepsilon$) ; see for example [1, 3, 4, 9, 8, 11]. The numerical method developed in [11] works in the case the domain Ω is rectangular. Here we extend this method to the case the domain Ω is a convex polygonal region with rational normals. This is possible due to the Lagrange multipliers introduced to approximate $\partial_\eta u_0$, resulting in a better error estimate for the H^1 broken semi-norm. The method presented here, in contrast to the methods in [1, 3, 4, 8], is based strongly on asymptotic expansions of u_ε .

We assume that $Y = [0,1] \times [0,1]$ and Ω is a bounded convex polygonal region in $\mathbb{R}^2$. More specifically, we assume that $\partial\Omega = \cup_{k=1}^N \Gamma^k$, where each Γ^k is a line segment with outward normal denoted by $N_k = (p_k, q_k)^t$, with p_k and q_k integers and relative prime. This hypothesis is required to guarantee periodicity of $a(x/\varepsilon)$ on the line containing Γ_k ; see [6].

[‡] Research supported in part by ANP.

[§] Research supported in part by CNPQ (Brazil) under grant 305539/2003-8 and by the U.S. National Science Foundation under grant CGR 9984404.

We use the standard notation $\|\cdot\|_s$, $\|\cdot\|_{s,p}$, and $\|\cdot\|_{s,h}$ for $H^s(\Omega)$ and $W_p^s(\Omega)$ norms, and for the H^s broken norms related to a regular partition $\mathcal{T}_h(\Omega) = K_1, K_2, ..., K_m$ of Ω, respectively. We always use the Einstein summation convention, i.e. repeated indices indicate summation, except for the index k. In what follows c denotes a generic constant independent of ε, h, and functions being evaluated.

2 Theoretical Approximation

2.1 The Asymptotic Expansion

The solution u_ε can be approximated by an asymptotic expansion. This approximation can be found using Equation (1) and the ansatz

$$u_\varepsilon(x) = u_0(x, x/\varepsilon) + \varepsilon u_1(x, x/\varepsilon) + \varepsilon^2 u_2(x, x/\varepsilon) + \cdots,$$

where the functions $u_j(x, y)$ are Y periodic in y. These terms are defined below; for more details see [2, 6].

Let $\chi^j \in H^1_{\text{per}}(Y)$ be the Y periodic weak solution with zero average over Y of

$$\nabla_y \cdot a(y)\nabla_y \chi^j = \nabla_y \cdot a(y)\nabla_y y_j = \frac{\partial}{\partial y_i} a_{ij}(y). \tag{2}$$

By regularity theory, we have that $\chi^j \in C^{2,\beta}_{\text{per}}(Y)$ when $a_{ij} \in C^{1,\beta}_{\text{per}}(Y)$. Define the matrix

$$A_{ij} = \frac{1}{|Y|} \int_Y a_{lm}(y) \frac{\partial}{\partial y_l}(y_i - \chi^i) \frac{\partial}{\partial y_m}(y_j - \chi^j) dy. \tag{3}$$

It is easy to see that the matrix A is symmetric positive definite. Define $u_0 \in H^1_0(\Omega)$ as the weak solution of

$$-\nabla . A\nabla u_0 = f \text{ in } \Omega, \tag{4}$$

and let $u_1(x, \frac{x}{\varepsilon}) = -\chi^j \left(\frac{x}{\varepsilon}\right) \frac{\partial u_0}{\partial x_j}(x)$. Note that $u_0 + \varepsilon u_1$ does not satisfy the zero Dirichlet boundary condition on $\partial\Omega$. In order to deal with this issue, a boundary corrector term $\theta_\varepsilon \in H^1(\Omega)$ is introduced as the weak solution of

$$-\nabla \cdot a(x/\varepsilon)\nabla\theta_\varepsilon = 0 \text{ in } \Omega, \qquad \theta_\varepsilon = -u_1(x, \frac{x}{\varepsilon}) \text{ on } \partial\Omega. \tag{5}$$

Therefore we obtain $u_0 + \varepsilon u_1 + \varepsilon\theta_\varepsilon \in H^1_0(\Omega)$.

2.2 Boundary Corrector Approximation

Note that the coefficients $a_{ij}(x/\varepsilon)$ and the boundary values $-u_1(x, \frac{x}{\varepsilon})$ of the Equation (5) are highly oscillatory. Hence it is not a trivial problem to obtain a good discretization for θ_ε. We propose an analytical approximation for θ_ε, denoted by ϕ_ε that satisfies the oscillating boundary condition and is more suitable for numerical approximation.

Observe that $u_0 = 0$ along $\partial\Omega$ implies $\nabla u_\varepsilon|_{\Gamma_k} = \eta^k \partial_{\eta^k} u_0$, where $\eta^k = N_k/|N_k|$. We then decompose $\theta_\varepsilon = \hat{\theta}_\varepsilon + \bar{\theta}_\varepsilon$ where

$$-\nabla \cdot a(x/\varepsilon)\nabla\hat{\theta}_\varepsilon = 0 \text{ in } \Omega, \quad \hat{\theta}_\varepsilon = -u_1 - \chi^* \partial_\eta u_0 \text{ on } \partial\Omega, \tag{6}$$

$$-\nabla \cdot a(x/\varepsilon)\nabla\bar{\theta}_\varepsilon = 0 \text{ in } \Omega, \quad \bar{\theta}_\varepsilon = \chi^* \partial_\eta u_0 \text{ on } \partial\Omega, \tag{7}$$

and $\chi^*|_{\Gamma_k} = \chi^*_k$ are properly chosen constants . In Remark 1, we will show that the Problems (6) and (7) are well posed. The approximation ϕ_ε for θ_ε is defined later as $\hat{\phi}_\varepsilon + \bar{\phi}_\varepsilon$, where $\hat{\phi}_\varepsilon \approx \hat{\theta}_\varepsilon$ and $\bar{\phi}_\varepsilon \approx \bar{\theta}_\varepsilon$. Next we define constants χ^*_k for which the approximation $\hat{\phi}_\varepsilon$ decays exponentially to zero away from the boundary.

Let $\tau^k = (\eta^k)^\perp$ be the $\pi/2$ rotation counterclockwise of η^k . We introduce the following normal and tangential coordinate system

$$\begin{pmatrix} y_1' \\ y_2' \end{pmatrix} = -\begin{pmatrix} \eta^k \cdot y \\ \tau^k \cdot y \end{pmatrix} \tag{8}$$

We observe that a function periodic in y with period 1 is periodic in y' with period $T_k = (p_k^2 + q_k^2)^{1/2}$. Associated to each side Γ_k of $\partial\Omega$, let $G_k = \{y \in \mathbb{R}^2;\ y_1' \leq 0;\ \text{and } 0 \leq y_2' \leq T_k\}$; $v_k \in H^1(G_k)$ is the weak solution of

$$\begin{aligned} &-\nabla_y \cdot a(y + \delta_\varepsilon\eta^k)\nabla_y v_k = 0 \text{ in } G_k \\ &v_k(y) = \chi^j(y + \delta_\varepsilon\eta^k)\eta^k_j \quad \text{on}\{y \in G_k, y_1' = 0\} \\ &v_k|_{y_2'=0} = v_k|_{y_2'=T_k}, \quad \text{for} \quad -\infty < y_1' < 0 \\ &\text{and} \quad \frac{\partial v_k}{\partial y_i}\exp(-\gamma y_1') \in L^2(G_k), \quad i = 1, 2, \end{aligned}$$

where $\delta_\varepsilon = T_k\left(s_k/\left(\varepsilon T_k\right) - \lfloor s_k/\left(\varepsilon T_k\right)\rfloor\right)$, and s_k is such that $\Gamma_k \subset \{x \in \mathbb{R}^2;\ x\cdot\eta^k = s_k\}$; ($\lfloor\cdot\rfloor$ denotes the integer part).

