Domain Decomposition Methods in Science and Engineering

Fourteenth International Conference on Domain Decomposition Methods

Cocoyoc, Mexico

Edited by: Ismael Herrera
David E. Keyes
Olof B. Widlund
Robert Yates

Published by National Autonomous University of Mexico (UNAM)

Domain Decomposition Methods in Science and Engineering

Fourteenth International Conference on Domain Decomposition Methods

Cocoyoc, Mexico

Domain Decomposition Methods in Science and Engineering

Fourteenth International Conference on Domain Decomposition Methods, Cocoyoc, Mexico

Edited by

Ismael Herrera
Mexico City, Mexico

David E. Keyes
Norfolk, USA

Olof B. Widlund
New York, USA

Robert Yates
Mexico City, Mexico

Published by National Autonomous University of Mexico (UNAM)
Mexico City, Mexico

Domain Decomposition Methods in Science and Engineering
I. Herrera, D. Keyes, O. Widlund, R. Yates (Eds.)

First Edition, June 2003

Printed and bound by Impretei S.A. de C.V., Almería 17, CP 03414, México D.F.

ISBN: 82-994951-1-3

Preface

The annual International Conference on Domain Decomposition Methods for Partial Differential Equations has been a major event in Applied Mathematics and Engineering for the last fifteen years. The proceedings of the Conferences have become a standard reference in the field, publishing seminal papers as well as the latest theoretical results and reports on practical applications.

The Fourteenth International Conference on Domain Decomposition Methods, was hosted by the Universidad Nacional Autónoma de México (UNAM) at Hacienda de Cocoyoc in Morelos, Mexico, January 6-12, 2002. It was organized by Ismael Herrera, Institute of Geophysics, of the National Autonomous University of Mexico (UNAM). He was assisted by a Local Organizing Committee headed by Robert Yates, with the active participation of Gustavo Ayala-Milian, Martin Diaz and Gerardo Zenteno.

This was the sixth of the meetings in this nearly annual conference to be hosted in the Americas, but the first such outside of the United States. It was stimulating and rewarding to have the participation of many practicing scientists and graduate students from Mexico's growing applied mathematics community. Approximately one hundred mathematicians, engineers, physical scientists, and computer scientists from 17 countries spanning five continents participated. This volume captures 52 of the 78 presentations of the Conference.

Since three parallel sessions were employed at the conference in order to accommodate as many presenters as possible, attendees and non-attendees alike may turn to this volume to keep up with the diversity of subject matter that the topical umbrella of "domain decomposition" inspires throughout the community. The interest of so many authors in meeting the editorial demands of this proceedings volume demonstrates that the common thread of domain decomposition continues to justify a regular meeting. "Divide and conquer" may be the most basic of algorithmic paradigms, but theoreticians and practitioners alike continue to seek — and find — incrementally more effective forms, and value the interdisciplinary forum provided by this proceedings series.

Domain decomposition is indeed a basic concept of numerical methods for partial differential equations (PDE's) in general, although this fact is not always recognized explicitly. It is enlightening to interpret many numerical methods for PDE's as domain decomposition procedures and, therefore, the advances in Domain Decomposition Methods are opening new avenues of research in this general area. This is exhibited in this volume. In particular, using a continuous approach an elegant general theory of domain decomposition methods (DDM's) is explained, which incorporates direct and a new class of indirect methods in a single framework. This general theory interprets DDM's as procedures for gathering a target of information, on the internal boundary -'the sought information'-, that is chosen beforehand and is sufficient for defining well-posed local problems in each one of the subdomains of the partition. There are two main procedures for gathering the 'sought information': the *direct method*, which applies local solutions of the original differential equation, and the *indirect method*, which uses local solutions of the adjoint differential equation. Several advantages of the 'indirect method' are exhibited.

Besides inspiring elegant theory, domain decomposition methodology satisfies the architectural imperatives of high-performance computers better than methods operating only on the finest scale of the discretization and over the global data set. These imperatives include: concurrency on the scale of the number of available processors, spatial data locality, temporal data locality, reasonably small communication-to-computation ratios, and reasonably infrequent process synchronization (measured by the number of useful floating-point operations performed between synchronizations). Spatial data locality refers to the proximity of the addresses of successively used elements, and temporal data locality refers to the proximity in time of successive references to a given element.

Spatial and temporal locality are both enhanced when a large computation based on nearest-neighbor updates is processed in contiguous blocks. On cache-based computers, subdomain blocks may be tuned for workingset sizes that reside in cache. On message-passing or cache-coherent nonuniform memory access (cc-NUMA) parallel computers, the concentration of gridpoint-oriented computations — proportional to subdomain volume — between external stencil edge-oriented communications — proportional to subdomain surface area, combined with a synchronization frequency of at most once per volume computation, gives domain decomposition excellent parallel scalability on a per iteration basis, over a range of problem size and concurrency. In view of these important architectural advantages for domain decomposition methods, it is fortunate, indeed, that mathematicians studied the convergence behavior aspects of the subject in advance of the wide availability of these cost-effective architectures, and showed how to endow domain decomposition iterative methods with algorithmic scalability, as well.

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. Since the most complex operation in a Schwarz-type domain decomposition iterative method — the application of the preconditioner — is logically equivalent in each subdomain to a conventional preconditioner applied to the global domain, 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.

The same arguments for reuse of existing serial methods in a parallel environment can be made for Schur-type or substructuring forms of domain decomposition, although in the substructuring case, there are additional types of operations to be performed on interfaces that are absent in the undecomposed original problem. Of course, treatment of the interface problem is where the art continues to undergo development, as the overall convergence depends upon this aspect when the subdomain problems are solved exactly.

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 impor-

tance 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, which can either be further subdivided or aggregated to fit the granularity of an available parallel computer, and have the correct topological and mathematical characteristics for scalability.

The organization of the present proceedings differs from that of previous volumes in that many of the papers are grouped into minisymposia, which provides a finer-grained topical grouping.

These proceedings will be of interest to mathematicians, computer scientists, and computational scientists, so we project its contents onto some relevant classification schemes below.

American Mathematical Society (AMS) 2000 subject classifications (`http://www.ams.org/msc/`) include:

65C20 Numerical simulation, modeling

65F10 Iterative methods for linear systems

65F15 Eigenvalue problems

65M55 Multigrid methods, domain decomposition for IVPs

65N30 Finite elements, Rayleigh-Ritz and Galerkin methods, finite methods

65N35 Spectral, collocation and related methods

65N55 Multigrid methods, domain decomposition for BVPs

65Y05 Parallel computation

68N99 Mathematical software

Association for Computing Machinery (ACM) 1998 subject classifications (`http://www.acm.org/class/1998/`) include:

D2 Programming environments, reusable libraries

F2 Analysis and complexity of numerical algorithms

G1 Numerical linear algebra, optimization, differential equations

G4 Mathematical software, parallel implementations, portability

J2 Applications in physical sciences and engineering

Applications for which domain decomposition methods have been specialized in this proceedings include:

fluids Stokes, Navier-Stokes, multiphase flow, dynamics of arteries, pipes, and rivers

materials phase change, composites

structures linear and nonlinear elasticity, fluid-structure interaction

other electrostatics, obstacle problems

For the convenience of readers coming recently into the subject of domain decomposition methods, a bibliography of previous proceedings is provided below, along with some major recent review articles and related special interest volumes. This list will inevitably be found embarrassingly incomplete. (No attempt has been made to supplement this list with the larger and closely related literature of multigrid and general iterative methods, except for the books by Hackbusch and Saad, which have significant domain decomposition components.)

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. T. F. Chan and T. P. Mathew, *Domain Decomposition Algorithms*, Acta Numerica, 1994, pp. 61-143.

3. 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.

4. 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.

5. T. Chan, T. Kako, H. Kawarada and O. Pironneau, eds., *Proc. Twelfth Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Chiba, 1999), DDM.org, Bergen, 2001.

6. N. Débit, M. Garbey, R. Hoppe, D. Keyes, Y. Kuznetsov and J. Périaux, eds., *Proc. Thirteenth Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Lyon, 2000), CINME, Barcelona, 2002.

7. C. Farhat and F.-X. Roux, *Implicit Parallel Processing in Structural Mechanics*, Computational Mechanics Advances **2**, 1994, pp. 1–124.

8. 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.

9. 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.

10. R. Glowinski, J. Périaux, Z.-C. Shi and O. B. Widlund, eds., *Eighth International Conference of Domain Decomposition Methods* (Beijing, 1995), Wiley, Strasbourg, 1997.

11. W. Hackbusch, *Iterative Methods for Large Sparse Linear Systems*, Springer, Heidelberg, 1993.

12. I. Herrera, R. Yates and M. Diaz, *General Theory of Domain Decomposition: Indirect Methods*, Numerical Methods for Partial Differential Equations, **18(3)**, pp 296-322, 2002.

13. 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.

14. D. E. Keyes, Y. Saad and D. G. Truhlar, eds., *Domain-based Parallelism and Problem Decomposition Methods in Science and Engineering*, SIAM, Philadelphia, 1995.

15. D. E. Keyes and J. Xu, eds. *Proc. Seventh Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (PennState, 1993), AMS, Providence, 1995.

16. C.-H. Lai, P. Bjørstad, M. Cross and O. Widlund, eds., *Proc. Eleventh Int. Conf. on Domain Decomposition Methods for Partial Differential Equations* (Greenwich, 1999), DDM.org, Bergen, 2000.

17. P. Le Tallec, *Domain Decomposition Methods in Computational Mechanics*, Computational Mechanics Advances **2**, 1994, pp. 121–220.

18. J. Mandel, ed., *Proc. Tenth Int. Conf. on Domain Decomposition Methods in Science and Engineering* (Boulder, 1998), AMS, Providence, 1999.

19. L. Pavarino and A. Toselli, *Recent Developments in Domain Decomposition Methods*, Volume 23 of *Lecture Notes in Computational Science & Engineering*, Springer Verlag, Heidelberg, 2002.

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

21. 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.

22. Y. Saad, *Iterative Methods for Sparse Linear Systems*, PWS, Boston, 1996.

23. 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.

24. B. I. Wolmuth, *Discretization Methods and Iterative Solvers Based on Domain Decomposition*, Volume 17 of *Lecture Notes in Computational Science & Engineering*, Springer Verlag, Heidelberg, 2001.

25. J. Xu, *Iterative Methods by Space Decomposition and Subspace Correction*, SIAM Review **34**, 1991, pp. 581-613.

We also mention the homepage for domain decomposition on the World Wide Web, `www.ddm.org`, maintained by Professor Martin Gander of McGill University. This site features links to conference, bibliographic, and personal information pertaining to domain decomposition, internationally.

Previous proceedings of the International Conferences on Domain Decomposition were published by SIAM, AMS, John Wiley and Sons and CIMNE. This time the publisher has been the National University of Mexico (UNAM), with the assistance of Impretei S.A. de C.V.

We wish to thank the members of the International Scientific Committee, and in particular the Chair, Ronald H.W. Hoppe, for their help in setting the scientific direction of the Conference. We are also grateful to the organizers of the mini-symposia for attracting high-quality presentations. The timely production of these Proceedings would not have been possible without the cooperation of the authors and the anonymous referees. We would like to thank them all for their graceful and timely response to our various demands.

The organizers of the Conference would like to acknowledge the sponsors of the Conference, namely UNAM through its Institute of Geophysics, the Instituto Nacional de Tecnología del Agua (IMTA) and the newly created Sociedad Mexicana de Métodos Numéricos en Ingeniería y Ciencia Aplicada (SMMNICA). Thanks are also due to Roland Glowinski and Yuri A. Kuznetsov, for their participation in the American Committee of the Conference, and to Alvaro Aldama, Fabian Garcia-Nocetti, Jaime Urrutia-Fucugauchi, Francisco Sanchez-Bernabe and Carlos Signoret-Poillon, for their participation in the Local Organizing Committee. Finally, we would like to express our appreciation to Ms. Marthita Cerrilla, the Secretary of the Conference, who made all the organizational details run smoothly, together with Martin Diaz and Ernesto Rubio, the Technical Editors of these Proceedings, who finalized the formatting of the papers in LATEX and prepared the whole book for printing.

Ismael Herrera
Mexico City, Mexico

David E. Keyes
Norfolk, USA

Olof B. Widlund
New York, USA

Robert Yates
Mexico City, Mexico

June 2003

Contents

Part I

Invited Plenary Lectures

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

1. Nonlinearity, numerics and propagation of information

A. A. Aldama[1]

1. Introduction. In the study of evolution equations that describe the dynamics of natural and man-made systems, it is always useful to determine the way in which information is propagated by the said equations. In other words, the manner in which different scales present in the solution of an evolution equation travel and decay through space and time. The ideal tool to determine the propagation properties of (continuous or discrete) evolution equations is Fourier or harmonic analysis. In the case of continuous systems, the study of propagation properties allows the understanding of their stability. On the other hand, much insight regarding the behavior of discrete approximations of partial differential equations may be gained by comparing the propagation properties of a continuous equation and its corresponding discrete analogue. Thus, so-called amplitude and phase portraits that respectively depict the ratio of numerical and analytical amplification factor amplitudes and the difference between analytical and numerical phases, both as functions of wavenumber, may be developed (see, for example, Abbot [1] and Vichenevsky and Bowles [17]). These portraits show in a very objective way the effects of "numerical diffusion" and "numerical dispersion" associated to each wave number. Furthermore, the determination of the stability of numerical approximations may be viewed as a by-product of their amplitude propagation properties. Interestingly enough, a similar approach may be applied to study of the convergence properties of iterative schemes for the solution of systems of equations, a fact that has been exploited by the champions of the multigrid approach (see, for instance, [9]). The author and his collaborators have demonstrated the power of Fourier techniques in the study of the propagation properties of non-orthodox approximations of the linear transport equation, via least-squares collocation (Bentley et al., [10]) and the Eulerian-Lagrangian localized adjoint method (Aldama and Arroyo, [6]). Moreover, they have established the existence of an ordinary differential analogy that simplifies the determination of the stability conditions for high order time discretizations of the linear transport equation (Aldama, [3], and Aldama and Aparicio, [5]). Finally, they have studied the convergence properties of a semi-iterative scheme for the solution of a coupled diffusion-reaction system that describes the decay of argon in rocks and minerals (Lee and Aldama, [15]).

Unfortunately, the application of Fourier methods is limited to linear and constant coefficient equations, subject to periodic boundary conditions or to linear and constant coefficient pure initial value problems occurring in infinite spatial domains. The author has developed an approach that allows the use of Fourier techniques in finite spatial domains, variable coefficient or nonlinear problems. Such approach consists of an asymptotic approximation that is constructed by employing Taylor-Fréchet expansions of the differential operators arising in evolution equations, the method of multiple scales and local analysis. Numerical experiments have shown excellent results of the application of the said approach. This paper reviews the general theory on which the approach is based and presents a number of applications made by the author and his

[1]Mexican Institute of Water Technology, Mexican Academy of Engineering and School of Engineering and National Autonomous University of Mexico, aaldama@tlaloc.imta.mx

collaborators that have produced excellent results.

2. Nonlinear evolution problems. Let us consider the following nonlinear evolution problem for the components of the N-dimensional vector $\mathbf{U} = \mathbf{U}(\mathbf{x}, t) \equiv [U_1(\mathbf{x}, t), U_2(\mathbf{x}, t), ..., U_N(\mathbf{x}, t)]^T$, dependent on the three-dimensional position vector $\mathbf{x}$ and time t:

$$\frac{\partial U_i}{\partial t} - N_i\left(U_j\right) \;=\; 0, \;\; \mathbf{x} \in \Omega, \; t > 0; \;\; i = 1, 2, ..., N \tag{2.1}$$

$$B_k(U_j) \;=\; 0, \;\; \mathbf{x} \in \partial\Omega, \;\; t > 0; \;\; k = 1, 2, ..., M \tag{2.2}$$

$$U_i(\mathbf{x}, 0) \;= F_j(\mathbf{x}), \; \mathbf{x} \in \Omega; \;\; i = 1, 2, ..., N \tag{2.3}$$

where (2.1) represents a set of N evolution equations, involving a like number of *spatial* differential operators, $N_i(\cdot)$, acting upon the components of $\mathbf{U}$; Ω is the spatial domain of interest and $\partial\Omega$ its boundary; equation (2.2) represents a set of M boundary conditions involving a like number of differential operators, $B_k(\cdot)$; equation (2.3) represents a set of N initial conditions, where $F_j(\mathbf{x})$ stands for a like number of prescribed functions. The number M is determined by the order of the operators $N_i(\cdot)$ and by the number N.

Examples of evolution equations of the kind represented by equation (2.1) abound. Take, for example, the celebrated Navier-Stokes equations for incompressible flow:

$$\frac{\partial u_i}{\partial t} + u_j \frac{\partial u_i}{\partial x_j} = -\frac{1}{\rho}\frac{\partial p}{\partial x_i} + \nu \frac{\partial^2 u_i}{\partial x_j \partial x_j} \tag{2.4}$$

where u_i $(i = 1, 2, 3)$ are the components of the velocity vector, p is the dynamic pressure, ρ is the density, ν is the kinematic viscosity, t is time, and x_i $(i = 1, 2, 3)$ are the components of the position vector; or the shallow water equations:

$$\begin{array}{l} \frac{\partial h}{\partial t} + \frac{\partial U h}{\partial x} + \frac{\partial V h}{\partial y} = 0 \\ \frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + V\frac{\partial U}{\partial y} - fV = -g\frac{\partial (z_b + h)}{\partial x} + \frac{1}{\rho h}\tau_{bx}(h, U, V) \\ \frac{\partial V}{\partial t} + U\frac{\partial V}{\partial x} + V\frac{\partial V}{\partial y} + fU = -g\frac{\partial (z_b + h)}{\partial y} + \frac{1}{\rho h}\tau_{by}(h, U, V) \end{array} \tag{2.5}$$

where U and V are the components of the velocity vector, h is the depth, ρ is the density, z_b is the bottom elevation, z_{bx} and z_{by} are the x and y components of the bottom shear stress, t is time, and x and y are the components of the position vector; or Richards equation:

$$S(\psi)\frac{\partial \psi}{\partial t} = \frac{\partial}{\partial x_j}\left[K(\psi)\frac{\partial}{\partial x_j}(\psi + z)\right] \tag{2.6}$$

where ψ is the pressure head, S is the specific moisture capacity, K is the unsaturated hydraulic conductivity, z is the vertical coordinate, t is time, and x_i $(i = 1, 2, 3)$ are the components of the position vector; or the two-species advection diffusion reaction system:

$$\begin{array}{l} \frac{\partial C_1}{\partial t} + V\frac{\partial C_1}{\partial x} = D\frac{\partial^2 C_1}{\partial x^2} - K_1(C_1)C_1 + f_1(C_2) \\ \frac{\partial C_2}{\partial t} + V\frac{\partial C_2}{\partial x} = D\frac{\partial^2 C_2}{\partial x^2} - K_2(C_2)C_2 + f_2(C_1) \end{array} \tag{2.7}$$

where C_1 and C_2 are the concentrations of species 1 and 2, V is the advective velocity, D is the diffusion/dispersion coefficient, $K_1(\cdot)$ and $K_2(\cdot)$ are nonlinear decay functions, $f_1(\cdot)$ and $f_2(\cdot)$are nonlinear source/sink functions, t is time, and x is the spatial coordinate.

Evidently, the problem (2.1)-(2.3) is continuous in space and time. Discrete analogues of such a problem may be developed through numerical approximations of the differential operators.

3. Taylor-Fréchet expansions of nonlinear operators. Let us decompose the dependent variable appearing in equation (2.1), U_i, as follows:

$$U_i = \bar{U}_i + u_i \tag{3.1}$$

where $\bar{U}_i$ represents a reference solution of problem (2.1)-(2.3) and u_i is a small perturbation around it, such that

$$\|u_i\| << \left\|\bar{U}_i\right\| \tag{3.2}$$

where $\|\cdot\|$ is a properly defined norm. The assumed nature of $\bar{U}_i$ implies that

$$\frac{\partial \bar{U}_i}{\partial t} - N_i\left(\bar{U}_j\right) \;=\; 0 \tag{3.3}$$

Substituting (3.1) in (2.1) yields:

$$\frac{\partial \bar{U}_i}{\partial t} + \frac{\partial u_i}{\partial t} - N_i\left(\bar{U}_j + u_j\right) \;=\; 0 \tag{3.4}$$

Employing a Taylor-Fréchet expansion (Milne, [16]) of the nonlinear operator that appears as the last term on the left hand side of the last equation results in:

$$N_i(\bar{U}_j + u_j) = N_j(\bar{U}_j) + \partial_{U_k} N_i(\bar{U}_j) \circ u_k + O(\,\|u_k u_k\|\,) \tag{3.5}$$

where $\partial_{U_k} N_i(\bar{U}_j) \circ (\cdot)$ stands for the first partial Fréchet derivative of the nonlinear differential operator $N_i(\cdot)$, which possesses a nonlinear parametric dependence on the reference solution and acts upon the perturbation u_k. It may be shown that first order Fréchet derivatives of nonlinear differential operators are themselves linear differential operators (Milne, [16]). Substituting (3.5) in (3.4) and accounting for (3.3) yields:

$$\frac{\partial u_i}{\partial t} - \partial_{U_k} N_i(\bar{U}_j) \circ u_k + O(\,\|u_k u_k\|\,) = 0 \tag{3.6}$$

As may be observed, to first order in u_i, equation (3.6) (for i=1,2,...,N) is linear, a fact that will be exploited later on.

4. Multiple scale analysis. Let $(\mathbf{x}_0, t_0)$ be a fixed reference point in space and time, with x_{0i} representing the components of $\mathbf{x}_0$. Thus, let us define "slow" space and time variables as follows:

$$\begin{aligned} \mathrm{X}_i &= \tfrac{x_i - x_{io}}{L} \\ \mathrm{T} &= \tfrac{t - t_o}{T} \end{aligned} \tag{4.1}$$

where L and T respectively represent characteristic "large" length and time scales present in U_i. Similarly, let us define "fast" space an time variables as follows:

$$\begin{array}{l} \chi_i = \frac{x_i - x_{io}}{\Lambda_x} \\ \tau = \frac{t - t_o}{\Lambda_t} \end{array} \tag{4.2}$$

where Λ_x and Λ_t respectively represent characteristic "small" length and time scales present in U_i. We will now assume that the following holds true:

$$\varepsilon = \frac{\Lambda_x}{L} = \frac{\Lambda_t}{T} << \ 1 \tag{4.3}$$

Now we are in position of introducing the *separation of scales hypothesis*:

$$\bar{U}_j = \bar{U}_j \left(\mathrm{X}_i, \mathrm{T} \right) \tag{4.4}$$

$$u_j = u_j \left(\chi_i, \tau \right) \tag{4.5}$$

Equations (4.4) and (4.5) express the assumption that the reference solution only depends on the slow variables, whereas the perturbation only depends on the fast variables. Hence the large and small length and time scales take on a more precise meaning. Indeed, L and T respectively represent the length and time scales characteristic of the reference solution, $\bar{U}_i$, and Λ_x and Λ_t respectively represent the length and time scales characteristic of the perturbation, u_i. As will be shown later, the separation of scales hypothesis (4.4)-(4.5) has proven to be valid in a number of specific cases. The reason for this is that it is very often the case that when stability or nonlinear iteration convergence are of interest, it is often the case that the most unstable or the most resistant modes correspond to small scale (i.e., high wavenumber) components of the solution, which through (4.4)-(4.5) may be isolated from a smoothly varying reference solution.

5. Localization. Now let us expand the reference solution, $\bar{U}_i(\mathbf{x}, t)$, and the perturbation, $u_i(\mathbf{x}, t)$, around the reference point $(\mathbf{x}_0, t_0)$, assuming the space and time displacements are of the same order of magnitude as Λ_x and Λ_t:

$$\begin{array}{l} \bar{U}_i(\mathbf{x}, t) = \bar{U}_i(\mathbf{x}_0, t_0) + (x_j - x_{0j}) \left. \frac{\partial \bar{U}_i}{\partial x_j} \right|_{(\mathbf{x_0}, t_0)} + (t - t_0) \left. \frac{\partial \bar{U}_i}{\partial t} \right|_{(\mathbf{x_0}, t_0)} + ... \\ u_i(\mathbf{x}, t) = u_i(\mathbf{x}_0, t_0) + (x_j - x_{0j}) \left. \frac{\partial u_i}{\partial x_j} \right|_{(\mathbf{x_0}, t_0)} + (t - t_0) \left. \frac{\partial u_i}{\partial t} \right|_{(\mathbf{x_0}, t_0)} + ... \end{array} \tag{5.1}$$

where

$$\frac{|\mathbf{x} - \mathbf{x}_0|}{\Lambda_x} = O(1) \tag{5.2}$$

$$\frac{|t - t_0|}{\Lambda_t} = O(1) \tag{5.3}$$

We may now introduce characteristic scales for the magnitudes of $\bar{U}_i$ and u_i:

$$\bar{U}_i = U \bar{U}_i^* \tag{5.4}$$

$$u_i = u u_i^* \tag{5.5}$$

where, say:

$$U = \left\| \bar{U}_i(\mathbf{x}_0, t_0) \right\| \tag{5.6}$$

$$u = \| u_i(\mathbf{x}_0, t_0) \| \tag{5.7}$$

and

$$\bar{U}_i^* = O(1) \tag{5.8}$$

$$u_i^* = O(1) \tag{5.9}$$

and $\bar{U}_i^* = O(1)$, $u_i^* = O(1)$. On account of equation (3.2), we may further assume that:

$$u = \varepsilon U \tag{5.10}$$

Now, from the separation of scales hypothesis (4.4)-(4.5), we get:

$$\begin{array}{l} \frac{\partial \bar{U}_i}{\partial x_j} = \frac{\partial \bar{U}_i}{\partial \mathrm{X}_k} \frac{\partial \mathrm{X}_k}{\partial x_j} = \frac{U}{L} \delta_{jk} \frac{\partial \bar{U}_i^*}{\partial \mathrm{X}_k} = \frac{U}{L} \frac{\partial \bar{U}_i^*}{\partial \mathrm{X}_j} \\ \frac{\partial \bar{U}_i}{\partial t} = \frac{\partial \bar{U}_i}{\partial \mathrm{T}} \frac{\partial \mathrm{T}}{\partial t} = \frac{U}{T} \frac{\partial \bar{U}_j^*}{\partial \mathrm{T}} \end{array} \tag{5.11}$$

$$\begin{array}{l} \frac{\partial u_i}{\partial x_j} = \frac{\partial u_i}{\partial \chi_k} \frac{\partial \chi_k}{\partial x_j} = \frac{u}{\Lambda_x} \delta_{jk} \frac{\partial u_i^*}{\partial \chi_k} = \frac{u}{\Lambda_x} \frac{\partial u_i^*}{\partial \chi_j} \\ \frac{\partial u_i}{\partial t} = \frac{\partial u_i}{\partial \tau} \frac{\partial \tau}{\partial t} = \frac{u}{\Lambda_t} \frac{\partial u_i^*}{\partial \tau} \end{array} \tag{5.12}$$

where (4.1), (4.2), (5.4) and (5.5) have been used. Employing now (5.2), (5.3), (5.8), (5.9), (5.11), and (5.12) in (5.1) it is readily shown that:

$$\bar{U}_i(\mathbf{x}, t) = \bar{U}_i(\mathbf{x}_0, t_0) \left[1 + O(\varepsilon) \right] \equiv U \left[1 + O(\varepsilon) \right] \tag{5.13}$$

$$u_i(\mathbf{x}, t) = u_i(\mathbf{x}_0, t_0) \left[1 + O(1) \right] \tag{5.14}$$

Equation (5.13) shows that whereas the reference solution, $\bar{U}_i$, may be localized in the neighborhood of the reference point $(\mathbf{x}_0, t_0)$ at space and time displacements commensurate with the small scales Λ_x and Λ_t, the perturbation, u_i, may not. In other words, an observer sensitive to the scales Λ_x and Λ_t, would only perceive the variations in the perturbation, and would view the reference solution as a constant.

6. Asymptotics. We now may seek an asymptotic solution to equation (3.6), of the form:

$$u_i = u_i^{(0)} + \varepsilon u_i^{(1)} + \varepsilon^2 u_i^{(2)} + ... \equiv \varepsilon U \left[u_i^{(0)*} + \varepsilon u_i^{(1)*} + \varepsilon^2 u_i^{(2)*} + ... \right] \tag{6.1}$$

where $u_i^{(k)*}$ (k=0,1,2,...) are dimensionless and of $O(1)$, and (5.5) and (5.10) have been accounted for. Substituting (5.13) and (6.1) in (3.6) we get the following evolution system for the zeroth order approximation $u_i^{(0)}$ (i=1,2,...,N):

$$\frac{\partial u_i^{(0)}}{\partial t} - \partial_{U_k} N_i(U_j) \circ u_i^{(0)} = 0 \; ; \;\; j = 1, 2, ..., N \tag{6.2}$$

It must be noted that equation (6.2) is linear and with *constant coefficients* that parametrically depend (alas, nonlinearly) on the constants U_j (j=1,2,...,N). Thus, equation (6.2) captures the dominant nonlinear behavior of equation (2.1) in the scales of Λ_x and Λ_t. Furthermore, the previously presented localization analysis was based on the assumption that:

$$|\chi_i| = |x_i - x_{io}| \, / \Lambda_x = |x_i - x_{io}| \, / \, (\varepsilon \, L) = O(1) \tag{6.3}$$

Therefore, as $\varepsilon \downarrow 0$, the domain corresponding to the zeroth order approximation $u_i^{(0)}$ (i=1,2,...,N) becomes unbounded.

7. Fourier analysis. In view of the above, the most general form of equation (6.2) may be written as follows in three-dimensional space:

$$\partial_t u_j^{(0)} = \sum_{r=1}^{N} \sum_{\mathbf{p} \in P} \alpha_{jr,\mathbf{p}} \partial_{\mathbf{p}} u_r^{(0)} \; ; \; j = 1, 2, ..., N; \text{ in } \mathbf{x} \in \Omega_\infty \tag{7.1}$$

where $\Omega_\infty \equiv (-\infty, \infty)^3$; $\partial_t \equiv \partial / \partial t$; $\mathbf{p} \equiv (p_1, p_2, p_3)$ represents a multi-index; $P \equiv \{(p_1, p_2, p_3 \, | 0 \leq p_1 + p_2 + p_3 \leq R\}$, where R is the maximum order of the spatial derivatives present in (7.1); $\alpha_{jr,\mathbf{p}}$ are constant coefficients; $\partial_{\mathbf{p}}(\cdot) \equiv \frac{\partial^{p_1+p_2+p_3}(\cdot)}{\partial x_1^{p_1} \partial x_2^{p_2} \partial x_3^{p_3}}$, and the summation convention is understood in $\mathbf{p}$.

Now, assuming the functions prescribed in the initial conditions (2.3) are of the form

$$F_j = \bar{F}_j + f_j, \;\; f_j / \bar{F}_j = O(\varepsilon) \; ; \;\; j = 1, 2, ..., N \tag{7.2}$$

it is consistent to write that the initial conditions that equation (7.1) is subject to, are:

$$u_j^{(0)} = f_j; \;\; j = 1, 2, ..., N \tag{7.3}$$

Equations (7.1) and (7.3) constitute a *pure initial value problem*, that may be tackled via Fourier methods. With that purpose in mind, the following Fourier representation may be used (Champeney, [12]):

$$u_j^{(0)}(\mathbf{x}, t) = \frac{1}{(2\pi)^{3/2}} \int\limits_{\Omega_\infty} \hat{u}_j^{(0)}(\mathbf{k}, t) \exp(-\mathrm{i} \, \mathbf{k} \cdot \mathbf{x}) d\mathbf{k} \tag{7.4}$$

where $\mathrm{i} \equiv \sqrt{-1}$, $\mathbf{k} \equiv (k_1, k_2, k_3)$ is the wavenumber vector, $d\mathbf{k} \equiv dk_1 dk_2 dk_3$, and the Fourier coefficients $\hat{u}_j^{(0)}$ are given by the following Fourier transforms:

$$\hat{u}_j^{(0)}(\mathbf{k}, t) \equiv \Im\left\{u_j^{(0)}(\mathbf{x}, t)\right\} = \frac{1}{(2\pi)^{3/2}} \int\limits_{\Omega_\infty} u_j^{(0)}(\mathbf{x}, t) \exp(\mathrm{i}\,\mathbf{k} \cdot \mathbf{x}) d\mathbf{x} \tag{7.5}$$

Now, it may be shown that the Fourier coefficients $\hat{u}_j^{(0)}$ may be determined by employing the initial conditions (7.3). Nevertheless, when the propagation properties of the equation (7.1) and, in particular, its stability are of interest, the initial values of $u_j^{(0)}$ are inconsequential. In effect, the stability of equation (7.1) is determined by finding whether $\hat{u}_j^{(0)}(\mathbf{k}, t)$ grows or decays in time.

8. Discrete systems. An analysis similar to that presented earlier may be performed for discrete systems, that may correspond to numerical approximations of partial differential evolution equations, such as equation (2.1). In such a case, the only additional aspect of the analysis that must be considered is the determination of the *modified partial differential equations* that are satisfied when the discrete equations in terms of the perturbation quantities are solved. This consideration allows a local analysis such as the one presented for the continuous case. In addition, instead of using a continuous Fourier pair, like (7.4)-(7.5), a semidiscrete one must be used (i.e., an integral representation for the physical space variables and a Fourier series representation for the wavenumber space variables). Examples of the use of such a technique follow.

9. The one-dimensional Richards equation. Let us consider the one - dimensional analogue of equation (2.6):

$$S(\psi)\frac{\partial \psi}{\partial t} = \frac{\partial}{\partial z}\left[K(\psi)\frac{\partial(\psi)}{\partial z}\right] + \frac{\partial K(\psi)}{\partial z} \tag{9.1}$$

The θ-central difference or θ-lumped finite element (with constant element size) approximation of equation (9.1) is:

$$\begin{aligned} &F(\psi_j^n) \equiv \overline{\theta}(S_j)\tfrac{\delta^{n+\frac{1}{2}}\psi_j}{\Delta t} - \overline{\theta}\left\{\tfrac{K_{j+\frac{1}{2}}\delta_{j+\frac{1}{2}}\psi - K_{j-\frac{1}{2}}\delta_{j-\frac{1}{2}}\psi}{\Delta z^2} + \tfrac{(\delta_{j+\frac{1}{2}} - \delta_{j-\frac{1}{2}})K}{2\Delta z}\right\} \\ &= 0 \end{aligned} \tag{9.2}$$

where $\bar{\theta}(\phi) = \theta(\phi^{n+1}) + (1-\theta)(\phi^n)$, $\delta^{n+\frac{1}{2}}\phi = \phi^{n+1} - \phi^n$, $\delta_{j+\frac{1}{2}}\phi = \phi_{j+1} - \phi_j$ and the usual notation for discrete approximations in space and time is employed.

Now, since Richards' equation is a nonlinear diffusion (i.e., parabolic) equation, a simple frozen coefficient analysis yields unconditional stability for the Crank-Nicolson scheme ($\theta = 1/2$). This result is contradicted by computational evidence, which shows that the said scheme often becomes unstable. This led the author to believe that the explanation for the emergence of instabilities should lie on nonlinear effects. Thus, it is apparent that the theory presented herein may be of use.

The solution of equation (9.2) may be decomposed as follows:

$$\psi_j^n = \tilde{\psi}_j^n + \varepsilon_j^n \tag{9.3}$$

where $\tilde{\psi}_j^n$ is the exact solution of equation (9.2) and ε_j^n a roundoff error. Substituting (9.3) in (9.2), employing a Taylor-Fréchet expansion and localizing the result yields the following equation for the roundoff error:

$$\begin{aligned} &S(\tilde{\psi}_0)\tfrac{\delta^{n+\frac{1}{2}}\varepsilon_j}{\Delta t} - K'(\tilde{\psi}_0)\left[2\left(\tfrac{\partial\tilde{\psi}}{\partial z}\right)_0 + 1\right]\overline{\theta}\left[\tfrac{(\delta_{j+\frac{1}{2}}+\delta_{j+\frac{1}{2}})\varepsilon}{2\Delta z}\right] = K(\tilde{\psi}_0)\times \\ &\times\overline{\theta}\left[\tfrac{(\delta_{j+\frac{1}{2}}-\delta_{j-\frac{1}{2}})\varepsilon}{\Delta z^2}\right] + K''(\tilde{\psi}_0)\left(\tfrac{\partial\tilde{\psi}}{\partial z}\right)_0\left[2\left(\tfrac{\partial\tilde{\psi}}{\partial z}\right)_0 + 1\right]\overline{\theta}\left[\tfrac{\varepsilon_{j+1}^{n+1}+\varepsilon_{j-1}^{n+1}}{2}\right] + \\ &+K'(\tilde{\psi}_0)\left(\tfrac{\partial^2\tilde{\psi}}{\partial z^2}\right)_0\overline{\theta}\left[\tfrac{\varepsilon_{j+\frac{1}{2}}+\varepsilon_{j-\frac{1}{2}}}{2}\right] - S'(\tilde{\psi}_0)\left(\tfrac{\partial\tilde{\psi}}{\partial z}\right)_0\bar{\theta}(\varepsilon_j) \end{aligned} \tag{9.4}$$

Since equation (9.4) is linear and with constant coefficients, without the loss of generality, the behavior of a single (but arbitrary) Fourier mode may be studied. Thus let us employ the following Fourier representation:

$$\varepsilon_j^n = E_k\xi_k^n\exp(\mathrm{i}j\beta_k) \tag{9.5}$$

where E_k is the amplitude associated with the wavenumber k, ξ_k is the corresponding amplification factor and $\beta_k \equiv k\Delta x$ is a dimensionless wavenumber. Substituting (9.5) in (9.4) results in:

$$\xi_k = \frac{1+(1-\theta)\mu_k}{1-\theta\mu_k} \tag{9.6}$$

where $\mu_k = (\mu_k)_R + \mathrm{i}(\mu_k)_R$ and

$$\begin{aligned} (\mu_k)_R &= \left\{\tfrac{K''(\tilde{\psi}_0)}{S(\tilde{\psi}_0)}\left(\tfrac{\partial\tilde{\psi}}{\partial z}\right)_0\left[\left(\tfrac{\partial\tilde{\psi}}{\partial z}\right)_0 + 1\right]\cos\beta_k + \tfrac{1}{2}\tfrac{K'(\tilde{\psi}_0)}{S(\tilde{\psi}_0)}\left(\tfrac{\partial^2\tilde{\psi}}{\partial z^2}\right)_0\times\right. \\ &\left.\times(1+\cos\beta_k) - \tfrac{S'(\tilde{\psi}_0)}{S(\tilde{\psi}_0)}\left(\tfrac{\partial\tilde{\psi}}{\partial z}\right)_0 - \tfrac{2}{\Delta z^2}\,\tfrac{K(\tilde{\psi}_0)}{S(\tilde{\psi}_0)}(1-\cos\beta_k)\right\}\Delta t \\ (\mu_k)_I &= \tfrac{K'(\tilde{\psi}_0)}{S(\tilde{\psi}_0)}\left[2\left(\tfrac{\partial\tilde{\psi}_0}{\partial z}\right)_0 + 1\right]\sin\beta_k\tfrac{\Delta t}{\Delta z} \end{aligned} \tag{9.7}$$

The stability condition for Crank-Nicolson scheme $\theta = 1/2$is $(\mu_k)_R \leq 0$, $\forall k$. Aldama and Aparicio ([5]) have shown that this condition is often violated in the numerical solution of Richards' equation. This explains the computational evidence that indicates that the Crank-Nicolson scheme becomes unstable in the solution of Richards' equation.

Since Richards equation (9.1) is nonlinear, its discrete analogue (9.2) generates an algebraic system of equations that is nonlinear as well. Thus, equation (9.2) must be solved in practice via an iterative scheme. The Picard or successive approximation iterative scheme for equation (9.2) may be written as follows:

$$\begin{array}{l}\left[\theta\, S_j^{n+1,m} + (1-\theta)\, S_j^n\right] \frac{\psi_j^{n+1,m+1} - \psi_j^n}{\Delta t} \\ -\theta \left\{ \frac{1}{2\Delta z^2} \left[\left(K_{j+1}^{n+1,m} + K_j^{n+1,m} \right) \left(\psi_{j+1}^{n+1,m+1} - \psi_j^{n+1,m+1} \right) \right.\right. \\ \left. - \left(K_j^{n+1,m} + K_{j-1}^{n+1,m} \right) \left(\psi_j^{n+1,m+1} - \psi_{j-1}^{n+1,m+1} \right) \right] \\ \left. + \frac{K_{j+1}^{n+1,m} - K_{j-1}^{n+1,m}}{2\Delta z} \right\} - (1-\theta) \left\{ \frac{1}{2\Delta z^2} \left[\left(K_{j+1}^n + K_j^n \right) \left(\psi_{j+1}^n - \psi_j^n \right) \right.\right. \\ \left. \left. - \left(K_j^n + K_{j-1}^n \right) \left(\psi_j^n - \psi_{j-1}^n \right) \right] + \frac{K_{j+1}^n - K_{j-1}^n}{2\Delta z} \right\} = 0 \end{array} \tag{9.8}$$

where the superindex m refers to iteration number. Now, a frozen coefficients analysis predicts unconditional convergence for scheme (9.8). This is not consistent with the observations of Huyarkon et al ([13]) and Celia et al ([11]), who have reported that the Picard scheme (9.8) sometimes diverges. In particular, it has been observed that it converges for small values of the time step, Δt, diverges for intermediate values and converges again for large values. This behavior would not be expected were the equation under study a linear one and, thus, may be attributed to nonlinearity.

In order to properly characterize the behavior of the Picard scheme applied to the solution of the discrete Richards equation, the theory presented in this paper may be applied. With that purpose in mind, let us express the $(m+1)$th iterate in equation (9.8) as follows:

$$\psi_j^{n+1,m+1} = \tilde{\psi}_j^{n+1} + \delta_j^{m+1} \tag{9.9}$$

where, as before, $\tilde{\psi}_j^{n+1}$ represents the exact solution of equation (9.2) and δ_j^{m+1}, the error corresponding to iteration $m+1$. Substituting (9.9) in equation (9.8), performing a Taylor-Fréchet expansion and localizing the result yields:

$$\begin{array}{l} S\left(\psi_0\right) \frac{\delta_j^{n+1}}{\Delta t} - \theta K'\left(\psi_0\right) \left(\frac{\partial \psi}{\partial z} \right)_0 \left(\frac{\delta_{j+1}^{m+1} - \delta_{j-1}^{m+1}}{2\Delta z} + \frac{\delta_{j+1}^m - \delta_{j-1}^m}{2\Delta z} \right) \\ -\theta K'\left(\psi_0\right) \frac{\delta_{j+1}^m - \delta_{j-1}^m}{2\Delta z} \theta K\left(\psi_0\right) \frac{\delta_{j+1}^{m+1} - \delta_j^{m+1} + \delta_{j-1}^{m+1}}{\Delta z^2} \\ +K''\left(\psi_0\right) \left(\frac{\partial \psi}{\partial z} \right)_0 \left[\left(\frac{\partial \psi}{\partial z} \right)_0 + 1 \right] + \frac{\delta_{j+1}^m + \delta_{j-1}^m}{2} \\ +K'\left(\psi_0\right) \left(\frac{\partial^2 \psi}{\partial z^2} \right)_0 \frac{\delta_{j+1}^m + 2\delta_j^m + \delta_{j-1}^m}{4} - +\theta S'\left(\psi_0\right) \left(\frac{\partial \psi}{\partial z} \right)_0 \left(\delta_j^{m+1} - \delta_j^m \right) \end{array} \tag{9.10}$$

Let us now study the behavior of a single (but arbitrary) Fourier mode in the solution of equation (9.10), by employing the following representation for the iteration error:

$$\delta_j^m = \Delta_k \xi_k^m \exp(\mathrm{i} j \beta_k) \tag{9.11}$$

where Δ_k is the amplitude associated with the wavenumber k, ξ_k is the corresponding amplification factor and $\beta_k \equiv k\Delta x$ is a dimensionless wavenumber. Substituting (9.5) in (9.4) results in:

$$\xi_k = \frac{\mu_{2,k}}{1 + \mu_{1,k}} \tag{9.12}$$

where $\mu_{1,k} = \mu_{1R,k} + \mathrm{i}\mu_{1I,k}$, $\mu_{2,k} = \mu_{2R,k} + \mathrm{i}\mu_{2I,k}$ and:

$$\begin{aligned}
&\mu_{1R,k} = 2\theta\tfrac{K(\psi_0)}{S(\psi_0)}\left(1-\cos\beta_k\right)\tfrac{\Delta t}{\Delta z^2} + \theta\tfrac{S'(\psi_0)}{S(\psi_0)}\left(\tfrac{\partial\psi}{\partial t}\right)_0\Delta t \\
&\mu_{1I,k} = -\theta\tfrac{K'(\psi_0)}{S(\psi_0)}\left(\tfrac{\partial\psi}{\partial z}\right)_0\tfrac{\Delta t}{\Delta z^2}\sin\beta_k\Delta t \\
&\mu_{2R,k} = \theta\tfrac{K''(\psi_0)}{S(\psi_0)}\left(\tfrac{\partial\psi}{\partial z}\right)_0\left[\left(\tfrac{\partial\psi}{\partial z}\right)_0 + 1\right]\cos\beta_k\Delta t + \tfrac{1}{2}\theta\tfrac{K'(\psi_0)}{S(\psi_0)} \\
&\times\left(+1\cos\beta_k\right)\Delta t\left(\tfrac{\partial^2\psi}{\partial z^2}\right)_0 - \theta\tfrac{S'(\psi_0)}{S(\psi_0)}\left(\tfrac{\partial\psi}{\partial t}\right)_0\Delta t \\
&\mu_{2I,k} = -\mu_{1I,k}\left[1+\left(\tfrac{\partial\psi}{\partial z}\right)_0^{-1}\right]
\end{aligned} \tag{9.13}$$

The convergence condition for the Picard iterative scheme may be written as follows:

$$|\,\xi_k| \;< 1 \quad \forall\, k \tag{9.14}$$

It may be shown that the above inequality leads to a quadratic inequality in Δt, which explains the observation that Picard iterations are sometimes convergent for "small" values of Δt, divergent for "intermediate" values, and convergent again for "large" values. Numerical experiments performed by Aldama and Paniconi ([8]) have validated such theoretical considerations.

10. The Saint-Venant equations. Another nonlinear evolution system that commonly arises in applications is the one constituted by the Saint-Venant equations that govern nonuniform, transient open channel flow:

$$\frac{\partial A}{\partial t} + \frac{\partial Q}{\partial x} = 0 \tag{10.1}$$

$$\frac{\partial Q}{\partial t} + \frac{\partial}{\partial x}\left(\frac{Q^2}{A}\right) + gA\frac{\partial h}{\partial x} + gA\frac{\partial z}{\partial x} + gS_f = 0 \tag{10.2}$$

where equation (10.1) expresses the conservation of mass principle and equation (10.2), the momentum principle. There, A represents the hydraulic area; Q, the discharge; h, the depth; z, the bottom elevation; S_f, the frictional slope; g, the acceleration of gravity; x, the spatial coordinate along the channel, and t, time. When Manning's formula is employed, the frictional slope may be expressed as follows:

$$S_f = \alpha\left(\frac{k_s}{R}\right)^{1/3}\frac{Q\,|Q|}{A^2R} \tag{10.3}$$

where $\alpha \cong 17/100$ (Aldama and Ocón, [7]); k_s is Nikuradse's equivalent roughness and R is the hydraulic radius.

The so-called generalized Preismann scheme ([1]) for the numerical solution of the Saint-Venant system (10.1)-(10.2) may be written as follows:

$$\frac{A_{j+1}^{n+1} - A_{j+1}^{n}}{\Delta t} + (1-\theta)\,\frac{Q_{j+1}^{n} - Q_{j}^{n}}{\Delta x} + \theta\frac{Q_{j+1}^{n+1} - Q_{j}^{n+1}}{\Delta x} = 0 \tag{10.4}$$

$$
\begin{array}{l}
(1-\psi)\,\frac{Q_j^{n+1}-Q_j^n}{\Delta t}+\psi\frac{Q_{j+1}^{n+1}-Q_{j+1}^n}{\Delta t}+(1-\theta)\,\frac{\left(\frac{Q^2}{A}\right)_{j+1}^{n}-\left(\frac{Q^2}{A}\right)_{j}^{n}}{\Delta x}+\theta\frac{\left(\frac{Q^2}{A}\right)_{j+1}^{n+1}-\left(\frac{Q^2}{A}\right)_{j}^{n+1}}{\Delta x}+ \\
\quad +g\left\{(1-\theta)\,\left[(1-\psi)\,A_j^n+\psi A_{j+1}^n\right]+\theta\left[(1-\psi)\,A_j^{n+1}+\psi A_{j+1}^{n+1}\right]\,\right\} \\
\left[(1-\theta)\,\frac{h_{j+1}^n-h_j^n}{\Delta x}+\theta\frac{h_{j+1}^{n+1}-h_j^{n+1}}{\Delta x}+\quad\frac{z_{j+1}-z_j}{\Delta x}\right]+(1-\theta)\left[(1-\psi)\,A_j^nS_{fj}^n+\psi A_{j+1}^nS_{fj+1}^n\right] \\
+\theta\left[(1-\psi)\,A_j^{n+1}S_{fj}^{n+1}+\psi A_{j+1}^{n+1}S_{fj+1}^{n+1}\right]=0
\end{array}
\tag{10.5}
$$

where $\psi \in [0\,,\,1]$ is a space weighting factor and $\theta \in [0\,,\,1]$ is a time weighting factor.

By applying the theory presented herein, it may be shown that the stability conditions for the generalized Preismann scheme (10.4)-(10.5) are:

$$
|\,V_e\,| \;\leq\; 1, \quad \psi \;=\; 0.5, \quad \theta \;\geq\; 0.5 \tag{10.6}
$$

where V_e is the Vedernikov number. The validity of the conditions (10.6) has been assessed via numerical experimentation (Aguilar, [2]).

11. The shallow water equations. The one-dimensional version of the shallow water equations may be written as follows:

$$
\begin{array}{l}
M_a(h,U) \equiv \frac{\partial h}{\partial t}+\frac{\partial Uh}{\partial x}=0 \\
M_0(h,U) \equiv \frac{\partial U}{\partial t}+U\frac{\partial U}{\partial x}+g\frac{\partial (z_b+h)}{\partial x}+gS_f=0
\end{array}
\tag{11.1}
$$

where $M_a(\cdot,\cdot)$ is the mass conservation operator and $M_o(\cdot,\cdot)$ is the momentum operator. The Generalized Wave Continuity Equation (GWCE) formulation was introduced in order to eliminate the spurious oscillations that arise in the numerical solution of the shallow water equations, in their primitive formulation (11.1), when collocated grids are used (see, for example Kinmark, [14]). The GWCE formulation introduces the following equation, which is derived from (11.1):

$$
W(h,U) \equiv \frac{\partial M_a(h,U)}{\partial t}-\frac{\partial M_o(h,U)}{\partial x}+GM_a(h,U)=0 \tag{11.2}
$$

where $W(\cdot,\cdot)$ is the so-called GWCE operator. The GWCE formulation consists of solving the coupled equations and $M_o(h,U)=0$. As is apparent, when $G \to \infty$, the GWCE formulation approaches the primitive formulation, and when $G \to 0$, the equation $W(h,U)=0$ approaches a nonlinear wave equation.

A number of investigators have become concerned with the fact that, apparently, the GWCE formulation does not possess good mass conservation properties (see Aldama et al., [4], for details). It may be shown, by applying the theory presented in this paper that such formulation does not satisfies the continuity equation and that the error is larger for high wavenumbers. This theoretical result is consistent with observations that indicate that relatively large mass conservation errors arise in refined grids.

12. Conclusions. A theory that consists of the Taylor-Fréchet expansion of nonlinear operators, multiple scale analysis, localization and asymptotic analysis has been presented in order to include dominant nonlinear effects in the study of the propagation properties (stability, amplitude and phase portraits, nonlinear iteration

convergence) of nonlinear evolution systems. The theory presented has been tested via a number of applications, a few of which are presented in this paper, with excellent results.

REFERENCES

[1] M. B. Abbot. *Computational hydraulics. Elements of the theory of free surface flows*. Pitman. London, 1979.

[2] A. Aguilar. *Propagation properties of numerical schemes for free flow simulation*. PhD thesis, UNAM. Mexico, 2002.

[3] A. Aldama. Stability analysis of discrete approximations of the advection diffusion equation through the use of an ordinary differential equation analogy. *Developments in Water Science*, (5):3–8, 1988.

[4] A. Aldama, A. Aguilar, J. Westerink, and R. Kolar. A mass conservation analysis of the GWCE formulation. In B. et al, editor, *XIII International Conference on Computational Methods in Water Resources*, pages 597–601 907–912. Balkema. Rotterdam, 2000.

[5] A. Aldama and J. Aparicio. The effect of nonlinearities in the stability of numerical solutions of Richards' equation. In B. et al, editor, *XII International Conference on Computational Methods in Water Resources. Volume I*, pages 289–296. Computational Mechanics Publications. Southampton, 1998.

[6] A. Aldama and V. Arroyo. Propagation properties of Eulerian Lagrangian Localized Adjoint Methods. In B. et al, editor, *XIII International Conference on Computational Methods in Water Resources*, pages 597–601. Vol 2. Balkema. Rotterdam, 2000.

[7] A. Aldama and A. Ocon. Flow resistance in open channels and Manning's formula limits of applicability (in spanish). *Ingenieria Hidraulica en Mexico*, XVII:107–115, Jan–Mar 2002.

[8] A. Aldama and C. Paniconi. An analysis of the convergence of picard iterations for implicit approximations of Richards' equation. In R. et al, editor, *IX International Conference on Computational Methods in Water Resources*, pages 521–528. Computational Mechanics Publications and Elsevier. Southampton and London., 1992.

[9] J. Aparicio and A. Aldama. On the efficient determination of stability properties for higher order approximations of the transport equation. In A. et al, editor, *XI International Conference on Computational Methods in Water Resources*, pages 29–36. Computational Mechanics Publications. Southampton, 1996.

[10] L. Bentley, A. Aldama, and G. Pinder. Fourier analysis of the Eulerian-Lagrangian-least-squares-collocation method. *International Journal for Numerical Methods in Fluids*, (11):427–444, 1990.

[11] M. Celia, E. T. Bouloutas, and R. L. Zarba. A general mass-conservative numerical solution for the unsaturated flow equation. *Water Resources Research*, (23):1483–1496, 1990.

[12] D. C. Champeney. *A Handbook of Fourier Theorems*. Cambridge University Press, 1989.

[13] P. Huyakon, S. Thomas, and B. Thompson. Techniques for making finite elements competitive in modelling flow in variably saturated porous media. *Water Resources Research*, (20):1099–1115, 1984.

[14] I. Kinmark. *The Shallow Water Wave Equations: Formulation, Analysis and Application*. Springer-Verlag. Berlin-Heidelberg, 1986.

[15] J. Lee and A. Aldama. Multipath diffusion: A general numerical model. *Computers and Geosciences*, (18):531–555, 1992.

[16] R. D. Milne. *Applied Functional Analysis*. Pitman. London, 1980.

[17] R. Vichnevetsky and J. B. Bowles. *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*. SIAM, Philadelphia, 1982.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

2. Non conforming domain decomposition: the Steklov-Poincaré operator point of view

S. Bertoluzza[1]

1. Introduction. One of the common approaches to solve the linear system arising in the domain decomposition method is to formally reduce it, by a Schur complement argument, to a lower dimensional linear system whose unknown is the value of the (discrete) solution on the interface of the decomposition. Solving such reduced linear system by any iterative technique implies the need of solving, at each iteration, independent discrete Dirichlet problems in the subdomains. Such Dirichlet problems constitute the most relevant part of the computational cost of such an approach and therefore attention needs to be paid in reducing the actual computational cost of the subdomain solvers. A key observation in this respect is that what one expects as an output of the iterative procedure is a (correct order) approximation of the trace of the solution u on the interface. There is no direct need of solving correctly the Dirichlet problems in the subdomains. The precision with which such problems are solved is only as relevant as its influence on the error on the trace of u on the interface. Only once the trace of u on the interface has been computed correctly, one will actually need to retrieve the solution in some or all of the subdomains.

In order to take advantage of this observation it is useful to look at the Schur complement linear system as non conforming discretization of the Steklov-Poicaré operator, mapping a function φ defined on the interface, to the jump of the normal derivative of its harmonic lifting (computed subdomain-wise). The non-conformity stems from replacing the harmonic lifting with its discretization. If we look at the Schur complement system from this point of view, a straightforward application of the first Strang Lemma, shows that the discretization in the subdomains needs to be designed in order to provide a correct order approximation of outer normal derivative, while there is no direct need to actually provide a good approximation of the solution u in the interior of the subdomains.

The aim of this paper is to formalise the above considerations in the case in which the starting domain decomposition formulation is the *three fields formulation*, and to provide a rigorous error estimate for the trace of u on the the interface, showing that the mesh can actually be chosen to be sensibly coarser in the interior of the subdomains without affecting the precision of the interface approximation, resulting in a sensible reduction in computational cost of the subdomain solvers.

2. The three fields formulation and the Steklov-Poincaré operator. Here and in the following we will use the notation $A \lesssim B$ and $A \gtrsim B$ to indicate that the quantity A is bounded from above – resp. from below – by a positive constant times the quantity B, the constant being independent of any relevant parameter, like the mesh size. The expression $A \simeq B$ will stand for $A \lesssim B \lesssim A$.

[1]IMATI-CNR, Pavia (Italy), silvia.bertoluzza@ian.pv.cnr.it

Let $\Omega \subset \mathbb{R}^2$ be a polygonal domain. We will consider the following simple model problem: given $f \in L^2(\Omega)$, find u satisfying

$$-\Delta u = f \text{ in } \Omega, \qquad u = 0 \text{ on } \partial\Omega. \tag{2.1}$$

To fix the ideas, we will consider consider the *three fields domain decomposition* formulation of such a problem [4]. We want to underline however that the general ideas presented here carry over to many other domain decomposition formulations, both conforming and non-conforming. Considering for simplicity a geometrically conforming decomposition $\Omega = \cup_k \Omega_k$, with Ω_k convex shape-regular polygons, $\Gamma_k = \partial\Omega_k$, and letting $\Sigma = \cup_k \Gamma_k$, we introduce the following functional spaces

$$V = \prod_k H^1(\Omega_k), \qquad \Lambda = \prod_k H^{-1/2}(\Gamma_k),$$
$$\Phi = \{\varphi \in L^2(\Sigma) : \text{there exists } u \in H^1_0(\Omega),\ u = \varphi \text{ on } \Sigma\} = H^1_0(\Omega)|_\Sigma,$$

respectively equipped with the norms:

$$\|u\|_V^2 = \sum_k \|u^k\|^2_{H^1(\Omega_k)}, \qquad \|\lambda\|_\Lambda^2 = \sum_k \|\lambda^k\|^2_{H^{-1/2}(\Gamma_k)},$$

and (see [2])

$$\|\varphi\|_\Phi^2 = \inf_{u \in H^1_0(\Omega) : u = \varphi \text{ on } \Sigma} \|u\|^2_{H^1(\Omega)} \simeq \sum_k |\varphi|^2_{H^{1/2}(\Gamma_k)}.$$

Let $a^k : H^1(\Omega_k) \times H^1(\Omega_k) \to \mathbb{R}$ denote the bilinear form corresponding to the Laplace operator:

$$a^k(w, v) = \int_{\Omega_k} \nabla w \nabla v.$$

The continuous three fields formulation of equation (2.1) is the following ([4]): find $(u, \lambda, \varphi) \in V \times \Lambda \times \Phi$ such that

$$\begin{cases} \forall k, \quad \forall v^k \in H^1(\Omega_k), \quad \forall \mu^k \in H^{-1/2}(\Gamma_k) : \\ \quad a^k(u^k, v^k) \ - \int_{\Gamma_k} v^k \lambda^k \qquad\qquad = \int_{\Omega_k} f v^k, \\ -\int_{\Gamma_k} u^k \mu^k \qquad\qquad + \int_{\Gamma_k} \mu^k \varphi = 0, \\ \\ \text{and } \forall \psi \in \Phi : \\ \qquad\qquad \sum_k \int_{\Gamma_k} \lambda^k \psi \qquad\qquad = 0. \end{cases} \tag{2.2}$$

It is known that this problem admits a unique solution (u, λ, φ), where u is indeed the solution of (2.1) and such that $\lambda^k = \partial u^k / \partial \nu^k$ on Γ_k, and $\varphi = u$ on Σ, where ν^k denotes the outer normal derivative to the subdomain Ω_k.

After choosing discretization spaces $V_h = \prod_k V_h^k \subset V$, $\Lambda_h = \prod_k \Lambda_h^k \subset \Lambda$ and $\Phi_h \subset \Phi$, equation (2.2) can be discretized by a Galerkin scheme, yielding the following problem: find $(u_h, \lambda_h, \varphi_h) \in V_h \times \Lambda_h \times \Phi_h$ such that

$$\left\{\begin{array}{l} \forall k,\ \ \forall v_h^k \in V_h^k,\ \ \forall \mu_h^k \in \Lambda_h^k : \\ \quad a^k(u_h^k, v_h^k) \quad - \int_{\Gamma_k} v_h^k \lambda_h^k \qquad\qquad\qquad = \int_{\Omega_k} f v_h^k, \\ -\int_{\Gamma_k} u_h^k \mu_h^k \qquad\qquad + \int_{\Gamma_k} \mu_h^k \varphi_h = 0, \\ \\ \text{and } \forall \psi_h \in \Phi_h : \\ \qquad\qquad\qquad \sum_k \int_{\Gamma_k} \lambda_h^k \psi_h \qquad\qquad = 0. \end{array}\right. \tag{2.3}$$

Existence, uniqueness and stability of the solution of the discretized problem rely on the validity of two *inf-sup* conditions,

$$\inf_{\lambda_h \in \Lambda_h} \sup_{u_h \in V_h} \frac{\sum_k \int_{\Gamma_k} \lambda_h^k u_h^k}{\|u_h\|_V \|\lambda_h\|_\Lambda} \geq \beta_1 > 0, \quad \inf_{\varphi_h \in \Phi_h} \sup_{\lambda_h \in \Lambda_h} \frac{\sum_k \int_{\Gamma_k} \lambda_h^k \varphi_h}{\|\varphi_h\|_\Phi \|\lambda_h\|_\Lambda} \geq \beta_2 > 0 \tag{2.4}$$

respectively coupling V_h with Λ_h, and Λ_h with Φ_h. Provided (2.4) holds, it is well known ([3]) that we can derive the following error estimate:

$$\|u-u_h\|_V + \|\lambda-\lambda_h\|_\Lambda + \|\varphi-\varphi_h\|_\Phi \lesssim \inf_{v_h \in V_h} \|u-v_h\|_V + \inf_{\mu_h \in \Lambda} \|\lambda-\mu_h\|_\Lambda + \inf_{\psi_h \in \Phi_h} \|\varphi-\psi_h\|_\Phi.$$

The linear system stemming from such an approximation takes the form

$$\begin{pmatrix} A & B^T & 0 \\ B & 0 & C^T \\ 0 & C & 0 \end{pmatrix} \cdot \begin{pmatrix} \underline{u}_h \\ \underline{\lambda}_h \\ \underline{\varphi}_h \end{pmatrix} = \begin{pmatrix} \underline{f} \\ 0 \\ 0 \end{pmatrix}, \tag{2.5}$$

($\underline{u}_h$, $\underline{\lambda}_h$, and $\underline{\varphi}_h$ being the vectors of the coefficients of u_h, λ_h and φ_h in the bases chosen for V_h, Λ_h and Φ_h respectively). The usual approach to the solution of such linear system is to reduce it, by a Schur complement argument, to the solution of a system in the unknown $\underline{\varphi}_h$, which takes the form

$$\mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T \, \underline{\varphi}_h = -\mathbf{C}\mathbf{A}^{-1} \begin{pmatrix} \underline{f} \\ 0 \end{pmatrix}, \quad \mathbf{C} = \begin{bmatrix} 0 & C \end{bmatrix}, \quad \mathbf{A} = \begin{pmatrix} A & B^T \\ B & 0 \end{pmatrix}. \tag{2.6}$$

The matrix $S = \mathbf{C}\mathbf{A}^{-1}\mathbf{C}^T$ does not need to be assembled. The system (2.6) is then solved by an iterative technique (like for instance a conjugate gradient method), for which only the action of S on a given vector needs to be implemented. In particular, multiplying by S implies the need for solving a linear system with matrix $\mathbf{A}$. This reduces, by a proper reordering of the unknowns, to independently solving a discrete Dirichlet problem with Lagrange multipliers in each subdomain. A key observation is that the significant unknown that one is looking for is φ, that is the trace on Σ of the solution u of the equation considered. The actual value of the function u_h and of the multiplier λ_h is only needed at the end of the iterative procedure and possibly only in some of the subdomains, namely the ones in which the end user is actually interested in computing the solution. Along the iterations, the precisions with which u_h and λ_h approximate u and λ respectively is only as important as its effect on the precision with which φ is approximated. From this point of view it would for

instance make sense to replace, along the iterations, the discretization spaces V_h and Λ_h with two other spaces V_h^* and Λ_h^* with $dim(V_h^* \oplus \Lambda_h^*) \ll dim(V_h \oplus \Lambda_h)$ – resulting in a reduction of CPU time in the solution of the discrete Dirichlet problems at each iteration – provided this does not reduce the precision of the approximation of the unknown φ. In this respect, the above mentioned error estimate is pessimistic. In order to obtain a sharper error estimate on the error $\|\varphi - \varphi_h\|_\Phi$ we can look at the linear system (2.6) as a non conforming discretization of the Steklov-Poincaré problem

$$\mathcal{S}\varphi = g \tag{2.7}$$

where we recall that the Steklov-Poincaré operator $\mathcal{S} : \Phi \to \Phi'$ is defined as

$$\langle \mathcal{S}\varphi, \psi \rangle = \sum_k \langle \partial_{\nu^k} \mathcal{L}_H^k \varphi, \psi \rangle$$

where $\mathcal{L}_H^k : H^{1/2}(\Gamma_k) \to H^1(\Omega_k)$ denotes the harmonic lifting:

$$-\Delta(\mathcal{L}_H^k \varphi) = 0, \quad \text{on } \Omega_k, \qquad \mathcal{L}_H^k \varphi = \varphi, \quad \text{on } \Gamma_k,$$

and where $g = g(f)$ is the jump along the interface of the normal derivative of the function u^f verifying $-\Delta u^f = f$ in each Ω_k and $u^f = 0$ on Σ.

The linear system (2.6) is indeed a discrete version of (2.7), the non conformity stemming from the fact that in the computation of the Steklov-Poincaré operator the Dirichlet problem is solved approximatively and the Lagrange multiplier is used to approximate the normal derivative. We can then introduce the notation

$$\mathcal{S}_h\varphi = \sum_k \langle \lambda_h^k(\varphi), \psi \rangle$$

where the $\lambda_h^k(\varphi)$'s are obtained by solving: find $u_h(\varphi) = (u_h^k(\varphi))_k \in V_h$, $\lambda_h(\varphi) = (\lambda_h^k(\varphi))_k \in \Lambda_h$ such that

$$\left\{ \begin{array}{l} \forall k, \quad \forall v_h \in V_h^k, \quad \forall \mu_h \in \Lambda_h^k \\ \quad \int_{\Omega_k} \nabla u_h^k(\varphi) \nabla v_h \quad - \quad \int_{\Gamma_k} \lambda_h^k(\varphi) v \quad = \quad 0 \\ \quad \int_{\Gamma_k} u_h^k(\varphi) \mu_h \qquad\qquad\qquad\qquad = \quad \int_{\Gamma_k} \varphi \mu_h. \end{array} \right. \tag{2.8}$$

In order to give an estimate on the φ component of the error we can use the first Strang Lemma ([5]), which yields

$$\|\varphi - \varphi_h\|_\Phi \lesssim \inf_{\zeta \in \Phi_h} \left\{ \|\varphi - \zeta\|_\Phi + \sup_{\psi_h \in \Phi_h} \frac{\langle (\mathcal{S} - \mathcal{S}_h)\zeta, \psi_h \rangle}{\|\psi_h\|_\Phi} + \sup_{\psi_h \in \Phi_h} \frac{\langle g - g_h, \psi_h \rangle}{\|\psi_h\|_\Phi} \right\}.$$

Let us better analyse the first consistency error term: setting $\lambda^k(\varphi) = \partial_{\nu^k} \mathcal{L}_H^k \varphi$ we have

$$\langle (\mathcal{S} - \mathcal{S}_h)\zeta, \psi_h \rangle = \sum_k \langle \lambda^k(\zeta) - \lambda_h^k(\zeta), \psi_h \rangle \lesssim \left(\sum_k \|\lambda^k(\zeta) - \lambda_h^k(\zeta)\|_{-1/2,\Gamma} \right)^{1/2} \|\psi_h\|_\Phi$$

which yields

$$\sup_{\psi_h\in\Phi_h}\frac{\langle(\mathcal{S}-\mathcal{S}_h)\zeta,\psi_h\rangle}{\|\psi_h\|_\Phi}\lesssim\left(\sum_k\|\lambda^k(\zeta)-\lambda_h^k(\zeta)\|_{-1/2,\Gamma}\right)^{1/2}.$$

It is not difficult to check that a similar result holds also for the second of the two consistency terms. The error $\|\varphi-\varphi_h\|_\Phi$ is thus not directly influenced by the precision with which the unknown u is approximated. The subdomain meshes should not necessarily be chosen by aiming at a good approximation of the whole u but only to a good approximation of its outer conormal derivative λ.

3. The mono-domain problem: local estimates. Let us from now on concentrate on one of the subdomain problems. For the sake of simplicity we will omit the subscript/superscript k. Ω will then denote a polygonal subdomain, Γ its boundary, and, given $\varphi\in H^{1/2}(\Gamma)$ and $f\in L^2(\Omega)$ we will consider the problem of finding $u\in H^1(\Omega)$ and $\lambda\in H^{-1/2}(\Gamma)$ such that

$$\left\{\begin{array}{l}\forall v\in H^1(\Omega),\quad\forall\mu\in H^{-1/2}(\Gamma)\\ \begin{array}{lcccl}\int_\Omega\nabla u\nabla v & - & \int_\Gamma\lambda v & = & \int_\Omega fv\\ \int_\Gamma u\mu & & & = & \int_\Gamma\varphi\mu.\end{array}\end{array}\right.\tag{3.1}$$

Again, we consider a Galerkin discretization: letting $V_h\in H^1(\Omega)$, $\Lambda_h\in H^{-1/2}(\Gamma)$ be two finite dimensional subspaces we look for $u_h\in V_h$, $\lambda_h\in\Lambda_h$ such that

$$\left\{\begin{array}{l}\forall v_h\in V_h,\quad\forall\mu_h\in\Lambda_h\\ \begin{array}{lcccl}\int_\Omega\nabla u_h\nabla v_h & - & \int_\Gamma\lambda_h v_h & = & \int_\Omega fv_h\\ \int_\Gamma u_h\mu_h & & & = & \int_\Gamma\varphi\mu_h.\end{array}\end{array}\right.\tag{3.2}$$

For the reasons explained in the previous section we are interested in giving a sharp bound on the λ component of the error. Under the usual classical assumptions needed for stability of the discrete problem (see (A4) in the following), the standard techniques yield estimates of the form

$$\begin{array}{rcl}\|\lambda-\lambda_h\|_{-1/2,\Gamma} & \leq & \|\lambda-\lambda_h\|_{-1/2,\Gamma}+\|u-u_h\|_{1,\Omega}\\ & \lesssim & \inf\limits_{\eta_h\in\Lambda_h}\|\lambda-\eta_h\|_{-1/2,\Gamma}+\inf\limits_{w_h\in V_h}\|u-w_h\|_{1,\Omega}.\end{array}$$

Such estimate provides a bound for the error on the multiplier λ depending not only on the regularity of λ and the approximation properties of the space Λ_h, but also on the overall regularity of the solution u and on the overall approximation property of the discretization space V_h. If we however try to estimate the error on λ directly, using a very simple argument, we could write

$$\begin{array}{rcl}\|\lambda-\lambda_h\|_{-1/2,\Gamma} & = & \sup\limits_{v\in H^{1/2}(\Gamma)}\dfrac{\int_\Gamma(\lambda-\lambda_h)v}{\|v\|_{1/2,\Gamma}}\\ & = & \sup\limits_{v\in H^{1/2}(\Gamma)}\left\{\dfrac{\int_\Gamma(\lambda-\lambda_h)(v-v_h)}{\|v\|_{1/2,\Gamma}}+\dfrac{\int_\Gamma(\lambda-\lambda_h)v_h}{\|v\|_{1/2,\Gamma}}\right\},\end{array}$$

where $v_h \in V_h|_\Gamma$ is the (unique) element such that $\int_\Gamma \mu_h(v - v_h) = 0$ for all μ_h in Λ_h, which exists and depends continuously on v, provided the standard inf-sup condition needed for stability of problem (3.2) holds. We can then easily bound the two terms on the right hand side thanks to the following bounds

$$\int_\Gamma (\lambda - \lambda_h)(v - v_h) = \int_\Gamma (\lambda - \mu_h)(v - v_h) \leq \|\lambda - \mu_h\|_{-1/2,\Gamma} \|v\|_{1/2,\Gamma}$$

which yields, thanks to the arbitrariness of μ_h,

$$\int_\Gamma (\lambda - \lambda_h)(v - v_h) \lesssim \inf_{\mu_h \in \Lambda_h} \|\lambda - \mu_h\|_{-1/2,\Gamma} \|v\|_{1/2,\Gamma}.$$

The second term can be bound by observing that for all $w_h \in V_h$, Galerkin orthogonality yields

$$\int_\Gamma (\lambda - \lambda_h) w_h = \int_\Omega \nabla(u - u_h) \nabla w_h.$$

We can then choose any (fixed) subdomain $\Omega_0 \subset \Omega$ such that $\Gamma \subset \partial\Omega_0$, construct a lifting $w_h \in V_h$ of v_h verifying

$$w_h|_\Gamma = v_h, \qquad supp w_h \subset \Omega_0, \qquad \|w_h\|_{1,\Omega} \lesssim \|v_h\|_{1/2,\Gamma}$$

(the constant in the last bound naturally depending on the subdomain Ω_0), and we would get

$$\int_\Gamma (\lambda - \lambda_h) v_h \lesssim \|u - u_h\|_{1,\Omega_0} \|v_h\|_{1/2,\Gamma}.$$

Now, we recall that we are dealing with the Galerkin solution an elliptic problem. If Ω_0 was an interior subdomain ($\bar{\Omega}_0 \subset\subset \Omega$) and letting Ω_1 be an intermediate subdomain, by applying a result by Nitsche and Schatz ([7]) we could bound $\|u - u_h\|_{1,\Omega_0}$ as

$$\|u - u_h\|_{1,\Omega_0} \lesssim h^{s-1} \|u\|_{1,\Omega_1} + \|u - u_h\|_{-p,\Omega}. \tag{3.3}$$

h being the mesh size of the discretization *relative to the subdomain* Ω_1 and p being any positive integer, arbitrary but fixed. Again the constants in the bound depends on the two subdomains Ω_0 and Ω_1. Since the global mesh size enters only through a negative norm of the error, and therefore, under suitable assumptions, with an higher order, its influence on the local error on Ω_0 is reduced.

In order to apply such kind of reasoning to the estimate of the error on the multiplier we need then to provide an estimate of the form (3.3) in the case in which Ω_0 is roughly speaking a strip all along the boundary. It turns out (see [1]) that in proving such an estimate we will also directly prove an estimate on the error $\|\lambda - \lambda_h\|_{-1/2,\Gamma}$ without need of using the above argument.

Let e_i, $i = 1, \cdots, N$ be the edges of Γ and let θ_i, $i = 1, \cdots, N$ be the interior angles. Let $\theta_0 = \max_i \theta_i$ be the maximum angle, and recall that the polygon is convex, that is $\theta_0 < \pi$. Assume that the discretization spaces V_h and Λ_h satisfy

(A1) **Global Approximation for** u**.** Let $1 \leq s \leq k_1$, $0 \leq \ell \leq r_1$. For each $u \in H^\ell(\Omega)$, there exists an element $w \in V_h$ such that

$$\|u - w\|_{s,\Omega} \lesssim H^{\ell-s}\|u\|_{\ell,\Omega};$$

Let now $\Omega_1 \subset \Omega$ be an open subdomain of Ω such that

$$\Gamma \subset \partial\Omega_1, \qquad \partial\Omega_1 \setminus \Gamma \text{ is of class } C^\infty.$$

(see figure 3.1) and assume that the space V_h has, when restricted to Ω_1, better approximation properties. More precisely assume that for any two open subdomains $G_0 \subset G \subseteq \Omega_1$ satisfying

$$\Gamma = \partial G_0 \cap \partial G, \qquad \partial G \setminus \Gamma \text{ and } \partial G_0 \setminus \Gamma \text{ are of class } C^\infty, \qquad \partial G_0 \setminus \Gamma \subset G$$

there exists an h_0 such that if $h \leq h_0$ then

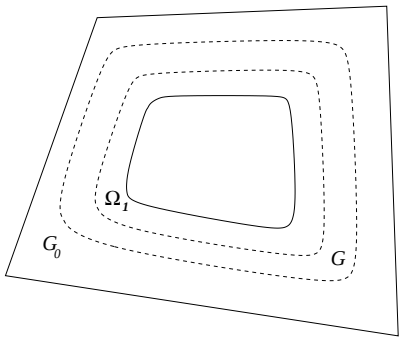


Figure 3.1: Subdomains $G_0 \subset G \subset \Omega_1$

(A2) **Local approximation for** u. Let $1 \leq s \leq k_1$, $s \leq \ell \leq r_1$. For each $u \in H^\ell(G)$, there exists an element $w \in V_h$ such that

$$\|u - w\|_{s,G} \lesssim h^{\ell-s}\|u\|_{\ell,G};$$

moreover if u is supported in G_0 then w can be chosen to be supported in G.

(A3) **Discrete commutator property**. Let $\omega \in C^\infty(G)$, $\omega = 0$ in $G \setminus G_0$, and let $v_h \in V_h$. Then there exists $w_h \in V_h$ such that $w_h = 0$ in $\Omega \setminus G_0$ and such that

$$\|\omega v_h - w_h\|_{1,G} \lesssim h\|v_h\|_{1,G}.$$

Remark 3.1 *Assumption A3 is a classical assumption that is usually made when some* localization *technique needs to be applied. It can be shown to hold under some standard assumptions, see [6]*

Finally, assume that the multiplier space Λ_h satisfies

(A4) **Stability conditions**. We have that

$$\inf_{\mu_h\in\Lambda_h}\sup_{v_h\in V_h}\frac{\int_\Gamma \mu_h v_h}{\|\mu_h\|_{-1/2,\Gamma}\|v_h\|_{1,\Omega}}\geq\alpha>0.$$

and for all v_h in V_h such that $\int_\Gamma v_h\mu_h=0$ forall $\mu_h\in\Lambda_h$, we have

$$\int_\Gamma|\nabla v_h|^2\gtrsim\|v_h\|^2_{1,\Omega}.$$

(A5) **Approximation for** λ. Let $-1/2<\ell\leq r_2$. For each $\lambda\in H^\ell(\Gamma)$, there exists an element $\mu\in\Lambda_h$ such that

$$\|\lambda-\mu\|_{-1/2,\Gamma}\lesssim h^{\ell+1/2}\sum_{i=1}^N\|\lambda\|_{\ell,e_i};$$

Let now $\Omega_0\subset\Omega_1$ be an open subdomain satisfying

$$\Gamma\subset\partial\Omega_0,\qquad\partial\Omega_0\setminus\Gamma\subset\Omega_1,\qquad\partial\Omega_0\setminus\Gamma\text{ is on class }C^\infty.$$

Under the previous assumptions we can prove the following theorem.

Theorem 3.1 *Suppose that A1–A5 are satisfied. Assume that $u\in H^s(\Omega)$, Then, for t_0 positive arbitrary but fixed verifying $t_0<s_0$, if h is sufficiently small the following bound holds*

$$\|u-u_h\|_{1,\Omega_0}+\|\lambda-\lambda_h\|_{-1/2,\Gamma}\lesssim(h^\tau+H^{\sigma+t_0})\|u\|_{s,\Omega}.$$

with $\tau=\min\{s-1,r_1-1,r_2+1/2\}$ and $\sigma=\min\{s,r_1,r_2+3/2\}$. where the implicit constant in the inequality depends on Ω_0, Ω_1 and t_0.

Trivially this yields the following corollary

Corollary 3.1 *Under the same assumptions of theorem 3.1 it holds*

$$\|\lambda-\lambda_h\|_{-1/2,\Gamma}\lesssim(h^\tau+H^{\sigma+t_0})\|u\|_{s,\Omega}.$$

By applying such corollary, it is clear that choosing a discretization satisfying assumptions $A1--A5$ with

$$H=h^{\tau/(\sigma+t_0)}$$

yields the optimal error estimate

$$\|\lambda-\lambda_h\|_{-1/2,\Gamma}\lesssim h^\tau\|u\|_{s,\Omega}.$$

In particular, the above results implies that, as far as the approximation of the Lagrange multiplier λ is concerned it is possible to chose the mesh in the interior of the subdomain sensibly coarser than the mesh that would be needed to approximate the function u with the same accuracy.

4. Numerical results. Let us test the theoreical results of the previous section on a simple example. Let $\Omega =]-1,1[^2$ and consider the following model problem:

$$-\Delta u = 13\sin(2x)\cos(3x), \text{ in } \Omega, \qquad u = \sin(2x)\cos(3y), \text{ on } \Gamma. \tag{4.1}$$

It is not difficult to verify that the solution of such a problem is the function $u = \sin(2x)\cos(3y)$ (see Figure 4.1).

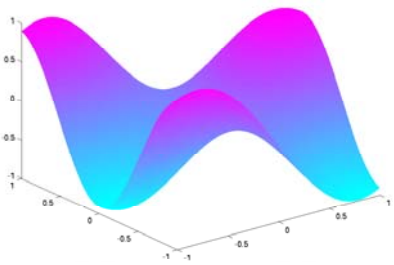


Figure 4.1: Solution of the model problem

In order to approximate u we consider a Lagrange multiplier formulation in the form (3.2) of the above problem, where V_h is chosen to be a P1 finite element space and Λ_h is defined as the trace of V_h on the boundary Γ. It is not difficult to check that if the triangulation on the boundary is quasi-uniform then assumptions A1–A5 are satisfied with $r_1 = 2$ and $r_2 = 1/2 - \varepsilon$ ($\varepsilon > 0$ arbitrary but fixed).

Letting $\delta \in]0,1[$ be a fixed parameter, we consider triangulations of Ω constructed in the following way: starting from a quasi uniform triangulation $\mathcal{T}_H$ of the whole Ω, set $\mathcal{T}_h^0 = \mathcal{T}_H$, and let $\mathcal{T}_h^j$ be obtained from $\mathcal{T}_h^{j-1}$ by "refining" (precisare) all those triangles T in $\mathcal{T}_h^{j-1}$ such that $suppT \cap \Omega \backslash]-1+\delta, 1-\delta[^2 \neq \emptyset$.

We compare the solution of problem (4.1) obtained with a quasi uniform triangulation of mesh-size $h = H/2^j$, with the one obtained using the triangulation $\mathcal{T}_H^j$ for $j = 1, \cdots, 4$ and for different values of the parameter δ. In the following figures we display both the $H^1(\Omega)$ and the $L^2(\Omega)$ norms of the error $u - u_h$, and the $L^2(\Omega)$ norm of the error $\lambda - \lambda_h$ (which for computational simplicity we prefer to the $H^{-1/2}(\Gamma)$). As one can expect, for the boundary refined triangulations, both the $H^1(\Omega)$ and the $L^2(\Omega)$ norms of the error on u are mainly influenced from coarse triangulations in the interior of Ω and do not sensibly vary as j increases, while they decrease with the expected rates when considering the quasi uniform mesh. Conversely, when considering the $L^2(\Gamma)$ norm of the error on λ, the boundary refined and the quasi uniform meshes display the same behaviour as j increases. However, the boundary refined meshes allows to get the same error with considerably less degrees of freedoms – and therefore with considerably lower computational cost.

REFERENCES

[1] S. Bertoluzza. Mesh design for the subdomain solvers arising in non conforming Steklov-Poincaré discretizations.

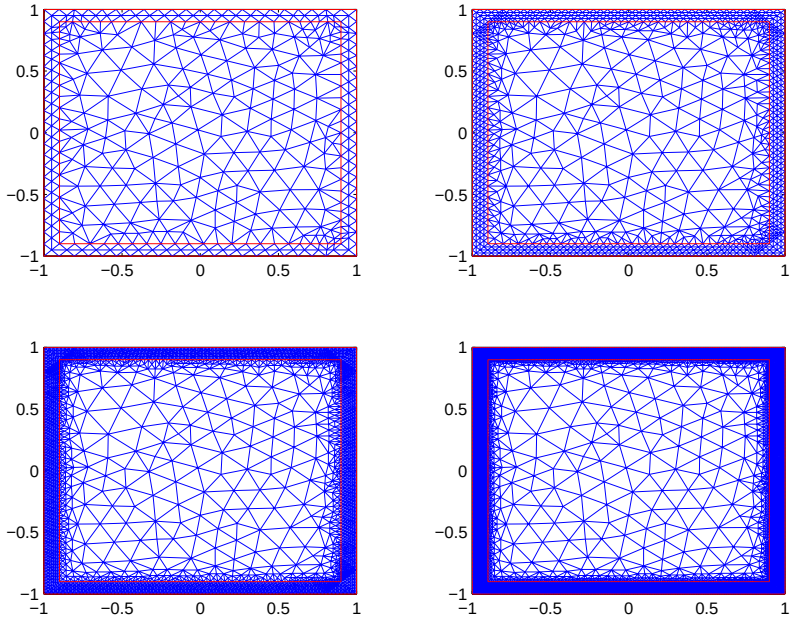


Figure 4.2: The triangulations used for the tests.

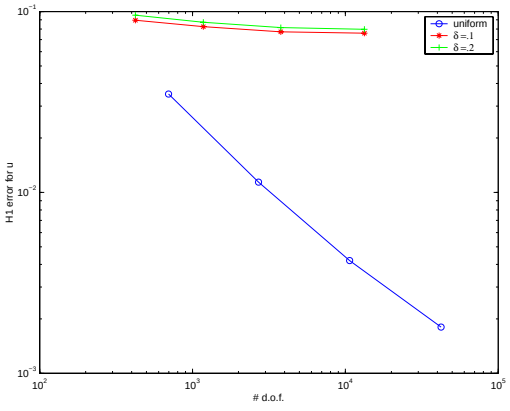


Figure 4.3: Error $\|u - u_h\|_{1,\Omega}$ vs. the number of degrees of freedom for the quasi uniform mesh and for the boundary refined mesh with δ resp. equals to .1 and .2

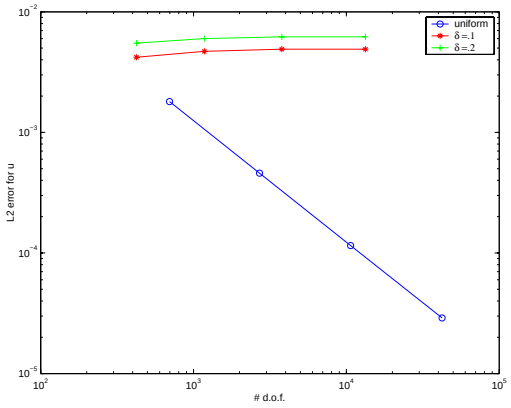


Figure 4.4: Error $\|u - u_h\|_{0,\Omega}$ vs. the number of degrees of freedom for the quasi uniform mesh and for the boundary refined mesh with δ resp. equals to .1 and .2

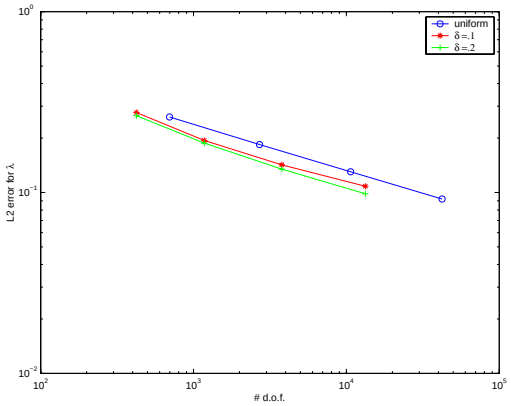


Figure 4.5: Error $\|\lambda - \lambda_h\|_{0,\Gamma}$ vs. the number of degrees of freedom for the quasi uniform mesh and for the boundary refined mesh with δ resp. equals to .1 and .2

[2] S. Bertoluzza. Analysis of a stabilized domain decomposition method. Technical report, I.A.N.-C.N.R. Pavia, 2000.

[3] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Springer-Verlag, New-York, 1991.

[4] F. Brezzi and L. D. Marini. A three fields domain decomposition method. *Contemp. Math.*, 157:27–34, 1994.

[5] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. North-Holland, Amsterdam, 1978.

[6] J. A. Nitsche. Ein kriterium für die quasi-optimalitaet des Ritzschen verfahrens. *Numer. Math.*, 11:346–348, 1968.

[7] J. A. Nitsche and A. H. Schatz. Interior estimates for Ritz-Galerkin methods. *Math. Comp.*, 28:937–958, 1974.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

3. A Generalized FETI - DP Method for a Mortar Discretization of Elliptic Problems

M. Dryja[1], O. B. Widlund[2]

1. Introduction. In this paper, an iterative substructuring method with Lagrange multipliers is proposed for discrete problems arising from approximations of elliptic problem in two dimensions on non-matching meshes. The problem is formulated using a mortar technique. The algorithm belongs to the family of dual-primal FETI (Finite Element Tearing and Interconnecting) methods which has been analyzed recently for discretization on matching meshes. In this method the unknowns at the vertices of substructures are eliminated together with those of the interior nodal points of these substructures. It is proved that the preconditioner proposed is almost optimal; it is also well suited for parallel computations.