Let

$$\chi^*_k = \frac{1}{(A\eta^k, \eta^k)T_k}\left(\int_0^{T_k}\left[\chi^l a_{ij}\left(\delta_{jm} - \frac{\partial\chi^m}{\partial y_j}\right)\eta^k_i\eta^k_m\eta^k_l\right]\Bigg|_{y_1'=\delta_\varepsilon} dy_2' \right. \\ \left. + \int_{G_k}(a(y + \delta_\varepsilon\eta^k)\nabla_y v_k \cdot \nabla_y v_k)dy\right).$$

It can be shown [6] that $v_k - \chi^*_k$ decays exponentially to zero when $y_1' \to -\infty$, i.e. $(v_k - \chi^*_k)\exp(-\gamma y_1') \in L^2(G_k)$.

We note by Remark 1 that $(u_1(x, \frac{x}{\varepsilon}) - \chi^*\partial_\eta u_0)|_{\Gamma_k} \in H^{1/2}_{00}(\Gamma_k)$. Thus we can split $\hat{\theta}_\varepsilon = \sum_{k=1}^{N}\hat{\theta}^k_\varepsilon$, where

$$L_\varepsilon\hat{\theta}^k_\varepsilon = 0 \text{ in } \Omega, \qquad \hat{\theta}^k_\varepsilon = \begin{cases} -u_1(x, \frac{x}{\varepsilon}) - \chi^*\partial_\eta u_0 & \text{on } \Gamma_k \\ 0 & \text{on } \partial\Omega \setminus \Gamma_k. \end{cases}$$

We approximate $\hat{\theta}^k_\varepsilon$ by $\hat{\phi}^k_\varepsilon$ given by

$$\hat{\phi}^k_\varepsilon(x_1, x_2) = \varphi_k(x)\left(v_k\left(\frac{x - s_k\eta^k}{\varepsilon}\right) - \chi^*_k\right)\nabla u_0 \cdot \eta^k, \tag{9}$$

where $\varphi_k(x)$ is a cut-off function such that $\varphi_k|_{\Gamma_k} = 1$, $\varphi_k|_{\partial\Omega\setminus\Gamma_k} = 0$, and $\varphi_k \nabla u_0 \cdot \eta^k \in W^{1,\infty}(\Omega)$ if $u_0 \in W^{2,\infty}(\Omega)$. For example, assume $\Gamma_k = \{x \in \mathbb{R}^2;\ x_1 = 0,\ 0 \leq x_2 \leq c\}$ and that x_1^+ is the inner normal direction. Let Γ_{k-1} and Γ_{k+1} be the edges with vertices at the point $(0,c)$ and $(0,0)$, respectively, and let $\alpha_k > 0$ and $\alpha_{k+1} < 0$ be the angles between the x_1 axis and Γ_{k-1} and Γ_{k+1} , respectively. Then we define

$$\varphi_k(x) = \begin{cases} 1 & \text{if } 0 \leq x_1 \leq \delta;\ 0 \leq x_2 \leq c \\ 1 - (x_2 - c)/(x_1 \tan\alpha_k) & \text{if } 0 \leq x_1 \leq \delta;\ x_2 > c \\ 1 + x_2/(x_1 \tan\alpha_{k+1}) & \text{if } 0 \leq x_1 \leq \delta;\ x_2 < 0 \\ \text{smooth} & \text{if } \delta \leq x_1 \leq 2\delta \\ 0 & \text{if } x_1 \geq 2\delta. \end{cases}$$

From [5], we obtain that $\varphi_k \in W^{1,\infty}_{\text{loc}}(\Omega)$. Since $\partial_{\eta^k} u_0 \in H^{1/2}_{00}(\Gamma_k)$ and assuming $u_0 \in W^{2,\infty}(\Omega)$, we obtain $\varphi_k \nabla u_0 \cdot \eta^k \in W^{1,\infty}(\Omega)$. Hence $\hat{\phi}_\varepsilon = \sum_{k=1}^{N} \hat{\phi}^k_\varepsilon$ approximates $\hat{\theta}_\varepsilon$, and $\hat{\phi}_\varepsilon = \hat{\theta}_\varepsilon$ on the boundary of Ω .

The boundary condition imposed in Equation (7) does not depend on ε . An effective approximation for $\bar{\theta}_\varepsilon$ is given by $\bar{\phi} \in H^1(\Omega)$ the solution of

$$-\nabla \cdot A \nabla \bar{\phi} = 0 \ \text{ in } \ \Omega, \qquad \bar{\phi} = \chi^* \partial_\eta u_0 \ \text{ on } \ \partial\Omega.$$

We define our theoretical approximation for u_ε as $u_0 + \varepsilon u_1 + \varepsilon \phi_\varepsilon$, where $\phi_\varepsilon = \hat{\phi}_\varepsilon + \bar{\phi}$. Note that $\phi_\varepsilon|_{\partial\Omega} = \theta_\varepsilon|_{\partial\Omega}$, therefore $u_0 + \varepsilon u_1 + \varepsilon \phi_\varepsilon = 0$ on $\partial\Omega$. In [10], we prove the following error bounds

Theorem 1. *Assume that $a_{ij} \in C^{1,\beta}_{per}(Y)$ and $u_0 \in W^{2,\infty}(\Omega)$. Then there exists a constant c , such that*

$$\|u_\varepsilon - u_0 - \varepsilon u_1 - \varepsilon \phi_\varepsilon\|_1 \leq c\varepsilon \|u_0\|_{2,\infty}.$$

Remark 1. Since u_0 satisfies zero Dirichlet boundary condition on $\partial\Omega$ and $u_0 \in H^2(\Omega)$, we have $\dfrac{\partial u_0}{\partial \eta^k} \in H^{1/2}_{00}(\Gamma_k)$ and $\|\chi^* \partial_\eta u_0\|_{H^{1/2}(\partial\Omega)} \leq c(\chi^*)\|u_0\|_2$; see [7]. Note also that $u_1(x, \dfrac{x}{\varepsilon}) = -\chi^j\left(\dfrac{x}{\varepsilon}\right)\dfrac{\partial u_0}{\partial x_j}(x)$, and since $\chi^j \in C^{2,\beta}(\mathbb{R}^2)$ and $u_0 \in H^2(\Omega) \cap H^1_0(\Omega)$, we get $u_1|_{\Gamma_k} \in H^{1/2}_{00}(\Gamma_k)$.

3 Finite Element Approximation

We now describe how to approximate the terms u_0 , u_1 , $\hat{\phi}_\varepsilon$ and $\bar{\phi}$ numerically.