We will consider a dual-primal FETI (FETI-DP) method, see [5], [9], and [6], for solving discrete problems arising from the approximation of the Dirichlet problem defined on a union of substructures Ω_i. Each substructure is the union of a number of elements of a coarse, shape-regular triangulation and the number of these triangles, which form such a substructure, is assumed to be uniformly bounded. The discretization is obtained by a mortar method on nonmatching meshes across the interface Γ; see [1], [2]. As in all other iterative substructuring methods, the unknowns corresponding to the interior nodal points are eliminated; in this dual-primal FETI method those at the vertices of Ω_i are eliminated as well. The remaining Schur complement system is solved by a FETI method; see Section 3 for details.

A full analysis of the convergence of several FETI-DP methods has been worked out for finite element approximations on matching meshes; see [9] for the two-dimensional case and [6] for three dimensions. This method, on nonmatching meshes and for the mortar discretizations in the 2-D case, was analyzed in [4]. The preconditioner used there is a standard one and the estimates are not optimal in the general case. In this paper, our analysis is extended to the preconditioner suggested in [7] for matching meshes. The results obtained for this method is better than those of [4]. The superiority of this method is consistent with the numerical results reported on in [11].

The remainder of this paper is organized as follows. In Section 2 differential and discrete problems are formulated while in Section 3 the dual-primal formulation is introduced. Sections 4 is are devoted to the analysis of the proposed preconditioner.

2. Differential and discrete problems. We will consider the following elliptic problem: find $u^* \in H_0^1(\Omega)$ such that

$$a(u^*, v) = f(v), \qquad v \in H_0^1(\Omega), \tag{2.1}$$

where

$$a(u, v) = \int_\Omega \nabla u \cdot \nabla v dx, \qquad f(v) = \int_\Omega f v \, dx$$

[1]Department of Mathematics, Warsaw University, Warsaw, Banacha 2, 02-097 Warsaw, Poland, E-mail: dryja@mimuw.edu.pl
[2]Courant Institute of Mathematical Sciences, New York University, 251 Mercer Street, New York, NY 10012, USA, E-mail: widlund@cs.nyu.edu

and Ω is a polygonal 2-D region which is a union of polygons Ω_i, $i = 1, \ldots, N$. These subregions form a coarse partitioning of Ω with subdomains with diameters on the order of H. In each Ω_i, we introduce a quasi-uniform, but otherwise arbitrary, triangulation of the subregion with a mesh parameter h_i; generally the resulting triangulations do not match across the edges of the Ω_i.

Let

$$W(\Omega) = W(\Omega_1) \times \cdots \times W(\Omega_N),$$

where $W(\Omega_i)$ are the finite element spaces of piecewise linear, continuous functions on the triangulation of Ω_i and which vanish on $\partial\Omega$ and let the interface be defined by $\Gamma = (\cup\partial\Omega_i)\backslash\partial\Omega$. We choose mortar and nonmortar edges of Γ, and denote them by $\gamma_{m(j)}$ and $\delta_{m(i)}$. In the analysis of the proposed preconditioner, we need a uniform bound on the ratios $h_{\gamma_{m(j)}}/h_{\delta_{m(i)}}$ where $h_{\gamma_{m(j)}}$ and $h_{\delta_{m(i)}}$ are the mesh parameters of $\gamma_{m(j)} \subset \partial\Omega_j$ and $\delta_{m(i)} \subset \partial\Omega_i$, $(\gamma_{m(j)} = \delta_{m(i)})$, respectively. The problem (2.1) is approximated in $X(\Omega)$, a subspace of $W(\Omega)$, of functions which satisfy the mortar condition, see [1], [2],

$$b(u, \psi) \equiv \sum_{i=1}^{N} \sum_{\delta_{m(i)} \subset \partial\Omega_i} \int_{\delta_{m(i)}} (u_i - u_j)\psi ds = 0, \qquad \psi \in M(\Gamma), \tag{2.2}$$

where $M(\Gamma) = \Pi_i \Pi_{\delta_{m(i)} \subset \partial\Omega_i} M(\delta_{m(i)})$ and $M(\delta_{m(i)})$ is the standard mortar space defined on $\delta_{m(i)}$, i.e., piecewise linear continuous functions which are constant on the elements which intersect $\partial\delta_{m(i)}$. Additionally, we assume that the functions of $X(\Omega)$ are continuous at the vertices of Ω_i, i.e., they take the same values, see [2]. In (2.2) $u_i \in W(\Omega_i)$ and $u_j \in W(\Omega_j)$ are the restrictions of u to $\delta_{m(i)}$ and $\gamma_{m(j)}$, respectively.

3. A dual-primal formulation of the problem. We will use some of the notations of [9], [6]. Let

$$K := diag_{j=1}^{N}(K^{(j)}), \tag{3.1}$$

where $K^{(j)}$ is the local stiffness matrix with respect to the standard basis functions of $W(\Omega_j)$. We eliminate the unknown variables corresponding to the interior nodal points and the vertices of Ω_i. A Schur complement $\tilde{S}$ results which is of the form:

$$\tilde{S} := K_{rr} - \begin{pmatrix} K_{ri} & K_{rc} \end{pmatrix} \begin{pmatrix} K_{ii} & K_{ic} \\ K_{ci} & K_{cc} \end{pmatrix}^{-1} \begin{pmatrix} K_{ir} \\ K_{cr} \end{pmatrix}. \tag{3.2}$$

Here,

$$\tilde{K} := \begin{pmatrix} K_{ii} & K_{ic} & K_{ir} \\ K_{ci} & K_{cc} & K_{cr} \\ K_{ri} & K_{rc} & K_{rr} \end{pmatrix},$$

where the rows correspond to the interior, vertex, and remaining (edge) nodal points, respectively. It is obtained from K by reordering the unknowns and taking into account that the functions of $X(\Omega)$ are continuous at the subdomain vertices.

Let

$$W(\Gamma) = W(\partial\Omega_1) \times \cdots \times W(\partial\Omega_N)$$

and let $W_r(\Gamma)$ denote the space of functions defined at the edge nodal points and which vanish at the vertices of Ω_i, and let $W_c(\Gamma)$ be the subspace of $W(\Gamma)$ of functions that are continuous at the vertices.

The dual-primal formulation of the mortar discretization of (2.1) is: find $u_r^* \in W_r(\Gamma)$ such that

$$J(u_r^*) = \min_{\substack{v_r \in W_r \\ Bv_r = 0}} J(v_r), \quad J(v) := 1/2\langle \tilde{S}v, v\rangle - \langle f_r, v\rangle, \tag{3.3}$$

where $< , >$ means the scalar product in l_2. B is defined by the mortar condition (2.2) as follows: on $\delta_{m(i)} \subset \partial\Omega_i$, $\delta_{m(i)} = \gamma_{m(j)}$, the matrix form of (2.2) is

$$B_{\delta_{m(i)}} \underline{u}_{i|\delta_{m(i)}} - B_{\gamma_{m(j)}} \underline{u}_{j|\gamma_{m(j)}} = 0. \tag{3.4}$$

Here,

$$B_{\delta_{m(i)}} = \{(\psi_l, \varphi_p)_{L^2(\delta_{m(i)})}\}, \qquad l, p = 1, ..., n_{m(i)},$$
$$\varphi_p \in W_i(\partial\Omega_i)_{|\delta_{m(i)}}, \psi_l \in M(\delta_{m(i)}) \ ,$$
$$B_{\gamma_{m(j)}} = \{(\psi_l, \varphi_k)_{L^2(\delta_{m(i)})}\}, \quad l = 1, ..., n_{m(i)}, \ \ k = 1, ..., n_{m(j)},$$

and $\varphi_k \in W_j(\partial\Omega_j)_{|\gamma_{m(j)}}; n_{m(i)}$ and $n_{m(j)}$ are the number of interior nodal points of $\delta_{m(i)}$ and $\gamma_{m(j)}$, respectively. Condition (3.4) can be rewritten as

$$\underline{u}_{i|\delta_{m(i)}} - B^{-1}_{\delta_{m(i)}} B_{\gamma_{m(j)}} \underline{u}_{j|\gamma_{m(j)}} = 0, \tag{3.5}$$

since the matrix $B_{\delta_{m(i)}} = B^T_{\delta_{m(i)}} > 0$. We note that $B_{\gamma_{m(j)}}$ is generally a rectangular matrix.

The matrix B is block-diagonal,

$$B = blockdiag\{D_{\delta_{m(i)}}\} \tag{3.6}$$

for i = 1,. . . ,N, and $\delta_{m(i)} \subset \partial\Omega_i$ where

$$D_{\delta_{m(i)}} \begin{pmatrix} \underline{u}_{i|\delta_{m(i)}} \\ \underline{u}_{j|\gamma_{m(j)}} \end{pmatrix} \equiv (I \ \ (-B^{-1}_{\delta_{m(i)}} B_{\gamma_{m(j)}})) \begin{pmatrix} \underline{u}_{i|\delta_{m(i)}} \\ \underline{u}_{j|\gamma_{m(j)}} \end{pmatrix}. \tag{3.7}$$

Introducing a space of Lagrange multipliers $V := Im(B)$ to enforce the constraints $Bv_r = 0$, we obtain a saddle point formulation of (3.3),

$$\begin{pmatrix} \tilde{S} & B^T \\ B & 0 \end{pmatrix} \begin{pmatrix} u_r^* \\ \lambda^* \end{pmatrix} = \begin{pmatrix} \tilde{f}_r \\ 0 \end{pmatrix}, \tag{3.8}$$

where $u_r^* \in W_r(\Gamma)$ and $\lambda^* \in V$. We obtain the problem

$$F\lambda^* = d, \tag{3.9}$$

where

$$F = B\tilde{S}^{-1}B^T, \qquad d = B\tilde{S}^{-1}\tilde{f}_r.$$

We now define a preconditioner for F. Let

$$S^{(j)} = K_{bb}^{(j)} - K_{bi}^{(j)}(K_{ii}^{(j)})^{-1}K_{ib}^{(j)}, \tag{3.10}$$

be the standard Schur complement of $K^{(j)}$ where $K_{ii}^{(j)}$ and $K_{bb}^{(j)}$ are the submatrices of $K^{(j)}$ corresponding to the interior and boundary unknowns of $\bar{\Omega}_j$, respectively. Let

$$S_{rr}^{(j)} = K_{rr}^{(j)} - K_{ri}^{(j)}(K_{ii}^{(j)})^{-1}K_{ir}^{(j)} \tag{3.11}$$

denote the Schur complement of $K^{(j)}$, without the rows and columns corresponding to the vertices. It is the restriction of $S^{(j)}$ to the space of functions which vanish at the vertices. Let

$$S := diag_{i=1}^{N}(S^{(i)}), \qquad S_{rr} := diag_{i=1}^{N}(S_{rr}^{(i)}).$$

We can take a preconditioner M of F of the form

$$M = (BS_{rr}B^T)^{-1}, \qquad M^{-1} = BS_{rr}B^T. \tag{3.12}$$

This preconditioner, called the standard one, was analyzed in [4] for two cases. In the first case there is Neumann-Dirichlet (N-D) ordering of substructures Ω_i; a Neumann substructure Ω_i is one where all sides are chosen as mortars while for a Dirichlet substructure all sides are nonmortars. In the second case, we do not have such ordering. For this preconditioner a bound was established for the condition number of FETI-DP method which is proportional to $(1+\log(H/h))^2$ in the first case while we need $(1+\log(H/h))^4$ in the second case.

We will now design a preconditioner for FETI-DP method which is similar to the one used in a FETI method on matching meshes in [7]. It is analyzed in the general case and a bound is obtained for the condition number of this method that is proportional to $(1+\log(H/h))^2$ only.

Let us introduce a scaling in $D_{\delta_{m(i)}}$, cf. (3.7), given by

$$\tilde{D}_{\delta_{m(i)}}\begin{pmatrix}\underline{u}_{i|\delta_{m(i)}}\\ \underline{u}_{j|\gamma_{m(j)}}\end{pmatrix} \equiv \{I \ \ (-\alpha_{ij}^{(m)}B_{\delta_{m(i)}}^{-1}B_{\gamma_{m(j)}})\}\begin{pmatrix}\underline{u}_{i|\delta_{m(i)}}\\ \underline{u}_{j|\gamma_{m(j)}}\end{pmatrix} \tag{3.13}$$

where $\alpha_{ij}^{(m)} = (h_{\delta_{m(i)}}/h_{\gamma_{m(j)}})$ and, cf. (3.6), let

$$\tilde{B} = blockdiag(\tilde{D}_{\delta_{m(i)}}) \tag{3.14}$$

for $i = 1, \ldots, N$, and $\delta_{m(i)} \subset \partial\Omega_i$. The preconditioner $\tilde{M}$ for F is of the form

$$\tilde{M}^{-1} = (B\tilde{B}^T)^{-1}\tilde{B}S_{rr}\tilde{B}^T(\tilde{B}B^T)^{-1}. \tag{3.15}$$

Remark We could also take

$$\widehat{M}^{-1} = diag(BB^T)^{-1}BS_{rr}B^T diag(BB^T)^{-1} \tag{3.16}$$

This corresponds to the preconditioner introduced in [8] for a FETI method on matching and nonmatching triangulations. To our knowledge, there is no full analysis of that method.

4. Convergence analysis. In this section we prove that the preconditioner $\tilde{M}$ is spectrally equivalent to F, except for a $(1+\log(H/h))^2$ factor; see Theorem 1. We follow the approach of [9], [6]. We first prove two auxiliary results.

Let us introduce the operator $P = \tilde{B}^T(B\tilde{B}^T)^{-1}B$ defined on W_r. We note that P is a projection, $P^2 = P$.

Lemma 1 Let $h_{\delta_{m(i)}} \sim h_{\gamma_{m(j)}}$, $\delta_{m(i)} \subset \partial\Omega_i$, i = 1,..., N be satisfied. Then for $w_r \in W_r$

$$|Pw_r|^2_{S_{rr}} \leq C(1+\log(H/h))^2 |w_r|^2_{\tilde{S}} \tag{4.1}$$

holds where the constant C is independent of $H = max_i H_i$ and $h = min_i h_i$.

Proof Let w be the discrete harmonic extension of w_r to the interior points and to the vertices in the sense of $< \tilde{S}u, u >$. We have

$$|w_r|^2_{\tilde{S}} = |w|^2_S, \quad w \in W_c. \tag{4.2}$$

Using this fact, we estimate $|Pw_r|_{S_{rr}}$ in terms of $|w|^2_S$. We construct $I^H w$ the function which is linear on the edges and which takes the values of w at the vertices. Setting $u \equiv w - I^H w$ and noting that $BI^H w = 0$, we have

$$|Pw_r|^2_{S_{rr}} = |Pu|^2_{S_{rr}} = \sum_{i=1}^{N} |Pu|^2_{S^{(i)}}. \tag{4.3}$$

We note that $Pu = 0$ at the vertices. Using that and setting $v = (B\tilde{B}^T)^{-1}Bu$, we have

$$|Pw_r|^2_{S^{(i)}} = |\tilde{B}^T v|^2_{S^{(i)}} \leq C\{ \sum_{\delta_{m(i)} \subset \partial\Omega_i} |\tilde{B}^T v|^2_{S_{\delta_{m(i)}}} + \sum_{\gamma_{m(i)} \subset \partial\Omega_i} |\tilde{B}^T v|^2_{S_{\gamma_{m(i)}}} \}, \tag{4.4}$$

where $S_{\delta_{m(i)}}$ and $S_{\gamma_{m(i)}}$ are matrix representations of the $H^{1/2}_{00}$ - norm on $\delta_{m(i)}$ and $\gamma_{m(i)}$, respectively; see Lemma 2 below. From the structure of $\tilde{B}$, see (3.13) and (3.14), it follows that

$$|\tilde{B}^T v|^2_{S_{\delta_{m(i)}}} = |v_i|^2_{S_{\delta_{m(i)}}} \tag{4.5}$$

and that

$$|\tilde{B}^T v|^2_{S_{\gamma_{m(i)}}} = |\tilde{B}^T_{ji} v_j|^2_{S_{\gamma_{m(i)}}}$$

where, here and below, $\tilde{B}_{ji} = \alpha^{(m)}_{ji} B^{-1}_{\delta_{m(j)}} B_{\gamma_{m(i)}} \equiv \alpha^{(m)}_{ji} B_{ji}$, $\gamma_{m(i)} = \delta_{m(j)}$, $\delta_{m(j)} \subset \partial\Omega_j$, and v_i and v_j are restrictions of v to $\bar{\Omega}_i$ and $\bar{\Omega}_j$, respectively.

We now prove that

$$|\tilde{B}^T_{ji} v_j|^2_{S_{\gamma_{m(i)}}} \leq C|v_j|^2_{S_{\delta_{m(j)}}}. \tag{4.6}$$

We note that $v = 0$ at the cross points. We have

$$|\tilde{B}^T_{ji} v_j|^2_{S_{\gamma_{m(i)}}} = \sup_{\varphi} \frac{| < S^{1/2}_{\gamma_{m(i)}} \tilde{B}^T_{ji} v_j, \varphi >_{\gamma_{m(i)}} |^2}{|\varphi|^2_{\gamma_{m(i)}}} =$$

$$= \sup_t \frac{| < v_j, \tilde{B}_{ji} t >_{\delta_{m(j)}} |^2}{|S^{-1/2}_{\gamma_{m(i)}} t|^2_{\gamma_{m(i)}}},$$

where $< \cdot, \cdot >_{\gamma_{m(i)}}$ and $< \cdot, \cdot >_{\delta_{m(j)}}$ are ℓ_2-inner products. Hence,

$$|\tilde{B}^T_{ji} v_j|^2_{S_{\gamma_{m(i)}}} \leq |S^{1/2}_{\delta_{m(j)}} v_j|^2_{\delta_{m(j)}} \sup_t \frac{|S^{-1/2}_{\delta_{m(j)}} \tilde{B}_{ji} t|^2_{\delta_{m(j)}}}{|S^{-1/2}_{\gamma_{m(i)}} t|^2_{\gamma_{m(i)}}}. \tag{4.7}$$

Let, here and below, $\pi_{\delta_{m(j)}}(t, 0)$ correspond to $B_{ji} t$ for a piecewise linear, continuous function, also denoted by t, and defined on $\gamma_{m(i)}$ by a vector t with components that vanish at the end of $\gamma_{m(i)}$. Using Lemma 2, below, and the $H^{-1/2}$-stability of $\pi_{\delta_{m(j)}}$, see [1], we get

$$|S^{-1/2}_{\delta_{m(j)}} \tilde{B}_{ji} t|^2_{\delta_{m(j)}} \leq C h^{-2}_{\gamma_{m(i)}} ||\pi_{\delta_{m(j)}}(t, 0)||^2_{H^{-1/2}(\delta_{m(j)})} \leq$$

$$\leq C h^{-2}_{\gamma_{m(i)}} \parallel t \parallel^2_{H^{-1/2}(\gamma_{m(i)})} \leq C |S^{-1/2}_{\gamma_{m(i)}} t|^2.$$

Here $H^{-1/2}$ is the dual to $H^{1/2}_{00}$. Using this bound in (4.7), we get

$$|\tilde{B}^T_{ji} v_j|^2_{S_{\gamma_{m(i)}}} \leq C |S^{1/2}_{\delta_{m(j)}} v_j|^2_{\delta_{m(j)}},$$

which proves (4.6). Using (4.5) and (4.6) in (4.4), we have

$$|\tilde{B}^T v|^2_{S^{(i)}} \leq C \{ \sum_{\delta_{m(i)} \subset \partial\Omega_i} |v_i|^2_{S_{\delta_{m(i)}}} + \sum_{\delta_{m(j)}} |v_j|^2_{S_{\delta_{m(j)}}} \}, \tag{4.8}$$

where the second sum is taken over $\delta_{m(j)} \subset \Omega_j$ such that $\gamma_{m(i)} = \delta_{m(j)}$ with $\gamma_{m(i)} \subset \partial\Omega_i$.

We now estimate the term $|S^{1/2}_{\delta_{m(i)}} v_i|^2$ of (4.8) as follows. We have

$$|v|^2_{S_{\delta_{m(i)}}} \leq 2 \{ |(B\tilde{B}^T)^{-1} B u - \frac{1}{2} B u|^2_{S_{\delta_{m(i)}}} + \frac{1}{4} |Bu|^2_{S_{\delta_{m(i)}}} \}. \tag{4.9}$$

We first estimate the second term. Using the structure of B, see (3.7), we have

$$|Bu|^2_{S_{\delta_{m(i)}}} \leq 2 \{ |u_i|^2_{S_{\delta_{m(i)}}} + |B_{ij} u_j|^2_{S_{\delta_{m(i)}}} \}, \tag{4.10}$$

where $\delta_{m(i)} = \gamma_{m(j)}$, $\gamma_{m(j)} \subset \Omega_j$. We note that

$$|B_{ij} u_j|^2_{S_{\delta_{m(i)}}} \leq C \parallel \pi_{\delta_{m(i)}}(u_j, 0) \parallel^2_{H^{1/2}_{00}(\delta_{m(i)})} \leq$$

$$\leq C |u_j|^2_{H^{1/2}_{00}(\gamma_{m(j)})} \leq C |u_j|^2_{S_{\gamma_{m(j)}}}.$$

Here we have used the $H^{1/2}_{00}$ - stability of $\pi_{\delta_{m(i)}}$, see [1]. Using this in (4.10), we have

$$|Bu|^2_{S_{\delta_{m(i)}}} \leq C \big\{ |u_i|^2_{S_{\delta_{m(i)}}} + |u_j|^2_{S_{\gamma_{m(j)}}} \big\}. \tag{4.11}$$

To estimate the first term of (4.9), we first use the fact that $|(B\tilde{B}^T)^{-1}| \leq 1$ since $B\tilde{B}^T = I_{\delta_{m(i)}} + B_{ij}\tilde{B}_{ij}^T$ on $\delta_{m(i)}$; this follows from the structure of B. Here $I_{\delta_{m(i)}}$ is the identity matrix of a dimension equal to the number of nodal points of $\delta_{m(i)}$. Using that and $S_{\delta_{m(i)}} \leq CI_{\delta_{m(i)}}$, we have

$$\begin{aligned} |(B\tilde{B}^T)^{-1}Bu - \frac{1}{2}Bu|^2_{S_{\delta_{m(i)}}} &\leq C|(B\tilde{B}^T)^{-1}(Bu - \frac{1}{2}(B\tilde{B}^T)Bu)|^2_{\ell_2} \\ &\leq C|Bu - \frac{1}{2}B\tilde{B}^T Bu|^2_{\ell_2}. \end{aligned} \tag{4.12}$$

Setting $z = Bu$ and noting that on $\delta_{m(i)}$

$$(z - \frac{1}{2}B\tilde{B}^T z)_{|\delta_{m(i)}} = \frac{1}{2}(z_i - B_{ij}\tilde{B}_{ij}^T z_i),$$

we have

$$|z - \frac{1}{2}B\tilde{B}^T z|^2_{\ell_2} = \frac{1}{4}|z_i - B_{ij}\tilde{B}_{ij}^T z_i|^2_{\ell_2}.$$

Let $g \equiv \tilde{B}_{ij}^T z_i$. We note that $z_i = \pi(z_i, 0)$ on $\delta_{m(i)}$. Using that

$$\begin{aligned} |z_i - B_{ij}g|^2_{\ell_2} &\leq \frac{C}{h_{\delta_{m(i)}}} \parallel z_i - \pi_{\delta_{m(i)}}(g,0) \parallel^2_{L^2(\delta_{m(i)})} = \\ &= \frac{C}{h_{\delta_{m(i)}}} \parallel \pi_{\delta_{m(i)}}(z_i - g, 0) \parallel^2_{L^2(\delta_{m(i)})} \leq \\ &\leq \frac{C}{h_{\delta_{m(i)}}} \parallel z_i - g \parallel^2_{L^2(\delta_{m(i)})}, \end{aligned} \tag{4.13}$$

in view of the L_2 - stability of $\pi_{\delta_{m(i)}}$; see [1].

The question is now how to estimate the right hand side of (4.13). We do that as follows. Let $\bar{z}_i$ be a piecewise constant function on $\delta_{m(i)}$ with respect to the triangulation on $\delta_{m(i)}$ and with values $z_i(x_k)$ at $x_k \in \delta_{m(i)h}$, the set of nodal points on $\delta_{m(i)}$. Using this, we get

$$\frac{1}{h_{\delta_{m(i)}}} \parallel z_i - g \parallel^2_{L^2(\delta_{m(i)})} \leq \frac{2}{h_{\delta_{m(i)}}} \parallel \bar{z}_i - g \parallel^2_{L^2(\delta_{m(i)})} + C|z_i|^2_{S_{\delta_{m(i)}}}, \tag{4.14}$$

since

$$\parallel z_i - \bar{z}_i \parallel^2_{L^2(\delta_{m(i)})} \leq Ch_{\delta_{m(i)}} \parallel z_i \parallel^2_{H^{1/2}_{00}(\delta_{m(i)})} \leq Ch_{\delta_{m(i)}}|z_i|^2_{S_{\delta_{m(i)}}}, \tag{4.15}$$

in view of a known estimate and Lemma 2.

There remains to prove that

$$\frac{1}{h_{\delta_{m(i)}}} \parallel \bar{z}_i - g \parallel^2_{L^2(\delta_{m(i)})} \leq C|z|^2_{S_{\delta_{m(i)}}}. \tag{4.16}$$

We do this as follows. Let $\bar{g}_\gamma$ be a piecewise constant function on $\gamma_{m(j)}$ with respect to the triangulation on $\gamma_{m(j)}$ and with values $g(x_k) = (\tilde{B}_{ij}^T z_i)_k$ at $x_k \in \gamma_{m(j)h}$, the set of nodal points on $\gamma_{m(j)}$. We have,

$$\frac{1}{h_{\delta_{m(i)}}} \parallel \bar{z}_i - g \parallel^2_{L^2} \leq \frac{2}{h_{\delta_{m(i)}}} \{ \parallel \bar{z}_i - \bar{g}_\gamma \parallel^2_{L^2} + \parallel g - \bar{g}_\gamma \parallel^2_{L^2} \}. \tag{4.17}$$

It is known that

$$\| g - \bar{g}_\gamma \|^2_{L^2(\gamma_{m(j)})} \leq C h_{\gamma_{m(j)}} \| g \|^2_{H^{1/2}_{00}(\gamma_{m(j)})} .$$

On the other hand,

$$\| g \|^2_{H^{1/2}_{00}(\gamma_{m(j)})} \leq C |z_i|^2_{S_{\delta_{m(i)}}},$$

in view of (4.6). Hence,

$$\frac{1}{h_{\delta_{m(i)}}} \| g - \bar{g}_\gamma \|^2_{L(\delta_{m(i)})} \leq C \frac{h_{\gamma_{m(j)}}}{h_{\delta_{m(i)}}} |z_i|^2_{S_{\delta_{m(i)}}} \leq C |z_i|^2_{S_{\delta_{m(i)}}}. \tag{4.18}$$

We now estimate $h^{-1}_{\delta_{m(i)}} \| \bar{z}_i - \bar{g}_\gamma \|^2_{L^2}$ of (4.17) as follows. We have

$$\| \bar{z}_i - \bar{g}_\gamma \|^2_{L^2(\delta_{m(i)})} = \sup_\varphi \frac{|(\bar{z}_i - \bar{g}_\gamma, \varphi)_{L^2}|^2}{\| \varphi \|^2_{L^2}}. \tag{4.19}$$

Let $Q_\delta \varphi$ and $Q_\gamma \varphi$ be the L_2 - projections on the spaces of piecewise constant functions on the triangulations of $\delta_{m(i)}$ and $\gamma_{m(j)}$, respectively. It is known that,

$$\| z_i - Q_\delta z_i \|^2_{L^2(\delta_{m(i)})} \leq C h_{\delta_{m(i)}} |z_i|^2_{H^{1/2}_{00}(\delta_{m(i)})}$$

and

$$\| z_i - Q_\gamma z_i \|^2_{L^2(\gamma_{m(j)})} \leq C h_{\gamma_{m(j)}} |z_i|^2_{H^{1/2}_{00}(\gamma_{m(j)})}.$$

Using the projections, we have

$$(\bar{z}_i - \bar{g}_\gamma, \varphi)_{L^2(\delta_{m(i)})} = (\bar{z}_i, Q_\delta \varphi)_{L^2(\delta_{m(i)})} - (\bar{g}_\gamma, Q_\gamma \varphi)_{L^2(\gamma_{m(j)})}. \tag{4.20}$$

We note that

$$(\bar{g}_\gamma, Q_\gamma \varphi)_{L^2(\gamma_{m(j)})} = h_{\gamma_{m(j)}} \sum_{x_k \in \gamma_{m(j)h}} g_\gamma(x_k)(Q_\gamma \varphi)(x_k) =$$

$$= \alpha^{(m)}_{ij} h_{\gamma_{m(j)}} (B^T_{ij} z_i, Q_\gamma \varphi)_{\ell_2} =$$

$$= h_{\delta_{m(i)}} (z_i, B_{ij} Q_\gamma \varphi)_{\ell_2} = (\bar{z}_i, \overline{B_{ij} Q_\gamma \varphi})_{L^2(\delta_{m(i)})},$$

where $\overline{B_{ij} Q_\gamma \varphi}$ is a piecewise constant function with respect to the $\delta_{m(i)}$ triangulation. Using this in (4.20), we have

$$(\bar{z}_i - \bar{g}_\gamma, \varphi)_{L^2(\delta_{m(i)})} = (\bar{z}_i, Q_\delta \varphi - \overline{B_{ij} Q_\gamma \varphi})_{L^2(\delta_{m(i)})}.$$

Hence,

$$\begin{aligned} (\bar{z}_i - \bar{g}_\gamma, \varphi)_{L^2(\delta_{m(i)})} &\leq \| z_i - \bar{z}_i \|_{L^2} \| Q_\delta \varphi - \overline{B_{ij} Q_\gamma \varphi} \|_{L^2} + \\ &+ \| z_i \|_{H^{1/2}_{00}(\delta_{m(i)})} \| Q_\delta \varphi - \overline{B_{ij} Q_\gamma \varphi} \|_{H^{-1/2}(\delta_{m(i)})} . \end{aligned} \tag{4.21}$$

We note that $\overline{B_{ij} Q_\gamma \varphi} = \overline{\pi_{\delta_{m(i)}}(Q_\gamma \varphi, 0)}$. Using that, we have

$$\| Q_\delta \varphi - \overline{B_{ij} Q_\gamma \varphi} \|_{H^{-1/2}(\delta_{m(i)})} \leq \| Q_\delta \varphi - \varphi \|_{H^{-1/2}(\delta_{m(i)})} +$$

$$\| \varphi - \pi_{\delta_{m(i)}}(Q_\gamma\varphi, 0) \|_{H^{-1/2}(\delta_{m(i)})} + \| \pi_{\delta_{m(i)}}(Q_\gamma\varphi, 0) - \overline{\pi_{\delta_{m(i)}}(Q_\gamma\varphi, 0)} \|_{H^{-1/2}(\delta_{m(i)})} .$$

Using known estimates for these terms, we get

$$\| Q_\delta\varphi - \overline{B_{ij}Q_\gamma\varphi} \|^2_{H^{-1/2}(\delta_{m(i)})} \leq C(h_{\delta_{m(i)}} + h_{\gamma_{m(j)}}) \| \varphi \|^2_{L^2(\delta_{m(i)})} . \tag{4.22}$$

It is easy to see that

$$\| Q_\delta\varphi - \overline{B_{ij}Q_\gamma\varphi} \|^2_{L^2(\delta_{m(i)})} \leq C \| \varphi \|^2_{L^2(\delta_{m(i)})} . \tag{4.23}$$

Using the estimates (4.22), (4.23), and (4.15) in (4.21), we get

$$(\bar{z}_i - \bar{g}_\gamma, \varphi)_{L^2(\delta_{m(i)})} \leq Ch_{\delta_{m(i)}} \| z_i \|_{H^{1/2}_{00}(\delta_{m(i)})} \| \varphi \|_{L^2(\delta_{m(i)})} .$$

In turn, substituting this into (4.19), we have

$$\| \bar{z}_i - \bar{g}_\gamma \|^2_{L^2(\delta_{m(i)})} \leq Ch_{\delta_{m(i)}} \| z_i \|^2_{H^{1/2}_{00}(\delta_{m(i)})} \leq Ch_{\delta_{m(i)}} |z_i|^2_{S_{\delta_{m(i)}}} .$$

Using this and (4.18) in (4.17) and the resulting inequality in (4.14), we get

$$\frac{1}{h_{\delta_{m(i)}}} \| z_i - g \|^2_{L^2(\delta_{m(i)})} \leq C|z_i|^2_{S_{\delta_{m(i)}}} .$$

In turn, using this estimate in (4.13) and the resulting inequality in (4.12), we have

$$|(B\tilde{B}^T)^{-1}Bu - 1/2Bu|^2_{S_{\delta_{m(i)}}} \leq C|Bu|^2_{S_{\delta_{m(i)}}} \leq C\{|u_i|^2_{S_{\delta_{m(i)}}} + |u_j|^2_{S_{\gamma_{m(j)}}}\};$$

we have also used (4.11). Using this and again (4.11) in (4.9) and the resulting inequality in (4.8), we get, cf. (4.4),

$$|Pw_r|^2_{S^{(i)}} \leq C\{ \sum_{\delta_{m(i)} \subset \partial\Omega_i} |u_i|^2_{S_{\delta_{m(i)}}} + \sum_{\gamma_{m(i)} = \delta_{m(j)}} |u_j|^2_{S_{\gamma_{m(j)}}} \}, \tag{4.24}$$

where the second sum is taken over $\gamma_{m(i)} \subset \Omega_i$. It is known that for $u = w - I^H w$ we have

$$|u_i|^2_{S_{\delta_{m(i)}}} \leq C(1 + \log(H/h))^2 |w_i|^2_{S_i}$$

Using this in (4.24) and summing the resulting inequality with respect i, we get (4.1), in view of (4.2). The proof is complete.

Lemma 2 Let $h_{\delta_{m(i)}} \sim h_{\gamma_{m(j)}}$. Then for $u \in W(\delta_{m(i)})$, which vanishes at the ends of $\delta_{m(i)}$ the following hold:

$$C_0 < S_{\delta_{m(i)}}u, u >_{\ell_2} \leq \| u \|^2_{H^{1/2}(\delta_{m(i)})} \leq C_1 < S_{\delta_{m(i)}}u, u >_{\ell_2} . \tag{4.25}$$

and

$$C_2 h^2_{\delta_{m(i)}} < S^{-1}_{\delta_{m(i)}}u, u > \leq \| u \|^2_{H^{-1/2}(\delta_{m(i)})} \leq C_3 h^2_{\delta_{m(i)}} < S^{-1}_{\delta_{m(i)}}u, u > \tag{4.26}$$

where C_i are positive constants independent of $h_{\delta_{m(i)}}$.

Proof The proof of (4.25) can be found for example in [3]. The proof of (4.26) follows from Proposition 7.5 in [10].
Cororally (see the proof of Lemma 1 in [4])

$$|B_{ij}t|^2_{S^{-1}_{\delta_{m(i)}}} \leq C|t|^2_{S^{-1}_{\gamma_{m(j)}}}. \tag{4.27}$$

Proof Let $\pi_{\delta_{m(i)}}(t,0)$ correspond to $B_{ij}t$ on $\delta_{m(i)}$ where t is a piecewise linear continuous function, also denoted by t, and defined by the vector t. Using (4.26), we have

$$h^2_{\delta_{m(i)}}|B_{ij}t|^2_{S^{-1}_{\delta_{m(i)}}} \leq C \parallel \pi_{\delta_{m(i)}}(t,0) \parallel^2_{H^{-1/2}(\delta_{m(i)})}. \tag{4.28}$$

We show below that

$$\parallel \pi_{\delta_{m(i)}}(t,0) \parallel^2_{H^{-1/2}(\delta_{m(i)})} \leq C(1+\frac{h_{\delta_{m(i)}}}{h_{\gamma_{m(j)}}}) \parallel t \parallel^2_{H^{-1/2}(\gamma_{m(j)})}. \tag{4.29}$$

Using this in (4.28), that $h_{\delta_{m(i)}} \sim h_{\gamma_{m(j)}}$, and (4.26), we get (4.27).

There remains to prove (4.29). We have

$$\begin{aligned} \parallel \pi_{\delta_{m(i)}}(t,0) \parallel_{H^{-1/2}(\delta_{m(i)})} \leq & \parallel t \parallel_{H^{-1/2}(\delta_{m(i)})} \\ & + \parallel \pi_{\delta_{m(i)}}(t,0) - t \parallel_{H^{-1/2}(\delta_{m(i)})} \end{aligned} \tag{4.30}$$

and

$$\begin{aligned} & \parallel \pi_{\delta_{m(i)}}(t,0) - t \parallel_{H^{-1/2}(\delta_{m(i)})} = \\ & = \max_g \frac{(\pi_{\delta_{m(i)}}(t,0) - t, g - Q_{\delta_{m(i)}}g)_{L^2(\delta_{m(i)})}}{\parallel g \parallel_{H^{1/2}_{00}(\delta_{m(i)})}}. \end{aligned} \tag{4.31}$$

Here $Q_{\delta_{m(i)}}$ is the L_2 orthogonal projection onto the mortar space $M(\delta_{m(i)})$. Using a known estimate for $g - Q_{\delta_{m(i)}}g$, the L_2 - stability of $\pi_{\delta_{m(i)}}$, and an inverse inequality, we get

$$\parallel \pi_{\delta_{m(i)}}(t,0) - t \parallel_{H^{-1/2}(\delta_{m(i)})} \leq C(\frac{h_{\delta_{m(i)}}}{h_{\gamma_{m(i)}}})^{1/2} \parallel t \parallel_{H^{-1/2}(\delta_{m(i)})}.$$

Using this bound in (4.30), we get (4.29). The proof is complete.
We now in the position to formulate and prove the main result.

Theorem 1 Let the assumptions of Lemma 1 be satisfied. Then for $\lambda \in V = Im(B)$

$$< \tilde{M}\lambda, \lambda > \leq < F\lambda, \lambda > \leq C(1+\log(H/h))^2 < \tilde{M}\lambda, \lambda > \tag{4.32}$$

holds, where C is independent of h and H.

Proof The right hand side of (4.32): We have, cf. [9],

$$< F\lambda, \lambda > = \max_{w_r \in W_r} \frac{| < \lambda, Bw_r > |^2}{|w_r|^2_{\tilde{S}}}.$$

Using Lemma 1, we get

$$< F\lambda, \lambda > \leq C(1+\log(H/h))^2 \max_{w_r} \frac{| < \lambda, Bw_r > |^2}{|Pw_r|^2_{S_{rr}}},$$

where $P = \tilde{B}^T(B\tilde{B}^T)^{-1}B$. In turn, by straightforward manipulations, see also (3.15), we have

$$< F\lambda, \lambda > \leq C(1+\log(H/h))^2 \max_{w_r} \frac{| < \lambda, Bw_r > |^2}{< (B\tilde{B}^T)^{-1}\tilde{B}S_{rr}\tilde{B}^T(B\tilde{B}^T)^{-1}Bw_r, Bw_r >} =$$

$$= C(1+\log(H/h))^2 \max_{w_r} \frac{| < \tilde{M}^{1/2}\lambda, \tilde{M}^{-1/2}Bw_r > |^2}{< \tilde{M}^{-1/2}Bw_r, \tilde{M}^{-1/2}Bw_r >} =$$

$$= C(1+\log(H/h))^2 < \tilde{M}\lambda, \lambda >$$

This proves the right hand side of (4.32).

The left hand side of (4.32): We have, cf. [9],

$$< F\lambda, \lambda > = \| \tilde{S}^{-1/2}B^T\lambda \|^2 = \max_v \frac{| < \lambda, Bv > |^2}{\| \tilde{S}^{1/2}v \|^2}.$$

Taking $v \in$ range (P) and using that $v = Pv$, and (4.2), we get

$$< F\lambda, \lambda > \geq \max_v \frac{< \lambda, Bv >}{< Pv, Pv >_{S_{rr}}}.$$

Setting $\mu = Bv$ and using the definition of P, we have

$$< F\lambda, \lambda > \geq \max_\mu \frac{| < \lambda, \mu > |^2}{< \tilde{M}^{-1}\mu, \mu >} =$$

$$= \max_\mu \frac{| < \tilde{M}^{1/2}\lambda, \tilde{M}^{-1/2}\mu > |^2}{< \tilde{M}^{-1/2}\mu, \tilde{M}^{-1/2}\mu} = < \tilde{M}\lambda, \lambda > .$$

This proves the left-hand side of (4.32). The proof of Theorem 1 is complete.

Acknowledgments: The work of the authors was supported in part by the National Science Foundation under Grant NSF - CCR - 9732208 and that of the first author also in part by the Polish Science Foundation under Grant 2 P03A 02116.

REFERENCES

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

[2] C. Bernardi, Y. Maday, and A. T. Patera. A new non conforming approach to domain decomposition: The mortar element method. In H. Brezis and J.-L. Lions, editors, *Collège de France Seminar*. Pitman, 1994. This paper appeared as a technical report about five years earlier.

[3] P. E. Bjørstad and O. B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23(6):1093–1120, 1986.

[4] M. Dryja and O. B. Widlund. A feti-dp for mortar discretization of elliptic problems. In *Proceedings ETH Zurich Workshop on Doman Decomposition Method. June 2001.Lecture Notes in Computational Science and Engineering*. Springer Verlag, 2002. to appear.

[5] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.

[6] A. Klawonn, O. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. Technical Report 815, Courant Institute of Mathematical Sciences, Department of Computer Science, April 2001. to appear in SIAM J. Numer. Anal.

[7] A. Klawonn and O. B. Widlund. FETI and Neumann–Neumann Iterative Substructuring Methods: Connections and New Results. *Comm. Pure Appl. Math.*, 54:57–90, January 2001.

[8] C. Lacour. Iterative substructuring preconditioner for the mortar finite element method. In P. E. Bjørstad, M. Espedal, and D. Keyes, editors, *Domain Decomposition Methods in Sciences and Engineering*. John Wiley & Sons, 1997. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.

[9] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.

[10] P. Peisker. On the numerical solution of the first biharmonic equation. *RAIRO Mathematical Modelling and Numerical Analysis*, 22(4):655–676, 1988.

[11] D. Stefanica and A. Klawonn. The FETI method for mortar finite elements. In C.-H. Lai, P. Bjorstad, M. Cross, and O.Widlund, editors, *Proceedings of 11th International Conference on Domain Decomposition Methods*, pages 121–129. DDM.org, 1999.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

4. Direct Domain Decomposition using the Hierarchical Matrix Technique

W. Hackbusch[1]

1. Introduction. In the time when the domain decomposition technique developed, direct solvers were quite common. We come back to direct methods; however, the term "direct" has another meaning. The usual understanding of a direct method is:

> Given a matrix A and a vector b,
> produce the solution x of $Ax = b$.

Here, we require a much more mighty procedure:

> Given a matrix A and a vector b,
> approximate the inverse A^{-1} and $x = \left(A^{-1}\right) * b$.

The technique of hierarchical matrices allows to perform this step with an almost optimal storage and operation cost $\tilde{O}(n)$ for $n \times n$ matrices related to elliptic operators. The symbol $\tilde{O}(n)$ means the order $O(n)$ up to a logarithmic factor, i.e., there is a (small) number α such that

$$\tilde{O}(n) = O\left(n \log^{\alpha} n\right).$$

In Section 2, we describe the underlying problem of a non-overlapping domain decomposition and the corresponding system of equations. It is interesting to remark that

- rough (exterior and) interior boundaries are allowed, i.e., no smoothness conditions on the subdomains or the interior boundary (skeleton) are necessary.
- inside of the subdomains, L^{∞}-coefficients are allowed (i.e., jumping coefficients, oscillatory coefficients, etc.). There is no need to place the skeleton along jump lines. A proof concerning robustness against rough boundaries and non-smooth coefficients is given in [1]. If, however, it happens that the coefficients are piecewise constant or analytic in the subdomains, a further improvement is possible using a new technique of Khoromskij and Melenk [7] (see Subsection 3.2).

The two items mentioned above allow to create the subdomains independent of smoothness considerations; instead we may use load balancing arguments.

Further advantages will be mentioned in the Section 3, where the direct solution is explained.

The basic of the solution method are the hierarchical matrices which are already described in several papers (cf. [4], [5]). An introduction is given in [2]. We give an outline of the method in Section 4.

[1]Max-Planck-Institut *Mathematik in den Naturwissenschaften*, wh@mis.mpg.de

Although the method of hierarchical matrices could be applied immediately to the global problem, the domain decomposition helps to achieve a parallelisation of the solution process. The details are discussed in Section 5.

We conclude this contribution in Section 6 with numerical example for the inversion of finite element stiffness matrices. We take an example with extremely non-smooth coefficients to support the remark above.

Figure 1.1: Domain decomposition with non-smooth interfaces

2. Non-Overlapping Domain Decomposition. Let the domain Ω be decomposed into p non-overlapping subdomains Ω_i, $i = 1, \ldots, p$ (cf. Figure 1.1). The skeleton Σ consists of the interior interfaces:

$$\Sigma := \left(\bigcup\nolimits_{i=1}^{p} \partial\Omega_i \right) \setminus \partial\Omega.$$

For a simpler finite element realisation we may assume (in 2D) that Ω_i are polygons. Then Σ is a union of straight lines. In 3D, Σ may consist of flat faces. As mentioned in the introduction, there is no need for Ω_i to form a regular macro element. Later, we will assume that all Ω_i contain a comparable number of degrees of freedom to achieve a load balance in the parallelisation process.

Let I_i be the index set of interior degrees of freedom in Ω_i (the precise definition of $j \in I_i$ is that the corresponding basis function b_j satisfies[2] $\operatorname{supp}(b_j) \subset \overline{\Omega_i}$). All remaining indices are associated with the skeleton and its set is denoted by I_Σ. Hence, we arrive at the decomposition of the global index set I into

$$I = I_1 \cup \cdots \cup I_p \cup I_\Sigma \qquad \text{(disjoint union)}.$$

As usual, the total dimension is denoted by

$$n := \#I. \tag{2.1}$$

[2]Note that by definition the support $\operatorname{supp}(b_j)$ is always in $\overline{\Omega}$, also if the nodal point lies on $\partial\Omega$.

The FE system $Au = f$ has the structure

$$\begin{bmatrix} A_{11} & O & \cdots & O & A_{1,\Sigma} \\ O & A_{22} & \cdots & O & A_{2,\Sigma} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ O & O & \cdots & A_{pp} & A_{p,\Sigma} \\ A_{\Sigma,1} & A_{\Sigma,2} & \cdots & A_{\Sigma,p} & A_{\Sigma\Sigma} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_p \\ u_\Sigma \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_p \\ f_\Sigma \end{bmatrix} \tag{2.2}$$

when we order the unknowns in the sequence $I_1, \ldots, I_p, I_\Sigma$.

As usual in domain decomposition, we assume that besides A all submatrices A_{ii} are invertible, i.e., the subdomain problems are solvable.

In the case of a non-matching domain decomposition (mortar FEM), the elimination of all slave nodes by means of the mortar condition yields again the system (2.2), where I_Σ is the index set of all mortar nodes (cf. [3]).

3. Direct Solution Process.

3.1. Description of Single Steps. The system (2.2) can be reduced to the Schur complement equation

$$Su_\Sigma = g_\Sigma, \tag{3.1}$$

where

$$S := A_{\Sigma\Sigma} - \sum_{i=1}^{p} A_{\Sigma,i} A_{ii}^{-1} A_{i,\Sigma}, \tag{3.2}$$

$$g_\Sigma := f_\Sigma - \sum_{i=1}^{p} A_{\Sigma,i} A_{ii}^{-1} f_i. \tag{3.3}$$

The remaining variables u_i are the result of

$$u_i := A_{ii}^{-1} \left(f_i - A_{i,\Sigma} u_\Sigma \right) \qquad \text{for } i = 1, \ldots, p. \tag{3.4}$$

An obvious solution method which is usually not used because one is afraid of the bad complexity up to $O(n^3)$ of standard solvers is the following:

Step 1a	produce the inverse matrix A_{ii}^{-1},
Step 1b	form the products $A_{\Sigma,i} * \left(A_{ii}^{-1}\right)$ and $\left(A_{\Sigma,i} A_{ii}^{-1}\right) * A_{i,\Sigma}$,
Step 1c	compute the vectors $\left(A_{\Sigma,i} A_{ii}^{-1}\right) * f_i$,
Step 2a	form the sum $S = A_{\Sigma\Sigma} - \sum_{i=1}^{p} \left(A_{\Sigma,i} A_{ii}^{-1} A_{i,\Sigma}\right)$,
Step 2b	compute the vector $g_\Sigma = f_\Sigma - \sum_{i=1}^{p} \left(A_{\Sigma,i} A_{ii}^{-1} f_i\right)$,
Step 3a	produce the inverse matrix S^{-1},
Step 3b	compute the vector $u_\Sigma = \left(S^{-1}\right) * g_\Sigma$,
Step 4	compute the vectors $u_i = \left(A_{ii}^{-1}\right) * \left[f_i - A_{i,\Sigma} * u_\Sigma\right]$.

Terms in round brackets are already computed quantities. The necessary operations are indicated by $\circ^{-1}$, $*$, $-$, $\sum$.

In the sequel we follow the Steps 1-4 with the following modifications: Steps 1a,1b,2a,3a are performed only approximately up to an error ε. Usually[3], one wants ε to be similar to the discretisation error, i.e.,

$$\varepsilon = O(h^\kappa) = O(n^{-\beta}), \tag{3.5}$$

where h is the step size (if there is a quasi-uniform one) and κ is the consistency order. Then $\beta = \kappa/d$ holds, where d is the spatial dimension:

$$\Omega \subset \mathbb{R}^d. \tag{3.6}$$

In the non-uniform finite element case, one expects an discretisation error $O(n^{-\beta})$ for an appropriate triangulation. In that case ignore the middle term in (3.5).

Allowing approximation errors of order $O(\varepsilon)$, the technique of hierarchical matrices explained in the next section will be able to perform all Steps 1a-4 with storage and computer time of order $\tilde{O}(n)$. Hence, the costs are similar to usual iterative DD methods. One of the advantages of the direct method is its robustness and the relative easy implementation. To be precise: It is not so simple to implement the hierarchical matrix method for the first time, but as soon as one has programmed this method, it can be used without modification for different FE applications as well as for the Schur equation $Su_\Sigma = g_\Sigma$ with the (fully populated) matrix S.

Finally we remark that A^{-1} can be computed in **Step 5**:

$$A^{-1} = \begin{bmatrix} \ddots & \vdots & \cdots & \vdots \\ \cdots & \delta_{ij}A_{ii}^{-1} + A_{ii}^{-1}A_{i,\Sigma}S^{-1}A_{\Sigma,j}A_{jj}^{-1} & \cdots & -A_{ii}^{-1}A_{i,\Sigma}S^{-1} \\ & \vdots & \ddots & \vdots \\ \cdots & -S^{-1}A_{\Sigma,j}A_{jj}^{-1} & \cdots & S^{-1} \end{bmatrix}.$$

However, we should make use of the representation by

$$\begin{aligned} A^{-1} = & \begin{bmatrix} A_{11}^{-1} & O & O & O \\ O & \ddots & O & O \\ O & O & A_{pp}^{-1} & O \\ O & O & O & O \end{bmatrix} \\ & + \begin{bmatrix} A_{11}^{-1}A_{1,\Sigma} \\ \vdots \\ A_{pp}^{-1}A_{p,\Sigma} \\ -I \end{bmatrix} \begin{bmatrix} S^{-1}A_{\Sigma,1}A_{11}^{-1} & \cdots & S^{-1}A_{\Sigma,p}A_{pp}^{-1} & -S^{-1} \end{bmatrix}. \end{aligned} \tag{3.7}$$

3.2. Improvement for Piecewise Smooth Coefficients. We mentioned that the approach from above works also efficiently if the subproblems corresponding to the FE matrices A_{ii} involve non-smooth coefficients of the elliptic differential equation. If, on the other hand, we know that the coefficients in one subdomain are constant (or

[3]If one performs only the Steps 1a,1b,2a and 3a to get a rough approximation of S^{-1} for the purpose of preconditioning, ε may be of fixed order $O(1)$, e.g., $\varepsilon = 1/10$.

analytic), one can exploit this fact by applying a more appropriate finite element discretisation. In [7] a so-called *boundary concentrated finite element method* is described which allows to solve the local problem with a number of unknowns proportional to $\text{area}(\partial\Omega_i)/h^{d-1}$. This number is usually smaller by a factor of h than the number of degrees of freedom in a classical FEM, although the same resolution is obtained at the boundary.

We do not discuss this modification in the subdomains in the following, i.e., we consider a traditional FEM.

4. Hierarchical Matrices. It is to be remarked that the method of hierarchical matrices does not apply to any matrix but only to those related to elliptic (pseudo-) differential operators. In our application, A_{ii} as well as their inverse matrices are related to the local elliptic problem, while S is a nicely behaving pseudo-differential operator composed from local Steklov operators. Nevertheless, the method is of black-box character since its description does not depend on specific features of the involved matrices. The success of this kind of approximation depends only on the ellipticity properties.

In the following, we give an introduction into the definition and construction of $\mathcal{H}$-matrices. The interested reader will find more details in [4], [5] and [2].

4.1. The Main Ingredients. We have to introduce

1. the index set I and the geometric properties of its indices;
2. the cluster tree $T(I)$;
3. the block-cluster tree $T(I \times I)$;
4. the admissibility criterion;
5. the (minimal admissible) partitioning of the matrix;
6. rank-k matrices;
7. the definition of an $\mathcal{H}$-matrix;
8. the (approximations of the) operations $A + B$, $A * B$, A^{-1};
9. the estimates for the storage and operation costs.

First, we give a preview of these topics. The *cluster tree* $T(I)$ describes how the whole index set can be partitioned into smaller pieces, which are needed, e.g., when we want to define a subblock of a vector. The *block-cluster tree* $T(I \times I)$ does the same for the matrix. Among the blocks contained in $T(I \times I)$ we can choose a collection of disjoint blocks covering $I \times I$. Then we get a *partitioning* of the matrix into various blocks. An example is given in Fig. 4.1.

The choice of this partitioning P is the essential part. It should contain as few blocks as possible to make the costs as low as possible. On the other hand, the approximation error must be sufficiently small. For this purpose, all blocks have to satisfy an *admissibility condition*. Then, filling all blocks (e.g., in Fig. 4.1) by matrices of rank smaller or equal some k, we obtain an $\mathcal{H}$-matrix from the class $\mathcal{H}(P, k)$. The results of $A + B$, $A * B$, A^{-1} for $A, B \in \mathcal{H}(P, k)$ will, in general, not be again in $\mathcal{H}(P, k)$, but they can be approximated in this class by a cost of $\mathcal{O}(n)$.

4.2. The Index Set and the Geometrical Data. As input for the algorithm we only need the description of the index set I (e.g., $\{1, \ldots, n\}$ or list of nodal points, etc.) and a characteristic subset $X(i) \subset \mathbb{R}^d$ associated with $i \in I$. For a collocation

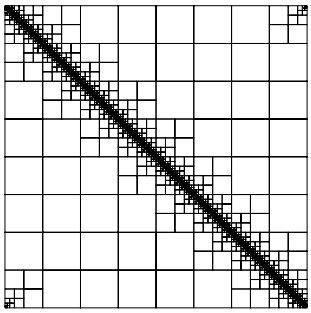


Figure 4.1: Block partitioning P for the unit circle

method, this may be the nodal point, i.e., $X(i) = \{x_i\}$. The appropriate choice for a Galerkin method is

$$X(i) = \operatorname{supp}(\phi_i), \text{ where } \phi_i \text{ is the FE-basis function associated with } i \in I. \quad (4.1)$$

4.3. The Cluster Tree $T(I)$. Formally, the cluster tree $T(I)$ has to satisfy

1. $T(I) \subseteq \mathcal{P}(I)$ (i.e., each node of $T(I)$ is a subset of the index set I).
2. I is the root of $T(I)$.
3. If $\tau \in T(I)$ is a leaf, then $\#\tau = 1$ (i.e., the leaves consist only one index, $\tau = \{i\}$).
4. If $\tau \in T(I)$ is *not* a leaf, then it has exactly two sons and is their disjoint union.

All nodes of $T(I)$ are called "clusters". For each $\tau \in T(I)$, we denote the set of its sons by $S(\tau) \subset T(I)$.

In practice, the condition $\#\tau = 1$ is replaced by $\#\tau \leq C_T$, e.g., with $C_T = 32$. The condition for a binary tree ("exactly two sons") in Part 4 can easily be generalised, although a binary tree is quite reasonable.

The sets $X(i)$ introduced above can immediately be generalised to all clusters by

$$X(\tau) = \bigcup_{i \in \tau} X(i) \subset \mathbb{R}^d \qquad \text{for all } \tau \in T(I). \quad (4.2)$$

Using the Euclidean metric in $\mathbb{R}^d$, we define the *diameter* of a cluster and the *distance* of a pair of clusters:

$$\begin{aligned} \operatorname{diam}(\tau) &= \sup\{|x-y| : x, y \in X(\tau)\} && \text{for } \tau \in T(I), \\ \operatorname{dist}(\tau, \sigma) &= \inf\{|x-y| : x \in X(\tau), y \in X(\sigma)\} && \text{for } \tau, \sigma \in T(I). \end{aligned}$$

The practical construction of $T(I)$ must take care that the clusters are as compact as possible, i.e., $\operatorname{diam}(\tau)$ should be as small as possible for a fixed number $\#\tau$ of indices. One possible construction is the recursive halving of bounding boxes as illustrated in Fig. 4.2. Note that this procedure applies to any irregular FE-triangulation in any spatial dimension.

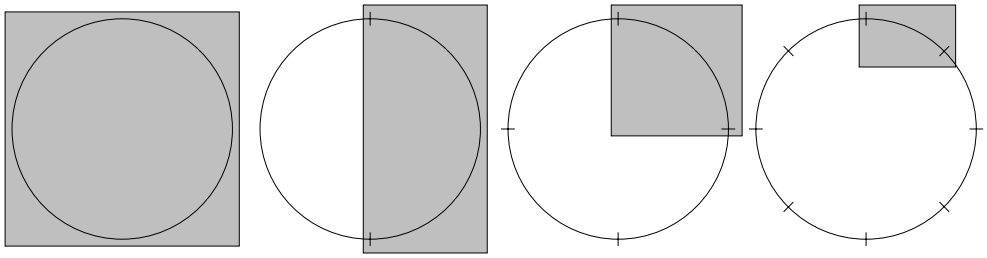


Figure 4.2: Dyadic clustering of the unit circle.

4.4. The Block-Cluster Tree $T(I\times I)$. The tree $T(I\times I)$ is completely defined by means of $T(I)$ when we use the following canonical choice. Let $I \times I$ belong to $T(I \times I)$. For all $\tau \times \sigma \in T(I \times I)$ with τ and σ not being leaves, assign the sons $\tau' \times \sigma'$ to $T(I \times I)$, where $\tau' \in S(\tau)$ and $\sigma' \in S(\sigma)$. Again, we write $S(\tau \times \sigma)$ for the set of sons of $\tau \times \sigma$.

Remark 4.1 *a) If $T(I)$ is a binary tree (as described by condition 4 from above), then $T(I \times I)$ is quad-tree.*

b) All "blocks" or "block-clusters" b from $T(I \times I)$ have the product form $b = \tau \times \sigma$ with $\tau, \sigma \in T(I)$. Indices $i \in \tau$ belong the rows of b, while $j \in \sigma$ are column indices.

The set $T(I \times I)$ provides a rich choice of larger and smaller blocks, which we can select to construct the partitioning of Subsection 4.6.

4.5. The Admissibility Condition. Let $b = \tau \times \sigma$ be a block from $T(I \times I)$. If τ or σ is a leaf in $T(I)$ (i.e., $\#\tau = 1$ or $\#\sigma = 1$), then also b is a leaf in $T(I \times I)$. In this case, b is accepted as "admissible". Otherwise, we recall diam and dist defined via (4.2) and require an admissibility condition like

$$\max\left(\operatorname{diam}(\tau), \operatorname{diam}(\sigma)\right) \geq 2\eta \operatorname{dist}(\tau, \sigma), \tag{4.3}$$

where, e.g., η may be chosen as $\frac{1}{2}$. Even the weaker requirement

$$\min\left(\operatorname{diam}(\tau), \operatorname{diam}(\sigma)\right) \geq 2\eta \operatorname{dist}(\tau, \sigma)$$

makes sense. Conditions of this form are known from panel clustering or from matrix compression in the case of wavelet bases.

It turns out that (4.3) is the appropriate condition to ensure that the rank-k matrices introduced below will lead to the desired accuracy.

4.6. The Partitioning. A partitioning of $I \times I$ is a set $P \subset T(I \times I)$, so that all elements (blocks) are disjoint and $I \times I = \cup_{b\in P} b$. The coarsest partitioning is $P = \{I \times I\}$, while the finest one consists of all leaves of $T(I \times I)$. In the first case we consider the matrix as one block, in the latter case each entry forms a one-by-one block.

We say that P is an admissible partitioning, if all $b \in P$ are admissible. The second of the trivial examples is such an admissible partitioning, since by definition

one-by-one blocks are admissible. However, the second example leads to the standard (costly) representation.

To obtain a representation which is as data-sparse as possible but still ensures the desired accuracy, we choose the admissible partitioning with the minimal number of blocks. The construction of this optimal P is as follows. Start with $P := \{I \times I\}$. Since $I \times I$ is definitely not admissible, we divide it into its sons $s \in S(I \times I)$ and replace $I \times I$ by the sons: $P := (P \setminus \{I \times I\}) \cup S(I \times I)$. Similarly, we check for every new $b \in P$, whether it is admissible. If not, $P := (P \setminus \{b\}) \cup S(b)$.

Under mild conditions, one proves that the construction of $T(I)$ by means of bounding boxes explained above, leads to $\#P = \tilde{O}(n)$.

4.7. Rk-Matrices. Except when $\#\tau = 1$ or $\#\sigma = 1$, we represent all block matrices as so-called Rk-matrices represented by $2k$ vectors $a_\iota \in \mathbb{R}^\tau$, $b_\iota^\top \in \mathbb{R}^\sigma$,

$$M = \sum\nolimits_{\iota \neq 1}^{k} a_\iota b_\iota^\top$$

or in matrix form: $M = AB^\top$ with $A \in \mathbb{R}^{\tau,k}$, $B \in \mathbb{R}^{k,\sigma}$. Note that all matrices of rank $\leq k$ can be represented in this form. The storage equals $k * (\#\tau + \#\sigma)$.

4.8. $\mathcal{H}(P,k)$-Matrices. For any partition P and all $k \in \mathbb{N}$, we define

$$\mathcal{H}(P,k) := \left\{ A \in \mathbb{R}^{I \times I} : \operatorname{rank}(A|_b) \leq k \text{ for all } b \in P \right\}$$

as the set of hierarchical matrices for the partitioning P of $I \times I$ and the maximal rank k. Here, $A|_b = (A_{ij})_{(i,j) \in b}$ is the block matrix corresponding to $b \in P$. $A|_b$ is represented as Rk-matrix.

There are generalisations i) where the integer k is replaced by a function $k : P \to \mathbb{N}$ (variable rank) and ii) where the condition $\operatorname{rank}(A|_b) \leq k$ is replaced by the stronger requirement that $A|_b$ belongs to a tensor space $V_\tau \otimes V_\sigma$ with $\min(\dim V_\tau, \dim V_\sigma) = k$ (see [6]).

4.9. $\mathcal{H}$-Matrix Operations. The simplest operation is the matrix-vector operation $(A, x) \mapsto A * x$. Obviously, subblocks of x must be multiplied by $A|_b$ and the partial results are summed up. Since $A|_b$ are Rk-matrices, $A|_b * x|_\sigma$ needs only simple scalar products. The overall cost is $\tilde{O}(n)$.

The addition of two $\mathcal{H}(P,k)$-matrices can be performed blockwise and yields a result in $\mathcal{H}(P,2k)$. Truncating all blocks to rank $\leq k$ (e.g., by means of SVD) gives the approximate result in $\mathcal{H}(P,k)$ with a cost of $\tilde{O}(n)$.

The approximative multiplication of two matrices can be performed recursively exploiting the hierarchical structure of the partitioning P (see [2]). The costs are again $\tilde{O}(n)$.

The block Gauss elimination (of a 2×2 block matrix) allows to reduce the inversion of the whole matrix to the inversion of the first block and Schur complement together with additions and multiplications. This yields a recursive algorithm for computing the inverse matrix approximately with cost $\tilde{O}(n)$.

5. Parallelisation.

5.1. First Approach. We recall the disjoint splitting of I into the subsets $I_1, \ldots, I_p, I_\Sigma$. For the purpose of load balance we assume that p processors are available and that the cardinalities $\#I_i$ $(i = 1, \ldots, p)$ are of similar size, i.e.,

$$\#I_i \sim \frac{n}{p} \qquad (i = 1, \ldots, p)\,. \tag{5.1}$$

The computations in Steps 1a-4 of Section 3 contain three different phases:

Phase I	Steps 1a-c
Phase II	Steps 2a-3b
Phase III	Step 4

Obviously, Phases I and III contain completely independent tasks for each $i = 1, \ldots, p$. Hence, assuming p processors, these phases are parallelisable without any communication. The work cost for each processor is $\tilde{O}(\#I_i) = \tilde{O}(\frac{n}{p})$ according to (5.1).

The summation $\sum_{i=1}^{p}$ in Steps 2a,b needs $\log_2(p)$ steps[4] to collect and add the terms. The computations of the Steps 3a,b are performed on one processor, i.e., no parallelisation is used in Phase II. The cost of Phase II amounts to $\tilde{O}(\#I_\Sigma)$.

In Phase III, u_Σ has to be copied to each processor. Then Step 4 can be performed with a cost of $\tilde{O}(\#I_i) = \tilde{O}(\frac{n}{p})$.

Similarly, the data can be distributed so that all p processors in Phase I,III need $\tilde{O}(\#I_i)$ storage, while the one processor of Phase II requires a storage of $\tilde{O}(\#I_\Sigma)$.

We may add a **Phase IV**, where Step 5 (computation of A^{-1}) is performed. For this purpose, the quantities $A_{ii}^{-1}A_{i,\Sigma}$, $S^{-1}\left(A_{\Sigma,i}A_{ii}^{-1}\right)$ from (3.7) are still to be computed, while $A_{\Sigma,i}A_{ii}^{-1}$ are already known from Step 1b.

In total, the whole computation of the phases I-III leads to a cost of $\tilde{O}(\frac{n}{p})+\tilde{O}(\#I_\Sigma)$. We next assume that subdomains related to I_i are determined such there surface is of minimal order, i.e., the set $I_{\Sigma,i} = \{j \in I_\Sigma :\ j \text{ neighboured to some } k \in I_i\}$ has a cardinality of $O\left((\#I_i)^{(d-1)/d}\right) = O\left((\frac{n}{p})^{(d-1)/d}\right)$. Hence,

$$O(\#I_\Sigma) = O\left(p\left(\frac{n}{p}\right)^{(d-1)/d}\right) = O\left(p^{1/d}n^{(d-1)/d}\right), \tag{5.2}$$

where d is the spatial dimension. Under the assumption (5.2), the work equals $W = \tilde{O}\left(\frac{n}{p} + p^{1/d}n^{(d-1)/d}\right)$. If n is fixed, the optimal number of processors is $p = O\left(n^{1/(d+1)}\right)$ and leads to $W = \tilde{O}\left(n^{d/(d+1)}\right)$. If, alternatively, the number p of processors is given, the right scaling of n yields $n = O\left(p^{d+1}\right)$.

We summarise in

Remark 5.1 *A parallel treatment in the Phases I and III with $p = O(n^{1/(d+1)})$ processors leads to a work $W = \tilde{O}(n^{d/(d+1)})$. The distributed memory requirements are also $\tilde{O}(n^{d/(d+1)})$. A possible Phase IV requires a work and local storage of the same size.*

[4]We remark that the $\log_2(p)$ factor can be ignored because of our definition of $\tilde{O}(\cdot)$.

5.2. Multiple DD Levels in Phase I. In the previous subsection it was assumed that the Steps 1a-c are performed by means of the $\mathcal{H}$-matrix arithmetic. An alternative is to compute A_{ii}^{-1} in Step 1a again by a DD approach using a further subdivision of I_i into $I_{i,j}$ $(j = 1, \ldots, q_i)$ and $I_{i,\Sigma}$. Due to the representation (3.7), the matrix multiplication in Step 1b can be parallel in q_i processors. The vector operation in Step 1c needs $O(p_i)$ communications to add up all partial results. The work needed to perform Steps 1a-c for a particular i is given by Remark 5.1: Under the natural assumptions from above about the subdivision into $I_{i,1}, \ldots, I_{i,q_i}, I_{i,\Sigma}$ and assuming $q_i = O((\#I_i)^{1/(d+1)})$, the work for Phase I is reduced to $\tilde{O}((\#I_i)^{d/(d+1)})$ (instead of $\tilde{O}(\#I_i)$).

Assume (5.1) and $q_i = q = O((\frac{n}{p})^{1/(d+1)})$ for all i, the total work is $W = \tilde{O}\left((\frac{n}{p})^{\frac{d}{d+1}} + p^{1/d} n^{\frac{d-1}{d}}\right)$, which is minimal for $p = n^{\frac{1}{d+1+d^2}}$, when $W = \tilde{O}(n^{\frac{d^2}{d+1+d^2}})$.

Remark 5.2 *a) The parallel two-level DD approach as described above reduces the work and storage to* $\tilde{O}(n^{\frac{d^2}{d+1+d^2}})$, *where* $P = pq = O(n^{\frac{d+1}{d+1+d^2}})$ *processors are used. These exponents are* $\frac{4}{7}$ *and* $\frac{3}{7}$ *in the case of* $d = 2$. *Note that three different kinds of parallelism appear: i) there are* $P = pq$ *problems to be solved in parallel for the index sets* $I_{i,j}$ $(i = 1, \ldots, p$ *and* $j = 1, \ldots, q)$, *ii)* p *tasks on* $I_{i,\Sigma}$, *iii) 1 task on* I_Σ.

b) There is an obvious generalisation to an L*-level DD approach. The exponents for the three-level case are* $W = \tilde{O}(n^{\frac{d^3}{(d+1)(1+d^2)}})$, $P = O(n^{\frac{d+1+d^2}{(d+1)(1+d^2)}})$. *The numbers for* $d = 2$ *and* $L = 3$ *are* $W = \tilde{O}(n^{\frac{8}{15}})$ *and* $P = O(n^{\frac{7}{15}})$. *For general* L, $W = \tilde{O}(n^{\omega_L})$ *and* $P = O(n^{1-\omega_L})$, *where the exponents* ω_L *converges to* $\lim \omega_L = \frac{d-1}{d}$, *i.e.,*

$$W \to \tilde{O}(n^{\frac{d-1}{d}}), \quad P \to O(n^{\frac{1}{d}}).$$

5.3. DD in Phase II. In the previous subsection, we have improved the performance in Phase I, while Phase II (Steps 2a-3b) remains unparallelised. The only side effect was that the number p of subdomains (of the first level) could be chosen smaller so that $\#I_\Sigma$ was decreasing.

Now we also parallelise Phase II, but it turns out that this approach is equivalent with the approach in Subsection 5.2. Consider a non-overlapping domain decomposition of Ω by Ω_i, $i = 1, \ldots, p$, which is organised in a hierarchical way, i.e., there is a coarser decomposition $\hat{\Omega}_k$, $k = 1, \ldots, K$, so that $\hat{\Omega}_k \supset \bigcup_{i \in J_k} \Omega_i$ for disjoint subsets satisfying $\bigcup_{k=1}^{K} J_k = \{1, \ldots, p\}$.

Ω_1	Ω_2	Ω_5	Ω_6
Ω_3	Ω_4	Ω_7	Ω_8
Ω_9	Ω_{10}	Ω_{13}	Ω_{14}
Ω_{11}	Ω_{12}	Ω_{15}	Ω_{16}

Coarse DD (double lines) and fine DD (single lines)

In the picture above, the first coarse subset is $\hat{\Omega}_1$ corresponding to the fine subsets Ω_i for $i \in J_1 = \{1, \ldots, 4\}$. The skeleton $\hat{\Sigma}$ of the coarse domain decomposition (double lines in the picture) is a subset of the skeleton Σ of the fine domain decomposition:

$\hat{\Sigma} \subset \Sigma$. The set $\Sigma \backslash \hat{\Sigma}$ consists of non-connected parts $\Sigma_k \subset \hat{\Omega}_k$, $k = 1, \ldots, K$. The Schur complement system corresponding to Σ has again the structure of system (2.2), where now the sets $I_1, \ldots, I_p, I_\Sigma$ correspond to $\Sigma_1, \ldots, \Sigma_p, \hat{\Sigma}$. Hence, the methods from Subsection 5.1 apply again. The multiple application can be done as in Subsection 5.2.

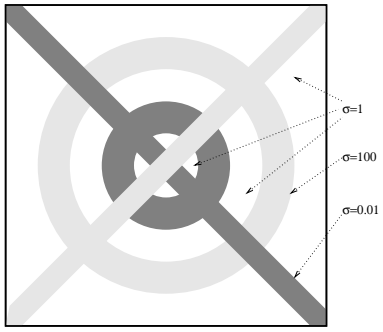
Remark 5.3 *The Schur complement system for the skeleton $\hat{\Sigma}$ from above can (identically) obtained in three different ways: a) eliminate directly all interior nodes in $\hat{\Omega}_k$, $k = 1, \ldots, K$; b) follow Subsection 5.2 and eliminate the interior nodes in $\hat{\Omega}_k$ by means of the secondary domain decomposition by Ω_i, $i \in J_k$; c) compute first the Schur complement system for the finer skeleton Σ and eliminate the nodes from Σ_k, $k = 1, \ldots, K$. The approaches b) and c) differ only in the ordering of the unknowns.*

6. Numerical Example. Since the critical question is the ability to compute the approximate inverse of a FE matrix, we give numerical results for this step. Furthermore, we choose an example with jumping coefficients.

Consider the differential equation

$$\begin{aligned} -\operatorname{div}(\sigma(x)\nabla u(x)) &= f(x) \qquad \text{in } \Omega = [0,1]^2, \\ u &= 0 \qquad \text{on } \Gamma = \partial\Omega, \end{aligned}$$

where the function $\sigma : \mathbb{R}^2 \to \mathbb{R}_{>0}$ defined on Ω has values depicted in the following figure:



$$\sigma(x,y) = \begin{cases} 0.01 & |x+y-1| < 0.05 \text{ or} \\ & (0.1 \le \|(x,y)\| < 0.2) \\ & \text{and } (|x-y| \ge 0.05) \\ 100 & |x-y| < 0.05 \text{ or} \\ & (0.3 \le \|(x,y)\| < 0.4) \\ & \text{and } (|x+y-1| \ge 0.05) \\ 1 & \text{otherwise} \end{cases}$$

We introduce a regular finite element discretisation which leads to the sparse $n \times n$ matrix, where $n \in \{32^2, 64^2, 128^2, 256^2\}$. The inversion algorithm applied to A yields the approximation $A_{\mathcal{H}}^{-1}$. The relative error $\|A^{-1} - A_{\mathcal{H}}^{-1}\|_2 / \|A^{-1}\|_2$ is $\le \|I - A_{\mathcal{H}}^{-1} A\|_2$. The later values are given in

	n = degree of freedom			
k	32^2	64^2	128^2	256^2
1	3.5+1	1.1+2	3.1+2	9.5+2
2	2.4-0	1.7+1	1.3+2	4.3+2
3	6.0-1	3.9-0	1.3+1	5.4+1
4	9.4-2	1.0-0	3.4-0	1.0+1
5	2.6-2	2.8-1	7.6-1	6.6-0
6	1.1-3	7.7-2	2.8-1	1.3-0
7	3.9-5	2.1-2	4.8-2	2.3-1
8	9.6-6	1.3-3	1.6-2	4.2-2
9	7.8-6	4.5-4	3.4-3	6.2-3
10	7.0-7	2.9-4	9.7-4	2.5-3
15	5.1-12	7.9-9	8.3-7	1.6-6
20	5.9-12	2.5-11	4.5-9	6.3-9

Due to the multiplication by A, these values increase with n like $\|I - A_{\mathcal{H}}^{-1}A\|_2 \approx \frac{n}{10} * 0.26^k$, confirming the exponential convergence with respect to the rank k. Note that equal approximation errors are obtained when k is chosen proportional to $\log n$.

Quite similar numbers as above are obtained in the case of a differential equation with smooth coefficient σ. This underlines that the smoothness or regularity of the boundary value problem does not deteriorate the approximation by $\mathcal{H}$-matrices. Tests with irregular triangulations in more complicated domains give again similar approximations.

Further examples can be seen in [1].

Acknowledgment. The numerical tests from the previous section are produced by Dr. L. Grasedyck (Leipzig).

REFERENCES

[1] M. Bebendorf and W. Hackbusch. Existence of $\mathcal{H}$-matrix approximants to the inverse FE-matrix of elliptic operators with L^∞-coefficients. Technical Report 21, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2002.

[2] S. Börm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. Technical Report 18, Max-Planck-Institut für Mathematik in den Naturwissenschaften, Leipzig, 2002.

[3] D. Braess, M. Dryja, and W. Hackbusch. Grid transfer for nonconforming FE-discretisations with application to non-matching grids. *Computing*, 63:1–25, 1999.

[4] W. Hackbusch. A sparse matrix arithmetic based on $\mathcal{H}$-matrices. Part I: Introduction to $\mathcal{H}$-matrices. *Computing*, 62:89–108, 1999.

[5] W. Hackbusch and B. Khoromskij. A sparse $\mathcal{H}$-matrix arithmetic. Part II: Application to multi-dimensional problems. *Computing*, 64:21–47, 2000.

[6] W. Hackbusch, B. Khoromskij, and S. A. Sauter. On $\mathcal{H}^2$-matrices. In H.-J. Bungartz, R. H. W. Hoppe, and C. Zenger, editors, *Lectures on Applied Mathematics*, pages 9–29. Springer-Verlag Berlin, 2000.

[7] B. Khoromskij and J. M. Melenk. Boundary concentrated finite element methods. Technical Report 45, Max-Planck-Institut für Mathematik in den Naturwissenschaften, 2001.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

5. The Indirect Approach To Domain Decomposition

I. Herrera[1], R. Yates,[2] M.A. Diaz[3]

1. Introduction. The main objective of DDM is, when a domain Ω and one of its partitions are given, to obtain the solution of a boundary value problem defined on it (the 'global problem'), by solving problems formulated on the subdomains of the partition (the 'local problems'), exclusively. This objective can be achieved if sufficient information about the global solution is known, on the internal boundary (which separates the subdomains from each other and to be denoted by Σ), for defining well-posed problems in each one of the subdomains of the partition. Herrera proposed recently a general and unifying theory [15],[14], in which DDM are interpreted as methods for gathering such information. According to it, one defines an information-target on Σ, referred as the *sought information* [15], and the objective of DDM is to obtain such information. There are two main procedures for gathering the *sought information*, which yield two broad categories of DDM: *direct methods* and *indirect* (or *Trefftz-Herrera*) *methods*. This paper belongs to a sequence of papers [15],[6],[5], [4],[21], included in this Proceedings, in which an overview of Herrera's unified theory is given. In particular, the present paper is devoted to a systematic presentation of indirect methods, and a companion paper deals with direct methods [6].

Herrera *et al.* [18],[9],[16], [10],[11],[17], [13] introduced indirect methods in numerical analysis. They are based on the Herrera's Algebraic Theory of boundary value problems [9],[10],[8]. Numerical procedures such as Localized Adjoint Methods (LAM) and Eulerian-Lagrangian LAM (ELLAM) are representative applications [17],[3]. A large number of transport problems in several dimensions have been treated using ELLAM [20]. Indirect Methods of domain decomposition stem from the following observation: when the method of weighted residuals is applied, the information about the exact solution that is contained in the approximate one is determined by the family of test functions that is used, exclusively [9],[16],[10]. This opens the possibility of constructing and applying a special kind of weighting functions, which have the property of yielding the *sought information* at the internal boundary Σ, exclusively, as it is done in Trefftz-Herrera Methods.

The construction of such weighting functions requires having available an instrument of analysis of the information supplied by different test functions. The natural framework for such analysis is given by Green's formulas. However, the conventional approach to this matter is not sufficiently informative for applications to domain decomposition methods. Indeed, in the usual approach [19], one considers the Green's formula

$$\int_\Omega w\mathcal{L}u dx = \int_\Omega u\mathcal{L}^* w dx \tag{1.1}$$

[1]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM) , iherrera@servidor.unam.mx
[2]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM), yates@altcomp.com.mx
[3]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM), mdiaz@tonatiuh.igeofcu.unam.mx

where Ω is a given region and, $\mathcal{L}$ and $\mathcal{L}^*$ are a differential operator and its adjoint, respectively. Then, given a family of functions $\left\{w^1, ..., w^N\right\}$, any approximate solution, $\hat{u}$, obtained with the method of weighted residuals, and with such family of test functions, fulfills

$$\int_\Omega w^\alpha \left(\mathcal{L}\hat{u} - f_\Omega\right) dx = \int_\Omega w^\alpha \left(\mathcal{L}\hat{u} - \mathcal{L}u\right) dx = \int_\Omega \left(\hat{u} - u\right) \mathcal{L}^* w^\alpha dx = 0 \tag{1.2}$$

In this manner, the conclusion is reached that the error $u - \hat{u}$ is orthogonal to the space spanned by the family of functions $\left\{\mathcal{L}^* w^1, ..., \mathcal{L}^* w^N\right\}$. However, this result is of little use when dealing with domain decomposition methods. For them, it is necessary to have a theory which is applicable to situations in which both trial and test functions may be discontinuous simultaneously. This was done introducing a kind of Green's formulas (*"Green-Herrera formulas"*) especially developed for operators defined on discontinuous fields (see [9],[16],[10]). They are based on the Herrera's abstract algebraic theory of boundary value problems, which possesses great generality; it was presented in a preliminary form in [8] and, later, further developed [9],[16],[10] and applied to the numerical treatment of differential equations [17],[12]. This kind of Green's formulas have been formulated in a special kind of function-spaces, in which their elements have jump discontinuities across the internal boundary. In particular, a special class of Sobolev spaces is constructed in this manner [2].

2. Notation. Consider a region Ω, with boundary $\partial\Omega$ and a partition $\{\Omega_1, ..., \Omega_E\}$ of Ω. Let

$$\Sigma \equiv \bigcup_{i \neq j} \left(\bar{\Omega}_i \cap \bar{\Omega}_j\right) \tag{2.1}$$

then Σ will be referred as the 'internal boundary' and $\partial\Omega$ as the 'external (or outer) boundary'. For each $i = 1, ..., E$, $D_1\left(\Omega_i\right)$ and $D_2\left(\Omega_i\right)$ will be two linear spaces of functions defined on Ω_i; then the spaces of *trial* (or base) and *test* (or weighting) functions are defined to be

$$\hat{D}_1\left(\Omega\right) \equiv D_1\left(\Omega_1\right) \oplus ... \oplus D_1\left(\Omega_E\right); \tag{2.2}$$

and

$$\hat{D}_2\left(\Omega\right) \equiv D_2\left(\Omega_1\right) \oplus ... \oplus D_2\left(\Omega_E\right); \tag{2.3}$$

respectively. In what follows we write $\hat{D}_1$ and $\hat{D}_2$, instead of $\hat{D}_1\left(\Omega\right)$ and $\hat{D}_2\left(\Omega\right)$, in order to simplify the notation. Functions belonging either to $\hat{D}_1$ and $\hat{D}_2$, are finite sequences of functions belonging to each one of the sub-domains of the partition. It will be assumed that for each $i = 1, ..., E$, and $\alpha = 1,\ 2$, the traces on Σ of functions belonging $D_\alpha\left(\Omega_i\right)$ exist, and the jump and average of test or weighting functions is defined by

$$[u] \equiv u_+ - u_-; \quad \textit{and} \quad \dot{u} \equiv (u_+ + u_-)/2; \tag{2.4}$$

where u_+ and u_- are the traces from one and the other side of Σ. Here, the unit normal vector to Σ is chosen arbitrarily, but the convention is such that it points towards the positive side of Σ. The special class of Sobolev spaces defined by

$$\hat{\mathbb{H}}^s(\Omega) \equiv \mathbb{H}^s(\Omega_1) \oplus ... \oplus \mathbb{H}^s(\Omega_E); \tag{2.5}$$

has special interest and was considered in [13].

3. Scope. It must be emphasized that the scope of the general theory presented in this paper, Herrera's unified theory of domain decomposition [15],[14], is quite wide, since it is applicable to any linear partial differential equation or system of such equations independently of its type. It handles problems with prescribed jumps on the internal boundary, Σ, and discontinuous equation coefficients, although every kind of equation has its own peculiarities. In particular, we would like to mention explicitly the following:

1. A SINGLE EQUATION
   - (a) Elliptic
     - i. Second Order
     - ii. Higher-Order
       - A. Biharmonic
   - (b) Parabolic
     - i. Heat Equation
   - (c) Hyperbolic
     - i. Wave Equation
2. SYSTEMS OF EQUATIONS
   - (a) Stokes Problems
   - (b) Mixed Methods (Raviart-Thomas)
   - (c) Elasticity

The general form of the boundary value problem with prescribed jumps (BVPJ), to be considered, is

$$\mathcal{L}u = \mathcal{L}u_\Omega \equiv f_\Omega; \quad in\ \Omega_i \quad i = 1, ..., E \tag{3.1}$$

$$B_j u = B_j u_\partial \equiv g_j; \quad on \quad \partial\Omega \tag{3.2}$$

and

$$[J_k u] = [J_k u_\Sigma] \equiv j_k; \quad on \quad \Sigma \tag{3.3}$$

where the $B'_j s$ and $J'_k s$ are certain differential operators (the $j's$ and $k's$ run over suitable finite ranges of natural numbers). Here, in addition, $u_\Omega \equiv \left(u_\Omega^1, ..., u_\Omega^E\right)$, u_∂ and u_Σ are given functions belonging to $\hat{D}_1$ (i.e., *'trial functions'*), which fulfill Eqs.(2.1), (2.2) and (2.3), respectively. Moreover, f_Ω, g_j and j_k may be defined by Eqs. (2.1) to (2.3).

In what follows, it will be assumed that the boundary conditions and jump conditions of this BVPJ can be brought into the point-wise variational form:

$$\mathcal{B}(u, w) = \mathcal{B}(u_\partial, w) \equiv g_\partial(w); \quad \forall w \in \hat{D}_2 \tag{3.4}$$

and

$$\mathcal{J}(u, w) = \mathcal{J}(u_\Sigma, w) \equiv j_\Sigma(w); \quad \forall w \in \hat{D}_2 \tag{3.5}$$

where $\mathcal{B}(u, w)$ and $\mathcal{J}(u, w)$, are bilinear functions defined point-wise.

4. Trefftz-Herrera Approach to DDM. Let us recall a few basic points of Herrera's unified theory (see [15]). The information that one deals with, when formulating and treating partial differential equations (i.e., the BVPJ), is classified in two broad categories: *'data of the problem'* and *'complementary information'*. In turn, three classes of data can be distinguished: data in the interior of the subdomains of the partition (given by the differential equation, which in the BVPJ is fulfilled in the interior of the subdomains, exclusively), the data on the external boundary ($B_j u, \quad on \quad \partial\Omega$) and the data on the internal boundary (namely, $[J_k u], \quad on \quad \Sigma$). The complementary information can be classified in a similar fashion: the values of the sought solution in the interior of the subdomains ($u_i \in D(\Omega_i)$, for $i = 1, ..., E$); the complementary information on the outer boundary (for example, the normal derivative in the case of Dirichlet problems for Laplace's equation); and the complementary information on the internal boundary Σ (for example, the average of the function and the average of the normal derivative across the discontinuity for elliptic problems of second order [5]). In the unified theory of DDM, a target of information, which is contained in the complementary information on Σ, is defined; it is called *'the sought information'*. It is required that the *sought information*, when complemented with the data of the problem, be sufficient for determining uniquely the solution of BVPJ in each one of the subdomains of the partition.

In general, however, the sought information may satisfy this property and yet be redundant, in the sense that if all of it is used simultaneously together with the data of the problem, ill-posed problems are obtained. Consider for example, a Dirichlet problem of an elliptic-type second order equation (see [5]), for which the jumps of the function and of its normal derivative have been prescribed. If for such problem the *sought information* is taken to be the average of the function -i.e., $(u_+ + u_-)/2$- and the average of the normal derivative -i.e., $\frac{1}{2}\partial(u_+ + u_-)/\partial n$, on Σ-, then it may be seen that it contains redundant information. Indeed, $u_+ = \frac{1}{2}(u_+ + u_-) + \frac{1}{2}(u_+ - u_-)$, $u_- = \frac{1}{2}(u_+ + u_-) - \frac{1}{2}(u_+ - u_-)$, and a similar relation holds for the normal derivatives. Therefore, if the *'sought information'* and the *'data of the problem'* are used simultaneously, one may derive not only the value of the BVPJ solution on the boundary of each one of the subdomains, but also the normal derivative, at least in a non-void section of those boundaries. As it is well known, this is an ill-posed problem, because

Dirichlet problem is already well-posed in each one of the subdomains. Thus, the *sought information* contains redundant information in this case.

Generally, in the numerical treatment of partial differential equations, efficiency requires eliminating redundant information. Due to this fact, when the choice of the *sought information* is such that there is a family of well-posed problems -one for each subdomain of the partition- which uses all the sought information, together with all the data of the BVPJ, such choice is said to be *'optimal'*. Once the information-target constituted by the *sought information* has been chosen, it is necessary to design a procedure for gathering it. There are two main ways of proceeding to achieve this goal: *direct methods* and *indirect* (or *Trefftz-Herrera*) *methods*. In the following Sections the general framework for designing indirect procedures is constructed.

Firstly, Green-Herrera formulas, which were originally derived in 1985 [9],[16],[10] will be presented. They are equations that relate the *'data of the problem'* with *'the complementary information'*. Then, a general variational principle of the usual kind, *in terms of the data of the problem*, which applies to any BVPJ, is introduced. Using Green-Herrera formula the variational formulation *in terms of the data of the problem*, is transformed into one *in terms of the complementary information*. Among the complementary information the *sought information* is singled out and the conditions that the test functions must satisfy in order to eliminate all the complementary information, except the *sought information*, are identified. When the variational principle *in terms of the complementary information* is applied, with weighting functions that fulfill such conditions, a variational principle which characterizes the sought information is derived. This principle provides a very general, although somewhat abstract, basis of Trefftz-Herrera Method (this is given by Theorem 7.1 Eq. 7.4).