- Let $\chi^j_{\hat{h}}$ be a numerical approximation of χ^j using a second order accurate conforming finite element on a mesh $\mathcal{T}_{\hat{h}}(Y)$.
- Define $A^{\hat{h}}_{ij} = \dfrac{1}{|Y|}\displaystyle\int_Y a_{lm}(y)\frac{\partial}{\partial y_l}(y_i - \chi^i_{\hat{h}})\frac{\partial}{\partial y_m}(y_j - \chi^j_{\hat{h}})dy.$
- Let $V^h(\Omega)$ be the space of $\mathcal{P}_1$ finite elements associated to a triangular mesh $\mathcal{T}_h(\Omega)$, and $V^h_0(\Omega) = V^h(\Omega) \cap H^1_0(\Omega)$. Define $u^{h,\hat{h}}_0 \in V^h_0(\Omega)$ as the solution of

$$\int_\Omega (A^{\hat{h}} \nabla u^{h,\hat{h}}_0, \nabla v^h)dx = \int_\Omega f v^h dx, \qquad \forall v^h \in V^h_0(\Omega).$$

- Let $Y_k^h = \{\lambda^h \in L^2(\Gamma_k);\ \lambda^h = \phi^h|_{\Gamma_k},\ \phi^h \in V^h(\Omega)\}$, and $Y_{0,k}^h = \{\lambda^h \in Y_k^h;\ \lambda^h = 0 \text{ at } \partial\Gamma_k\}$. Define $\lambda_k^h \in Y_{0,k}^h$, as the solution of

$$\int_{\Gamma_k} \lambda_k^h \phi^h d\sigma = \int_\Omega A_{ij}^{\hat{h}} \partial_i u_0^{h,\hat{h}} \partial_j \phi^h dx - \int_\Omega f \phi^h dx, \tag{10}$$

  $\forall \phi^h \in V^h(\Omega);\ \phi^h|_{\partial\Omega\setminus\Gamma_k} = 0$. Observe that λ_k^h approximates $A\nabla u_0 \cdot \eta^k$ and that $u_0 \in H_0^1(\Omega) \cap H^2(\Omega)$ implies $\nabla u_0 \cdot \tau^k|_{\Gamma_k} = 0$. Hence define $\nu^{h,\hat{h}}$ by

$$A^{\hat{h}} \nu^{h,\hat{h}} \cdot \eta^k = \lambda_k^h,$$
$$\nu^{h,\hat{h}} \cdot \tau^k = 0,$$

  and then approximate $\partial_{\eta^k} u_0$ by $\mu^{h,\hat{h}} = \nu^{h,\hat{h}} \cdot \eta^k$.
- Define the approximation for ∇u_0 as $\Psi^{h,\hat{h}} = \nabla u_0^{h,\hat{h}} + \sum_{k=1}^N E_k(\mu^{h,\hat{h}} - \nabla u_0^{h,\hat{h}} \cdot \eta^k)\eta^k$, where given $g \in L^2(\Gamma_k)$ is such that $g|_{K_i \cap \Gamma_k} \in V^h(\Omega)|_{K_i\cap\Gamma_k},\ \forall\ K_i \in \mathcal{T}_h(\Omega);\ K_i \cap \Gamma_k \neq \emptyset$. $E_k(g)$ denotes the extension by zero to Ω of g satisfying $E_k(g)|_{K_i} \in V^h(\Omega)|_{K_i}$, $\forall\ K_i \in \mathcal{T}_h(\Omega)$.
- Define $u_1^{h,\hat{h}}(x, x/\varepsilon) = -\Psi_j^{h,\hat{h}}(x)\chi_{\hat{h}}^j(x/\varepsilon)$. Note that this leads to a nonconforming approximation for u_1 on the partition $\mathcal{T}_h(\Omega)$.
- Let p be a positive integer and $G_k^p = \{y \in \mathbb{R}^2; y_1' \leq 0,\ |y_1'| \leq p;\ \text{and } 0 \leq y_2' \leq T_k\}$. Define $\tilde{v}_k \in H^1(G_k^p)$ as the weak solution of

$$\begin{aligned}
&-\nabla_y \cdot a(y + \delta_\varepsilon \eta^k)\nabla_y \tilde{v}_k = 0 \ \text{ in } \ G_k^p \\
&\tilde{v}_k(y) = \chi_{\hat{h}}^j(y + \delta_\varepsilon \eta^k)\eta_j^k, \text{ on } \{y \in G_k, y_1' = 0\} \\
&\partial_\eta \tilde{v}_k = 0, \ \text{ on } \{y \in G_k^p;\ |y_1'| = p\} \\
&\text{and } v_k|_{y_2'=0} = v_k|_{y_2'=T_k}, \ \text{ for } \ |y_1'| < p.
\end{aligned}$$

  Let $v_k^{\hat{h},p}$ be a numerical approximation of $\tilde{v}_k$ using a second order accurate conforming finite element on a mesh $\mathcal{T}_{\hat{h}}(G_e^p)$.
- Define

$$\chi_k^{*,\hat{h},p} = \frac{1}{(A^{\hat{h}}\eta^k, \eta^k)T_k} \left(\int_0^{T_k} \left[\chi_{\hat{h}}^l a_{ij} \left(\delta_{jm} - \frac{\partial \chi_{\hat{h}}^m}{\partial y_j} \right) \eta_i^k \eta_m^k \eta_l^k \right] \Big|_{y_1' = \delta_\varepsilon} dy_2' \right.$$
$$\left. + \int_{G_k} (a(y + \delta_\varepsilon \eta^k)\nabla_y v_k^{\hat{h},p}) \cdot \nabla_y v_k^{\hat{h},p} dy \right),$$

- Observe that the term $v_k\left((x - s_k\eta^k)/\varepsilon\right)$ appears in Equation (9). Since the approximation $v_k^{\hat{h},p}$ is defined on G_k^p , we can calculate $v_k^{\hat{h},p}\left((x - s_k\eta^k)/\varepsilon\right)$ only if $|x_1' - s_k| \leq \varepsilon p$. The functions $v_k - \chi_k^*$ decays exponentially to zero in the $-\eta^k$ direction, hence its is natural to consider the following approximation

$$\hat{\phi}_\varepsilon^{e,h,\hat{h},p}(x_1, x_2) =$$
$$\begin{cases} \left(v_k^{\hat{h},p}\left(\dfrac{x - s_k\eta^k}{\varepsilon}\right) - \chi_k^{*,\hat{h},p} \right) \varphi_k \Psi^{h,\hat{h}} \cdot \eta^k & \text{if } \ |x_1' - s_k| < \varepsilon p \\ 0 & \text{if } \ |x_1' - s_k| \geq \varepsilon p. \end{cases}$$

Let $\hat{\phi}_\varepsilon^{h,\hat{h},p} = \sum_{k=1}^{N} \hat{\phi}_\varepsilon^{k,h,\hat{h},p}$.

- Let $\bar{\phi}^{h,\hat{h},p}$ be a second order accurate finite element approximation on a mesh of size h for the following equation

$$-\nabla A^{\hat{h}} \nabla \varrho = 0, \qquad \varrho = \chi^{*,\hat{h},p} \mu^{h,\hat{h}} \text{ on } \partial\Omega. \tag{11}$$

Remark 2. By construction $\mu^{h,\hat{h}} = 0$ at the corners of Ω, therefore $\chi^* \mu^{h,\hat{h}} \in H^{1/2}(\partial\Omega)$. This implies that Equation (11) is well posed. In addition $\chi^* \mu^{h,\hat{h}} \in V^h(\Omega)|_{\partial\Omega}$; hence we can look for a numerical solution of Equation (11) in $V^h(\Omega)$.

- Approximate θ_ε by $\phi_\varepsilon^{h,\hat{h},p} = \hat{\phi}_\varepsilon^{h,\hat{h},p} + \bar{\phi}^{h,\hat{h},p}$ and finally define the numerical solution for Equation (1) by $u_\varepsilon^{h,\hat{h},p} = u_0^{h,\hat{h}} + \varepsilon u_1^{h,\hat{h}} + \varepsilon \phi_\varepsilon^{h,\hat{h},p}$.

4 Error Analysis

When $p \to \infty$ and $\hat{h} \to 0$, we prove in [10] the following estimates.