5. Green-Herrera Formulas. To start, let $\mathcal{L}$ and $\mathcal{L}^*$ be a differential operator and its formal adjoint; then there exists a vector-valued bilinear function $\underline{\mathcal{D}}$, which satisfies

$$w\mathcal{L}u - u\mathcal{L}^*w \equiv \nabla \cdot \underline{\mathcal{D}}(u,w) \tag{5.1}$$

It will also be assumed that there are bilinear functions $\mathcal{B}(u,w)$, $\mathcal{C}(w,u)$, $\mathcal{J}(u,w)$ and $\mathcal{K}(w,u)$, the first two defined on $\partial\Omega$ and the last two on Σ, such that

$$\underline{\mathcal{D}}(u,w)\cdot\underline{n} = \mathcal{B}(u,w) - \mathcal{C}(w,u); \quad \text{on } \partial\Omega \tag{5.2}$$

and

$$-[\underline{\mathcal{D}}(u,w)\cdot\underline{n}] = \mathcal{J}(u,w) - \mathcal{K}(w,u); \quad \text{on } \Sigma \tag{5.3}$$

Generally, the definitions of $\mathcal{B}$ and $\mathcal{C}$ depend on the kind of boundary conditions and the "smoothness criterion" of the specific problem considered [9],[16]. For the case when the coefficients of the differential operators are continuous, Herrera has given very general formulas for $\mathcal{J}$ and $\mathcal{K}$ [18]; they are:

$$\mathcal{J}(u,w) = -\underline{\mathcal{D}}([u],\dot{w})\cdot\underline{n} \quad \text{and} \quad \mathcal{K}(w,u) = \underline{\mathcal{D}}(\dot{u},[w])\cdot\underline{n} \tag{5.4}$$

Applying the generalized divergence theorem [2], this implies the following Green-Herrera formula [18]:

$$\begin{aligned} &\int_\Omega w\mathcal{L}u dx - \int_{\partial\Omega} \mathcal{B}(u,w)dx - \int_\Sigma \mathcal{J}(u,w)dx \\ &= \int_\Omega u\mathcal{L}^* w dx - \int_{\partial\Omega} \mathcal{C}^*(u,w)dx - \int_\Sigma \mathcal{K}^*(u,w)dx \end{aligned} \tag{5.5}$$

Introduce the following notation:

$$\langle Pu, w\rangle = \int_\Omega w\mathcal{L}u dx; \quad \langle Q^*u, w\rangle = \int_\Omega u\mathcal{L}^* w dx \tag{5.6}$$

$$\langle Bu, w\rangle = \int_{\partial\Omega} \mathcal{B}(u,w)dx; \quad \langle C^*u, w\rangle = \int_{\partial\Omega} \mathcal{C}^*(u,w)dx \tag{5.7}$$

$$\langle Ju, w\rangle = \int_\Sigma \mathcal{J}(u,w)dx; \quad \langle K^*u, w\rangle = \int_\Sigma \mathcal{K}^*(u,w)dx \tag{5.8}$$

With these definitions, each one of P, B, J, Q^*, C^* and K^*, are real-valued bilinear functionals defined on $\hat{D}_1 \times \hat{D}_2$, and Eq.(5.5) can be written as

$$\langle (P - B - J)u, w\rangle \equiv \langle (Q^* - C^* - K^*)u, w\rangle; \quad \forall (u,w) \in \hat{D}_1 \times \hat{D}_2 \tag{5.9}$$

or more briefly

$$P - B - J \equiv Q^* - C^* - K^*; \tag{5.10}$$

6. Variational Formulations of the Problem with Prescribed Jumps. A weak formulation of the BVPJ is

$$\langle (P - B - J)u, w\rangle \equiv \langle f - g - j, w\rangle; \quad \forall w \in \hat{D}_2 \tag{6.1}$$

where f, g and $j \in D_2^*$. This equation is equivalent to

$$\langle (Q^* - C^* - K^*)u, w\rangle \equiv \langle f - g - j, w\rangle; \quad \forall w \in \hat{D}_2 \tag{6.2}$$

by virtue of Green-Herrera formula of Eq. (5.10). Necessary conditions for the existence of solution of this problem is that there exist $u_\Omega \in \hat{D}_1$, $u_\partial \in \hat{D}_1$ and $u_\Sigma \in \hat{D}_1$, such that:

$$f \equiv Pu_\Omega, \quad g \equiv Bu_\partial \quad and \quad j \equiv Ju_\Sigma \tag{6.3}$$

Thus, it is assumed that such functions exist. From now on, the following notation is adopted: $u \in \hat{D}_1$ will be a solution of the BVPJ, which is assumed to exist and to be unique; therefore, $u \in \hat{D}_1$ fulfills Eq. (6.1). Observe that Eqs. (6.1) and (6.2) supply two different but equivalent variational formulations of the BVPJ. The first one will be referred as the '*variational formulation in terms of the data*', while the second one will be referred as the '*variational formulation in terms of the complementary information*' (this latter variational principle was introduced in [18] with the title "variational formulation in terms of the sought information" but it is more convenient to reserve such name for another formulation that will be introduced later).

Eqs. (6.1) and (6.2), can also be written as equalities between linear funtionals:

$$(P - B - J)u = f - g - j; \tag{6.4}$$

and

$$(Q^* - C^* - K^*)u = f - g - j; \tag{6.5}$$

respectively.

7. Variational Formulation of Trefftz-Herrera Method. A first step to derive Trefftz-Herrera procedures is to use the variational formulation in terms of the complementary information of Eq.(6.2) to establish conditions that a weighting function must fulfill in order to yield information on the internal boundary Σ, exclusively. What is required is to eliminate the terms containing Q^*u and C^*u in that equation. This is achieved if the test functions satisfy $Qw = 0$ and $Cw = 0$, simultaneously, because $\langle Q^*u, w\rangle \equiv \langle Qw, u\rangle$ and $\langle C^*u, w\rangle \equiv \langle Cw, u\rangle$. Thus, in view of Eq. (6.2), one has

$$-\langle K^*u, w\rangle = \langle f - g - j, w\rangle; \quad \forall w \in N_Q \cap N_C \subset \hat{D}_2 \tag{7.1}$$

where N_Q and N_C are null subspaces of the operators Q and C respectively.

Observe that the left-hand side of Eq.(7.1) involves the complementary information on Σ, exclusively, as desired. Generally, the complementary information on Σ, K^*u, is sufficient to define well-posed problems in each one of the subdomains of the domain decomposition, when the boundary data is added to it. However, it can be seen through specific examples that the complementary information K^*u is more than what is essential to achieve this goal and handling excessive information, in general, requires carrying too many degrees of freedom in the computational process, which in many cases is inconvenient. Thus, generally, to develop numerical methods of optimal efficiency, it is better to eliminate part of such information.

The general procedure for carrying out such elimination consists in introducing a '*weak decomposition*' $\{S, R\}$ of the bilinear functional K (for a definition of weak decomposition, see [10]). Then, S and R are bilinear functionals and fulfill

$$K \equiv S + R; \tag{7.2}$$

Then '*the sought information*' is defined to be S^*u, where $u \in \hat{D}_1$ is the solution of the BVPJ. In particular, a function $\tilde{u} \in \hat{D}_1$ is said to '*contain the sought information*' when $S^*\tilde{u}$=S^*u.

Let $\tilde{N}_2 \subset \hat{D}_2$ be defined by $\tilde{N}_2 \equiv N_Q \cap N_C \cap N_R$. An auxiliary concept, quite useful for formulating Trefftz-Herrera domain decomposition procedures, which was originally introduced in 1980 [7], is the following (see [1]).

Definition 7.1.- *A subset of weighting functions, $\mathcal{E} \subset \tilde{N}_2 \equiv N_Q \cap N_C \cap N_R$, is said to be TH-complete for S^*, when for any $\hat{u} \in \hat{D}_1$, one has:*

$$\langle S^*\hat{u}, w\rangle = 0, \forall w \in \mathcal{E} \Rightarrow S^*\hat{u} = 0; \tag{7.3}$$

Clearly, a necessary and sufficient condition for the existence of TH-complete systems, is that $\tilde{N}_2 \equiv N_Q \cap N_C \cap N_R$ be, itself, a TH-complete system.

Theorem 7.1 *Let $\mathcal{E} \subset \tilde{N}_2$ be a system of TH-complete weighting functions for S^*, and let $u \in \hat{D}_1$ be the solution of the BVPJ. Then, a necessary and sufficient condition for $\hat{u} \in \hat{D}_1$ to contain the sought information, is that*

$$-\langle S^*\hat{u}, w\rangle = \langle f - g - j, w\rangle\,; \quad \forall w \in \mathcal{E} \tag{7.4}$$

Proof. If $u \in \hat{D}_1$ is the solution of the BVPJ, one has

$$-\langle S^*u, w\rangle = \langle f - g - j, w\rangle\,; \quad \forall w \in \mathcal{E} \tag{7.5}$$

Hence

$$-\langle S^*(\hat{u} - u), w\rangle = 0; \quad \forall w \in \mathcal{E} \tag{7.6}$$

and, therefore, $S^*\hat{u}$=S^*u. ■

Theorem 7.1, supplies a very **General Formulation of Indirect Methods (or Trefftz-Herrera Methods) of Domain Decomposition** which can be applied to any linear equation or system of such equations. When $u_p \in \hat{D}_1$ is a function satisfying $Pu_p = f$ and $Bu_p = g$ then Eq.(7.4) can be replaced by

$$-\langle S^*\hat{u}, w\rangle = -\langle S^*u_p, w\rangle + \langle J(u_p - u_\Sigma), w\rangle\,; \quad \forall w \in \mathcal{E} \tag{7.7}$$

In applications, Eq.(7.7) determines the average of the solution and/or its derivatives on Σ.

8. Unified Approach to DDM: Abstract Formulation. The concepts and notations of the previous Sections, can be used to give an abstract expression to the unified formulation of Domain Decomposition Methods.

In this Section a pair of weak decomposition $\{S_J, R_J\}$ and $\{S, R\}$ of J and K, respectively, will be considered. This assumption implies that [10]

$$J = S_J + R_J \tag{8.1}$$

in addition to Eq. (7.2). Even more, under the above assumption a function $\hat{u} \in \hat{D}_1$ fulfills Eq. (6.4), if and only if

$$(P - B - R_J)\,\hat{u} = Pu_\Omega - Bu_\partial - R_J u_\Sigma \tag{8.2}$$

and

$$S_J\hat{u} = S_J u_\Sigma \tag{8.3}$$

It has interest to consider the case when Eq. (8.3) can be replaced by the condition that $\hat{u}$ contains the sought information; i.e., when Eq. (8.3) can be replaced by

$$S^*\hat{u} = S^*u \tag{8.4}$$

because this leads to a quite general formulation of DDM.

Definition 8.1.- *The pair of weak decompositions* $\{S_J, R_J\}$ *and* $\{S, R\}$ *of* J *and* K*, respectively, is said to be optimal, when given any* $u_\Omega \in \hat{D}_1$*,* $u_\partial \in \hat{D}_1$*,* $u_\Sigma \in \hat{D}_1$ *and* $u_I \in \hat{D}_1$*, the problem of finding* $\hat{u} \in \hat{D}_1$*, such that*

$$(P - B - R_J)\,\hat{u} = Pu_\Omega - Bu_\partial - R_J u_\Sigma \tag{8.5}$$

and

$$S^*\hat{u} = S^* u_I \tag{8.6}$$

is local and well-posed.

Lemma 8.1 *Assume* $\hat{u} \in \hat{D}_1$ *is solution of the local problems defined by Eqs. (8.5) and (8.6), for some* $u_I \in \hat{D}_1$*, then the following assertions are equivalent*

i).- u_I contains the sought information,

ii).- $J\hat{u} = Ju$,

iii).- $\hat{u}$ is the solution of the BVPJ.

Proof. First, we show that *ii)* and *iii)* are equivalent. To this end, assuming *ii)* observe that Eq. (8.5) together with *ii)* imply that $\hat{u} \in \hat{D}_1$ is the solution of the BVPJ. Conversely, if $\hat{u} \in \hat{D}_1$ is solution of the BVPJ, then $J\hat{u} = Ju$. The equivalence between *i)* and *iii)* is immediate. Indeed, assume *iii)* then $S^*u_I = S^*\hat{u} = S^*u$; i.e., u_I contains the sought information. If *i)* holds, then

$$(P - B - R_J)\,\hat{u} = (P - B - R_J)\,u \tag{8.7}$$

together with

$$S^*\hat{u} = S^*u \tag{8.8}$$

and *iii)* follows from the uniqueness of solution of the local problems. ■

Definition 8.2 (Steklov-Poincaré Operator).- *Given any* $v \in \hat{D}_1$*, define* $\tau : D_1 \to D_2^*$ *, by*

$$\tau(v) = J\hat{v} \tag{8.9}$$

where $\hat{v} \in \hat{D}_1$ *is the solution of the local boundary value problems with* $u_I = v$ *.*

Lemma 8.2 *A function* $\hat{u} \in \hat{D}_1$*, contains the sought information if and only if*

$$\tau(\hat{u}) = Ju \tag{8.10}$$

Proof. It is immediate in view of the previous Lemma. ■

9. The Second Order Elliptic Equation. As an illustration, consider the BVPJ for an elliptic operator of second order

$$\mathcal{L}u \equiv -\nabla \cdot (\underline{\underline{a}} \cdot \nabla u) + \nabla \cdot (\underline{b} u) + cu = f_\Omega; \quad in \quad \Omega, \tag{9.1}$$

subjected to the boundary conditions

$$u = u_\partial; \quad on \quad \partial\Omega, \tag{9.2}$$

and the jump conditions

$$[u] = [u_\Sigma] \equiv j^0_\Sigma \quad and \quad [\underline{\underline{a}} \cdot \nabla u] \cdot \underline{n} = [\underline{\underline{a}} \cdot \nabla u_\Sigma] \cdot \underline{n} \equiv j^1_\Sigma; \quad on \quad \Sigma, \tag{9.3}$$

The numerical treatment of this problem, using both a Direct Method and a Trefftz-Herrera Method, is explained in [6] and [5]. When the differential operator is given as in Eq. (9.1), then,

$$w\mathcal{L}u - u\mathcal{L}^* w \equiv \nabla \cdot \underline{\mathcal{D}}(u, w) \tag{9.4}$$

where

$$\underline{\mathcal{D}}(u, w) \equiv u(\underline{a}_n \cdot \nabla w + b_n w) - w\underline{a}_n \cdot \nabla u \tag{9.5}$$

Define the following bilinear functionals:

$$\mathcal{B}(u, w) \equiv u(\underline{a}_n \cdot \nabla w + b_n w) \cdot \underline{n}, \quad \mathcal{C}(w, u) \equiv w\underline{a}_n \cdot \nabla u \tag{9.6}$$

$$\mathcal{J}(u, w) \equiv \dot{w}[\underline{a}_n \cdot \nabla u] - [u]\,\dot{\overline{(\underline{a}_n \cdot \nabla w + b_n w)}} \tag{9.7}$$

$$\mathcal{K}(w, u) \equiv \dot{u}[\underline{a}_n \cdot \nabla w + b_n w] - [w]\,\dot{\overline{(\underline{a}_n \cdot \nabla u)}} \tag{9.8}$$

$$\mathcal{S}_J(u, w) \equiv \dot{w}[\underline{a}_n \cdot \nabla u], \quad \mathcal{R}_J(u, w) \equiv -[u]\,\dot{\overline{(\underline{a}_n \cdot \nabla w + b_n w)}} \tag{9.9}$$

$$\mathcal{S}(w, u) \equiv \dot{u}[\underline{a}_n \cdot \nabla w + b_n w] \quad and \quad \mathcal{R}(w, u) \equiv -[w]\,\dot{\overline{(\underline{a}_n \cdot \nabla u)}} \tag{9.10}$$

In addition, define the bilinear functionals S_J, R_J, S and R in a similar fashion to Eqs. (5.6)-(5.8), by means of corresponding integrals.

Then Green-Herrera formula of Eq. (5.10) holds. Even more, Eqs. (7.2) and (8.1) are fulfilled and the pair $\{S_J, R_J\}$ and $\{S, R\}$ constitute an optimal pair of weak decompositions, because the local problems are well posed. Indeed, Eq.(8.2) is the BVPJ of Eqs.(9.1) to (9.3) except that the jump condition associated with this latter equation has been omitted. However, the jump of the function, of Eq.(9.2), is indeed prescribed. This problem has many solutions. However, with the above definition of $\mathcal{S}$, the *sought information* is the average of the function on the internal boundary Σ. When this information is complemented with the jump of the function, which is the data given by Eq.(9.2), the values of the function on both sides of Σ are determined by the identities

$$u_+ \equiv \dot{u} + 1/2\,[u] \quad \textit{and} \quad u_- \equiv \dot{u} - 1/2\,[u] \tag{9.11}$$

This information together with the boundary conditions in the external boundary permits establishing well-posed problems in each one of the subdomains of the partition.

10. Optimal Interpolation. The Indirect Method yields information on the internal boundary Σ, exclusively. To extend that information into the interior of the subdomains of the partition, it is necessary to solve the local problems [5],[21]. The following results will be useful in applications, to carry out such step.

Let $\tilde{N}_1 \subset \hat{D}_1$ be defined by $\tilde{N}_1 \equiv N_P \cap N_B \cap N_{R_J}$.

Theorem 10.1 *Let $u_P \in \hat{D}_1$ be such that*

$$Pu_P = Pu_\Omega, \quad Bu_P = Bu_\partial \quad \textit{and} \quad R_J u_P = R_J u_\Sigma. \tag{10.1}$$

Then there exists $v \in \tilde{N}_1$ such that

$$-\langle S^* v, w\rangle = \langle S_J\,(u_P - u_\Sigma)\,, w\rangle\,; \quad \forall w \in \tilde{N}_2 \tag{10.2}$$

In addition, define $\hat{u} \in \hat{D}_1$ by $\hat{u} \equiv u_P + v$. Then $\hat{u} \in \hat{D}_1$ contains the sought information. Even more, $\hat{u} \equiv u$, where u is the solution of the BVPJ.

Proof. Take $u \in \hat{D}_1$ as in the Theorem, then this function contains the sought information and, in view of Eq. (10.1), Eq. (10.2) can be applied, with $\hat{u} \equiv u$. Define $v \equiv u - u_P$, then

$$-\langle S^* v, w\rangle = \langle J\,(u_P - u_\Sigma)\,, w\rangle = \langle S_J\,(u_P - u_\Sigma)\,, w\rangle\,; \quad \forall w \in \tilde{N}_2 \tag{10.3}$$

because $R_J\,(u_P - u_\Sigma) = 0$. However, from Eq. (10.1), it follows that $v \in \tilde{N}_1 \equiv N_P \cap N_B \cap N_{R_J}$. When $\tilde{u} \in \hat{D}_1$ is defined as in the Theorem, then it fulfills

$$(P - B - R_J)\,(\hat{u} - u) = 0 \quad \textit{and} \quad S^*(\hat{u} - u) = 0 \tag{10.4}$$

Therefore, $\hat{u} - u = 0$, since the problem of Eqs. (10.4), is well-posed. ∎

The Symmetric Case: In this case $\hat{D}_1 = \hat{D}_2 \equiv \hat{D}$, $P = Q$, $B = C$, $J = K$, $S \equiv S_J$ and $R \equiv R_J$. Then $\tilde{N} \equiv \tilde{N}_2 \equiv N_Q \cap N_C \cap N_R = N_P \cap N_B \cap N_{R_J} \equiv \tilde{N}_1$. If it is further assumed that the bilinear functional $-\langle S^* u, w\rangle$ is symmetric and positive definite $\forall u, w \in \tilde{N}$, it can be shown that the quadratic functional $-\langle S^* \tilde{u}, \tilde{u}\rangle - 2\,\langle f - g - j, \tilde{u}\rangle$ attains its minimum over $\tilde{N}$, at $\tilde{u} \in \tilde{N}$, if and only if $\tilde{u} \in \tilde{N}$ contains the sought information.

REFERENCES

[1] H. Begehr and R. Gilbert. *Transformations, Transmutations and Kernel Functions*. Longman Scientific and Technical, 1992.

[2] R. Berlanga and I. Herrera. The Gauss Theorem for Domain Decompositions in Sobolev Spaces. *Applicable Analysis: An International Journal*, 76(1-2):67–81, 2000.

[3] M. Celia, T. Russell, I. Herrera, and R. Ewing. An Eulerian-Lagrangian Localized Adjoint Method for the Advection-Diffusion Equation. *Advances in Water Resources*, 13(4):187–206, 1990.

[4] M. Diaz and I. Herrera. Indirect Method of Collocation for the Biharmonic Equation. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[5] M. Diaz, I. Herrera, and R. Yates. Indirect Method of Collocation: Second Order Equations. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[6] F. Garcia-Nocetti, I. Herrera, R. Yates, E. Rubio, and L. Ochoa. The Direct Approach to Domain Decomposition Methods. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[7] I. Herrera. Boundary Methods. A Criterion for Completeness. In *Proc. National Academy of Sciences, USA*, volume 77, pages 4395–4398, 1980.

[8] I. Herrera. *Boundary Methods: An Algebraic Theory*. Pitman Advance Publishing Program, Boston, London, Melbourne, 1984.

[9] I. Herrera. Unified Approach to Numerical Methods. Part 1. Green's Formulas for Operators in Discontinuous Fields. *Numerical Methods for Partial Differential Equations*, 1(1):12–37, 1985.

[10] I. Herrera. *Some Unifying Concepts in Applied Mathematics*, pages 79–88. The Merging of Disciplines: New Directions in Pure, Applied, and Computational Mathematics. Edited by R.E. Ewing, K.I. Gross and C.F. Martin. Springer Verlag, New York, 1986.

[11] I. Herrera. The Algebraic Theory Approach for Ordinary Differential Equations: Highly Accurate Finite Differences. *Numerical Methods for Partial Differential Equations*, 3(3):199–218, 1987.

[12] I. Herrera. *Localized Adjoint Methods: A New Discretization Methodology*, chapter 6, pages 66–77. Computational Methods in Geosciences. SIAM, 1992.

[13] I. Herrera. Trefftz-Herrera Domain Decomposition. *Advances in Engineering Software*, 24:43–56, 1995.

[14] I. Herrera. Trefftz Method: A General Theory. *Numerical Methods for Partial Differential Equations*, 16(6):561–580, 2000.

[15] I. Herrera. Unified theory of domain decomposition methods. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[16] I. Herrera, L. Chargoy, and G. Alducin. Unified Approach to Numerical Methods. Part 3. Finite Differences and Ordinary Differential Equations. *Numerical Methods for Partial Differrential Equations*, 1:241–258, 1985.

[17] I. Herrera, R. Ewing, M. Celia, and T. Russell. Eulerian-Lagrangian Localized Adjoint Method: The Theoretical Framework. *Numerical Methods for Partial Differential Equations*, 9(4):431–457, 1993.

[18] I. Herrera, R. Yates, and M. Diaz. General Theory of Domain Decomposition: Indirect Methods. *Numerical Methods for Partial Differential Equations*, 18(3):296–322, 2002.

[19] J. T. Oden. *Finite Elements of Non-linear Continua*. McGraw-Hill, New York, 1972.

[20] T. Russell and M. Celia. Eulerian-Lagrangian Localized Adjoint Methods (ELLAM): From Cocoyoc to the Present. *Eos Trans. AGU*, 82(47), 2001.

[21] R. Yates and I. Herrera. Parallel Implementation of Indirect Collocation Method. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

6. Applications of Domain Decomposition and Partition of Unity Methods in Physics and Geometry

M. Holst[1]

1. Introduction. In this article we consider a class of adaptive multilevel domain decomposition-like algorithms, built from a combination of adaptive multilevel finite element, domain decomposition, and partition of unity methods. These algorithms have several interesting features such as very low communication requirements, and they inherit a simple and elegant approximation theory framework from partition of unity methods. They are also very easy to use with highly complex sequential adaptive finite element packages, requiring little or no modification of the underlying sequential finite element software. The parallel algorithm can be implemented as a simple loop which starts off a sequential local adaptive solve on a collection of processors simultaneously.

We first review the Partition of Unity Method (PUM) of Babuška and Melenk in Section 2, and outline the PUM approximation theory framework. In Section 3, we describe a variant we refer to here as the Parallel Partition of Unity Method (PPUM), which is a combination of the Partition of Unity Method with the parallel adaptive algorithm from [4]. We then derive two global error estimates for PPUM, by exploiting the PUM analysis framework it inherits, and by employing some recent local estimates of Xu and Zhou [22]. We then discuss a duality-based variant of PPUM in Section 4 which is more appropriate for certain applications, and we derive a suitable variant of the PPUM approximation theory framework. Our implementation of PPUM-type algorithms using the FEtk and MC software packages is described in Section 5. We then present a short numerical example in Section 6 involving the Einstein constraints arising in gravitational wave models.

2. The Partition of Unity Method (PUM) of Babuška and Melenk. We first briefly review the partition of unity method (PUM) of Babuška and Melenk [1]. Let $\Omega \subset \mathbb{R}^d$ be an open set and let $\{\Omega_i\}$ be an open cover of Ω with a bounded local overlap property: For all $x \in \Omega$, there exists a constant M such that

$$\sup_i \{\ i \mid x \in \Omega_i\ \} \leq M. \tag{2.1}$$

A Lipschitz *partition of unity* $\{\phi_i\}$ subordinate to the cover $\{\Omega_i\}$ satisfies the following five conditions:

$$\sum_i \phi_i(x) \equiv 1, \quad \forall x \in \Omega, \tag{2.2}$$

$$\phi_i \in C^k(\Omega) \quad \forall i, \quad (k \geq 0), \tag{2.3}$$

$$\text{supp } \phi_i \subset \overline{\Omega}_i, \quad \forall i, \tag{2.4}$$

$$\|\phi_i\|_{L^\infty(\Omega)} \leq C_\infty, \quad \forall i, \tag{2.5}$$

$$\|\nabla \phi_i\|_{L^\infty(\Omega)} \leq \frac{C_G}{\text{diam}(\Omega_i)}, \quad \forall i. \tag{2.6}$$

[1]UC San Diego, mholst@ucsd.edu

Several explicit constructions of partitions of unity satisfying (2.2)–(2.6) exist. The simplest construction in the case of a polygon $\Omega \subset \mathbb{R}^d$ employs global C^0 piecewise linear finite element basis functions defined on a simplex mesh subdivision $\mathcal{S}$ of Ω. The $\{\Omega_i\}$ are first built by first constructing a disjoint partitioning $\{\Omega_i^\circ\}$ of $\mathcal{S}$ using e.g. spectral or inertial bisection [4]. Each of the disjoint Ω_i° are extended to define Ω_i by considering all boundary vertices of Ω_i°; all simplices of neighboring Ω_j°, $j \neq i$ which are contained in the boundary vertex 1-rings of Ω_i° are added to Ω_i° to form Ω_i. This procedure produces the smallest overlap for the $\{\Omega_i\}$, such that the properties (2.2)–(2.5) are satisfied by the resulting $\{\phi_i\}$ built from the nodal C^0 piecewise linear finite element basis functions. Property (2.6) is also satisfied, but C_G will depend on the diameter of the overlap simplices. More sophisticated constructions with superior properties are possible; see e.g. [8, 19].

The *partition of unity method (PUM)* builds an approximation $u_{ap} = \sum_i \phi_i v_i$ where the v_i are taken from the local approximation spaces:

$$V_i \subset C^k(\Omega \cap \Omega_i) \subset H^1(\Omega \cap \Omega_i), \quad \forall i, \quad (k \geq 0). \tag{2.7}$$

The following simple lemma makes possible several useful results.

Lemma 2.1 *Let $w, w_i \in H^1(\Omega)$ with supp $w_i \subseteq \overline{\Omega \cap \Omega_i}$. Then*

$$\begin{aligned} \sum_i \|w\|^2_{H^k(\Omega_i)} &\leq M\|w\|^2_{H^k(\Omega)}, \qquad k = 0, 1 \\ \|\sum_i w_i\|^2_{H^k(\Omega)} &\leq M \sum_i \|w_i\|^2_{H^k(\Omega \cap \Omega_i)}, \qquad k = 0, 1 \end{aligned}$$

Proof. The proof follows from (2.1) and (2.2)–(2.6); see [1]. ∎

The basic approximation properties of PUM following from 2.1 are as follows.

Theorem 2.1 (Babuška and Melenk [1]) *If the local spaces V_i have the following approximation properties:*

$$\begin{aligned} \|u - v_i\|_{L^2(\Omega \cap \Omega_i)} &\leq \epsilon_0(i), \quad \forall i, \\ \|\nabla(u - v_i)\|_{L^2(\Omega \cap \Omega_i)} &\leq \epsilon_1(i), \quad \forall i, \end{aligned}$$

then the following a priori *global error estimates hold:*

$$\begin{aligned} \|u - u_{ap}\|_{L^2(\Omega)} &\leq \sqrt{M} C_\infty \left(\sum_i \epsilon_0^2(i) \right)^{1/2}, \\ \|\nabla(u - u_{ap})\|_{L^2(\Omega)} &\leq \sqrt{2M} \left(\sum_i \left(\frac{C_G}{diam(\Omega_i)} \right)^2 \epsilon_1^2(i) + C_\infty^2 \epsilon_0^2(i) \right)^{1/2}. \end{aligned}$$

Proof. This follows from Lemma 2.1 by taking $u - u_{ap} = \sum_i \phi_i (u - v_i)$ and then by using $w_i = \phi_i(u - v_i)$ in Lemma 2.1. ∎

Consider now the following linear elliptic problem:

$$\begin{aligned} -\nabla \cdot (a \nabla u) &= f \text{ in } \Omega, \\ u &= 0 \text{ on } \partial\Omega, \end{aligned} \tag{2.8}$$

where $a_{ij} \in W^{1,\infty}(\Omega)$, $f \in L^2(\Omega)$, $a_{ij}\xi_i\xi_j \geq a_0 > 0$, $\forall \xi_i \neq 0$, where $\Omega \subset \mathbb{R}^d$ is a convex polyhedral domain. A weak formulation is:

$$\text{Find } u \in H_0^1(\Omega) \text{ such that } \langle F(u), v \rangle = 0, \quad \forall v \in H_0^1(\Omega), \tag{2.9}$$

where

$$\langle F(u), v \rangle = \int_\Omega a \nabla u \cdot \nabla v \; dx - \int_\Omega f v \; dx.$$

A general Galerkin approximation is the solution to the subspace problem:

$$\text{Find } u_{ap} \in V \subset H_0^1(\Omega) \text{ s.t. } \langle F(u_{ap}), v \rangle = 0, \; \forall v \in V \subset H_0^1(\Omega). \tag{2.10}$$

With PUM, the subspace V for the Galerkin approximation is taken to be the globally coupled *PUM space* (cf. [8]):

$$V = \left\{ \; v \mid v = \sum_i \phi_i v_i, \; v_i \in V_i \; \right\} \subset H^1(\Omega),$$

If error estimates are available for the quality of the local solutions produced in the local spaces, then the PUM approximation theory framework given in Theorem 2.1 guarantees a global solution quality.

3. A Parallel Partition of Unity Method (PPUM). A new approach to the use of parallel computers with adaptive finite element methods was presented recently in [4]. The following variant of the algorithm in [4] is described in [9], which we refer to as the *Parallel Partition of Unity Method* (or PPUM). This variant replaces the final global smoothing iteration in [4] with a reconstruction based on Babuška and Melenk's original Partition of Unity Method [1], which provides some additional approximation theory structure.

Algorithm (PPUM - Parallel Partition of Unity Method [4, 9])

1. *Discretize and solve the problem using a global coarse mesh.*
2. *Compute* a posteriori *error estimates using the coarse solution, and decompose the mesh to achieve equal error using weighted spectral or inertial bisection.*
3. *Give the entire mesh to a collection of processors, where each processor will perform a completely independent multilevel adaptive solve, restricting local refinement to only an assigned portion of the domain. The portion of the domain assigned to each processor coincides with one of the domains produced by spectral bisection with some overlap (produced by conformity algorithms, or by explicitly enforcing substantial overlap). When a processor has reached an error tolerance locally, computation stops on that processor.*
4. *Combine the independently produced solutions using a partition of unity subordinate to the overlapping subdomains.*

While the PPUM algorithm seems to ignore the global coupling of the elliptic problem, recent results on local error estimation [22], as well as some not-so-recent results on interior estimates [17], support this as provably good in some sense. The principle idea underlying the results in [17, 22] is that while elliptic problems are globally coupled, this global coupling is essentially a "low-frequency" coupling, and can be handled on the initial mesh which is much coarser than that required for approximation accuracy considerations. This idea has been exploited, for example, in [21, 22], and is why the construction of a coarse problem in overlapping domain decomposition methods is the key to obtaining convergence rates which are independent of the number of subdomains (c.f. [20]). An example showing the types of local refinements that occur within each subdomain is depicted in Figure 3.1.

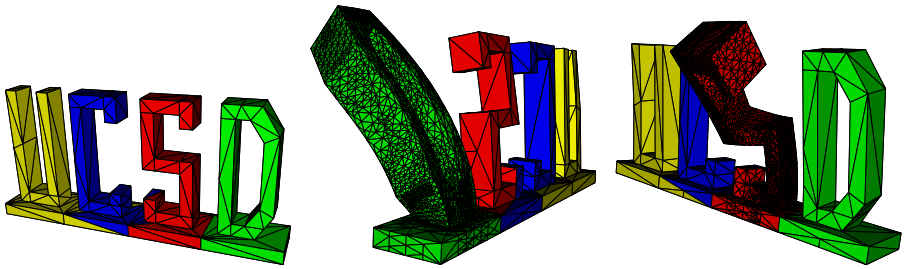


Figure 3.1: Example showing the types of local refinements created by PPUM.

To illustrate how PPUM can produce a quality global solution, we will give a global error estimate for PPUM solutions. This analysis can also be found in [9]. We can view PPUM as building a PUM approximation $u_{pp} = \sum_i \phi_i v_i$ where the v_i are taken from the local spaces:

$$V_i = \mathcal{X}_i V_i^g \subset C^k(\Omega \cap \Omega_i) \subset H^1(\Omega \cap \Omega_i), \quad \forall i, \quad (k \geq 0), \tag{3.1}$$

where $\mathcal{X}_i$ is the characteristic function for Ω_i, and where

$$V_i^g \subset C^k(\Omega) \subset H^1(\Omega), \quad \forall i, \quad (k \geq 0). \tag{3.2}$$

In PPUM, the global spaces V_i^g in (3.1)–(3.2) are built from locally enriching an initial coarse global space V_0 by locally adapting the finite element mesh on which V_0 is built. (This is in contrast to classical overlapping Schwarz methods where local spaces are often built through enrichment of V_0 by locally adapting the mesh on which V_0 is built, and then removing the portions of the mesh exterior to the adapted region.) The PUM space V is then

$$\begin{aligned} V &= \left\{ v \mid v = \sum_i \phi_i v_i, \ v_i \in V_i \right\} \\ &= \left\{ v \mid v = \sum_i \phi_i \mathcal{X}_i v_i^g = \sum_i \phi_i v_i^g, \ v_i^g \in V_i^g \right\} \subset H^1(\Omega). \end{aligned}$$

In contrast to the approach in PUM where one seeks a global Galerkin solution in the PUM space as in (2.10), the PPUM algorithm described here and in [9] builds a global approximation u_{pp} to the solution to (2.9) from decoupled *local* Galerkin solutions:

$$u_{pp} = \sum_i \phi_i u_i = \sum_i \phi_i u_i^g, \tag{3.3}$$

where each u_i^g satisfies:

$$\text{Find } u_i^g \in V_i^g \text{ such that } \langle F(u_i^g), v_i^g \rangle = 0, \quad \forall v_i^g \in V_i^g. \tag{3.4}$$

We have the following global error estimate for the approximation u_{pp} in (3.3) built from (3.4) using the local PPUM parallel algorithm.

Theorem 3.1 *Assume the solution to (2.8) satisfies $u \in H^{1+\alpha}(\Omega)$, $\alpha > 0$, that quasi-uniform meshes of sizes h and $H > h$ are used for Ω_i^0 and $\Omega \backslash \Omega_i^0$ respectively, and that $diam(\Omega_i) \geq 1/Q > 0 \ \ \forall i$. If the local solutions are built from C^0 piecewise linear finite elements, then the global solution u_{pp} in (3.3) produced by Algorithm PPUM satisfies the following global error bounds:*

$$\begin{aligned} \|u - u_{pp}\|_{L^2(\Omega)} &\leq \sqrt{PM} C_\infty \left(C_1 h^\alpha + C_2 H^{1+\alpha} \right), \\ \|\nabla(u - u_{pp})\|_{L^2(\Omega)} &\leq \sqrt{2PM(Q^2 C_G^2 + C_\infty^2)} \left(C_1 h^\alpha + C_2 H^{1+\alpha} \right), \end{aligned}$$

where $P =$ number of local spaces V_i. Further, if $H \leq h^{\alpha/(1+\alpha)}$ then:

$$\begin{aligned} \|u - u_{pp}\|_{L^2(\Omega)} &\leq \sqrt{PM} C_\infty \max\{C_1, C_2\} h^\alpha, \\ \|\nabla(u - u_{pp})\|_{L^2(\Omega)} &\leq \sqrt{2PM(Q^2 C_G^2 + C_\infty^2)} \max\{C_1, C_2\} h^\alpha, \end{aligned}$$

so that the solution produced by Algorithm PPUM is of optimal order in the H^1-norm.

Proof. Viewing PPUM as a PUM gives access to the *a priori* estimates in Theorem 2.1; these require local estimates of the form:

$$\begin{aligned} \|u - u_i\|_{L^2(\Omega \cap \Omega_i)} = \|u - u_i^g\|_{L^2(\Omega \cap \Omega_i)} &\leq \epsilon_0(i), \\ \|\nabla(u - u_i)\|_{L^2(\Omega \cap \Omega_i)} = \|\nabla(u - u_i^g)\|_{L^2(\Omega \cap \Omega_i)} &\leq \epsilon_1(i). \end{aligned}$$

Such local *a priori* estimates are available for problems of the form (2.8) [17, 22]. They can be shown to take the following form:

$$\|u - u_i^g\|_{H^1(\Omega_i \cap \Omega)} \leq C \left(\inf_{v_i^0 \in V_i^0} \|u - v_i^0\|_{H^1(\Omega_i^0 \cap \Omega)} + \|u - u_i^g\|_{L^2(\Omega)} \right)$$

where

$$V_i^0 \subset C^k(\Omega_i^0 \cap \Omega) \subset H^1(\Omega_i \cap \Omega),$$

and where

$$\Omega_i \subset\subset \Omega_i^0, \qquad \Omega_{ij} = \Omega_i^0 \bigcap \Omega_i^0, \qquad |\Omega_{ij}| \approx |\Omega_i| \approx |\Omega_j|.$$

Since we assume $u \in H^{1+\alpha}(\Omega)$, $\alpha > 0$, and since quasi-uniform meshes of sizes h and $H > h$ are used for Ω_i^0 and $\Omega \backslash \Omega_i^0$ respectively, we have:

$$\begin{aligned} \|u - u_i^g\|_{H^1(\Omega_i \cap \Omega)} &= \left(\|u - u_i^g\|^2_{L^2(\Omega_i \cap \Omega)} + \|\nabla(u - u_i^g)\|^2_{L^2(\Omega_i \cap \Omega)} \right)^{1/2} \\ &\leq C_1 h^\alpha + C_2 H^{1+\alpha}. \end{aligned}$$

I.e., in this setting we can use $\epsilon_0(i) = \epsilon_1(i) = C_1 h^\alpha + C_2 H^{1+\alpha}$. The *a priori* PUM estimates in Theorem 2.1 then become:

$$\begin{aligned} \|u - u_{pp}\|_{L^2(\Omega)} &\leq \sqrt{M} C_\infty \left(\sum_i (C_1 h^\alpha + C_2 H^{1+\alpha})^2 \right)^{1/2}, \\ \|\nabla(u - u_{pp})\|_{L^2(\Omega)} &\leq \sqrt{2M} \\ &\cdot \left(\left[\sum_i \left(\frac{C_G}{\operatorname{diam}(\Omega_i)} \right)^2 + C_\infty^2 \right] (C_1 h^\alpha + C_2 H^{1+\alpha})^2 \right)^{1/2}. \end{aligned}$$

If $P =$ number of local spaces V_i, and if $\operatorname{diam}(\Omega_i) \geq 1/Q > 0 \ \ \forall i$, this is simply:

$$\begin{aligned} \|u - u_{pp}\|_{L^2(\Omega)} &\leq \sqrt{PM} C_\infty \left(C_1 h^\alpha + C_2 H^{1+\alpha} \right), \\ \|\nabla(u - u_{pp})\|_{L^2(\Omega)} &\leq \sqrt{2PM(Q^2 C_G^2 + C_\infty^2)} \left(C_1 h^\alpha + C_2 H^{1+\alpha} \right). \end{aligned}$$

If $H \leq h^{\alpha/(1+\alpha)}$ then u_{pp} from PPUM is asymptotically as good as a global Galerkin solution when the error is measured in the H^1-norm. ∎

Local versions of Theorem 3.1 appear in [22] for a variety of related parallel algorithms. Note that the local estimates in [22] hold more generally for nonlinear versions of (2.8), so that Theorem 3.1 can be shown to hold in a more general setting. Finally, it should be noted that improving the estimates in the L^2-norm is not generally possible; the required local estimates simply do not hold. Improving the solution quality in the L^2-norm generally requires more global information. However, for some applications one is more interested in a quality approximation of the *gradient* or the *energy* of the solution rather than to the solution itself.

4. Duality-based PPUM. We first briefly review a standard approach to the use of duality methods in error estimation. (cf. [6, 7] for a more complete discussion). Consider the weak formulation (2.9) involving a possibly nonlinear differential operator $F : H_0^1(\Omega) \mapsto H^{-1}(\Omega)$, and a Galerkin approximation u_{ap} satisfying (2.10). If $F \in C^1$, the generalized Taylor expansion exists:

$$F(u + h) = F(u) + \left\{ \int_0^1 DF(u + \xi h) d\xi \right\} h.$$

With $e = u - u_{ap}$, and with $F(u) = 0$, leads to the linearized error equation:

$$F(u_{ap}) = F(u - e) = F(u) + \mathcal{A}(u_{ap} - u) = -\mathcal{A}e,$$

where the linearization operator $\mathcal{A}$ is defined as:

$$\mathcal{A} = \int_0^1 DF(u + \xi h) d\xi.$$

Assume now we are interested in a linear functional of the error $l(e) = \langle e, \psi \rangle$, where ψ is the (assumed accessible) Riesz-representer of $l(\cdot)$. If $\phi \in H_0^1(\Omega)$ is the solution to the linearized dual problem:

$$\mathcal{A}^T \phi = \psi,$$

then we can exploit the linearization operator $\mathcal{A}$ and its adjoint $\mathcal{A}^T$ to give the following identity:

$$\langle e, \psi \rangle = \langle e, \mathcal{A}^T \phi \rangle = \langle \mathcal{A} e, \phi \rangle = -\langle F(u_{ap}), \phi \rangle. \tag{4.1}$$

If we can compute an approximation $\phi_{ap} \in \mathcal{V} \subset H_0^1(\Omega)$ to the linearized dual problem then we can estimate the error by combining this with the (computable) residual $F(u_{ap})$:

$$|\langle e, \psi \rangle| = |\langle F(u_{ap}), \phi \rangle| = |\langle F(u_{ap}), \phi - \phi_{ap} \rangle|,$$

where the last term is a result of (2.10). The term on the right is then estimated locally using assumptions on the quality of the approximation ϕ_{ap} and by various numerical techniques; cf. [6]. The local estimates are then used to drive adaptive mesh refinement. This type of duality-based error estimation has been shown to be useful for certain applications in engineering and other areas where accuracy in a linear functional of the solution is important, but accuracy in the solution itself is not (cf. [7]).

Consider now this type of error estimation in the context of domain decomposition and PPUM. Given a linear or nonlinear weak formulation as in (2.9), we are interested in the solution u as well as in the error in PPUM approximations u_{pp} as defined in (3.3)–(3.4). If a global linear functional $l(u - u_{pp})$ of the error $u - u_{pp}$ is of interest rather than the error itself, then we can formulate a variant of the PPUM parallel algorithm which has in some sense a more general approximation theory framework than that of the previous section. There are no assumptions beyond solvability of the local problems and of the global dual problems with localized data, and perhaps some minimal smoothness assumptions on the dual solution. In particular, the theory does not require local *a priori* error estimates; the local *a priori* estimates are replaced by solving global dual problem problems with localized data, and then incorporating the dual solutions explictly into the *a posteriori* error estimate. As a result, the large overlap assumption needed for the local estimates in the proof of Theorem 3.1 is unnecessary. Similarly, the large overlap assumption needed to achieve the bounded gradient property (2.6) is no longer needed.

The following result gives a global bound on a linear functional of the error based on satisfying local computable *a posteriori* bounds involving localized dual problems.

Theorem 4.1 *Let $\{\phi_i\}$ be a partition of unity subordinate to a cover $\{\Omega_i\}$. If ψ is the Riesz-representer for a linear functional $l(u)$, then the functional of the error in the PPUM approximation u_{pp} from (3.3) satisfies*

$$l(u - u_{pp}) = -\sum_{k=1}^{p} \langle F(u_i^g), \omega_i \rangle,$$

where u_i^g are the solutions to the subspace problems in (3.4), and where the ω_i are the solutions to the following global dual problems *with localized data:*

$$\textit{Find } \omega_i \in H_0^1(\Omega) \textit{ such that } (A^T \omega_i, v)_{L^2(\Omega)} = (\phi_i \psi, v)_{L^2(\Omega)}, \quad \forall v \in H_0^1(\Omega). \tag{4.2}$$

Moreover, if the local residual $F(u_i^g)$, weighted by the localized dual solution ω_i, satisfies the following error tolerance in each subspace:

$$|\langle F(u_i^g), \omega_i \rangle| < \frac{\epsilon}{p}, \qquad i = 1, \ldots, p \tag{4.3}$$

then the linear functional of the global error $u - u_{pp}$ satisfies

$$|l(u - u_{pp})| < \epsilon. \tag{4.4}$$

Proof. With $l(u - u_{pp}) = (u - u_{pp}, \psi)_{L^2(\Omega)}$, the localized representation comes from:

$$(u - u_{pp}, \psi)_{L^2(\Omega)} = (\sum_{k=1}^{p} \phi_i u - \sum_{i=1}^{p} \phi_i u_i^g, \psi)_{L^2(\Omega)} = \sum_{k=1}^{p} (\phi_i (u - u_i^g), \psi)_{L^2(\Omega \cap \Omega_i)}.$$

From (4.1) and (4.2), each term in the sum can be written in terms of the local residual $F(u_i^g)$ as follows:

$$\begin{aligned} (\phi_i (u - u_i^g), \psi)_{L^2(\Omega \cap \Omega_i)} &= (u - u_i^g, \phi_i \psi)_{L^2(\Omega \cap \Omega_i)} \\ &= (u - u_i^g, \mathcal{A}^T \omega_i)_{L^2(\Omega)} \\ &= (\mathcal{A}(u - u_i^g), \omega_i)_{L^2(\Omega)} \\ &= -(F(u_i^g), \omega_i)_{L^2(\Omega)}. \end{aligned}$$

This gives then

$$|(u - u_{pp}, \psi)_{L^2(\Omega)}| \leq \sum_{k=1}^{p} |\langle F(u_i^g), \psi \rangle| < \sum_{k=1}^{p} \frac{\epsilon}{p} = \epsilon.$$

■

We will make a few additional remarks about the parallel adaptive algorithm which arises naturally from Theorem 4.1. Unlike the case in Theorem 3.1, the constants C_∞ and C_G in (2.5) and (2.6) do not impact the error estimate in Theorem 4.1, removing the need for the *a priori* large overlap assumptions. Moreover, local *a priori* estimates are not required either, removing a second separate large overlap assumption that must be made to prove results such as Theorem 3.1. Using large overlap of *a priori* unknown size to satisfy the requirements for Theorem 3.1 seems unrealistic for implementations. On the other hand, no such *a priori* assumptions are required to use the result in Theorem 4.1 as the basis for a parallel adaptive algorithm. One simply solves the local dual problems (4.2) on each processor independently, adapts the mesh on each processor independently until the computable local error estimate satisfies the tolerance (4.3), which then guarantees that the functional of the global error meets the target in (4.4).

Whether such a duality-based approach will produce an efficient parallel algorithm is not at all clear; however, it is at least a mechanism for decomposing the solution to an elliptic problem over a number of subdomains. Note that ellipticity is not used in Theorem 4.1, so that the approach is also likely reasonable for other classes of PDE. These questions, together with a number of related duality-based decomposition algorithms are examined in more detail in [5]. The analysis in [5] is based on a different approach involving estimates of Green function decay rather than through partition of unity methods.

5. Implementation in FETK **and** MC**.** Our implementations are performed using FETK and MC (see [9] for a more complete discussion of MC and FETK). MC is the adaptive multilevel finite element software kernel within FETK, a large collection of collaboratively developed finite element software tools based at UC San Diego (see `www.fetk.org`). MC is written in ANSI C (as is most of FETK), and is designed to produce highly accurate numerical solutions to nonlinear covariant elliptic systems of tensor equations on 2- and 3-manifolds in an optimal or nearly-optimal way. MC employs *a posteriori* error estimation, adaptive simplex subdivision, unstructured algebraic multilevel methods, global inexact Newton methods, and numerical continuation methods. Several of the features of MC are somewhat unusual, allowing for the treatment of very general nonlinear elliptic systems of tensor equations on domains with the structure of (Riemannian) 2- and 3-manifolds. Some of these features are:

- *Abstraction of the elliptic system*: The elliptic system is defined only through a nonlinear weak form over the domain manifold, along with an associated linearization form, also defined everywhere on the domain manifold (precisely the forms $\langle F(u), v\rangle$ and $\langle DF(u)w, v\rangle$ in the discussions above).

- *Abstraction of the domain manifold*: The domain manifold is specified by giving a polyhedral representation of the topology, along with an abstract set of coordinate labels of the user's interpretation, possibly consisting of multiple charts. MC works only with the topology of the domain, the connectivity of the polyhedral representation. The geometry of the domain manifold is provided only through the form definitions, which contain the manifold metric information.

- *Dimension independence*: Exactly the same code paths in MC are taken for both two- and three-dimensional problems (as well as for higher-dimensional problems). To achieve this dimension independence, MC employs the simplex as its fundamental geometrical object for defining finite element bases.

As a consequence of the abstract weak form approach to defining the problem, the complete definition of a complex nonlinear tensor system such as large deformation nonlinear elasticity requires writing only a few hundred lines of C to define the two weak forms. Changing to a different tensor system (e.g. the example later in the paper involving the constraints in the Einstein equations) involves providing only a different definition of the forms and a different domain description.

A datastructure referred to as the *ringed-vertex* (cf. [9]) is used to represent meshes of d-simplices of arbitrary topology. This datastructure is illustrated in Figure 5.1. The ringed-vertex datastructure is similar to the winged-edge, quad-edge, and edge-facet datastructures commonly used in the computational geometry community for representing 2-manifolds [15], but it can be used more generally to represent arbitrary d-manifolds, $d \geq 2$. It maintains a mesh of d-simplices with near minimal storage, yet for shape-regular (non-degenerate) meshes, it provides $O(1)$-time access to all information necessary for refinement, un-refinement, and Petrov-Galerkin discretization of a differential operator. The ringed-vertex datastructure also allows for dimension independent implementations of mesh refinement and mesh manipulation, with one implementation (the same code path) covering arbitrary dimension d. An interesting feature of this datastructure is that the C structures used for vertices, simplices,

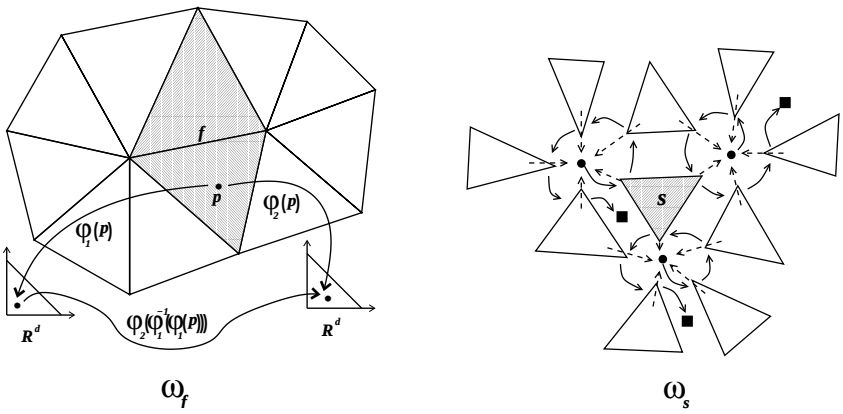


Figure 5.1: Polyhedral manifold representation. The figure on the left shows two overlapping polyhedral (vertex) charts consisting of the two rings of simplices around two vertices sharing an edge. The region consisting of the two darkened triangles around the face f is denoted ω_f, and represents the overlap of the two vertex charts. Polyhedral manifold topology is represented by MC using the *ringed-vertex* (or *RIVER*) datastructure. The datastructure is illustrated for a given simplex s in the figure on the right; the topology primitives are vertices and d-simplices. The collection of the simplices which meet the simplex s at its vertices (which then includes those simplices that share faces as well) is denoted as ω_s.

and edges are all of fixed size, so that a fast array-based implementation is possible, as opposed to a less-efficient list-based approach commonly taken for finite element implementations on unstructured meshes. A detailed description of the ringed-vertex datastructure, along with a complexity analysis of various traversal algorithms, can be found in [9].

Our modifications to MC to implement PPUM are minimal, and are described in detail in [4]. These modifications involve primarily forcing the error indicator to ignore regions outside the subdomain assigned to the particular processor. The implementation does not form an explicit partition of unity or a final global solution; the solution must be evaluated locally by locating the disjoint subdomain containing the physical region of interest, and then by using the solution produced by the processor assigned to that particular subdomain. Note that forming a global conforming mesh as needed to build a global partition of unity is possible even in a very loosely coupled parallel environment, due to the deterministic nature of the bisection-based algorithms we use for simplex subdivision (see [9]). For example, if bisection by longest edge (supplemented with tie-breaking) is used to subdivide any simplex that is refined on any processor, then the progeny types, shapes, and configurations can be predicted in a completely determinstic way. If two simplices share faces across a subdomain boundary, then they are either compatible (their triangular faces exactly match), or one of the simplices has been bisected more times than its neighbor. By exchanging only the generation numbers between subdomains, a global conforming mesh can be reached using only additional bisection.

6. Example 1: The Einstein Constraints in Gravitation. The evolution of the gravitational field was conjectured by Einstein to be governed by twelve coupled first-order hyperbolic equations for the metric of space-time and its time derivative, where the evolution is constrained for all time by a coupled four-component elliptic system. The theory basically gives what is viewed as the correct interpretation of the graviational field as a bending of space and time around matter and energy, as opposed to the classical Newtonian view of the gravitational field as analogous to the electrostatic field; cf. Figure 6.1. The four-component elliptic constraint system

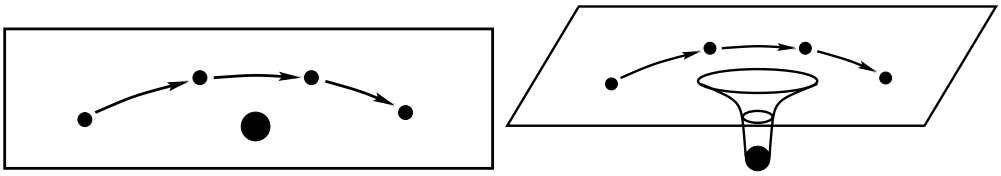


Figure 6.1: Newtonian versus general relativistic explanations of gravitation: the small mass simply follows a geodesic on the curved surface created by the large mass.

consists of a nonlinear scalar *Hamiltonian constraint*, and a linear 3-vector *momentum* constraint. The evolution and constraint equations, similar in some respects to Maxwell's equations, are collectively referred to as the *Einstein equations*. Solving the constraint equations numerically, separately or together with the evolution equations, is currently of great interest to the physics community due to the recent construction of a new generation of gravitational wave detectors (cf. [12, 11] for more detailed discussions of this application).

Allowing for both Dirichlet and Robin boundary conditions on a 3-manifold $\mathcal{M}$ with boundary $\partial\mathcal{M} = \partial_0\mathcal{M} \cup \partial_1\mathcal{M}$, as typically the case in black hole and neutron star models (cf. [12, 11]), the strong form of the constraints can be written as:

$$\begin{aligned}
\hat{\Delta}\phi &= \frac{1}{8}\hat{R}\phi + \frac{1}{12}(\mathrm{tr}K)^2\phi^5 && (6.1)\\
&\quad -\frac{1}{8}(\overset{*}{\hat{A}}_{ab} + (\hat{L}W)_{ab})^2\phi^{-7} - 2\pi\hat{\rho}\phi^{-3} \ \text{ in } \mathcal{M},\\
\hat{n}_a\hat{D}^a\phi + c\phi &= z \text{ on } \partial_1\mathcal{M}, && (6.2)\\
\phi &= f \text{ on } \partial_0\mathcal{M}, && (6.3)\\
\hat{D}_b(\hat{L}W)^{ab} &= \frac{2}{3}\phi^6\hat{D}^a\mathrm{tr}K + 8\pi\hat{j}^a \ \text{ in } \mathcal{M}, && (6.4)\\
(\hat{L}W)^{ab}\hat{n}_b + C^a_{\ b}W^b &= Z^a \text{ on } \partial_1\mathcal{M}, && (6.5)\\
W^a &= F^a \ \text{ on } \partial_0\mathcal{M}, && (6.6)
\end{aligned}$$

where the following standard notation has been employed:

$$\begin{aligned}
\hat{\Delta}\phi &= \hat{D}_a\hat{D}^a\phi,\\
(\hat{L}W)^{ab} &= \hat{D}^aW^b + \hat{D}^bW^a - \frac{2}{3}\hat{\gamma}^{ab}\hat{D}_cW^c,\\
\mathrm{tr}K &= \gamma^{ab}K_{ab},\\
(C_{ab})^2 &= C^{ab}C_{ab}.
\end{aligned}$$

In the tensor expressions above, there is an implicit sum on all repeated indices in products, and the covariant derivative with respect to the fixed background metric $\hat{\gamma}_{ab}$ is denoted as $\hat{D}_a$ The remaining symbols in the equations ($\hat{R}$, K, ${}^*\hat{A}_{ab}$, $\hat{\rho}$, $\hat{j}^a$, z, Z^a, f, F^a, c, and C^a_b) represent various physical parameters, and are described in detail in [12, 11] and the referenences therein. Stating the system as set of tensor equations comes from the need to work with domains which generally have the structure of 3-manifolds rather than single open sets in $\mathbb{R}^3$ (cf. [9]).

Equations (6.1)–(6.6) are known to be well-posed only for certain problem data and manifold topologies [16, 13]. Note that if multiple solutions in the form of folds or bifurcations are present in solutions of (6.1)–(6.6) then path-following numerical methods will be required for numerical solution [14]. For our purposes here, we select the problem data and manifold topology such that the assumptions for the two general well-posedness results in [12] hold for (6.1)–(6.6). The assumptions required for the two results in [12] are quite weak, and are, for the most part, minimal assumptions beyond those required to give a well-defined weak formulation in L^p-based Sobolev spaces.

In [9], two quasi-optimal *a priori* error estimates are established for Galerkin approximations to the solutions to (6.1)–(6.6). These take the form (see Theorems 4.3 and 4.4 in [9]):

$$\|u - u_h\|_{H^1(\mathcal{M})} \leq C \inf_{v \in V_h} \|u - v\|_{H^1(\mathcal{M})} \tag{6.7}$$

$$\|u - u_h\|_{L^2(\mathcal{M})} \leq C a_h \inf_{v \in V_h} \|u - v\|_{H^1(\mathcal{M})}, \tag{6.8}$$

where $V_h \subset H^1(\mathcal{M})$ is e.g. a finite element space. In the case of the momentum constraint, there is a restriction on the size of the elements in the underlying finite element mesh for the above results to hold, characterized above by the parameter a_h. This restriction is due to the fact that the result is established through of the Gårding inequality result due to Schatz [18]. In the case of the Hamiltonian constraint, there are no restrictions on the approximation spaces.

To use MC to calculate the initial bending of space and time around a single massive black hole by solving the above constraint equations, we place a spherical object of unit radius in space, and infinite space is truncated with an enclosing sphere of radius 100. (This outer boundary may be moved further from the object to improve the accuracy of boundary condition approximations.) Reasonable choices for the remaining functions and parameters appearing in the equations are used below to completely specify the problem for use as an illustrative numerical example. (More careful examination of the various functions and parameters appear in [12], and a number of detailed experiments with more physically meaningful data appear in [11].)

We then generate an initial (coarse) mesh of tetrahedra inside the enclosing sphere, exterior to the spherical object within the enclosing sphere. The mesh is generated by adaptively bisecting an initial mesh consisting of an icosahedron volume filled with tetrahedra. The bisection procedure simply bisects any tetrahedron which touches the surface of the small spherical object. When a reasonable approximation to the surface of the sphere is obtained, the tetrahedra completely inside the small spherical object are removed, and the points forming the surface of the small spherical object are projected to the spherical surface exactly. This projection involves solving a linear

elasticity problem, together with the use of a shape-optimization-based smoothing procedure. The smoothing procedure locally optimizes the shape measure function described in [9] in an iterative fashion. A much improved binary black hole mesh generator has been developed by D. Bernstein; the new mesh generator is described in [11] along with a number of more detailed examples using MC.

The initial coarse mesh is shown in Figure 6.2, generated using the procedure described above, has approximately 30,000 tetrahedral elements and 5,000 vertices. To solve the problem on a 4-processor computing cluster using a PPUM-like algorithm, we begin by partitioning the domain into four subdomains (shown in Figure 6.3) with approximately equal error using the recursive spectral bisection algorithm described in [4]. The four subdomain problems are then solved independently by MC, starting from the complete coarse mesh and coarse mesh solution. The mesh is adaptively refined in each subdomain until a mesh with roughly 50000 vertices is obtained (yielding subdomains with about 250000 simplices each).

The refinement performed by MC is confined primarily to the given region as driven by the weighted residual error indicator described in [9], with some refinement into adjacent regions due to the closure algorithm which maintains conformity and shape regularity. The four problems are solved completely independently by the sequential adaptive software package MC. One component of the solution (the conformal factor ϕ) of the elliptic system is depicted in Figures 6.4 (the subdomain 0 and subdomain 1 solutions).

A number of more detailed examples involving the contraints, using more physically meaningful data, appear in [11]. Application of PPUM to massively parallel simulations of microtubules and other extremely large and complex biological structures can be found in [3, 2]. The results in [3, 2] demonstrate both good parallel scaling of PPUM as well as quality approximation of the gradient of electrostatic potentials (solutions to the Poisson-Boltzmann equation; cf. [10]).

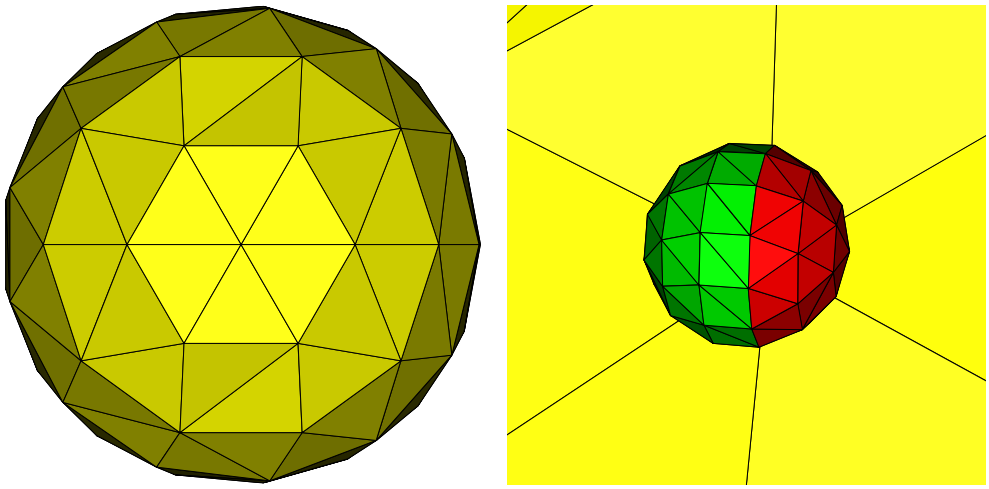


Figure 6.2: Recursize spectral bisection of the single hole domain into four subdomains (boundary surfaces of three of the four subdomains are shown).

REFERENCES

[1] I. Babuška and J. M. Melenk. The partition of unity finite element method. *Internat. J. Numer. Methods Engrg.*, 40:727–758, 1997.

[2] N. Baker, D. Sept, M. J. Holst, and J. A. McCammon. The adaptive multilevel finite element solution of the Poisson-Boltzmann equation on massively parallel computers. *IBM Journal of Research and Development*, 45:427–438, 2001.

[3] N. Baker, D. Sept, S. Joseph, M. J. Holst, and J. A. McCammon. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proc. Natl. Acad. Sci. USA*, 98:10037–10041, 2001.

[4] R. E. Bank and M. J. Holst. A new paradigm for parallel adaptive mesh refinement. *SISC*, 22(4):1411–1443, 2000.

[5] D. Estep, M. J. Holst, and M. Larson. *Solution Decomposition using Localized Influence Functions*. In Preparation.

[6] D. Estep, M. J. Holst, and D. Mikulencak. Accounting for stability: *a posteriori* error estimates for finite element methods based on residuals and variational analysis. *Communications in Numerical Methods in Engineering*, 18(1):15–30, 2002.

[7] M. B. Giles and E. Süli. Adjoint methods for pdes: a posteriori error analysis and postprocessing by duality. *Acta Numerica*, 2002.

[8] M. Griebel and M. A. Schweitzer. A particle-partition of unity method for the solution of elliptic, parabolic, and hyperbolic PDEs. *SISC*, 22(3):853–890, 2000.

[9] M. J. Holst. Adaptive numerical treatment of elliptic systems on manifolds. *Advances in Computational Mathematics*, 15:139–191, 2001.

[10] M. J. Holst, N. Baker, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I: algorithms and examples. *J. Comput. Chem.*, 21:1319–1342, 2000.

[11] M. J. Holst and D. Bernstein. *Adaptive Finite Element Solution of the Initial Value Problem in General Relativity I. Algorithms*. In Preparation.

[12] M. J. Holst and D. Bernstein. *Some results on non-constant mean curvature solutions to the Einstein constraint equations*. In Preparation.

[13] J. Isenberg and V. Moncrief. A set of nonconstant mean curvature solutions of the Einstein constraint equations on closed manifolds. *Classical and Quantum Gravity*, 13:1819–1847, 1996.

[14] H. B. Keller. *Numerical Methods in Bifurcation Problems*. Tata Institute of Fundamental Research, 1987.

[15] E. P. Mücke. *Shapes and Implementations in Three-Dimensional Geometry*. PhD thesis, Department of Computer Science, University of Illinois at Urbana-Champaign, 1993.

[16] N. Murchadha and J. W. York. Existence and uniqueness of solutions of the Hamiltonian constraint of general relativity on compact manifolds. *J. Math. Phys.*, 14(11):1551–1557, 1973.

[17] J. A. Nitsche and A. H. Schatz. Interior estimates for Ritz-Galerkin methods. *Math. Comp.*, 28:937–958, 1974.

[18] A. H. Schatz. An observation concerning Ritz-Galerkin methods with indefinite bilinear forms. *Math. Comp.*, 28(128):959–962, 1974.

[19] D. S. Shepard. A two-dimensional interpolation function for irregularly spaced data. In *Proceedings of the 1968 ACM National Conference, New York*, pages 517–524, 1968.

[20] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, December 1992.

[21] J. Xu. Two-grid discretization techniques for linear and nonlinear pdes. *SIAM J. N. A.*, 33:1759–1777, 1996.

[22] J. Xu and A. hui Zhou. Local and parallel finite element algorithms based on two-grid discretizations. *Math. Comput.*, 69:881–909, 2000.

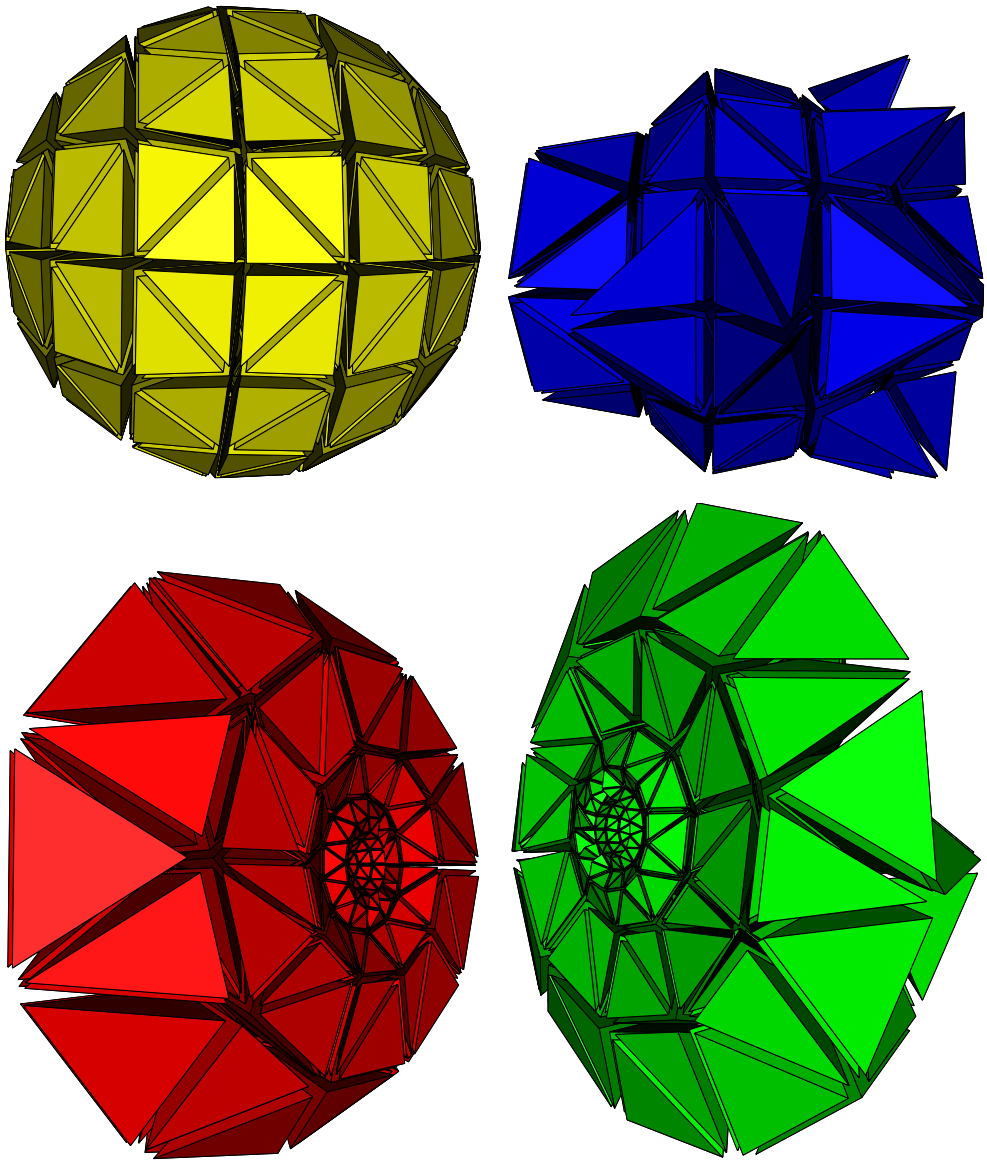


Figure 6.3: Recursize spectral bisection of the single hole domain into four subdomains.

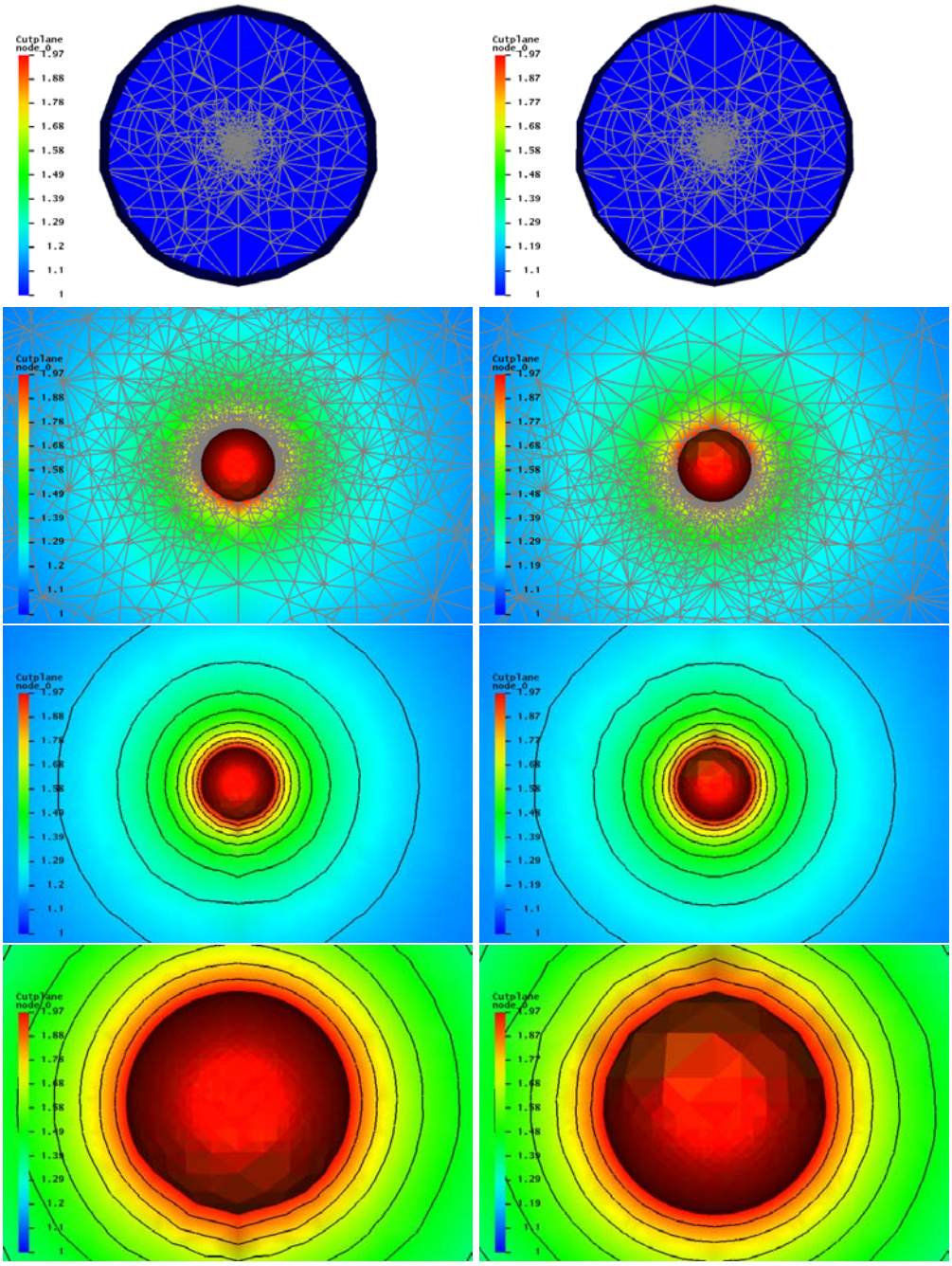


Figure 6.4: Decoupling of the scalar conformal factor in the initial data using PPUM; domain 0 in the left column, and domain 1 on the right.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

7. Domain Decomposition in the Mainstream of Computational Science

D. E. Keyes[1] [2]

1. Introduction. Computational peak performance on full-scale scientific applications, as tracked by the Gordon Bell prize, has increased by four orders of magnitude since the prize was first awarded in 1988 — twenty-five times faster than can accounted for by Moore's Law alone. The extra factor comes from process concurrency, which is as much as 8,192-fold on the $100M "ASCI White" machine at Lawrence Livermore, currently ranked as the world's second most powerful, after the new Japanese "Earth Simulator". The latter was recently clocked at more than 35 trillion floating point operations per second (Tflop/s) on the LINPACK benchmark and at 26.6 Tflop/s on a climate application [27]. Though architectural concurrency is easy to achieve, algorithmic concurrency to match is less so in scientific codes. Intuitively, this is due to global domains of influence in many problems presented to the computer as implicitly discretized operator equations — implicitness being all but legislated for the multiscale systems of global climate, transonic airliners, petroleum reservoirs, tokamaks, etc., the simulation of which justifies expenditures for the highest-end machines.

A key requirement of candidate solution algorithms is mathematical optimality. This means a convergence rate as independent as possible of discretization parameters. In practice, linear systems require a hierarchical, multilevel approach to obtain rapid linear convergence. Nonlinear systems require a Newton-like approach to obtain asymptotically quadratic convergence. The concept of optimality can also be extended into the physical modeling regime to include continuation schemes and physics-informed preconditioning, so that multiple scale problems are attacked with a manageable number of scales visible to the numerics at any given stage.

In this context, optimal parallel algorithms for PDE simulations of Jacobian-free Newton-Krylov type, preconditioned with Schwarz and Schur domain decompositions, including multilevel generalizations of Schwarz, are coming into prominence. One of the main benefits of the Jacobian-free Newton-Krylov approach is the exploitation of multiple discrete representations of the underlying continuous operator, the idea being to converge fully to a representation of high fidelity through a series of inexpensive and stable steps based on representations of lower fidelity. Simultaneous advances in object-oriented software engineering have enabled the construction of internally complex software systems in which these algorithmic elements can be combined modularly, recursively, and relatively efficiently in parallel, while presenting a programming environment that allows the user to function at a rather high level. For large systems with strong nonlinearities robustification techniques have been developed, including pseudo-transient continuation, parameter continuation, grid sequencing, model sequencing,

[1]Mathematics & Statistics Department, Old Dominion University, Norfolk, VA 23529-0077 and ISCR, Lawrence Livermore Nat. Lab., Livermore, CA 94551-9989 and ICASE, NASA Langley Res. Ctr., Hampton, VA 23681-2199, keyes@icase.edu

[2]Supported in part by the U.S. Department of Energy under SciDAC subcontract FC02-01ER25476 to Old Dominion University, by Lawrence Livermore National Laboratory under ASCI Level-2 subcontract B347882 to Old Dominion University, and by NASA under contract NAS1-19480 to ICASE.

and nonlinear preconditioning. These improvements in nonlinear rootfinding have made it possible for large-scale PDE-constrained optimization problems (e.g., design, control, parameter identification — usually the ultimate problems behind the proximate PDEs) to be placed into the same domain-decomposed algorithmic framework as the PDE, itself.

The architecture of the terascale systems available in the United States, built around hierarchical distributed memory, appears hostile to conventional sequential optimal PDE algorithms, but is ultimately suitable apart from reservations about limited memory bandwidth. The distributed aspects must be overcome with judicious combinations of message-passing and/or shared memory program models. The hierarchical aspects must be overcome with register blocking, cache blocking, and prefetching. Algorithms for these PDE-based simulations must be highly concurrent, straightforward to load balance, latency tolerant, cache friendly (with strong temporal and spatial locality of reference), and highly scalable (in the sense of convergence rate) as problem size and processor number are increased in proportion. The goal for algorithmic scalability is to fill up the memory of arbitrarily large machines while preserving constant (or at most logarithmically growing) running times with respect to a proportionally smaller problem on one processor. Domain-decomposed multilevel methods are natural for all of these *desiderata*. Domain decomposition is also natural for the software engineering of simulation codes: valuable extent code designed for a sequential PDE analysis can often be "componentized" and made part of an effective domain-decomposed, operator-split preconditioner.

For a pair of web-downloadable full-scale reviews documenting these themes more fully, see [17, 18]. This page-limited chapter skims these reviews at a high level, emphasizing the importance of domain decomposition to large-scale scientific computing.

2. The Newton-Krylov-Schwarz Family of Algorithms. Many problems in engineering and applied physics can be written in the form

$$V\frac{\partial \mathbf{u}}{\partial t} + \mathcal{F}(\mathbf{u}) = 0, \tag{2.1}$$

where $\mathbf{u}$ is a vector of functions depending upon spatial variables $\mathbf{x}$ and t, $\mathcal{F}$ is a vector of spatial differential operators acting on $\mathbf{u}$, and V is a diagonal scaling matrix with nonnegative diagonal entries. If all of the equations are "prognostic" then V has strictly positive diagonal entries; but we may also accommodate the case of entirely steady-state equations, $V = 0$, or some combination of positive and zero diagonal entries, corresponding to prognostic equations for some variables and steady-state constraints for others. Steady-state equations often arise from *a priori* equilibrium assumptions designed to suppress timescales faster than those of dynamical interest, e.g., acoustic waves in aerodynamics, gravity waves in geophysics, Alfvén waves in magnetohydrodynamics, etc.

Semidiscretizing in space to approximate $\mathcal{F}(\mathbf{u})$ with $\mathbf{f}(\mathbf{u})$, and in time with implicit Euler, we get the algebraic system:

$$(\frac{V}{\tau^\ell})\mathbf{u}^\ell + \mathbf{f}(\mathbf{u}^\ell) = (\frac{V}{\tau^\ell})\mathbf{u}^{\ell-1}. \tag{2.2}$$

Higher-order temporal schemes are easily put into this framework with the incorporation of additional history vectors with appropriate weights on the right-hand side. We

are not directly concerned with discretization or the adaptivity of the discretization to the solution in this chapter. However, the achievement of nonlinear consistency by Newton's method on each time step is motivated by a desire to go to higher order than the pervasive standard of no better than first-order in time and second-order in space. Because $\mathbf{f}$ may be highly nonlinear, even a steady-state numerical analysis is often made to follow a pseudo-transient continuation until the ball of convergence for Newton's method for the steady-state problem is reached. In this case, time accuracy is not an issue, and τ^ℓ becomes a parameter to be placed at the service of the algorithm [16].

Whether discretized time accurately or not, we are left at each time step with a system of nonlinear algebraic equations (2.2), written abstractly as $\mathbf{F}^\ell(\mathbf{u}^\ell) = 0$. We solve these systems in sequence for each set of discretized spatial grid functions, $\mathbf{u}^\ell$, using an inexact Newton method. The resulting linear systems for the Newton corrections involving the Jacobian of $\mathbf{F}^\ell$ with respect to instantaneous or lagged iterates $\mathbf{u}^{\ell,k}$, are solved with a Krylov method, relying only on Jacobian-vector multiplications. (Here, $\mathbf{u}^{\ell,0} \equiv \mathbf{u}^{\ell-1}$, and $\mathbf{u}^{\ell,k} \rightarrow \mathbf{u}^\ell$, as $k \rightarrow \infty$ in a Newton iteration loop on inner index k.) The Krylov method needs to be preconditioned for acceptable inner iteration convergence rates, and the preconditioning is the "make-or-break" aspect of an implicit code. The other phases possess high concurrency and parallelize well already, if properly load balanced, being made up of vector updates, inner products, and sparse matrix-vector products.

The job of the preconditioner is to approximate the action of the Jacobian inverse in a way that does not make it the dominant consumer of memory or cycles in the overall algorithm and (most importantly) does not introduce idleness through chained data dependencies, as in Gaussian elimination. The true inverse of the Jacobian is usually dense, reflecting the global Green's function of the continuous linearized PDE operator it approximates, and it is not obvious that a good preconditioner approximating this inverse action can avoid extensive global communication. A good preconditioner saves time and space by permitting fewer iterations in the Krylov loop and smaller storage for the Krylov subspace than would be required in its absence. An additive Schwarz preconditioner accomplishes this in a localized manner, with an approximate solve in each subdomain of a partitioning of the global PDE domain. Applying any subdomain preconditioner within an additive Schwarz framework tends to increases floating point rates over the same preconditioner applied globally, since the smaller subdomain blocks maintain better cache residency. Combining a Schwarz preconditioner with a Krylov iteration method inside an inexact Newton method leads to a synergistic parallelizable nonlinear boundary value problem solver with a classical name: Newton-Krylov-Schwarz (NKS). In the remainder of this section, we build up NKS from the outside inwards.

Inexact Newton Methods. We use the term "inexact Newton method" to denote any nonlinear iterative method for solving $\mathbf{F}(\mathbf{u}) = 0$ through a sequence $\mathbf{u}^k = \mathbf{u}^{k-1} + \lambda^k \delta\mathbf{u}^k$, where $\delta\mathbf{u}^k$ approximately satisfies the true Newton correction equation

$$\mathbf{F}'(\mathbf{u}^{k-1})\delta\mathbf{u}^k = -\mathbf{F}(\mathbf{u}^{k-1}), \tag{2.3}$$

in the sense that the linear residual norm $||\mathbf{F}'(\mathbf{u}^{k-1})\delta\mathbf{u}^k + \mathbf{F}(\mathbf{u}^{k-1})||$ is sufficiently small. Typically the right-hand side of the linear Newton correction equation, which is the nonlinear residual $\mathbf{F}(\mathbf{u}^{k-1})$, is evaluated to full precision, so the inexactness

arises from incomplete convergence employing the true Jacobian, freshly evaluated at $\mathbf{u}^{k-1}$, or from the employment of an inexact Jacobian for $\mathbf{F}'(\mathbf{u}^{k-1})$.

Newton-Krylov Methods. A Newton-Krylov (NK) method uses a Krylov method, such as GMRES [26], to solve (2.3) for $\delta\mathbf{u}^{\ell}$. From a computational point of view, one of the most important characteristics of a Krylov method for the linear system $Ax = b$ is that information about the matrix A needs to be accessed only in the form of matrix-vector products in a relatively small number of carefully chosen directions. When the matrix A represents the Jacobian of a discretized system of PDEs, each of these matrix-vector products is similar in computational and communication cost to a stencil update phase (or "global flux balance") of an explicit method applied to the same set of discrete conservation equations, or to a single finest-grid "work unit" in a multigrid method. NK methods are suited for nonlinear problems in which it is unreasonable to compute or store a true full Jacobian, where the action of A can be approximated by discrete directional derivatives.

Newton-Krylov-Schwarz Methods. A Newton-Krylov-Schwarz (NKS) method combines a Newton-Krylov method, such as Newton-GMRES [6], with a Krylov-Schwarz (KS) method, such as restricted additive Schwarz [9]. If the Jacobian A is ill-conditioned, the Krylov method will require an unacceptably large number of iterations. In order to control the number of Krylov iterations, while obtaining concurrency proportional to the number of processors, they are preconditioned with domain-decomposed additive Schwarz methods [28]. The system is transformed into the equivalent form $B^{-1}Ax = B^{-1}b$ through the action of a preconditioner, B, whose inverse action approximates that of A, but at smaller cost. It is in the choice of preconditioning that the battle for low computational cost and scalable parallelism is usually won or lost. In KS methods, the preconditioning is introduced on a subdomain-by-subdomain basis through a conveniently computable approximation to a local Jacobian. Such Schwarz-type preconditioning provides good data locality for parallel implementations over a range of parallel granularities, allowing significant architectural adaptability.

Schwarz Methods. Schwarz methods [7, 11, 28, 32] create concurrency at a desired granularity algorithmically and explicitly through partitioning, without the necessity of any code dependence analysis or special compiler. Generically, in continuous or discrete settings, Schwarz partitions a solution space into n subspaces, possibly overlapping, whose union is the original space, and forms an approximate inverse of the operator in each subspace. Algebraically, to solve the discrete linear system, $Ax = f$, let Boolean rectangular matrix R_i extract the i^{th} subset of the elements of x defining an algebraic subspace: $x_i = R_i x$, and let $A_i \equiv R_i A R_i^T$ be invertible within the i^{th} subspace. Then the additive Schwarz approximate inverse is defined as

$$B_{ASM}^{-1} = \sum_i {R_i}^T A_i^{-1} R_i. \tag{2.4}$$

From the PDE perspective, subspace decomposition is domain decomposition. B^{-1} is formed out of (approximate) local solves on (possibly overlapping) subdomains.

In the grid-based context of a PDE, Boolean operators R_i and R_i^T, $i = 1, \ldots, n$, represent gather and scatter (communication) operations, mapping between a global vector and its i^{th} subdomain support. When A derives from an elliptic operator and R_i is the characteristic function of unknowns in a subdomain, optimal convergence

(independent of $\dim(x)$ and the number of partitions) can be proved, with the addition of a coarse grid, which is denoted with subscript "0": $B_{ASM}^{-1} = R_0^T A_0^{-1} R_0 + \sum_{i>0} R_i^T A_i^{-1} R_i$. Here, R_0 is a conventional geometrically based multilevel interpolation operator. It is an important freedom in practical implementations that the coarse grid space need not be related to the fine grid space or to the subdomain partitioning. The $A_i^{-1} (i > 0)$ in B^{-1} are often replaced with inexact solves in practice, such as a multigrid V-cycle. The exact forward matrix-vector action of A in $B^{-1}A$ is still required, even if inexact solves are employed in the preconditioner.