Theorem 2. *Assume that $a_{ij} \in C_{per}^{1,\beta}(Y)$, $\beta > 0$ and $u_0 \in W^{2,\infty}(\Omega)$. Then there exists a constant c, such that*

$$\|u_\varepsilon - u_h\|_{1,h} \le c(h+\varepsilon)\|u_0\|_{2,\infty},$$
$$\|u_\varepsilon - u_h\|_0 \le c(h^2 + \varepsilon + \varepsilon h)\|u_0\|_{2,\infty}.$$

5 Numerical Experiments

Consider

$$a(x) = \left(\frac{2 + P\sin(2\pi x_1/\varepsilon)}{2 + P\cos(2\pi x_2/\varepsilon)} + \frac{2 + \sin(2\pi x_2/\varepsilon)}{2 + P\sin(2\pi x_1/\varepsilon)} \right) I_{2\times 2}, \quad \text{and} \quad f(x) = -1.$$

We compare the solution obtained by our method with the solution obtained by a second order accurate finite element method on a fine mesh with size h_f, which we call u_ε^*. Table 1 provide absolute errors estimates for $u_\varepsilon^* - u_\varepsilon^{h,\hat{h},p}$. We have used $p = 2$, $\hat{h} = 1/64$, $h_f = 1/2048$, and a triangular mesh with continuous piecewise linear functions to approximate $\chi_{\hat{h}}^j$ and $v_k^{\hat{h},p}$.

From Table 1, we see that for $\varepsilon << h$ we have errors of order $O(h^2)$ and $O(h)$ for the L^2 norm and H^1 semi norm , respectively. We observe that when we fix h and decrease ε the errors almost do not change. This is evidence that in this case the dominant error term is $O(h)$. Also looking at the diagonal values in this table we see clearly that the numerical error agrees with the theoretical rates from Theorem 2.

Table 2 shows the improvement obtained in the final approximation by considering the numerical approximation for the boundary corrector. We observe a better improvement on the $\|\cdot\|_0$ norm rather then on $|\cdot|_{1,h}$ semi norm. The reason for this is that $\bar{\phi}$ is obtained through the homogenized equation associated to Problem (7). Therefore it is a good approximation for $\bar{\theta}_\varepsilon$ in the $L^2(\Omega)$ norm but not in $|\cdot|_1$ semi norm. The term $\hat{\phi}_\varepsilon$ is defined in a thin boundary layer that primarily force the approximation to satisfy the zero Dirichlet boundary condition.

Table 1. $u_\varepsilon^* - u_\varepsilon^{h,\hat{h},p}$ error

$\|\cdot\|_0$ error

$\varepsilon \downarrow \quad h \rightarrow$	1/8	1/16	1/32	1/64
1/16	2.3863e-04	1.5793e-04		
1/32	2.3241e-04	8.0169e-05	1.7773e-05	
1/64	2.3540e-04	5.4314e-05	1.6020e-05	1.5601e-05

$|\cdot|_{1,h}$ error

1/16	0.0097	0.0067		
1/32	0.0086	0.0051	0.0036	
1/64	0.0086	0.0044	0.0025	0.0018

Table 2.

$\varepsilon = 1/64$, $h = 1/32$, $h_f = 1/1024$

	$\|\cdot\|_0$	$|\cdot|_{1,h}$
$u_\varepsilon^* - u_0^{h,\hat{h}}$	0.0287	0.0215
$u_\varepsilon^* - u_0^{h,\hat{h}} - \varepsilon u_1^{h,\hat{h}}$	0.0213	0.0026
$u_\varepsilon^* - u_0^{h,\hat{h}} - \varepsilon u_1^{h,\hat{h}} - \varepsilon\bar{\phi}^{h,\hat{h},p}$	5.0450e-05	0.0026
$u_\varepsilon^* - u_0^{h,\hat{h}} - \varepsilon u_1^{h,\hat{h}} - \varepsilon(\bar{\phi}^{h,\hat{h},p} + \hat{\phi}_\varepsilon^{h,\hat{h},p})$	5.1865e-05	0.0025

6 Conclusions

We propose a new method for approximating numerically the solution of Equation (1). This method is based strongly on the periodicity of the coefficients a_{ij} , and for this reason it has relative low computational cost with an optimal error convergence rate.

References

1. G. Allaire and R. Brizzi, *A multiscale finite element method for numerical homogenization*, Multiscale Model. Simul., 4 (2005), pp. 790–812.
2. A. Bensoussan, J. L. Lions, and G. Papanicolaou, *Asymptotic Analysis for Periodic Structures*, North Holland, New York, 1980.
3. Y. R. Efendiev, T. Y. Hou, and X.-H. Wu, *Convergence of a nonconforming multiscale finite element method*, SIAM J. Numer. Anal., 37 (2000), pp. 888–910.

4. T. Y. Hou and X.-H. Wu, *A multiscale finite element method for elliptic problems in composite materials and porous media*, J. Comp. Phys., (1997), pp. 169–189.
5. P.-L. Lions, *On the Schwarz alternating method. I.*, in First International Symposium on Domain Decomposition Methods for Partial Differential Equations, R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, eds., Philadelphia, PA, 1988, SIAM, pp. 1–42.
6. S. Moskow and M. Vogelius, *First-order corrections to the homogenized Eigenvalues of a periodic composite medium. a convergence proof*, Proc. Edinb. Math. Soc. Sect. A, 127 (1997), pp. 1263–1299.
7. P. Peisker, *On the numerical solution of the first biharmonic equation*, RAIRO Mathematical Modelling and Numerical Analysis, 22 (1988), pp. 655–676.
8. G. Sangalli, *Capturing small scales in elliptic problems using a residual-free bubbles finite element method*, Multiscale Model. Simul., 1 (2003), pp. 485–503.
9. C. Schwab and A.-M. Matache, *Generalized FEM for Homogenization Problems*, vol. 20 of Lecture Notes in Computational Science and Engineering, Springer, 2002, pp. 197–238.
10. H. M. Versieux and M. Sarkis, *Convergence analysis of numerical boundary correctors for elliptic equations with rapidly oscillating periodic coefficients.* In preparation.
11. ———, *Numerical boundary correctors for elliptic equations with rapidly oscillating periodic coefficients*, Commun. Numer. Methods Engrg., (2006).

A FETI Domain Decomposition Method Applied to Contact Problems with Large Displacements

Vít Vondrák [1] , Zdeněk Dostál [1] , Jiří Dobiáš [2] , and Svatopluk Pták [2]

[1] Dept. of Appl. Math., Technical University of Ostrava, 17. listopadu 15, CZ-70833, Ostrava-Poruba, The Czech Republic. vit.vondrak@vsb.cz, zdenek.dostal@vsb.cz

[2] Inst. of Thermomechanics, Czech Academy of Sciences, Dolejškova 5, CZ-18200, Prague 8, The Czech Republic. jdobias@it.cas.cz, svata@it.cas.cz

Summary. The solution of contact problems between solid bodies poses difficulties to solvers because in general neither the distributions of the contact tractions throughout the areas currently in contact nor the configurations of these areas are known a priori. This implies that the contact problems are inherently strongly nonlinear. Probably the most popular solution method is based on direct iterations with the non-penetration conditions imposed by the penalty method ([7] or [6]). The method enables us easily to enhance to include other non-linearity such as in the case of large displacements.

In this paper we are concerned with application of a variant of the FETI domain decomposition method that enforces feasibility of Lagrange multipliers by the penalty [1]. The dual penalty method, which has been shown to be optimal for small displacements is used in inner loop of the algorithm that treats large displacements. We give results of numerical experiments that demonstrate high efficiency of the FETI method

1 The Primal Penalty Method

The boundary conditions generated by bodies in contact are formally of the same form as the boundary conditions induced by externally applied surface tractions. However, the difficulties with the contact tractions is that in general we know neither their distributions throughout the areas currently in contact, nor the shapes and magnitude of these areas until we have solved the problem. Finding them has to be part of the solution.

From now on, we will consider the frictionless contact. There exist two basic methods to remove the contact constraints. The first one is the Lagrange multiplier method and the second one the penalty method, or in the sense of this paper the primal penalty method. With the latter method constraints are enforced by penalization. The penalization of the Kuhn–Tucker conditions in the normal direction is established by introducing a penalty parameter ε_n in

$$f_n = \varepsilon_n \langle g \rangle \tag{1}$$

where f_n stands for the normal contact force, g denotes the depth of interpenetration of the bodies in contact and $\langle . \rangle = 0.5[(.) + |.|]$ is known as the Macauley bracket. It returns the non-negative part of its operand. The normal penalty can be seen as the stiffness of a spring placed between corresponding contacting surfaces. The penalty method yields an exact solution if the penalty tends to infinity, but otherwise permits certain violation of the constraint that the interpenetration has to be zero. In practice it is necessary to estimate the magnitude of the penalty parameter to limit the penetration, yet it should not be too large to avoid ill-conditioning. The penalty parameter should be increased if the grid is refined.

2 Application of FETI to Contact Problems

Let us briefly outline the fundamental formulae of the FETI method. Consider solid bodies in contact, discretized on a finite element mesh and in addition decomposed into sub-domains. The numerical approximation to the problem in terms of the finite element discretization and auxiliary domain decomposition can be expressed as

$$\min \frac{1}{2} u^\top K u - f^\top u \quad \text{subject to} \quad B^I u \leq 0 \quad \text{and} \quad B^E u = 0 \tag{2}$$

where A stands for a positive semi-definite stiffness matrix, B^I and B^E denote the full rank matrices which enforce the discretized inequality constraints describing conditions of non-interpenetration of bodies and inter-subdomain equality constraints, respectively, and f stands for the discrete analogue of the linear form $\ell(u)$.

Denoting

$$\lambda = \begin{bmatrix} \lambda^I \\ \lambda^E \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} B^I \\ B^E \end{bmatrix},$$

we can write the Lagrangian associated with problem (2) as

$$L(u, \lambda) = \frac{1}{2} u^\top K u - f^\top u + \lambda^\top B u.$$

It is well known that (2) is equivalent to the saddle point problem

$$\text{Find} \quad (\overline{u}, \overline{\lambda}) \quad \text{s.t.} \quad L(\overline{u}, \overline{\lambda}) = \sup_{\lambda_I \geq 0} \inf_u L(u, \lambda). \tag{3}$$

After eliminating the primal variables u from (3), we obtain the minimization problem

$$\min \ \Theta(\lambda) \quad \text{s.t.} \quad \lambda_I \geq 0 \quad \text{and} \quad R^\top (f - B^\top \lambda) = 0, \tag{4}$$

where

$$\Theta(\lambda) = \frac{1}{2} \lambda^\top B K^\dagger B^\top \lambda - \lambda^\top B K^\dagger f, \tag{5}$$

$K^{\dagger}$ denotes a generalized inverse that satisfies $KK^{\dagger}K = K$, and R denotes the full rank matrix whose columns span the kernel of K.

Even though problem (4) is much more suitable for computations than (2), further improvement may be achieved by adopting some simple observations and the results of Farhat, Mandel, Roux and Tezaur [3, 5]. Let us denote

$$F = BK^{\dagger}B^{\top}, \quad \widetilde{G} = R^{\top}B^{\top}, \quad \widetilde{e} = R^{\top}f, \quad \widetilde{d} = BK^{\dagger}f,$$

and let $\widetilde{\lambda}$ solve $\widetilde{G}\widetilde{\lambda} = \widetilde{e}$, so that we can transform the problem (4) to the minimization on a subset of the vector space by looking for the solution in the form $\lambda = \mu + \widetilde{\lambda}$. Since

$$\frac{1}{2}\lambda^{\top}F\lambda - \lambda^{\top}\widetilde{d} = \frac{1}{2}\mu^{\top}F\mu - \mu^{\top}(\widetilde{d} - F\widetilde{\lambda}) + \frac{1}{2}\widetilde{\lambda}^{\top}F\widetilde{\lambda} - \widetilde{\lambda}^{\top}\widetilde{d},$$

problem (4) is, after returning to the old notation, equivalent to

$$\min \quad \frac{1}{2}\lambda^{\top}F\lambda - \lambda^{\top}d \quad \text{s.t.} \quad G\lambda = 0 \quad \text{and} \quad \lambda^{I} \geq -\widetilde{\lambda}^{I} \tag{6}$$

where $d = \widetilde{d} - F\widetilde{\lambda}$ and G denotes a matrix arising from the orthonormalization of the rows of $\widetilde{G}$.

Our final step is based on the observation that the problem (6) is equivalent to

$$\min \quad \frac{1}{2}\lambda^{\top}PFP\lambda - \lambda^{\top}Pd \quad \text{s.t.} \quad G\lambda = 0 \quad \text{and} \quad \lambda^{I} \geq -\widetilde{\lambda}^{I} \tag{7}$$

where

$$Q = G^{\top}G \qquad \text{and} \qquad P = I - Q$$

denote the orthogonal projectors on the image space of $G^{\top}$ and on the kernel of G, respectively. Enhancing the equality constraints in (7) by the penalty into the function

$$\Theta_{\rho}(\lambda) = \frac{1}{2}\lambda^{\top}(PFP + \rho Q)\lambda - \lambda^{\top}Pd, \tag{8}$$

we can approximate the solution of (7) by the solution of

$$\min \quad \Theta_{\rho}(\lambda) \quad \text{s.t.} \quad \lambda^{I} \geq -\widetilde{\lambda}^{I} \tag{9}$$

with a sufficiently large penalty parameter ρ. Note that the image spaces of the projectors P and Q are invariant subspaces of the Hessian $H^{\rho} = PFP + \rho Q$ of $\Theta_{\rho}(\lambda)$.

3 A Scalable algorithm based on optimal dual penalty

In this section we shall describe a scalable algorithm for (9). The basic ingredient of the theoretical development is the estimate by Mandel and Tezaur [5] who proved that under an assumption on regularity of the discretization and boundedness of H/h, there is a lower bound $\alpha > 0$ on the eigenvalues of PFP restricted to the range of P that is independent of h and H, so that for any vector λ

$$\lambda^{\top}PFP\lambda \geq \alpha\|P\lambda\|^{2}. \tag{10}$$

Denoting $\alpha_1 = \min\{\alpha, \rho_0\}$, it follows that for any $\delta, \rho_0 > 0$ and $\rho \geq \rho_0$

$$\delta^\top H^\rho \delta \geq \delta^\top H^{\rho_0} \delta \geq \alpha_1 \|\delta\|^2. \tag{11}$$

Another important ingredient is a recently proposed algorithm for bound constrained quadratic programming called modified proportioning with reduced gradient projections (MPRGP) [2]. The MPRGP algorithm with the choice of parameters $\Gamma = 1$ and $\overline{\alpha} \in (0, \|H^\rho\|^{-1}]$ generates the iterates $\{\lambda^k\}$ for the unique solution $\overline{\lambda}$ of (9) so that the rate of convergence in the energy norm defined by $\|\lambda\|^2_{H^\rho} = \lambda^\top H^\rho \lambda$ is given by

$$\|\lambda^k - \overline{\lambda}\|^2_{H^\rho} \leq \frac{2\eta^k}{\alpha_1} \left(\Theta_\rho(\lambda^0) - \Theta_\rho(\overline{\lambda})\right), \quad \eta = 1 - \frac{\overline{\alpha}\alpha_1}{4}. \tag{12}$$

Theorem 1. *Let C, ρ and ε denote given positive numbers, and let $\{\lambda^i_{H,h}\}$ denote the iterations generated by the MPRPG algorithm with the initial approximation $\lambda^0 = 0$ for the solution $\overline{\lambda}_{H,h}$ of the problem (9) arising from a sufficiently regular discretization of the continous problem with the decomposition, discretization and penalization parameters H, h and ρ . Then there is an integer k independent of h and H such that $H/h \leq C$ implies*

$$\|\lambda^k_{H,h} - \overline{\lambda}_{H,h}\| \leq \varepsilon \|Pd\|. \tag{13}$$

Proof. See [1].