Table 2.1: *Theoretical condition number estimates $\kappa(B^{-1}A)$, for self-adjoint positive-definite elliptic problems [28] and corresponding iteration count estimates for Krylov-Schwarz based on an idealized isotropic partitioning of the domain in dimensions 2 or 3.*

Preconditioning	$\kappa(B^{-1}A)$	*2D Iter.*	*3D Iter.*
Point Jacobi	$\mathcal{O}(h^{-2})$	$\mathcal{O}(N^{1/2})$	$\mathcal{O}(N^{1/3})$
Domain Jacobi	$\mathcal{O}((hH)^{-1})$	$\mathcal{O}((NP)^{1/4})$	$\mathcal{O}((NP)^{1/6})$
1-level Additive Schwarz	$\mathcal{O}(H^{-2})$	$\mathcal{O}(P^{1/2})$	$\mathcal{O}(P^{1/3})$
2-level Additive Schwarz	$\mathcal{O}(1)$	$\mathcal{O}(1)$	$\mathcal{O}(1)$

Condition number estimates for $B^{-1}A$ are given in the first column of Table 1 in terms of the quasi-uniform mesh parameter h, and subdomain parameter H. The two-level Schwarz method with generous overlap has a condition number that is independent of the fineness of the discretization and the granularity of the decomposition, which implies perfect algorithmic scalability. However, there is an increasing implementation overhead in the coarse-grid solution required in the two-level method that offsets this perfect algorithmic scalability. In practice, a one-level method is often used, since it is amenable to a perfectly scalable implementation. Alternatively, a two-level method is used but the coarse level is solved only approximately, in a tradeoff that depends upon the application and the architecture. These condition number results are extensible to nonself-adjointness, mild indefiniteness, and inexact subdomain solvers. The theory requires a "sufficiently fine" coarse mesh, H, for the first two of these extensions, but computational experience shows that the theory is often pessimistic.

The *restricted* additive Schwarz Method (RASM) eliminates interprocess communication during the interpolation phase of the additive Schwarz technique [9]. In particular, if we decompose a problem into a set of overlapping subdomains Ω_i, the conventional additive Schwarz method is a three-phase process consisting of first collecting data from neighboring subdomains via global-to-local restriction operators R_i, then performing a local linear solve on each subdomain A_i^{-1}, and finally sending partial solutions to neighboring subdomains via the local-to-global prolongation operators ${R_i}^T$. The RASM preconditioner performs a complete restriction operation but does not use any communication during the interpolation phase, denoted instead as ${R'_i}^T$. This provides the obvious benefit of a 50% reduction in nearest-neighbor communication overhead. In addition, experimentally, it preconditions better than the original additive Schwarz method over a broad class of problems [9], for reasons that are beginning to be understood [8].

Although the spectral radius, $\rho(I - B^{-1}A)$, may exceed unity, the spectrum, $\sigma(B^{-1}A)$, is profoundly clustered, so Krylov acceleration methods work well on the preconditioned solution of $B^{-1}Ax = B^{-1}f$. Krylov-Schwarz methods typically converge in a number of iterations that scales as the square-root of the condition number of the Schwarz-preconditioned system. For convergence scalability estimates, assume one subdomain per processor in a d-dimensional isotropic problem, where $N = h^{-d}$ and $P = H^{-d}$. Then iteration counts may be estimated as in the last two columns of Table 1.

The proof of these estimates is generally approached via an algebra of projection operators, $P_i \equiv {R_i}^T A_i^{-1} R_i A$. The ratio of upper bound to lower bound of the spectrum of the sum of the orthogonal projections P_i is an estimate of the condition number for $B^{-1}A = \sum_i P_i$. Since $||P_i|| \leq 1$, the upper bound follows easily from the geometry of the decomposition and is a generally a constant related to the number of colors required to color the subdomains. The lower bound depends crucially upon the partitioning of the solution space. Without a coarse subspace to support the solution at subdomain boundaries, the fine space contributions must fall rapidly to zero from finite values in the subdomain interiors, resulting in high H^1 "energy" inversely proportional to the overlap distance over which the solutions must decay.

For simple intuition behind this table consider the following: errors propagate from the interior to the boundary in steps that are proportional to the largest implicit aggregate in the preconditioner, whether pointwise (in N) or subdomainwise (in P). The use of overlap in going from Domain Jacobi to Additive Schwarz avoids the introduction of high energy at near discontinuities at subdomain boundaries. The two-level method projects out low-wavenumber errors rapidly at the price of solving a global problem.

Only the two-level method scales perfectly in convergence rate (constant, independent of N and P), like a traditional multilevel iterative method [4, 5, 14, 30]. However, the two-level method shares with multilevel methods a nonscalable cost-per-iteration from the necessity of solving a coarse-grid system of size $\mathcal{O}(P)$. Unlike recursive multilevel methods, a two-level Schwarz method may have a rather fine coarse grid, for example, $H = \mathcal{O}(h^{1/2})$, which potentially makes it less scalable overall. Parallelizing the coarse grid solve is necessary. Neither extreme of a fully distributed or a fully redundant coarse solve is optimal, but rather something in between. When reuse is possible, storing a parallel inverse can be cost-effective [31].

When it appears additively in the Schwarz preconditioner, the coarse grid injects some work that potentially spoils the "single-program, multiple data" (SPMD) parallel programming paradigm, in which each processor runs an identical image over local data. For instance, the SPMD model would not hold if one subset of processors worked on the coarse grid problem concurrently to the others each working on subdomains. Therefore, in two-level SPMD implementations, other Schwarz preconditioner polynomials than the purely additive are used in practice. A preconditioner may be defined that solves the fine subdomains concurrently in the standard way, and then assembles a new residual and solves the coarse grid in a separate phase. This leads to the method denoted "Hybrid II" in [28]:

$$B^{-1} = A_0^{-1} + (I - A_0^{-1}A)(\sum_{i=1}^{n} A_i^{-1}).$$

The subspace inverses are typically done approximately, as in the purely additive case.

Readers uncomfortable with the appearance of the Schwarz formula $A^{-1} \approx \sum_i {R_i}^T A_i^{-1} R_i$, implying that the inverse of the sum is well approximated by the sum of the inverses in subspaces, may benefit from recalling an exact result from eigenanalysis. Let $\{r_i\}_{i=1}^N$ be a complete set of orthonormal row (left) eigenvectors for an SPD matrix A. Then $r_i A = a_i r_i$, or $a_i = r_i A r_i^T$, for corresponding eigenvalues a_i. Then, we have the representations of A and A^{-1} as sums over subspaces,

$$A = \sum_{i=1}^{N} r_i^T a_i r_i \quad \text{and} \quad A^{-1} = \sum_{i=1}^{N} r_i^T a_i^{-1} r_i = \sum_{i=1}^{N} r_i^T (r_i A r_i^T)^{-1} r_i.$$

The latter is nothing but a special case of the Schwarz formula! In practice, invariant subspaces are far too expensive to obtain for practical use in Schwarz, and their basis vectors are general globally dense, resulting in too much storage and communication in forming restrictions and prolongations. Characteristic subspaces of subdomains, in contrast, provide locality and sparsity, but are not invariant upon multiplication by A, since the stencils overlap subdomain boundaries. Choosing good decompositions is a balance between conditioning and parallel complexity, in practice.

Contrast of Domain Decomposition with Other Decompositions. It is worthwhile to emphasize the architectural advantages of Schwarz-type domain decomposition methods *vis-à-vis* other mathematically useful decompositions.

Given the operator equation $\mathcal{L}u = f$ in Ω, and a desire for either concurrent or sequential "divide-and-conquer," one can devise operator decompositions $\mathcal{L} = \sum_j \mathcal{L}_j$, function-space decompositions $u = \sum_j u_j \phi_j$, or domain decompositions $\Omega = \cup_j \Omega_j$.

Let us contrast an example of each on the parabolic PDE in two space dimensions

$$\frac{\partial u}{\partial t} + [\mathcal{L}_x + \mathcal{L}_y]u = f(x, y, t) \text{ in } \Omega, \tag{2.5}$$

with $u = 0$ on $\partial\Omega$, where $\mathcal{L}_x \equiv -\frac{\partial}{\partial x} a_x(x, y) \frac{\partial}{\partial x} + b_x(x, y) \frac{\partial}{\partial x}$, $(a_x > 0)$ and with a corresponding form for $\mathcal{L}_y$. Upon implicit time discretization

$$\left[\frac{I}{\Delta t} + \mathcal{L}_x + \mathcal{L}_y\right] u^{(\ell+1)} = \left[\frac{I}{\Delta t}\right] u^{(\ell)} + f \equiv \tilde{f},$$

we get an elliptic problem at each time step.

The Alternating Direction Implicit (ADI) method is an example of operator decomposition. Proceeding in half-steps, one each devoted to the x- and y-directions, we write

$$\left[\frac{I}{\Delta t/2} + \mathcal{L}_x\right] u^{(\ell+1/2)} = \left[\frac{I}{\Delta t/2} - \mathcal{L}_y\right] u^{(\ell)} + f$$

$$\left[\frac{I}{\Delta t/2} + \mathcal{L}_y\right] u^{(\ell+1)} = \left[\frac{I}{\Delta t/2} - \mathcal{L}_x\right] u^{(\ell+1/2)} + f.$$

The overall iteration matrix mapping $u^{(\ell)}$ to $u^{(\ell+1)}$ is factored into four sequential substeps per time step: two sparse matrix-vector multiplies and two sets of unidirectional bandsolves. If the data is alternately laid out in unidirectional slabs on the processors, so as to allow each set of unidirectional bandsolves to be executed independently,

then we have perfect parallelism *within* substeps, but, global data exchanges *between* substeps. In other words, computation and communication each scale with the bulk size of the data of the problem.

A Fourier or spectral method is an example of a function-space decomposition. We expand

$$u(x,y,t) = \sum_{j=1}^{N} a_j(t)\phi_j(x,y).$$

Enforcing Galerkin conditions on (2.5) with the basis functions ϕ_i, we get

$$\frac{d}{dt}(\phi_i, u) = (\phi_i, \mathcal{L}u) + (\phi_i, f), \quad i = 1, \ldots, N.$$

Plugging the expansion into the Galerkin form,

$$\sum_{j=1}^{N}(\phi_i, \phi_j)\frac{da_j}{dt} = \sum_{j=1}^{N}(\phi_i, \mathcal{L}\phi_j)a_j + (\phi_i, f), i = 1, \ldots, N.$$

Inverting the mass matrix, $M \equiv [(\phi_j, \phi_i)]$ on both sides, and denoting the stiffness matrix by $K \equiv [(\phi_j, \mathcal{L}\phi_i)]$, we get a set of ordinary differential equations for the expansion coefficients:

$$\dot{a} = M^{-1}Ka + M^{-1}g.$$

If the basis functions are orthogonal and diagonalize the operator, then M and K are diagonal, and these equations perfectly decouple, creating N-fold concurrency for the evolution of the spectral components. However, in applications, it is necessary to frequently reconstitute the physical variable u. This is true for interpreting or visualizing the model and also for handling possible additional terms of the PDE in physical space in a "pseudo-spectral" approach, since it is unlikely that practically arising operators readily lead to orthogonal eigenfunctions for which there are fast transforms. Transforming back and forth from physical to spectral space on each iteration leads, again, to an algorithm where the computation and the communication together scale with the problem size, and there is all-to-all communication.

An additive Schwarz domain decomposition method for this problem has been described already. We replace $Au = f$ by $B_{ASM}^{-1}Au = B_{ASM}^{-1}f$ and solve by a Krylov method. There are several Krylov steps per time step, each requiring a matrix-vector multiplies with $B_{ASM}^{-1}A$. Due to the concurrency implied by the sum, there is parallelism on each subregion. However the dominant communication is nearest-neighbor data exchange, whose size scales as the perimeter (resp., surface in three dimensions), compared to the computation, whose size scales as the area (resp., volume). Therefore, domain decomposition possesses excellent scalability properties with respect to implementation on distributed memory computers. There is a need for a small global sparse linear system solve in some problems, to obtain mathematical optimality. (This is not necessary for the parabolic problem considered above.) Though this small problem requires global communication (either to set up redundant instances, solved concurrently, or to carry out a collaborative solution) and demands analysis and extreme care to keep subdominant, it escapes the bulk communication burdens of the other approaches.

Physics-based Preconditioning. An important class of preconditioners for the Jacobian-free Newton-Krylov method, complementary to the domain-split parallelism of Schwarz, is physics-based operator splitting. The operator notation for the right-preconditioned, matrix-free form of the method is:

$$J(\mathbf{u})B_{split}^{-1}\mathbf{v} \approx \frac{\mathbf{F}(\mathbf{u}+\epsilon B_{split}^{-1}\mathbf{v}) - \mathbf{F}(\mathbf{u})}{\epsilon}, \tag{2.6}$$

where "*split*" denotes a preconditioning process handled in an operator-split manner. Many operator-split time integration methods have been developed based on insight from the physics of the underlying system. It is well understood that operator-split methods have limitations as solvers, thus they most likely also have limitations as preconditioners. However, they still provide an interesting class of preconditioners for the Jacobian-free Newton-Krylov method.

The essential insight of physics-based preconditioning is that preconditioner in a Newton-Krylov method maps a nonlinear residual to an approximate state-vector correction, namely, the Newton update. Such a map implicitly resides in most interative procedures of computational physics. The use of operator-split solvers as preconditioners for Jacobian-free Newton-Krylov appears not to have a long history, but is rapidly developing. See instances for time-independent reaction diffusion equations [24], time-dependent MHD equations [10], steady state incompressible Navier-Stokes equations [19, 25], and time-dependent incompressible Navier-Stokes equations [20, 25]. Also in [20], a standard approximate linearization method used for phase-change heat conduction problems, has been employed as a preconditioner for a JFNK solution of phase-change heat conduction problems.

3. Parallel Implementation of NKS Using PETSc. To implement NKS methods on distributed memory parallel computers, we employ the "Portable, Extensible Toolkit for Scientific Computing" (PETSc) [1, 2], a library that attempts to handle through a uniform interface, in a highly efficient way, the low-level details of the distributed memory hierarchy. Examples of such details include striking the right balance between buffering messages and minimizing buffer copies, overlapping communication and computation, organizing node code for strong cache locality, preallocating memory in sizable chunks rather than incrementally, and separating tasks into one-time and every-time subtasks using the inspector/executor paradigm. The benefits to be gained from these and from other numerically neutral but architecturally sensitive techniques are so significant that it is efficient in both the programmer-time and execution-time senses to express them in general purpose code. Among other important packages implementing Newton-Krylov in parallel, we mention Aztec [15], KINSOL [29], NITSOL [23], and the Iterative Template Library (ITL) [21].

PETSc is a large and versatile package integrating distributed vectors, distributed matrices in several sparse storage formats, Krylov subspace methods, preconditioners, and Newton-like nonlinear methods with built-in trust region or linesearch strategies and continuation for robustness. It has been designed to provide the numerical infrastructure for application codes involving the implicit numerical solution of PDEs, and it sits atop MPI for portability to most parallel machines. The PETSc library is written in C, but may be accessed from user codes written in C, FORTRAN, and C++. PETSc version 2, first released in June 1995, has been downloaded thousands

of times by users worldwide. PETSc has many features relevant to PDE analysis, including matrix-free Krylov methods, blocked forms of parallel preconditioners, and various types of time-stepping.

When well tuned, large-scale PDE codes spend almost all of their time in two phases: flux computations to evaluate conservation law residuals, where one aims to have such codes spent almost *all* their time, and sparse linear algebraic kernels, which are a fact of life in implicit methods. Altogether, four basic groups of tasks can be identified based on the criteria of arithmetic concurrency, communication patterns, and the ratio of operation complexity to data size within the task. These four distinct phases, present in most implicit codes, are vertex-based loops, edge-based loops, recurrences, and global reductions. Each of these groups of tasks has a distinct proportion of work to datasize to communication requirements and each stresses a different subsystem of contemporary high-performance computers. In the language of a vertex-centered code, in which the data is

- Vertex-based loops
  - state vector and auxiliary vector updates
- Edge-based "stencil op" loops
  - residual evaluation, Jacobian evaluation
  - Jacobian-vector product (often replaced with matrix-free form, involving residual evaluation)
  - interpolation between grid levels
- Sparse, narrow-band recurrences
  - (approximate) factorization, back substitution, relaxation/smoothing
- Vector inner products and norms
  - orthogonalization/conjugation
  - convergence progress checks and stability heuristics

Vertex-based loops are characterized by work closely proportional to datasize, pointwise concurrency, and no communication.

Edge-based "stencil op" loops have a large ratio of work to datasize, since each vertex is used in many discrete stencil operations, and each degree of freedom at a point (momenta, energy, density, species concentration) generally interacts with all others in the conservation laws—through constitutive and state relationships or directly. There is concurrency at the level of the number of edges between vertices (or, at worst, the number of edges of a given "color" when write consistency needs to be protected through mesh coloring). There is local communication between processors sharing ownership of the vertices in a stencil.

Sparse, narrow-band recurrences involve work closely proportional to data size, the matrix being the largest data object and each of its elements typically being used once. Concurrency is at the level of the number of fronts in the recurrence, which may vary with the level of exactness of the recurrence. In a preconditioned iterative method, the recurrences are typically broken to deliver a prescribed process concurrency; only the quality of the preconditioning is thereby affected, not the final result. Depending upon whether one uses a pure decomposed Schwarz-type preconditioner, a truncated incomplete solve, or an exact solve, there may be no, local only, or global communication in this task.

Vector inner products and norms involve work closely proportional to data size, mostly pointwise concurrency, and global communication.

Based on these characteristics, one anticipates that vertex-based loops, recurrences, and inner products will be *memory bandwidth-limited*, whereas edge-based loops are likely to be only *load/store-limited*. However, edge-based loops are vulnerable to *internode bandwidth* if the latter does not scale. Inner products are vulnerable to *internode latency* and *network diameter*. Recurrences can resemble some combination of edge-based loops and inner products in their communication characteristics if preconditioning fancier than simple Schwarz is employed. For instance, if incomplete factorization is employed globally or a coarse grid is used in a multilevel preconditioner, global recurrences ensue.

Analysis of a parallel aerodynamics code reimplemented in PETSc [13] shows that, after tuning, as expected, the linear algebraic kernels run at close to the aggregate memory bandwidth limit on performance, the flux computations are bounded either by memory bandwidth or instruction scheduling (depending upon the ratio of load/store units to floating-point units in the CPU), and parallel efficiency is bounded primarily by slight load imbalances at synchronization points.

4. Terascale Optimal PDE Simulations (TOPS). Under the Scientific Discovery through Advanced Computing (SciDAC) initiative of the U.S. Department of Energy (`http://www.science.doe.gov/scidac/`), a nine-institution team is building an integrated software infrastructure center (ISIC) that focuses on developing, implementing, and supporting optimal or near optimal schemes for PDE simulations and closely related tasks, including optimization of PDE-constrained systems, eigenanalysis, and adaptive time integration, as well as implicit linear and nonlinear solvers. The Terascale Optimal PDE Simulations (TOPS) Center is researching and developing and will deploy a toolkit of open source solvers for the nonlinear partial differential equations that arise in many application areas, including fusion, accelerator design, global climate change, and the collapse of supernovae. These algorithms — primarily multilevel methods — aim to reduce computational bottlenecks by one or more orders of magnitude on terascale computers, enabling scientific simulation on a scale heretofore impossible.

Along with usability, robustness, and algorithmic efficiency, an important goal of this ISIC is to attain the highest possible computational performance in its implementations by accommodating to the memory bandwidth limitations of hierarchical memory architectures.

PDE simulation codes require implicit solvers for multiscale, multiphase, multiphysics phenomena from hydrodynamics, electromagnetism, radiation transport,

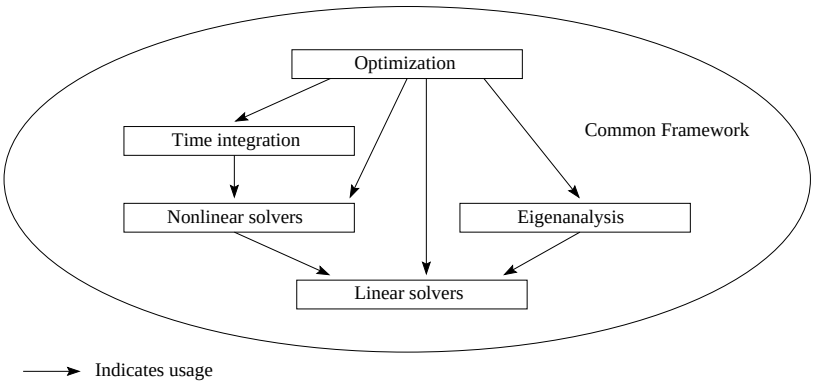


Figure 4.1: *An arrow from A to B indicates that A typically uses B. Optimization of systems governed by PDEs requires repeated access to a PDE solver. The PDE system may be steady-state or time-dependent. Time-dependent PDEs are typically solved with implicit temporal differencing. After choice of the time-integration scheme, they, in turn, require the same types of nonlinear solvers that are used to solve steady-state PDEs. Many algorithms for nonlinear problems of high dimension generate a sequence of linear problems, so linear solver capability is at the core. Eigenanalysis arises inside of or independently of optimization. Like direct PDE analysis, eigenanalysis generally depends upon solving a sequence of linear problems. All of these five classes of problems, in a PDE context, share grid-based data structures and considerable parallel software infrastructure. Therefore, it is compelling to undertake them together.*

chemical kinetics, and quantum chemistry. Problem sizes are typically now in the millions of unknowns; and with emerging large-scale computing systems and inexpensive clusters, we expect this size to increase by a factor of a thousand over the next five years. Moreover, these simulations are increasingly used for design optimization, parameter identification, and process control applications that require many repeated, related simulations.

The TOPS ISIC is concerned with five PDE simulation capabilities: adaptive time integrators for stiff systems, nonlinear implicit solvers, optimization, linear solvers, and eigenanalysis. The relationship between these areas is depicted in Figure 4.1. In addition, TOPS emphasizes two cross-cutting topics: software integration (or interoperability) and high-performance coding techniques for PDE applications.

Optimal (and nearly optimal) complexity numerical algorithms almost invariably depend upon a hierarchy of approximations to "bootstrap" to the required highly accurate final solution. Generally, an underlying continuum (infinite dimensional) high fidelity mathematical model of the physics is discretized to "high" order on a "fine" mesh to define the top level of the hierarchy of approximations. The representations of the problem at lower levels of the hierarchy may employ other models (possibly of lower physical fidelity), coarser meshes, lower order discretization schemes, inexact linearizations, and even lower floating-point precisions. The philosophy that underlies our algorithmics and software is the same as that of this chapter — to make the majority of progress towards the highly resolved result through possibly low-resolution

stages that run well on high-end distributed hierarchical memory computers.

The ingredients for constructing hierarchy-of-approximations-based methods are remarkably similar, be it for solving linear systems, nonlinear problems, eigenvalue problems, or optimization problems, namely:

1. A method for generating several discrete problems at different resolutions (for example on several grids),

2. An inexpensive (requiring few floating point operations, loads, and stores per degree of freedom) method for iteratively improving an approximate solution at a particular resolution,

3. A means of interpolating (discrete) functions at a particular resolution to the next finer resolution,

4. A means of transferring (discrete) functions at a particular resolution to the next coarser resolution (often obtained trivially from interpolation).

Software should reflect the simplicity and uniformity of these ingredients over the five problem classes and over a wide range of applications. With experience we expect to achieve a reduction in the number of lines of code that need to be written and maintained, because the same code can be reused in many circumstances.

The efforts defined for TOPS, the co-PIs joining to undertake them, and the alliances proposed with other groups have been chosen to exploit the present opportunity to revolutionize large-scale solver infrastructure, and lift the capabilities of dozens of DOE's computational science groups as an outcome. The co-PIs' current software (e.g., Hypre [12], PETSc [1], ScaLAPACK [3], SuperLU [22]), though not algorithmically optimal in many cases, and not yet as interoperable as required, is in the hands of thousands of users, and has created a valuable experience base. Just as we expect the user community to drive research and development, we expect to significantly impact the scientific priorities of users by emphasizing optimization (inverse problems, optimal control, optimal design) and eigenanalysis as part of the solver toolkit.

Optimization subject to PDE-constraints is a particularly active subfield of optimization because the traditional means of handling constraints in black-box optimization codes — with a call to a PDE solver in the inner loop — is too expensive. We are emphasizing "simultaneous analysis and design" methods in which the cost of doing the optimization is a small multiple of doing the simulation and the simulation data structures are actually part of the optimization data structures.

Likewise, we expect that a convenient software path from PDE analysis to eigenanalysis will impact the scientific approach of users with complex applications. For instance, a PDE analysis can be pipelined into the scientific added-value tasks of stability analysis for small perturbations about a solution and reduced dimension representations (model reduction), with reuse of distributed data structures and solver components.

The motivation behind TOPS is that most PDE simulation is ultimately a part of some larger scientific process that can be hosted by the same data structures and carried out with many of the same optimized kernels as the simulation, itself. We intend to make the connection to such processes explicit and inviting to users, and this

will be a prime metric of our success. The organization of the effort flows directly from this program of "holistic simulation": Terascale software for PDEs should extend from the analysis to the scientifically important auxiliary processes of sensitivity analysis, modal analysis and the ultimate "prize" of optimization subject to conservation laws embodied by the PDE system.

5. Conclusions. The emergence of the nonlinearly implicit Jacobian-free Newton-Krylov-Schwarz family of methods has provided a pathway towards terascale simulation of PDE-based systems. Domain decomposition is desirable for possessing a communication cost that is subdominant to computation – even optimal order computation, linear in the problem size – and fixed in ratio, as problem size and processor count are scaled in proportion.

Large-scale implicit computations have matured to a point of practical use on distributed/shared memory architectures for static-grid problems. More sophisticated algorithms, including solution adaptivity, inherit the same features *within* static-grid phases, of course, but require extensive additional infrastructure for dynamic parallel adaptivity, rebalancing, and maintenance of efficient, consistent distributed data structures.

While mathematical theory has been crucial in the development of NKS methods, their most successful application also depends upon a more-than-superficial understanding of the underlying architecture and of the physics being modeled. In the future, as we head towards petascale simulation and greater integration of complex physics codes in full system analysis and optimization, we expect that this interdisciplinary interdependence will only increase.

Acknowledgements The author thanks Xiao-Chuan Cai, Omar Ghattas, Bill Gropp, Dana Knoll, and Barry Smith for long-term collaborations on parallel algorithms, and Satish Balay, Paul Hovland, Dinesh Kaushik, and Lois McInnes from the PETSc team at Argonne National Laboratory (along with Gropp and Smith and others) for their wizardry in implementation.

REFERENCES

[1] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object-oriented numerical software libraries. In *Modern Software Tools in Scientific Computing*, pages 163–201. Birkhauser, 1997.

[2] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. The Portable, Extensible Toolkit for Scientific Computing, version 2.3.1. `http://www.mcs.anl.gov/petsc/`, 2002.

[3] L. S. Blackford, J. Choi, A. Cleary, E. D'Azevedo, J. Demmel, I. Dhillon, J. Dongarra, S. Hammarling, G. Henry, A. Petitet, K. Stanley, D. Walker, and R. C. Whaley. *ScaLAPACK Users' Guide*. SIAM, 1997.

[4] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Math. Comp.*, 31:333, 1977.

[5] A. Brandt. Multigrid Techniques: 1984 Guide with Applications to Fluid Dynamics. Technical report, von Karman Institute, 1984.

[6] P. N. Brown and Y. Saad. Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Stat. Comput.*, 11:450–481, 1990.

[7] X.-C. Cai. Some domain decomposition algorithms for nonselfadjoint elliptic and parabolic partial differential equations. Technical Report 461, Courant Institute, 1989.

[8] X.-C. Cai, M. Dryja, and M. Sarkis. RASHO: A restricted additive Schwarz preconditioner with harmonic overlap. In *Proceedings of the 13th International Conference on Domain Decomposition Methods*. Domain Decomposition Press, 2002.

[9] X.-C. Cai and M. Sarkis. A restricted additive Schwarz preconditioner for general sparse linear systems. *SIAM J. Sci. Comput.*, 21:792–797, 1999.

[10] L. Chacon, D. A. Knoll, and J. M. Finn. An implicit nonlinear reduced resistive MHD solver. *J. Comput. Phys.*, 178:15–36, 2002.

[11] M. Dryja and O. B. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, Department of Computer Science, Courant Institute, 1987.

[12] R. D. Falgout and U. M. Yang. Hypre: a library of high performance preconditioners. In *Lecture Notes in Computer Science, vol. 2331*, pages 632–641. Springer-Verlag, 2002.

[13] W. D. Gropp, D. K. K. D. E. Keyes, and B. F. Smith. High performance parallel implicit CFD. *Parallel Computing*, 27:337–362, 2001.

[14] W. Hackbusch. *Iterative Methods for Large Sparse Linear Systems*. Springer, 1993.

[15] S. A. Hutchinson, J. N. Shadid, and R. S. Tuminaro. Aztec user's guide: Version 1.1. Technical Report SAND95-1559, Sandia National Laboratories, October 1995.

[16] C. T. Kelley and D. E. Keyes. Convergence analysis of pseudo-transient continuation. *SIAM J. Numer. Anal.*, 35:508–523, 1998.

[17] D. E. Keyes. Terascale implicit methods for partial differential equations. In *Recent Advances in Numerical Methods for Partial Differential Equations and Applications*. AMS, 2002.

[18] D. A. Knoll and D. E. Keyes. Jacobian-free Newton-Krylov methods: A survey of approaches and applications. submitted to J. Comput. Phys., 2002.

[19] D. A. Knoll and V. Mousseau. On Newton-Krylov multigrid methods for the incompressible Navier-Stokes equations. *J. Comput. Phys.*, 163:262–267, 2000.

[20] D. A. Knoll, W. B. VanderHeyden, V. A. Mousseau, and D. B. Kothe. On preconditioning Newton-Krylov methods in solidifying flow applications. *SIAM J. Sci. Comput.*, 23:381–397, 2001.

[21] L.-Q. Lee and A. Lumsdaine. The iterative template library. submitted to ACM Transactions on Mathematical Software, 2002.

[22] X. S. Li and J. W. Demmel. SuperLU_DIST: A scalable distributed-memory sparse direct solver for unsymmetric linear systems. submitted to ACM Transactions on Mathematical Software; also available as Lawrence Berkeley National Laboratory tech report LBNL-49388, 2002.

[23] H. F. W. Michael Pernice. NITSOL: A Newton iterative solver for nonlinear systems. *SIAM J. Sci. Stat. Comput.*, 19:302–318, 1998.

[24] V. Mousseau, D. A. Knoll, and W. Rider. Physics-based preconditioning and the Newton-Krylov method for non-equilibrium radiation diffusion. *J. Comput. Phys.*, 160:743–765, 2000.

[25] M. Pernice and M. D. Tocci. A multigrid-preconditioned Newton-Krylov method for the incompressible Navier-Stokes equations. *SIAM J. Sci. Comput.*, 23:398–418, 2001.

[26] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comput.*, 7:856–869, July 1986.

[27] S. Shingu, H. Takahara, H. Fuchigami, M. Yamada, Y. Tsuda, W. Ohfuchi, Y. Sasaki, K. Kobayashi, T. Hagiwara, S.-I. Habata, M. Yokokawa, H. Itoh, and K. Otsuka. A 26.58 Tflop/s global atmospheric simulation with the spectral transform method on the earth simulator. Proceedings of SC2002, 2002.

[28] B. F. Smith, P. Bjørstad, and W. D. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[29] A. G. Taylor and A. C. Hindmarsh. User documentation for KINSOL: a nonlinear solver for sequential and parallel computers. Technical Report UCRL-ID-131185, Lawrence Livermore National Laboratory, July 1998.

[30] U. Trottenberg, A. Schuller, and C. Oosterlee. *Multigrid*. Academic Press, 2000.

[31] H. M. Tufo and P. Fischer. Fast parallel direct solvers for coarse grid problems. *J. Par. Dist. Comput.*, 61:151–177, 2001.

[32] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34:581–613, 1991.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

8. Nonlinearly Preconditioned Newton's Method

S. H. Lui[1]

1. Introduction. Many challenging problems in science and engineering are large and nonlinear. Typically they are solved by Newton's method or its many variations. If parallel computers are available, the solution process can be sped up by the use of domain decomposition techniques. The traditional domain decomposition approach for nonlinear PDEs is to use the classical Newton's method and apply classical domain decomposition techniques such as the additive Schwarz preconditioner ([6]) to the resulting linear systems. This is often referred to as the Newton-Krylov-Schwarz method ([1], [2]). For most nonlinear equations, this works very well. However, for more difficult problems, the lack of a good initial guess means that the Newton-Krylov-Schwarz iteration may not converge or may converge very slowly. Often, the failure may be traced to boundary layers, singularities (corners/cusps) in the domain, and/or multi–physics domains (fluid–structure interaction problems for instance). These problematic regions slow down global convergence or cause stagnation in the iteration. There are of course many papers on the application of domain decomposition methods to nonlinear problems, especially those in fluid mechanics. Many references can be found in the proceedings of the annual conference on domain decomposition methods, starting with [7].

Meanwhile, other workers have begun to look at applying Schwarz methods directly on the nonlinear subdomain problems: [5], [17], [18], [11], [12], and [13]. These nonlinear Schwarz methods have the nice property that difficult regions are isolated in a small number of subdomains where special techniques (finer grid, asymptotics, etc.) may be brought to bear without interfering with the convergence in other parts of the domain. However, they still require a good initial guess for convergence and their rate of convergence is usually slow (linear).

Recently, Cai and Keyes ([3]) have proposed a new method which is a marriage of the Newton-Krylov-Schwarz and nonlinear Schwarz methods. Their idea is to nonlinearly precondition the given nonlinear equations $F(u) = 0$ so that the resultant equations $\mathcal{F}(u) = 0$ are closer to linear equations and so amenable to solution by Newton's method without the necessity of a good initial guess. The nonlinear preconditioner is a nonlinear additive Schwarz preconditioner which requires the solution of a nonlinear subdomain PDE. The new system $\mathcal{F}(u) = 0$ is solved using a modified Newton's method where the Jacobian has the same form as in the Newton-Krylov-Schwarz algorithm. In particular, it reduces to the additive Schwarz algorithm when F is linear. In [3], they illustrate the impressive robustness of this new method with the driven cavity flow problem where Newton's method stagnates at a moderate Reynolds number while the nonlinearly preconditioned method is able to compute to a considerably larger Reynolds number and still maintain fast quadratic convergence.

In this paper, we carry out some preliminary convergence analysis of this nonlinearly preconditioned method as well as estimate crudely its radius of quadratic convergence, that is, the radius of the ball where the iterates converge quadratically. This is compared to the corresponding quantity for the classical Newton's method. The discussion is in the context of semilinear elliptic PDEs which are described in the next section. In section three, we shall examine two types of convergence theories: classical q-quadratic convergence and r-quadratic convergence assuming data only at the initial guess. In the last section, we carry out some numerical experiments on some quasi-linear two-point boundary value problems and conclude.

[1]Department of Mathematics, University of Manitoba, Winnipeg, Manitoba, Canada luish@cc.umanitoba.ca. This work was supported in part by a grant from NSERC of Canada.

2. Nonlinearly preconditioned PDEs. In this section, we apply the nonlinear preconditioner to a class of semilinear elliptic PDEs and see its relation with the Newton-Krylov-Schwarz and nonlinear Schwarz methods.

Let Ω be a bounded domain in $\mathbf{R}^N$ with a smooth boundary. We consider the PDE

$$-\triangle u = f(x,u) \text{ on } \Omega \tag{2.1}$$

for the solution $u \in H_0^1(\Omega)$. For simplicity, we write $f(u)$ for $f(x,u)$. Throughout this paper, we assume that this PDE has the unique solution u.

Suppose for some fixed integer $m > 1$, $\Omega = \Omega_1 \cup \cdots \cup \Omega_m$, where the subdomains Ω_i have smooth boundaries and are overlapping, meaning that $H_0^1(\Omega) = H_0^1(\Omega_1) + \cdots + H_0^1(\Omega_m)$. In this paper, a function in $H_0^1(\Omega_i)$ is considered as a function in $H_0^1(\Omega)$ by extension by zero. Let $\|\cdot\|$ be the norm on $H_0^1(\Omega)$, that is,

$$\|v\|^2 = \int_\Omega |\nabla v|^2$$

and $\|\cdot\|_{-1}$ be the norm on the dual space $H^{-1}(\Omega)$. Let P_i denote the projection $P_i H_0^1(\Omega) = H_0^1(\Omega_i)$ in the $H_0^1(\Omega)$–norm.

It is more convenient to express the PDE as the nonlinear operator equation

$$F(u) \equiv u + \triangle^{-1} f(u) = 0,$$

where $F : H_0^1(\Omega) \to H_0^1(\Omega)$, and $\triangle^{-1} : H^{-1}(\Omega) \to H_0^1(\Omega)$ denotes the inverse of the Laplacian operator on Ω with homogeneous Dirichlet boundary conditions. Define the new nonlinear equations ([3])

$$\mathcal{F}(u) \equiv \sum_{i=1}^m T_i(u) = 0$$

where $T_i : H_0^1(\Omega) \to H_0^1(\Omega_i)$ satisfies

$$P_i F(v + T_i(v)) = 0, \qquad v \in H_0^1(\Omega).$$

It is assumed that this solution exists and is unique given v. One can think of $T_i(v)$ as a correction to the current guess v obtained by solving a nonlinear subdomain PDE. Let $y_i = T_i(v)$ and using the definition of F, we obtain

$$y_i + P_i \triangle^{-1} f(v + y_i) = -P_i v \tag{2.2}$$

or

$$-\triangle_i y_i - f(v + y_i) = \triangle v \text{ on } \Omega_i. \tag{2.3}$$

This nonlinear subdomain PDE is very much like that in nonlinear Schwarz algorithms mentioned above.

The nonlinearly preconditioned method solves the new nonlinear equations using Newton's method. That is given $u^{(0)}$, it produces the sequence

$$u^{(n+1)} = u^{(n)} - \mathcal{F}'(u^{(n)})^{-1} \mathcal{F}(u^{(n)}).$$

In practice, a Krylov subspace method such as GMRES ([15]) is used to solve the above linear equations. These methods only require that we supply a procedure to compute $\mathcal{F}'(u^{(n)})w$ for an arbitrary $w \in H_0^1(\Omega)$. Let us look at this in a little more detail. Let

$$\mathcal{F}'(u^{(n)})w = \sum_{i=1}^m z_i, \qquad z_i = \frac{\partial T_i(u^{(n)})}{\partial v} w \in H_0^1(\Omega_i).$$

From (2.2) and abbreviating $T_i(u^{(n)})$ by y_i,

$$z_i + \triangle_i^{-1} f'(u^{(n)} + y_i)(w + z_i) = -P_i w$$

or equivalently

$$[-\triangle_i - f'(u^{(n)} + y_i)]\, z_i = [\triangle + f'(u^{(n)} + y_i)]\, w. \tag{2.4}$$

This scheme will be referred to as the nonlinearly preconditioned Newton's method (NP1). Other variations are possible. The original scheme of Cai and Keyes (henceforth called NP0) replaces $\mathcal{F}'(u^{(n)})$ by $\mathcal{F}'(\tilde{u}^{(n)})$ whose action on w yields

$$[-\triangle_i - f'(u^{(n)})]\, z_i = [\triangle + f'(u^{(n)})]\, w.$$

Note that this has the same form as applying the additive Schwarz preconditioner to solve a linear system for operator $F'(u^{(n)})$. While this gives a nice connection to the well-understood Newton-Krylov-Schwarz algorithm, it does not use the most up-to-date information ($T_i(u^{(n)})$) and this sometimes compromises the robustness of the algorithm. For some examples, see the section on numerical experiments.

A third variation (NP2) replaces (2.4) by

$$[-\triangle_i - f'(u^{(n)} + y)]\, z_i = [\triangle + f'(u^{(n)} + y)]\, w$$

where $y = \sum_{i=1}^{m} y_i = \mathcal{F}(u^{(n)})$. The reasoning here is that y incorporates information from neighboring subdomains and may lead to a better estimate. We assume that z_i exists and is unique in all three cases.

The following is a version of the partition lemma ([14], [10]) for bounded linear operators.

Lemma 2.1 *Let A be a bounded linear operator on a Hilbert space H. Suppose $H = H_1 + \cdots + H_m$ and $A = A_1 + \cdots + A_m$ where H_i are Hilbert spaces and A_i are bounded linear operators on H_i. Then there is some constant C_m such that*

$$\|A\| \geq C_m \sum_{i=1}^{m} \|A_i\|.$$

Finally, we collect together the definitions of all constants which will appear later. Let

1. r denote the radius of $B_r(u)$, the open ball with center at u;
2. $\alpha_0(u^{(0)})$ denote the eigenvalue of $F'(u^{(0)}) = I + \triangle^{-1} f'(u^{(0)})$ of smallest magnitude and $\alpha_0 = \alpha_0(u)$;
3. γ denote the Lipschitz constant for f':
$$\|f'(w) - f'(v)\|_{-1} \leq \gamma \|w - v\|, \qquad w, v \in B_r(u);$$
4. α_i denote the eigenvalue of $I + \triangle_i^{-1} f'(u)$ on Ω_i of smallest magnitude with corresponding eigenfunction $\phi_i \in H_0^1(\Omega_i)$ and $|\alpha_{max}| = \max\limits_{1 \leq i \leq m} |\alpha_i|$;
5. $\alpha_i(u^{(0)})$ denote the eigenvalue of $I + \triangle_i^{-1} f'(u^{(0)} + T_i(u^{(0)}))$ on Ω_i of smallest magnitude and $|\alpha_{max}(u^{(0)})| = \max\limits_{1 \leq i \leq m} |\alpha_i(u^{(0)})|$; note $\alpha_i = \alpha_i(u)$;
6. $\rho_i = \sup\limits_{w \in B_r(u)} \| [I + \triangle_i^{-1} f'(w)]^{-1} P_i [I + \triangle^{-1} f'(w)] \|$ and $\rho_{max} = \max\limits_{1 \leq i \leq m} \rho_i$;
7. $\beta_i = \sup\limits_{w \in B_r(u)} \| [I + \triangle_i^{-1} f'(w)]^{-1} \|$ and $\beta_{max} = \max\limits_{1 \leq i \leq m} \beta_i$.

Note that r must be sufficiently small so that the Newton iteration is well defined and all iterates remain in $B_r(u)$.

3. Convergence Theory. Newton's method is one of the oldest, simplest and most efficient methods for solving nonlinear equations. Most of the best algorithms today are modifications of Newton's method. It is not surprising that many types of convergence theories exist, depending on the hypotheses and the convergence result. We shall examine two such theories. Recall that iterates $\{e^{(n)}\}$ converge q-quadratically to 0 if $\|e^{(n+1)}\| \leq c \|e^{(n)}\|^2$ for some constant c while it converges r-quadratically to 0 if $\|e^{(n)}\| \leq c_n$ where $\{c_n\}$ converges q-quadratically to 0.

The first theory is well known and is concerned with the q-quadratic convergence of Newton's method. The second theory is rather special in that all assumptions are at one point, the initial iterate – there is no Lipschitz condition in a region which is required in the other theory. It is unfortunate that we are unable to do much analysis for the nonlinearly preconditioned method in regard to this theory and must resort to some numerical experiments. For the first theory, we attempt to contrast the rate of convergence of the nonlinearly preconditioned method versus that of the classical Newton's method, and the radii of quadratic convergence of the two methods.

3.1. q-quadratic convergence. The first convergence theory assumes that $F'(u)$ has a bounded inverse with $\|F'(u)^{-1}\| \leq \eta$ and $\|F'(u) - F'(v)\| \leq \Gamma \|u - v\|$ for all $v \in B_r(u)$. Then the (classical) Newton iterates $\{u^{(n)}\}$ for $F(u) = 0$ are well defined and the error $e_N^{(n)} = u^{(n)} - u$ satisfies

$$\|e_N^{(n+1)}\| \leq \eta\Gamma \|e_N^{(n)}\|^2$$

provided the initial iterate $u^{(0)} \in B_{\epsilon_N}(u)$, where

$$\epsilon_N = \min\left(r, \frac{1}{2\eta\Gamma}\right).$$

See, for instance, [4] or [9]. We use the subscript N to describe the relevant quantity for the classical Newton's method. Hence ϵ_N is a lower bound of the radius of the ball where quadratic convergence takes place.

Applying the classical Newton's method to our semilinear elliptic PDE $F(u) = 0$, we find that $\eta = |\alpha_0|^{-1}$ and for any $v \in B_r(u)$,

$$\begin{aligned} \|F'(u) - F'(v)\| &= \|\triangle^{-1}(f'(u) - f'(v))\| \\ &= \|f'(u) - f'(v)\|_{-1} \\ &\leq \gamma \|u - v\|. \end{aligned}$$

Thus provided $u^{(0)} \in B_{\epsilon_N}(u)$,

$$\|e_N^{(n+1)}\| \leq \frac{\gamma}{|\alpha_0|} \|e_N^{(n)}\|^2, \qquad \epsilon_N = \min\left(r, \frac{|\alpha_0|}{2\gamma}\right). \tag{3.1}$$

Now we compute these same quantities for the nonlinear preconditioned Newton's method which employs the classical Newton's method to solve $\mathcal{F}(u) = 0$. By some straightforward calculations,

$$\|\mathcal{F}'(u)\| \geq C_m |\alpha_0| \sum_{i=1}^{m} \frac{1}{|\alpha_i|}$$

and for $v \in B_r(u)$,

$$\|\mathcal{F}'(u) - \mathcal{F}'(v)\| \leq \gamma \sum_{i=1}^{m} \frac{(1 + \rho_i)^2}{|\alpha_i|} \|u - v\|.$$

Putting everything together, we obtain the error relation for the nonlinearly preconditioned Newton's method

$$\|e_{NP1}^{(n+1)}\| \leq \frac{\gamma \sum_{i=1}^{m} \frac{(1+\rho_i)^2}{|\alpha_i|}}{C_m \, |\alpha_0| \sum_{i=1}^{m} \frac{1}{|\alpha_i|}} \|e_{NP1}^{(n)}\|^2 \leq \frac{\gamma}{|\alpha_0|} \frac{(1+\rho_{max})^2}{C_m} \|e_{NP1}^{(n)}\|^2,$$

provided that $u^{(0)} \in B_{\epsilon_{NP1}}(u)$, where

$$\epsilon_{NP1} = \min\left(r, \frac{|\alpha_0|}{2\gamma} \frac{C_m}{(1+\rho_{max})^2}\right).$$

These can be compared directly with (3.1), unfortunately to the detriment of NP1. To obtain a sharper estimate, we believe that it is necessary to restrict the class of PDEs. Note if $f' \equiv 0$, then $\rho_{max} = 1$ while if $f' \leq 0$, then $\rho_{max} \leq C$ for some constant C. With a suitable finite element discretization, C is independent of the mesh size but can increase with the number of subdomains.