♯

Theorem 1 shows that we can generate efficiently a value of λ that is near to the solution of (9). Its feasibility error is considered in the next theorem.

Theorem 2. *Let C_1 and ρ denote given positive numbers. Then there is a positive constant C such that if $\varepsilon > 0$ and $\lambda_{H,h,\rho}$ denotes an approximate solution of the problem (9) arising from a sufficiently regular discretization of the continuous problem with the decomposition, discretization and penalization parameters H, h and ρ , respectively, $H/h \leq C_1$, $\rho \geq \rho_0$ and $\|\nu(\lambda_{H,h,\rho})\| \leq \varepsilon \|Pd\|$, then*

$$\|G\lambda_{H,h,\rho}\| \leq C \frac{1+\varepsilon}{\sqrt{\rho}} \|Pd\|. \tag{14}$$

Moreover, there is a constant $C_{H,h}$ that depends on H, h such that for any ρ

$$\|G\lambda_{H,h,\rho}\| \leq C_{H,h} \frac{1+\varepsilon}{\rho} \|Pd\|. \tag{15}$$

Proof. See [1].

♯

Theorem 2 shows that a prescribed bound on the relative feasibility error (14) may be achieved with the penalty parameter ρ independent of the discretization parameter h . Thus we have shown that *we can get an approximate solution of the problem (7) with prescribed precision in a number of steps that does not depend on the discretization parameter h* . Let us recall that even though large penalty parameters may destroy conditioning of the Hessian of the Lagrangian, they *need not* slow down the convergence of the conjugate gradient based methods.

4 Contact problems with large displacements

While the FETI method is directly applicable to the solution to contact problems of linearly elastic bodies with small displacements, any other non-linearity necessitates application of additional methods for the solution of nonlinear problems. The nonlinearity we take into account, apart from the contact, is the one caused by large displacements and finite rotations. To this end we use the total Lagrangian formulation which includes all kinematic non-linear effects. As a strain measure we make use of the Green–Lagrange tensor and as the stress measure the second Piola–Kirchhoff tensor which is work–conjugate with the previously mentioned strain tensor.

The Modified Newton-Raphson method was used as a tool for solving these nonlinear problems. Hence, the following algorithm is proposed:

```
Initial step:
    Assembling of stiffness matrix $K$ and matrix of
    continuity conditions between subdomains $B^E$
Step 1
    Assembling of external nodal forces vector $f_{ext}$
    Prescribing conditions of non-interpenetration of bodies
    in current configuration $B^I$, $c^I$ .
Step 2
    Evaluation of internal forces vector $f_{int}$ stemming from
    stresses
Step 3
    FETI solution of contact problem

        $\min \frac{1}{2} u^T K u - u^T f \; s.t. \; B^I u \le c^I \; and \; B^E u = 0$

    where the vector f is the residual between the external forces $f_{ext}$
    and the contact and internal forces $f_{int}$ .
Step 4
    Test of convergence.
    In negative case go to Step 1, otherwise stop.
```

The relative change of nodal displacements can be chosen as a suitable stopping criterion.

5 Hertzian Problem of Contact of Two Cylinders

Consider a classic Hertzian problem, i.e. a frictionless and elastic one, of two cylinders with parallel axes in contact as in Figure 1. The radius of the upper cylinder is

$R_1 = 1000$ mm and that of the lower cylinder is infinite, which means that the lower body is a half-space. The material properties of the two bodies are as follows: Young's modulus $E = 2.0 \times 10^{11}$ Pa and Poisson's ratio $\nu = 0.3$. The load $Q = 400$ MN/m is applied along the axis of the upper cylinder. The problem is two-dimensional from a mathematical point of view, but it was modelled with tri-linear elements as a three-dimensional problem considering bodies of finite length. The boundary conditions are imposed in such a way that they generated a plane strain problem. The complete mesh is shown in Figure 1 as well as its detail along the surfaces potentially in contact.

The analytical solution by McEwen can be found in [4]. The results yielded by both the FETI method in terms of the dual penalty approach and the analytical solution are shown in Figure 2b. It shows the distribution of the normal contact stress along one half of the contact surface of the lower cylinder from the plane of symmetry upwards. It is obvious that the difference between the two solutions is small. Let us notice that the various values of the dual penalty varies from 1e+0 to 1e+4 without significant change of the solution. For comparison Figure 2a depicts solution of the same problem but in terms of the primal penalty method. The problem is not semi-coercive but coercive in this case because the primal penalty method cannot treat problems with sub-domains undergoing the rigid body motions. The penalty method is applied with five different values of the penalty parameters. It can clearly be seen how the quality of solution degrades progressively as the penalty parameter is reduced.

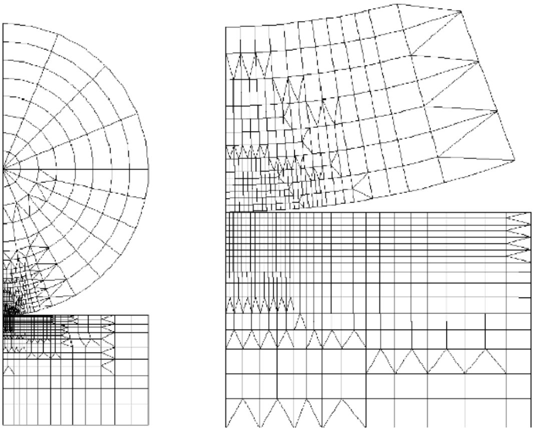


Fig. 1. Hertzian contact problem.

We solve the problem with a load (400 MN/m); the displacements cannot be regarded as small. Therefore we had to iterate in the outer loop of the algorithm in section 5, because of the large displacements. The total load was applied in two steps for better convergence.

Figure 3a depicts the number of conjugate gradients of the inner problem solver needed for convergence at each cycle of the outer loop. Figure 3b demonstrates the

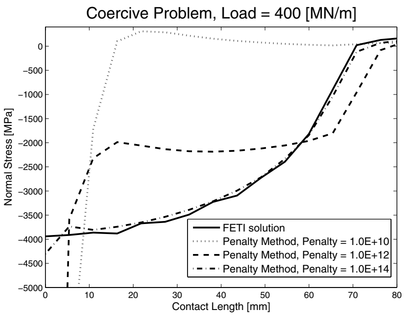


Fig. 2a. Normal contact stresses: Primal penalty method

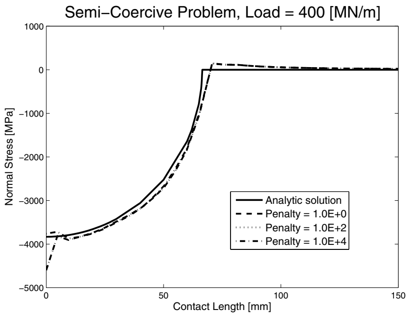


Fig. 2b. Normal contact stresses: Dual penalty (FETI) method.

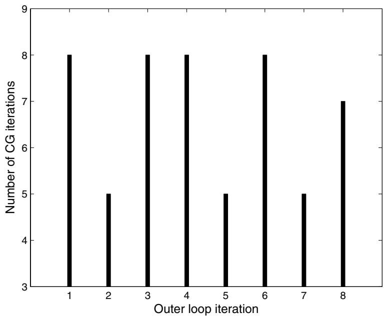


Fig. 3a. Number of iterations of CG.

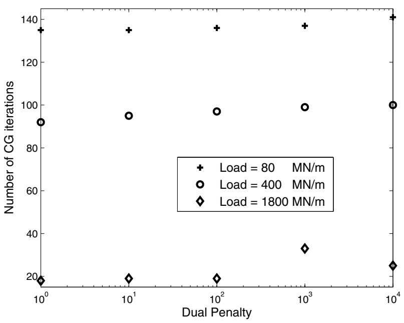


Fig. 3b. Convergence rate.

independence of the number of conjugate gradients for different choices of the value of dual penalty.

Acknowledgement. This research is supported by Grant Agency of the Czech Republic by grant #101/05/0423 and grant of Ministry of Education of the Czech Republic "KONTAKT ME 641".

References

1. Z. Dostál and D. Horák, *Scalable FETI with optimal dual penalty for a variational inequality*, Numer. Linear Algebra Appl., 11 (2004), pp. 455–472.

2. Z. Dostál and J. Schöberl, *Minimizing quadratic functions over non-negative cone with the rate of convergence and finite termination*, Comput. Optim. Appl., 30 (2005), pp. 23–43.
3. C. Farhat, J. Mandel, and F.-X. Roux, *Optimal convergence properties of the FETI domain decomposition method*, Comput. Methods Appl. Mech. Engrg., 115 (1994), pp. 365–385.
4. K. L. Johnson, *Contact Mechanics*, Cambridge University Press, Cambridge, 1985.
5. J. Mandel and R. Tezaur, *Convergence of a substructuring method with Lagrange multipliers*, Numer. Math., 73 (1996), pp. 473–487.
6. P. Wriggers, *Computational contact mechanics*, John Wiley & Sons, Ltd., Chichester, West Sussex, England, 2002.
7. Z.-H. Zhong, *Finite element procedures for contact-impact problems*, Oxford University Press, 1993.

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

Vol. 1 D. Funaro, *Spectral Elements for Transport-Dominated Equations.* 1997. X, 211 pp. Softcover. ISBN 3-540-62649-2

Vol. 2 H. P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 1999. XXIII, 682 pp. Hardcover. ISBN 3-540-65274-4

Vol. 3 W. Hackbusch, G. Wittum (eds.), *Multigrid Methods V.* Proceedings of the Fifth European Multigrid Conference held in Stuttgart, Germany, October 1-4, 1996. 1998. VIII, 334 pp. Softcover. ISBN 3-540-63133-X

Vol. 4 P. Deuflhard, J. Hermans, B. Leimkuhler, A. E. Mark, S. Reich, R. D. Skeel (eds.), *Computational Molecular Dynamics: Challenges, Methods, Ideas.* Proceedings of the 2nd International Symposium on Algorithms for Macromolecular Modelling, Berlin, May 21-24, 1997. 1998. XI, 489 pp. Softcover. ISBN 3-540-63242-5

Vol. 5 D. Kröner, M. Ohlberger, C. Rohde (eds.), *An Introduction to Recent Developments in Theory and Numerics for Conservation Laws.* Proceedings of the International School on Theory and Numerics for Conservation Laws, Freiburg / Littenweiler, October 20-24, 1997. 1998. VII, 285 pp. Softcover. ISBN 3-540-65081-4

Vol. 6 S. Turek, *Efficient Solvers for Incompressible Flow Problems.* An Algorithmic and Computational Approach. 1999. XVII, 352 pp, with CD-ROM. Hardcover. ISBN 3-540-65433-X

Vol. 7 R. von Schwerin, ***M**ulti **B**ody **S**ystem **SIM**ulation.* Numerical Methods, Algorithms, and Software. 1999. XX, 338 pp. Softcover. ISBN 3-540-65662-6

Vol. 8 H.-J. Bungartz, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.* Proceedings of the International FORTWIHR Conference on HPSEC, Munich, March 16-18, 1998. 1999. X, 471 pp. Softcover. ISBN 3-540-65730-4

Vol. 9 T. J. Barth, H. Deconinck (eds.), *High-Order Methods for Computational Physics.* 1999. VII, 582 pp. Hardcover. ISBN 3-540-65893-9

Vol. 10 H. P. Langtangen, A. M. Bruaset, E. Quak (eds.), *Advances in Software Tools for Scientific Computing.* 2000. X, 357 pp. Softcover. ISBN 3-540-66557-9

Vol. 11 B. Cockburn, G. E. Karniadakis, C.-W. Shu (eds.), *Discontinuous Galerkin Methods.* Theory, Computation and Applications. 2000. XI, 470 pp. Hardcover. ISBN 3-540-66787-3

Vol. 12 U. van Rienen, *Numerical Methods in Computational Electrodynamics.* Linear Systems in Practical Applications. 2000. XIII, 375 pp. Softcover. ISBN 3-540-67629-5

Vol. 13 B. Engquist, L. Johnsson, M. Hammill, F. Short (eds.), *Simulation and Visualization on the Grid.* Parallelldatorcentrum Seventh Annual Conference, Stockholm, December 1999, Proceedings. 2000. XIII, 301 pp. Softcover. ISBN 3-540-67264-8

Vol. 14 E. Dick, K. Riemslagh, J. Vierendeels (eds.), *Multigrid Methods VI.* Proceedings of the Sixth European Multigrid Conference Held in Gent, Belgium, September 27-30, 1999. 2000. IX, 293 pp. Softcover. ISBN 3-540-67157-9

Vol. 15 A. Frommer, T. Lippert, B. Medeke, K. Schilling (eds.), *Numerical Challenges in Lattice Quantum Chromodynamics.* Joint Interdisciplinary Workshop of John von Neumann Institute for Computing, Jülich and Institute of Applied Computer Science, Wuppertal University, August 1999. 2000. VIII, 184 pp. Softcover. ISBN 3-540-67732-1

Vol. 16 J. Lang, *Adaptive Multilevel Solution of Nonlinear Parabolic PDE Systems.* Theory, Algorithm, and Applications. 2001. XII, 157 pp. Softcover. ISBN 3-540-67900-6

Vol. 17 B. I. Wohlmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition.* 2001. X, 197 pp. Softcover. ISBN 3-540-41083-X

Vol. 18 U. van Rienen, M. Günther, D. Hecht (eds.), *Scientific Computing in Electrical Engineering.* Proceedings of the 3rd International Workshop, August 20-23, 2000, Warnemünde, Germany. 2001. XII, 428 pp. Softcover. ISBN 3-540-42173-4

Vol. 19 I. Babuška, P. G. Ciarlet, T. Miyoshi (eds.), *Mathematical Modeling and Numerical Simulation in Continuum Mechanics.* Proceedings of the International Symposium on Mathematical Modeling and Numerical Simulation in Continuum Mechanics, September 29 - October 3, 2000, Yamaguchi, Japan. 2002. VIII, 301 pp. Softcover. ISBN 3-540-42399-0

Vol. 20 T. J. Barth, T. Chan, R. Haimes (eds.), *Multiscale and Multiresolution Methods.* Theory and Applications. 2002. X, 389 pp. Softcover. ISBN 3-540-42420-2

Vol. 21 M. Breuer, F. Durst, C. Zenger (eds.), *High Performance Scientific and Engineering Computing.* Proceedings of the 3rd International FORTWIHR Conference on HPSEC, Erlangen, March 12-14, 2001. 2002. XIII, 408 pp. Softcover. ISBN 3-540-42946-8

Vol. 22 K. Urban, *Wavelets in Numerical Simulation.* Problem Adapted Construction and Applications. 2002. XV, 181 pp. Softcover. ISBN 3-540-43055-5

Vol. 23 L. F. Pavarino, A. Toselli (eds.), *Recent Developments in Domain Decomposition Methods.* 2002. XII, 243 pp. Softcover. ISBN 3-540-43413-5

Vol. 24 T. Schlick, H. H. Gan (eds.), *Computational Methods for Macromolecules: Challenges and Applications.* Proceedings of the 3rd International Workshop on Algorithms for Macromolecular Modeling, New York, October 12-14, 2000. 2002. IX, 504 pp. Softcover. ISBN 3-540-43756-8

Vol. 25 T. J. Barth, H. Deconinck (eds.), *Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics.* 2003. VII, 344 pp. Hardcover. ISBN 3-540-43758-4

Vol. 26 M. Griebel, M. A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations.* 2003. IX, 466 pp. Softcover. ISBN 3-540-43891-2

Vol. 27 S. Müller, *Adaptive Multiscale Schemes for Conservation Laws.* 2003. XIV, 181 pp. Softcover. ISBN 3-540-44325-8

Vol. 28 C. Carstensen, S. Funken, W. Hackbusch, R. H. W. Hoppe, P. Monk (eds.), *Computational Electromagnetics.* Proceedings of the GAMM Workshop on "Computational Electromagnetics", Kiel, Germany, January 26-28, 2001. 2003. X, 209 pp. Softcover. ISBN 3-540-44392-4

Vol. 29 M. A. Schweitzer, *A Parallel Multilevel Partition of Unity Method for Elliptic Partial Differential Equations.* 2003. V, 194 pp. Softcover. ISBN 3-540-00351-7

Vol. 30 T. Biegler, O. Ghattas, M. Heinkenschloss, B. van Bloemen Waanders (eds.), *Large-Scale PDE-Constrained Optimization.* 2003. VI, 349 pp. Softcover. ISBN 3-540-05045-0

Vol. 31 M. Ainsworth, P. Davies, D. Duncan, P. Martin, B. Rynne (eds.), *Topics in Computational Wave Propagation.* Direct and Inverse Problems. 2003. VIII, 399 pp. Softcover. ISBN 3-540-00744-X

Vol. 32 H. Emmerich, B. Nestler, M. Schreckenberg (eds.), *Interface and Transport Dynamics.* Computational Modelling. 2003. XV, 432 pp. Hardcover. ISBN 3-540-40367-1

Vol. 33 H. P. Langtangen, A. Tveito (eds.), *Advanced Topics in Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 2003. XIX, 658 pp. Softcover. ISBN 3-540-01438-1

Vol. 34 V. John, *Large Eddy Simulation of Turbulent Incompressible Flows.* Analytical and Numerical Results for a Class of LES Models. 2004. XII, 261 pp. Softcover. ISBN 3-540-40643-3

Vol. 35 E. Bänsch (ed.), *Challenges in Scientific Computing - CISC 2002.* Proceedings of the Conference *Challenges in Scientific Computing*, Berlin, October 2-5, 2002. 2003. VIII, 287 pp. Hardcover. ISBN 3-540-40887-8

Vol. 36 B. N. Khoromskij, G. Wittum, *Numerical Solution of Elliptic Differential Equations by Reduction to the Interface.* 2004. XI, 293 pp. Softcover. ISBN 3-540-20406-7

Vol. 37 A. Iske, *Multiresolution Methods in Scattered Data Modelling.* 2004. XII, 182 pp. Softcover. ISBN 3-540-20479-2

Vol. 38 S.-I. Niculescu, K. Gu (eds.), *Advances in Time-Delay Systems.* 2004. XIV, 446 pp. Softcover. ISBN 3-540-20890-9

Vol. 39 S. Attinger, P. Koumoutsakos (eds.), *Multiscale Modelling and Simulation.* 2004. VIII, 277 pp. Softcover. ISBN 3-540-21180-2

Vol. 40 R. Kornhuber, R. Hoppe, J. Périaux, O. Pironneau, O. Wildlund, J. Xu (eds.), *Domain Decomposition Methods in Science and Engineering.* 2005. XVIII, 690 pp. Softcover. ISBN 3-540-22523-4

Vol. 41 T. Plewa, T. Linde, V.G. Weirs (eds.), *Adaptive Mesh Refinement – Theory and Applications.* 2005. XIV, 552 pp. Softcover. ISBN 3-540-21147-0

Vol. 42 A. Schmidt, K.G. Siebert, *Design of Adaptive Finite Element Software.* The Finite Element Toolbox ALBERTA. 2005. XII, 322 pp. Hardcover. ISBN 3-540-22842-X

Vol. 43 M. Griebel, M.A. Schweitzer (eds.), *Meshfree Methods for Partial Differential Equations II.* 2005. XIII, 303 pp. Softcover. ISBN 3-540-23026-2

Vol. 44 B. Engquist, P. Lötstedt, O. Runborg (eds.), *Multiscale Methods in Science and Engineering.* 2005. XII, 291 pp. Softcover. ISBN 3-540-25335-1

Vol. 45 P. Benner, V. Mehrmann, D.C. Sorensen (eds.), *Dimension Reduction of Large-Scale Systems.* 2005. XII, 402 pp. Softcover. ISBN 3-540-24545-6

Vol. 46 D. Kressner (ed.), *Numerical Methods for General and Structured Eigenvalue Problems.* 2005. XIV, 258 pp. Softcover. ISBN 3-540-24546-4

Vol. 47 A. Boriçi, A. Frommer, B. Joó, A. Kennedy, B. Pendleton (eds.), *QCD and Numerical Analysis III.* 2005. XIII, 201 pp. Softcover. ISBN 3-540-21257-4

Vol. 48 F. Graziani (ed.), *Computational Methods in Transport.* 2006. VIII, 524 pp. Softcover. ISBN 3-540-28122-3

Vol. 49 B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, R. Skeel (eds.), *New Algorithms for Macromolecular Simulation.* 2006. XVI, 376 pp. Softcover. ISBN 3-540-25542-7

Vol. 50 M. Bücker, G. Corliss, P. Hovland, U. Naumann, B. Norris (eds.), *Automatic Differentiation: Applications, Theory, and Implementations.* 2006. XVIII, 362 pp. Softcover. ISBN 3-540-28403-6

Vol. 51 A.M. Bruaset, A. Tveito (eds.), *Numerical Solution of Partial Differential Equations on Parallel Computers* 2006. XII, 482 pp. Softcover. ISBN 3-540-29076-1

Vol. 52 K.H. Hoffmann, A. Meyer (eds.), *Parallel Algorithms and Cluster Computing.* 2006. X, 374 pp. Softcover. ISBN 3-540-33539-0

Vol. 53 H.-J. Bungartz, M. Schäfer (eds.), *Fluid-Structure Interaction.* 2006. VII, 388 pp. Softcover. ISBN 3-540-34595-7

Vol. 54 J. Behrens, *Adaptive Atmospheric Modeling.* 2006. XX, 314 pp. Softcover. ISBN 3-540-33382-7

Vol. 55 O.B. Widlund, D.E. Keyes (eds.), *Domain Decomposition Methods in Science and Engineering XVI.* 2007. XXI, 780 pp. Softcover. ISBN 3-540-34468-3

Monographs in Computational Science and Engineering

Vol. 1 J. Sundnes, G.T. Lines, X. Cai, B.F. Nielsen, K.-A. Mardal, A. Tveito, *Computing the Electrical Activity in the Heart.* 2006. XI, 318 pp. Hardcover. ISBN 3-540-33432-7

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

Vol. 1 H. P. Langtangen, *Computational Partial Differential Equations.* Numerical Methods and Diffpack Programming. 2nd Edition 2003. XXVI, 855 pp. Hardcover. ISBN 3-540-43416-X

Vol. 2 A. Quarteroni, F. Saleri, *Scientific Computing with MATLAB and Octave.* 2nd Edition 2006. XIV, 318 pp. Hardcover. ISBN 3-540-32612-X

Vol. 3 H. P. Langtangen, *Python Scripting for Computational Science*. 2nd Edition 2006. XXIV, 736 pp. Hardcover. ISBN 3-540-29415-5

For further information on these books please have a look at our mathematics catalogue at the following URL: `www.springer.com/series/5151`