It is not difficult to deduce similar estimates for NP0, the original scheme of Cai and Keyes:

$$\|e_{NP0}^{(n+1)}\| \leq K\left(1 + \gamma \sum_{i=1}^{m} \beta_i (1+\rho_i)\rho_i\right) \|e_{NP0}^{(n)}\|^2$$

for some constant K.

We have examined two other r-quadratic convergence theories that are similar to the first theory above. They differ in the Lipschitz condition ([19], [21]) or the assumption that $F'(u^{(0)})$ is invertible (rather than $F'(u)$) ([8], [20]). The results of the analysis are similar to those of the first theory and will be reported elsewhere.

3.2. r-quadratic convergence. In this theory due to Smale [16], we no longer assume a Lipschitz condition in a ball. Instead, all assumptions are at the initial point $u^{(0)}$ of the iteration. However, we need to assume that F is an analytic operator. Define

$$\omega(u^{(0)}) = \|F'(u^{(0)})^{-1} F(u^{(0)})\| \sup_{j>1} \left\| \frac{F'(u^{(0)})^{-1} F^{(j)}(u^{(0)})}{j!} \right\|^{\frac{1}{j-1}},$$

where $F^{(j)}$ denotes the jth derivative of F. If $\omega(u^{(0)}) < \omega_0 = .13\cdots$ which is a universal constant, then Newton's method for $F(u) = 0$ with initial guess $u^{(0)}$ converges quadratically in the manner

$$\|u^{(n)} - u\| \leq \left(\frac{1}{2}\right)^{2^{n-1}} \frac{7\,\|u^{(1)} - u^{(0)}\|}{4}.$$

This theory is extremely interesting. It is more practical in the sense that no Lipschitz condition in a region is necessary. However, the computation of ω can be a daunting task. For some problems, the nonlinearity is quadratic (Navier-Stokes equations, for instance) and the supremum in the definition of ω is taken over $j = 2$ only.

For Newton's method applied to our semilinear elliptic PDE,

$$\omega(u^{(0)}) \leq \| \, [I + \triangle^{-1} f'(u^{(0)})]^{-1} \, [u^{(0)} + \triangle^{-1} f(u^{(0)})] \, \| \sup_{j>1} \left\| \frac{f^{(j)}(u^{(0)})}{|\alpha_0(u^{(0)})| \, j!} \right\|_{-1}^{\frac{1}{j-1}}$$

which can usually be worked out in practice. However, for the nonlinear preconditioned Newton's method,

$$\omega(u^{(0)}) = \|\mathcal{F}'(u^{(0)})^{-1} \mathcal{F}(u^{(0)})\| \sup_{j>1} \left\| \frac{\mathcal{F}'(u^{(0)})^{-1} \mathcal{F}^{(j)}(u^{(0)})}{j!} \right\|^{\frac{1}{j-1}}$$

	N	NP0	NP1	NP2
f_1	19, 19	5, 7	4, 6	4, 9
f_2	F, F	8, F	6, 6	6, 6
f_3	12, 12	6, 6	4, 5	4, 5
f_4	8, 8	F, F	4, 4	4, 4
f_5	40, 40	F, F	F, F	F, F
f_6	15, 15	9, 7	8, 6	8, 5

Table 4.1: Comparison of the number of Newton iterations to convergence. F denotes not converged after 100 iterations. The first entry of each pair refers to the number of iterations for an overlap of one point while the second refers to that for an overlap of 10 points.

and we are unable to give a more explicit expression.

Another related result in [16] states that if

$$\|e_N^{(0)}\| < \chi, \qquad \chi \equiv \frac{3-\sqrt{7}}{2}\left(\sup_{j>1}\left\|\frac{F'(u)^{-1}F^{(j)}(u)}{j!}\right\|^{\frac{1}{j-1}}\right)^{-1}, \tag{3.2}$$

then the Newton iteration converges r-quadratically:

$$\|e_N^{(n)}\| \le \left(\frac{1}{2}\right)^{2^{n-1}} \|e_N^{(0)}\|.$$

We shall evaluate χ numerically in the next section.

4. Numerical Experiments and Discussions. We have performed some numerical experiments in MATLAB to solve two-point boundary value problems of the form

$$-u'' = f(x, u, u') \text{ on } (0,1) \tag{4.1}$$

with homogeneous Dirichlet boundary conditions. The ODEs are discretized using the usual second-order finite difference scheme with step size $h = 1/160$ and the resultant nonlinear equations are solved using four methods: classical Newton's method (N), and the three variations of the nonlinearly preconditioned Newton's methods NP0, NP1, and NP2. For the latter three, the domain is split into two overlapping subdomains. Two domain decompositions were tested: one with an overlap of one grid point and the other with an overlap of 10 grid points. Throughout, we employ Newton's method (rather than an inexact Newton's method in [3]) and a simple backtracking algorithm where the length of the Newton step is halved until a sufficient decrease in the residual (Algorithm 6.3.5 in [4]). ([3] uses cubic backtracking.) For a fair comparison, all methods use the same stopping criteria: the nonlinear residual $\|v'' + f(x, v, v')\|_{L^2} < 10^{-8}$ and the L^2-norm of the Newton step is smaller than $h^2 \approx 4 \times 10^{-5}$. (It would be more natural for the nonlinearly preconditioned methods to base the stopping criteria on $\mathcal{F}$ rather than on F.) The initial iterate is always the zero function.

We display the results for six functions

1. $f_1 = (10\sin(10x) - u^3u')/.02$;

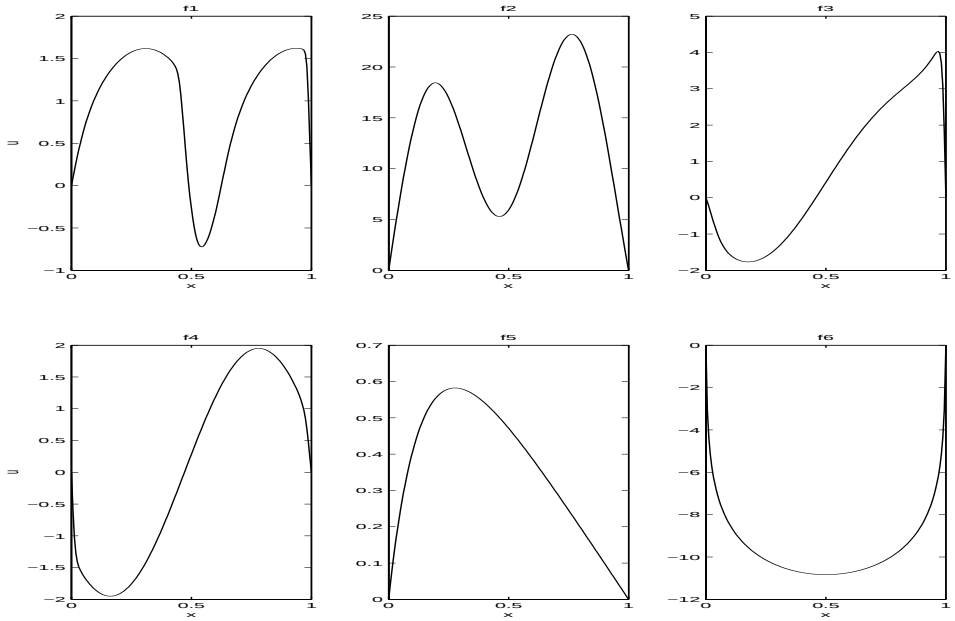


Figure 4.1: Solutions of boundary value problems

2. $f_2 = 100e^{\cdot 1u/(1+u'^2)} + 1000\sin(10x)$;

3. $f_3 = (10\cos(10x) - uu' + e^u)/.03$;

4. $f_4 = (10\cos(10x) - uu')/.01$;

5. $f_5 = \left[u'^2 - u + 1 - \left(\frac{e^{\frac{-x}{\sqrt{\epsilon}}}}{\sqrt{\epsilon}} + e^{\frac{-1}{\sqrt{\epsilon}}} - 1\right)^2 + x(e^{\frac{-x}{\sqrt{\epsilon}}} - 1)\right]/\epsilon, \qquad \epsilon = .02$;

6. $f_6 = -10^6 e^u$;

whose solutions are illustrated in Figure 4.1. Actually we tried other functions too. Most of them were too easy and all four methods converged rapidly. Table 4.1 shows the number of Newton iterations for the different methods and functions. Tables 4.2 tabulates the average number of GMRES iterations to solve each global linear system (ignoring the number of GMRES iterations in solving nonlinear subdomain problems). The number of Newton iterations to solve each nonlinear subdomain ODE is typically four or five. Figure 4.2 shows the convergence history of the methods for f_1.

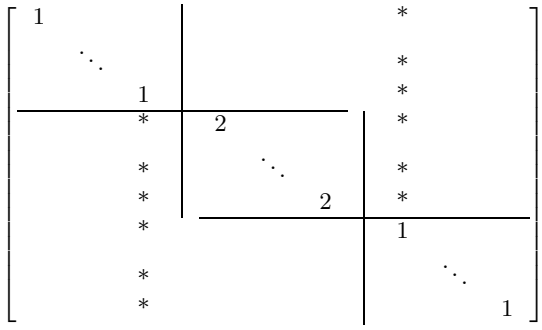
For f_2, the Newton iteration failed to converge after 100 iterations. The residuals decreased at an extremely slow rate. NP0 also failed to converge here (for an overlap of ten) as well as failing for f_4 due to non-convergence of the nonlinear subdomain ODE solver. Here, the algorithm neglects the most up-to-date data ($T_i(u^{(0)})$) causing one iterate to stray too far away. For f_5, Newton's method had some difficulty but eventually converged while all three nonlinearly preconditioned methods failed. The cause of the failures was that the Newton iteration for the subdomain nonlinear equation did not converge, mainly because the initial iterate is too far from the exact solution. Note that Newton's method fails to converge if the

	N	NP0	NP1	NP2
f_1	3.0, 4.0	3.0, 4.0	3.0, 4.0	3.0, 4.0
f_2	F, F	3.0, F	3.0, 4.0	3.0, 4.0
f_3	3.0, 4.0	3.0, 4.0	3.0, 4.0	3.0, 4.0
f_4	3.0, 4.0	F, F	3.0, 4.0	3.0, 4.0
f_5	3.0, 4.0	F, F	F, F	F, F
f_6	2.5, 2.9	2.9, 2.9	3.0, 3.0	3.0, 3.0

Table 4.2: Comparison of the average number of GMRES iterations per Newton step.

constant .02 in f_5 is replaced by .01. Except for f_5, NP1 and NP2 converge with between 1/2 and 1/4 of the iterations required by Newton's method.

In general, the number of GMRES iterations increases from three to four as the overlap increases from one to ten. This can be explained as follows. The matrix approximation of $\mathcal{F}'$ has a rather simple structure:



with non-zero diagonal entries plus two non-zero columns indicated by $*$. Note that the middle block corresponds to the unknowns in the overlapping region. This matrix has at most four distinct eigenvalues, including 1 and 2. Thus GMRES converges in at most four iterations. In the special case that the overlap is one, the the middle block does not appear and so the matrix has a 2×2 block structure and has at most three distinct eigenvalues, including 1. (Note that some authors ([3]) call this case the non-overlapping case.) We stress that this is independent of the step size h.

Next, we numerically evaluate the radii of quadratic convergence for the first convergence theory (q-quadratic convergence). We choose the ODE

$$-u'' = f(u), \qquad f(u) \equiv -\lambda(u+1)(u+2) \tag{4.2}$$

to facilitate this calculation. Initially, we take $\lambda = 100$. For this nonlinearity, γ can be evaluated analytically, equal to $200/\pi^2$. Thus from (3.1),

$$\epsilon_N = \frac{|\alpha_0|\pi^2}{400}.$$

Note that r cannot be much larger than 1.7 because the Jacobian for N can become singular beyond this point. As for NP1, the matrix approximation of $\mathcal{F}'(u)$ is computed explicitly while the Lipschitz constant is estimated numerically. The interval $[0, 1]$ is divided into 160 subintervals in this calculation and overlaps of two and twenty points are considered.

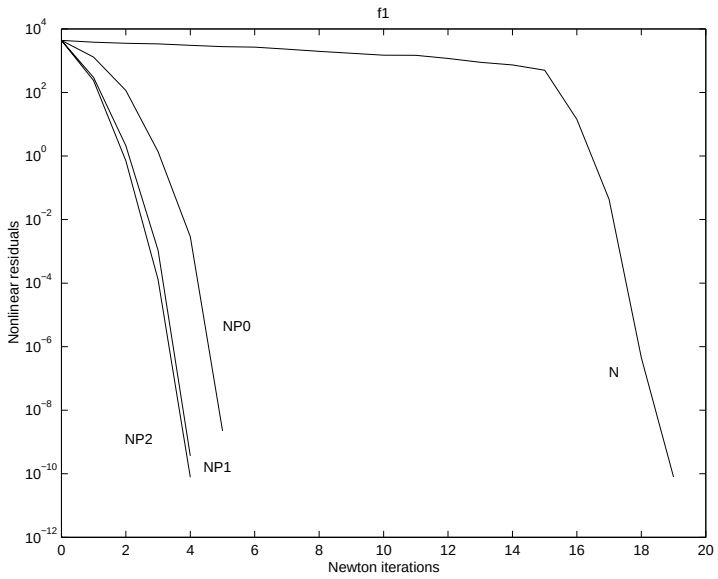


Figure 4.2: Convergence history.

The results are shown in the left diagram of Figure 4.3, which indicates that the radius of quadratic convergence of NP1 is larger than that of Newton's method. This observation should be viewed with some caution because these radii are only lower bounds for the true radii of quadratic convergence. It would be desirable to come up with a sharp upper bound of these radii for comparison.

We also repeated the calculation for $\lambda = 1$ (right diagram in Figure 4.3). Note that for a small overlap, the radius of quadratic convergence of NP1 is actually smaller than that of Newton's method. Any theory must take this into account.

Finally, we report on numerical evaluations of some quantities in Smale's theory for (4.2) with $\lambda = 100$. The main difficulty is in the computation of the supremum term in χ. Currently, we compute all terms up to $j = 20$ in (3.2) and then extrapolate the result (a least squares fit of a rational function) to infinity, a highly speculative process! We obtain $\chi \approx .3$ for NP1 in contrast with the corresponding value of χ for N which is .07. Thus, the estimated radius of quadratic convergence of NP1 is four to five times larger than that of the classical Newton's method. For $\lambda = 1$, the results are qualitative similar, in contrast with the first convergence theory. This may indicate the result of the first theory is not as sharp as Smale's.

Based on these limited experiments, the classical Newton's method does well. Note that each iteration of a nonlinearly preconditioned method costs about twice as much as one iteration of a classical Newton's method in terms of execution time because of the extra nonlinear subdomain solves. NP1 and NP2 are better than NP0 in terms of both speed and robustness. Assuming a parallel computing environment where each processor is assigned to a subdomain, then the addition of y_i in (2.4) involves no communication while replacing y_i by y (NP2) entails communications with all adjacent neighbors. This should not be of much concern since y has to be formed anyway because it is the nonlinear residual $\mathcal{F}(u^{(n)})$. Clearly, many more numerical experiments on nonlinear PDEs are necessary before any definitive conclusion can be reached.

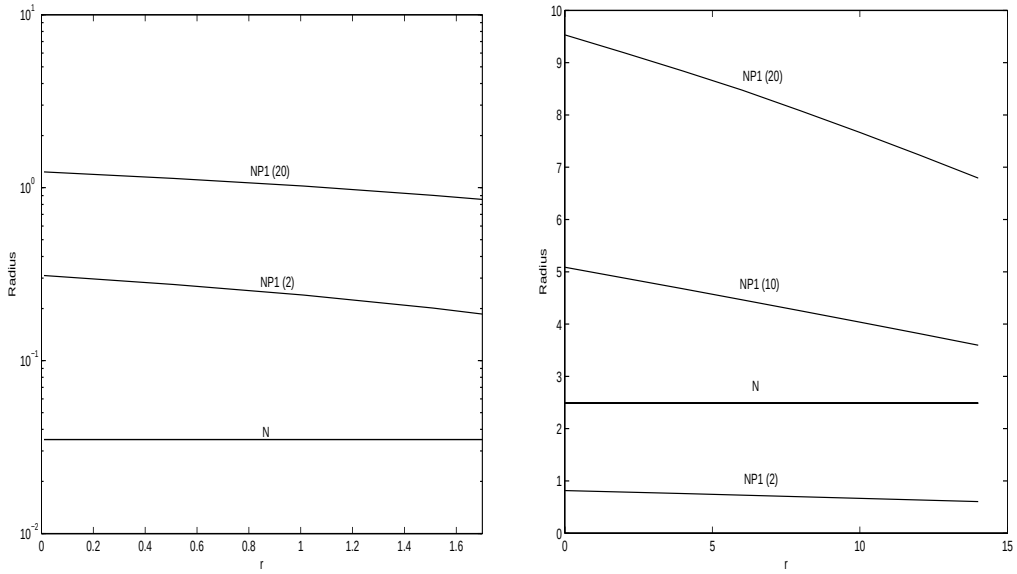


Figure 4.3: Numerical evaluation of ϵ_N and ϵ_{NP1}, the radii of q-quadratic convergence for N and NP1 (with overlap of two, ten, and twenty points) for $u^{(0)} \in B_r(u)$ and $\lambda = 100$ (left), $\lambda = 1$ (right).

While nonlinearly preconditioned Newton's methods are undoubtedly more robust for some problems, they can breakdown when the classical Newton's method works. The main reason is they require the solution of nonlinear subdomain problems which typically involves another Newton's iteration where there is a chance of non-convergence. This can be due to the lack of a good initial guess or may be the subdomain nonlinear problem has no solution or multiple solutions! It is not difficult to write down specific examples where NP1 will fail in the first iteration. This will be discussed in a future report.

REFERENCES

[1] X. C. Cai and M. Dryja. Domain decomposition methods for monotone nonlinear elliptic problems. In D. Keyes and J. Xu, editors, *Domain decomposition methods in scientific and engineering computing*, pages 335–360. AMS, 1994.

[2] X. C. Cai, W. D. Gropp, D. E. Keyes, R. G. Melvin, and D. P. Young. Parallel Newton-Krylov-Schwarz algorithms for the transonic full potential equation. *SISC*, 19:245–265, 1998.

[3] X. C. Cai and D. Keyes. Nonlinearly preconditioned inexact Newton algorithms. *SISC*, 24:183–200, 2002.

[4] J. E. Dennis, Jr and R. B. Schnabel. *Numerical Methods for Unconstrained Optimization and Nonlinear Equations*. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 1996.

[5] M. Dryja and W. Hackbusch. On the nonlinear domain decomposition method. *BIT*, pages 296–311, 1997.

[6] M. Dryja and O. B. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987.

[7] R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors. *Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988. SIAM. Proceedings of

the First International Symposium on Domain Decomposition Methods for Partial Differential Equations, Paris, France, January 1987.

[8] L. V. Kantorovich and G. P. Akilov. *Functional Analysis*. Pergamon Press, New York, 1982.

[9] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Frontiers in applied mathematics. SIAM, 1995.

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

[11] S. H. Lui. On Schwarz alternating methods for nonlinear elliptic PDEs. *SISC*, 21:1506–1523, 2000.

[12] S. H. Lui. On Schwarz alternating methods for the incompressible Navier-Stokes equations. *SISC*, 22:1974–1986, 2001.

[13] S. H. Lui. On linear monotone and Schwarz methods for nonlinear elliptic PDEs. *Numer. Math.*, 2002.

[14] A. M. Matsokin and S. V. Nepomnyaschikh. A Schwarz alternating method in a subspace. *Soviet Mathematics*, 29(10):78–84, 1985.

[15] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.

[16] S. Smale. Newton's method estimates from data at one point. In R. Ewing, K. Gross, and C. Martin, editors, *The Merging of Disciplines: New Directions in Pure, Applied and Computational Math*, pages 185–196, NY, 1986. Springer.

[17] X.-C. Tai and M. S. Espedal. Rate of convergence of some space decomposition method for linear and non-linear elliptic problems. *SIAM J. Numer. Anal.*, 35:1558–1570, 1998.

[18] X. C. Tai and J. Xu. Global convergence of subspace correction methods for convex optimization problems. *Math. Comput.*, 2001.

[19] J. Traub and H. Wozniakowski. Convergence and complexity of Newton iteration. *J. Assoc. Comput. Math.*, 29:250–258, 1979.

[20] X. Wang. Convergence of Newton's method and inverse function theorem in Banach space. *Math. Comp.*, 68:169–186, 1999.

[21] X. Wang. Convergence of Newton's method and uniqueness of the solution of equations in Banach space. *IMA J. Numer. Anal.*, 20:123–134, 2000.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

9. Iterative Substructuring with Lagrange Multipliers for Coupled Fluid-Solid Scattering

Jan Mandel[1]

1. Introduction. In [9], we have proposed an iterative method for the solution of linear systems arizing from finite element discretization of the time harmonic acoustics of coupled fluid-solid systems in fluid pressure and solid displacement formulation. The method extended the FETI-H method for the Helmholtz equation [4, 6, 7, 12] to coupled fluid-elastic acoustics. In this paper, we investigate a stabilization of the discrete coupled system for the case when the solid scatterer is at resonance and investigate computationally the convergence of the iterative substructuring method for the modified system.

The main idea of the method of [9] is as follows. The fluid and the solid domains are decomposed into non-overlapping subdomains. Continuity of the solution is enforced by Lagrange multipliers. To prevent singular or nearly singular subdomain matrices due to resonance, the continuity conditions betweeen the subdomains are replaced by artificial radiation-like conditions. Because original degrees of freedom are coupled across the wet interface, the system is augmented by duplicating the degrees of freedom on the wet interface and adding equations enforcing the equality of the original and the duplicate degrees of freedoom. The original degrees of freedom can then be eliminated subdomain by subdomain and the resulting system is solved by Krylov iterations preconditioned by a Galerkin correction on a subspace consisting of plane waves in each subdomain. In each iteration, the method requires the solution of one independent acoustic problem per subdomain, and the solution of a coarse problem with several degrees of freedom per subdomain. The number of iterations in was most cases about the same as the number of iterations of the FETI-H method for the related Helmholtz problem with Neumann boundary condition instead of an elastic scatterer, which was explained by numerical decoupling of the fluid and the elastic fields in the stiff scatterer limit.

In this article, we propose a new artificial radiation-like condition on the wet interface, and we observe in computational tests that that it it prevents deterioration of convergence in the case of one solid subdomain at resonance. We also investigate the sensitivity of the method to variants of artificial radiation condition between the elastic subdomains.

Our radiation-like condition between elastic subdomains has been inspired by [2], which generalized the alternating method of [5] to elasticity. Iterative methods consisting of alternating solution in the fluid and the solid region are known [1, 3]. In [3], the alternating method of [5] was extended to the coupled problem, with the wet interface conditions replaced by their complex linear combinations. The resulting iterative algorithm needs either access to normal derivatives or additional variables on the wet interface. Our radiation-like condition on the wet interface is obtained by a simple modification of the coupled system matrix, resulting in an equivalent algebraic system. Since this process is unrelated to the substructuring method at hand, it may be of independent interest.

2. The scattering problem. We need to describe the scattering problem and the discretization used. This material is standard [11, 13] and it is included only for completeness and to introduce the notation.

We consider an acoustic scattering problem with an elastic scatterer completely immersed in a fluid. Let Ω and Ω_e be bounded domains in $\Re^n$, $= 2, 3$, $\overline{\Omega}_e \subset \Omega$, and let $\Omega_f = \Omega \setminus \Omega_e$, cf., Figure 5.1. Let ν denote the exterior normal of Ω_e. Let $\partial\Omega$ be decomposed into disjoint subsets, $\partial\Omega = \Gamma_d \cup \Gamma_n \cup \Gamma_a$. The domain Ω_f is filled with a fluid. The acoustic pressure at

[1]University of Colorado at Denver and University of Colorado at Boulder, jmandel@colorado.edu

time t is assumed to be of the form $\mathrm{Re}\, pe^{i\omega t}$, where p is complex amplitude independent of t. The amplitude p is governed by the Helmholtz equation

$$\Delta p + k^2 p = 0 \quad \text{in} \quad \Omega_f, \tag{2.1}$$

with the boundary conditions

$$p = p_0 \text{ on } \Gamma_d, \quad \frac{\partial p}{\partial \nu} = 0 \text{ on } \Gamma_n, \quad \frac{\partial p}{\partial \nu} + ikp = 0 \text{ on } \Gamma_a, \tag{2.2}$$

where $k = \omega/c_f$ is the wave number and c_f is the speed of sound in the fluid. The boundary conditions (2.2) model excitation, sound hard boundary, and outgoing boundary, respectively. The amplitude of the displacement u of the elastic body occupying the domain Ω_e satisfies the elastodynamic equation

$$\nabla \cdot \tau + \omega^2 \rho_e u = 0 \quad \text{in} \quad \Omega_e, \tag{2.3}$$

where τ is the stress tensor and ρ_e is the density of the solid. For simplicity, we consider an isotropic homogeneous material with

$$\tau = \lambda I(\nabla \cdot u) + 2\mu e(u), \quad e_{ij}(u) = \frac{1}{2}(\frac{\partial u_i}{\partial x_j} + \frac{\partial u_j}{\partial x_i}), \tag{2.4}$$

where λ and μ are the Lamé coefficients of the solid.

Let $\Gamma = \partial\Omega_e$ be the wet interface. On Γ, the fluid pressure and the solid displacement satisfy

$$\nu \cdot u = \frac{1}{\rho_f \omega^2} \frac{\partial p}{\partial \nu}, \quad \nu \cdot \tau \cdot \nu = -p, \quad \nu \times \tau \cdot \nu = 0, \tag{2.5}$$

where ρ_f is the fluid density. The interface conditions (2.5) model the continuity of normal displacement, the balance of normal forces, and zero tangential tension, respectively.

We use the following variational form. Define the spaces $V_f = \{q \in H^1(\Omega_f) \mid q = 0 \text{ on } \Gamma_d\}$, $V_e = (H^1(\Omega_e))^n$, where H^1 is the Sobolev space of generalized functions with square integrable generalized first derivatives. Assuming that p_0 on Γ_d is extended to a function in $H^1(\Omega_f)$, multiplying equation (2.1) by a test function $q \in V_f$, equation (2.3) by a test function $u \in V_e$, and integrating by parts, we obtain the following variational form of (2.1) – (2.5): Find p such that $p - p_0 \in V_f$, and $u \in V_e$ such that for all $q \in V_f$ and all $v \in V_e$,

$$\begin{aligned} -\int_{\Omega_f} \nabla p \nabla q + k^2 \int_{\Omega_f} pq - ik \int_{\Gamma_a} pq - \omega^2 \int_{\Gamma} \rho_f (\nu \cdot u) q = 0, \\ -\int_{\Omega_e} \big(\lambda(\nabla \cdot u)(\nabla \cdot v) + 2\mu e(u) : e(v)\big) + \omega^2 \int_{\Omega_e} \rho_e u \cdot v - \int_{\Gamma} p(\nu \cdot v) = 0. \end{aligned}$$

We replace V_f and V_e with conforming finite element spaces and obtain the algebraic system

$$\begin{bmatrix} -\mathbf{K}_f + k^2 \mathbf{M}_f - ik\mathbf{G}_f & -\rho_f \omega^2 \mathbf{T} \\ -\mathbf{T}' & -\mathbf{K}_e + \omega^2 \mathbf{M}_e \end{bmatrix} \begin{bmatrix} \mathbf{p} \\ \mathbf{u} \end{bmatrix} = \begin{bmatrix} \mathbf{r} \\ 0 \end{bmatrix}. \tag{2.6}$$

In the coupled system (2.6), $\mathbf{p}$ and $\mathbf{u}$ are the vectors of the (values of) degrees of freedom of p and u, i.e., p and u are the finite element interpolations of $\mathbf{p}$ and $\mathbf{u}$, respectively. The

matrix blocks in (2.6) are defined by

$$\mathbf{p}'\mathbf{K}_f\mathbf{q} = \int_{\Omega_f} \nabla p \cdot \nabla q, \qquad \mathbf{p}'\mathbf{M}_f\mathbf{q} = \int_{\Omega_f} pq,$$
$$\mathbf{p}'\mathbf{G}_f\mathbf{q} = \int_{\Gamma_a} pq, \qquad \mathbf{u}'\mathbf{K}_e\mathbf{v} = \int_{\Omega_e} \big(\lambda(\nabla\cdot u)(\nabla\cdot v) + 2\mu e(u) : e(v)\big),$$
$$\mathbf{u}'\mathbf{M}_e\mathbf{v} = \int_{\Omega_e} \rho_e(u\cdot v), \qquad \mathbf{p}'\mathbf{T}\mathbf{v} = \int_{\Gamma} p(\nu\cdot v).$$

3. Iterative Substructuring. In this section, we summarize the iterative method following [9]. Further details and a development of the method starting from FETI-H can be found in [9]. The present method differs in the more general choice of artificial radiation condition between elastic subdomains.

The fluid and solid domains are decomposed into nonoverlapping subdomains that consist of unions of elements,

$$\overline{\Omega}_f = \bigcup_{s=1}^{N_f} \overline{\Omega}_e^s, \qquad \overline{\Omega}_e = \bigcup_{s=1}^{N_e} \overline{\Omega}_e^s. \tag{3.1}$$

The fields and vectors of degrees of freedom corresponding to Ω_f^s and Ω_e^s are denoted by p^s, u^s, $\mathbf{p}^s$ and $\mathbf{u}^s$, respectively. The normal vector to $\partial\Omega^s$ is denoted by ν^s.

The Helmholtz equation (2.1) is then equivalent to the same equation in each of the subdomains Ω_f^s, with the interface conditions

$$p^s = p^t, \quad \frac{\partial p^s}{\partial \nu^s} + \frac{\partial p^t}{\partial \nu^t} = 0, \quad \text{on } \partial\Omega_f^s \cap \partial\Omega_f^t. \tag{3.2}$$

Similarly, the elastodynamic equation (2.3) is equivalent to the same equation in each of the subdomains Ω_e^s, with the continuity of the displacement and the traction on the intersubdomain interfaces,

$$u^s = u^t, \quad \tau(u^s)\nu^s + \tau(u^t)\nu^t = 0, \quad \text{on } \partial\Omega_e^s \cap \partial\Omega_e^t. \tag{3.3}$$

The continuity of the pressure and the displacement will be enforced by Lagrange multipliers.

Define subdomain matrices by subassembly,

$$\mathbf{p}^{\mathbf{s}\prime}\mathbf{K}_f^s\mathbf{q} = \int_{\Omega_f^s} \nabla p \cdot \nabla q, \qquad \mathbf{p}^{\mathbf{s}\prime}\mathbf{M}_f^s\mathbf{q}^s = \int_{\Omega_f^s} pq,$$
$$\mathbf{p}^{\mathbf{s}\prime}\mathbf{G}_f^s\mathbf{q}^s = \int_{\partial\Omega_f^s\cap\Gamma_a} pq, \qquad \mathbf{u}^{\mathbf{s}\prime}\mathbf{K}_e^s\mathbf{v}^s = \int_{\Omega_e^s} \lambda(\nabla\cdot u)(\nabla\cdot v) + 2\mu e(u) : e(v),$$
$$\mathbf{u}^{\mathbf{s}\prime}\mathbf{M}_e^s\mathbf{v} = \int_{\Omega_e^s} \rho_e(u\cdot v), \qquad \mathbf{p}^{\mathbf{r}\prime}\mathbf{T}^{rs}\mathbf{v}^s = \int_{\partial\Omega_f^r\cap\partial\Omega_e^s} p(\nu\cdot v).$$

We will use vectors consisting of all subdomain degrees of freedom,

$$\hat{\mathbf{p}} = \begin{bmatrix} \mathbf{p}^1 \\ \vdots \\ \mathbf{p}^{N_f} \end{bmatrix}, \qquad \hat{\mathbf{u}} = \begin{bmatrix} \mathbf{u}^1 \\ \vdots \\ \mathbf{u}^{N_e} \end{bmatrix},$$

and the corresponding partitioned matrices,

$$\hat{\mathbf{K}}_f = \mathrm{diag}(\mathbf{K}_f^s) = \begin{bmatrix} \mathbf{K}_f^1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{K}_f^{N_f} \end{bmatrix}, \qquad \hat{\mathbf{K}}_e = \mathrm{diag}(\mathbf{K}_e^s) = \begin{bmatrix} \mathbf{K}_e^1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{K}_e^{N_e} \end{bmatrix}.$$

The matrices $\hat{\mathbf{M}}_f$, $\hat{\mathbf{G}}_f$, and $\hat{\mathbf{M}}_e$ are defined similarly, and

$$\hat{\mathbf{T}} = (\mathbf{T}^{rs})_{rs} = \begin{bmatrix} \mathbf{T}^{11} & \dots & \mathbf{T}^{1,N_e} \\ \vdots & \ddots & \vdots \\ \mathbf{T}^{N_f,1} & \dots & \mathbf{T}^{N_f,N_e} \end{bmatrix}.$$

Let $\mathbf{N}_f$ and $\mathbf{N}_e$ be the matrices with $0, 1$ entries of the global to local maps corresponding to the decompositions of Ω_f and Ω_e, respectively, cf., (3.1), so that

$$\mathbf{K}_f = \mathbf{N}_f' \hat{\mathbf{K}}_f \mathbf{N}_f, \qquad \mathbf{K}_e = \mathbf{N}_e' \hat{\mathbf{K}}_e \mathbf{N}_e.$$

Let $\mathbf{B}_f = (\mathbf{B}_f^1, \dots, \mathbf{B}_f^{N_f})$ and $\mathbf{B}_e = (\mathbf{B}_e^1, \dots, \mathbf{B}_e^{N_e})$ be matrices of full rank such that the conditions $\mathbf{B}_f \hat{\mathbf{p}} = 0$ and $\mathbf{B}_e \hat{\mathbf{u}} = 0$ express the constraint that the values of the same degrees of freedom on two different subdomains coincide, that is, $\mathbf{B}_f \hat{\mathbf{p}} = 0 \iff \hat{\mathbf{p}} = \mathbf{N}_f \mathbf{p}$ for some $\mathbf{p}$, and $\mathbf{B}_e \hat{\mathbf{u}} = 0 \iff \hat{\mathbf{u}} = \mathbf{N}_e \mathbf{u}$ for some $\mathbf{u}$. See [8] for details on the construction of such matrices with entries $0, \pm 1$. Here, we use the matrices from [8] and orthogonalize their rows for numerical stability; the resulting matrices are still sparse.

Multiplying the second equation in (2.6) by $\omega^2 \rho_f$ to symmetrize the off-diagonal block and introducing Lagrange multipliers λ_f and λ_e for the constraints $\mathbf{B}_f \mathbf{p} = 0$ and $\mathbf{B}_e \mathbf{u} = 0$, we get the system of linear equations in block form,

$$\begin{bmatrix} -\hat{\mathbf{K}}_f + k^2 \hat{\mathbf{M}}_f - ik\hat{\mathbf{G}} & -\omega^2 \rho_f \hat{\mathbf{T}} & \mathbf{B}_f' & 0 \\ -\omega^2 \rho_f \hat{\mathbf{T}}' & \omega^2 \rho_f (-\hat{\mathbf{K}}_e + \omega^2 \hat{\mathbf{M}}_e) & 0 & \mathbf{B}_e' \\ \mathbf{B}_f & 0 & 0 & 0 \\ 0 & \mathbf{B}_e & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{u}} \\ \lambda_f \\ \lambda_e \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{r}} \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.4}$$

where $\mathbf{N}' \hat{\mathbf{r}} = \mathbf{r}$. Similarly as in [8], it can be shown that the system (3.4) is equivalent to (2.6) in the sense that $(\mathbf{p}, \mathbf{u})$ is a solution of (2.6) if and only if $(\hat{\mathbf{p}}, \hat{\mathbf{u}}, \lambda_f, \lambda_e)$ with $\hat{\mathbf{p}} = \mathbf{N}_f \mathbf{p}$, $\hat{\mathbf{u}} = \mathbf{N}_e \mathbf{u}$, is a solution of (3.4) for some λ_f and λ_e.

Using the properties of the global to local maps $\mathbf{N}_f$ and $\mathbf{N}_e$, it is easy to see that $(\hat{\mathbf{p}}, \hat{\mathbf{u}}, \lambda_f, \lambda_e)$ is a solution of (3.4) if and only if $\hat{\mathbf{p}} = \mathbf{N}_f \mathbf{p}$ and $\hat{\mathbf{u}} = \mathbf{N}_e \mathbf{u}$, where $(\mathbf{p}, \mathbf{u})$ solves (2.6).

We will want to eventually eliminate the variables $\hat{\mathbf{p}}$ and $\hat{\mathbf{u}}$. But the matrices $-\hat{\mathbf{K}}_f + k^2 \hat{\mathbf{M}}_f$ and $-\hat{\mathbf{K}}_e + \omega^2 \hat{\mathbf{M}}_e$ may be singular due to resonance. For this reason, the continuity of normal derivative and traction between subdomains are replaced by artificial radiation conditions,

$$p^s + i\sigma^{st} k \frac{\partial p^s}{\partial \nu^s} = u^t + i\sigma^{ts} k \frac{\partial p^t}{\partial \nu^t} \text{ on } \partial\Omega_f^s \cap \partial\Omega_f^t \tag{3.5}$$

and

$$u^s + i\sigma^{ts} \alpha \tau(u^s) \nu^s = u^t + i\sigma^{st} \alpha \tau(u^t) \nu^t, \text{ on } \partial\Omega_e^s \cap \partial\Omega_e^t. \tag{3.6}$$

Here, $\sigma^{st} = \pm 1$ or 0, $\sigma^{st} = -\sigma^{ts}$, and

$$\alpha = \alpha_0 \omega \sqrt{\rho_e (\lambda + 2\mu)}. \tag{3.7}$$

If $\sigma^{st} = \pm 1$, the interface condition (3.5) allows a plane wave to pass in one normal direction through the interface between the subdomains. Similarly, by a simple computation, the condition (3.6) with $\alpha_0 = 1$ is satisfied by the plane pressure wave

$$u(x) = de^{i\frac{\omega}{c_p} d \cdot x}, \quad |d| = 1, \quad c_p = \sqrt{\frac{\lambda + 2\mu}{\rho_e}}, \tag{3.8}$$

in one of the normal directions, $d = \pm\nu^s$. An alternative form of (3.6) is

$$\nu^s(u^s \cdot \nu^s) + i\sigma^{ts}\alpha\tau(u^s)\nu^s = \nu^t(u^t \cdot \nu^t) + i\sigma^{st}\alpha\tau(u^t)\nu^t \text{ on } \partial\Omega_f^s \cap \partial\Omega_f^t, \tag{3.9}$$

which, for α from (3.7) with $\alpha_0 = 1$, is also satisfied by the pressure wave (3.8) in normal direction. In [9], the condition (3.9) with $\alpha = \omega\rho_e$ was used.

This change of intersubdomain interface conditions corresponds to replacing the subdomain matrices $-\hat{\mathbf{K}}_f + k^2\hat{\mathbf{M}}_f$ and $-\hat{\mathbf{K}}_e + \omega^2\hat{\mathbf{M}}_e$ by *regularized matrices*

$$\begin{aligned} \hat{\mathbf{A}}_f &= -\hat{\mathbf{K}}_f + k^2\hat{\mathbf{M}}_f + ik\hat{\mathbf{G}}_f + \hat{\mathbf{R}}_f, \\ \hat{\mathbf{A}}_e &= -\hat{\mathbf{K}}_e + \omega^2\hat{\mathbf{M}}_e + \hat{\mathbf{R}}_e, \end{aligned}$$

where the regularization matrices are given by

$$\hat{\mathbf{R}}_f = \text{diag}(\mathbf{R}_f^s), \qquad \mathbf{p}^{s\prime}\mathbf{R}_f^s\mathbf{q}^s = ik\sum_{\substack{t=1 \\ t\neq s}}^{N_f} \sigma^{st} \int\limits_{\partial\Omega_f^s \cap \partial\Omega_f^t} pq$$

between fluid subdomains, and

$$\hat{\mathbf{R}}_e = \text{diag}(\mathbf{R}_e^s), \qquad \mathbf{u}^{s\prime}\mathbf{R}_e^s\mathbf{v}^s = i\alpha\sum_{\substack{t=1 \\ t\neq s}}^{N_e} \sigma^{st} \int\limits_{\partial\Omega_e^s \cap \partial\Omega_e^t} u \cdot v, \tag{3.10}$$

between elastic subdomains for the interface condition (3.6) and by

$$\hat{\mathbf{R}}_e = \text{diag}(\mathbf{R}_e^s), \qquad \mathbf{u}^{s\prime}\mathbf{R}_e^s\mathbf{v}^s = i\alpha\sum_{\substack{t=1 \\ t\neq s}}^{N_e} \sigma^{st} \int\limits_{\partial\Omega_e^s \cap \partial\Omega_e^t} (\nu^s \cdot u)(\nu^s \cdot v), \tag{3.11}$$

if (3.9) is used.

It is shown in [6] for the Helmholtz equation that if for a given s, all $\sigma^{st} \geq 0$ or all $\sigma^{st} \leq 0$ with some $\sigma^{st} \neq 0$, then $\hat{\mathbf{A}}_f^s$ is invertible. The case of elastic subdomains is similar. For details on strategies for choosing σ^{st} to guarantee this, see [6]. In our computations, we simply choose $\sigma^{st} = +1$ if $s > t$, $\sigma_f^{st} = -1$ if $s < t$.

Because

$$\mathbf{N}_f'\hat{\mathbf{R}}_f\mathbf{N}_f = 0, \qquad \mathbf{N}_e'\hat{\mathbf{R}}_e\mathbf{N}_e = 0,$$

the effect of adding the matrices $\hat{\mathbf{R}}_f$, $\hat{\mathbf{R}}_f$ cancels in the assembled system, and the system (3.4) is equivalent to

$$\begin{bmatrix} \hat{\mathbf{A}}_f & -\omega^2\rho_f\hat{\mathbf{T}} & \mathbf{B}_f' & 0 \\ -\omega^2\rho_f\hat{\mathbf{T}}' & \omega^2\rho_f\hat{\mathbf{A}}_e & 0 & \mathbf{B}_e' \\ \mathbf{B}_f & 0 & 0 & 0 \\ 0 & \mathbf{B}_e & 0 & 0 \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{u}} \\ \lambda_f \\ \lambda_e \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{r}} \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.12}$$

Eliminating the original degrees of freedom at this point does not result in independent computation in each subdomain, because of coupling of degrees of freedom across the wet

interface by the matrix $\hat{\mathbf{T}}$. Hence, we first duplicate the interface degrees of freedom as follows. Since the value of $\hat{\mathbf{T}}\hat{\mathbf{u}}$ depends on the values of $\hat{\mathbf{u}}$ on Γ only, we have

$$\hat{\mathbf{T}}\hat{\mathbf{u}} = \hat{\mathbf{T}}\mathbf{J}_e\hat{\mathbf{u}}_\Gamma, \qquad \hat{\mathbf{u}}_\Gamma = \mathbf{J}'_e\hat{\mathbf{u}},$$

where $\hat{\mathbf{J}}_e$ is the matrix of the operator of embedding a subvector that corresponds to degrees of freedom on Γ into $\hat{\mathbf{u}}$ by adding zero entries. Similarly,

$$\hat{\mathbf{T}}'\hat{\mathbf{p}} = \hat{\mathbf{T}}'\mathbf{J}_f\hat{\mathbf{p}}_\Gamma, \qquad \hat{\mathbf{p}}_\Gamma = \mathbf{J}'_f\hat{\mathbf{p}}.$$

Therefore, we obtain the augmented system equivalent to (3.12),

$$\begin{bmatrix} \hat{\mathbf{A}}_f & 0 & \mathbf{B}'_f & 0 & 0 & -\omega^2\rho_f\hat{\mathbf{T}}\mathbf{J}_e \\ 0 & \omega^2\rho_f\hat{\mathbf{A}}_e & 0 & \mathbf{B}'_e & -\omega^2\rho_f\hat{\mathbf{T}}'\mathbf{J}_f & 0 \\ \mathbf{B}_f & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathbf{B}_e & 0 & 0 & 0 & 0 \\ \mathbf{J}'_f & 0 & 0 & 0 & -\mathbf{I} & 0 \\ 0 & \mathbf{J}'_e & 0 & 0 & 0 & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \hat{\mathbf{p}} \\ \hat{\mathbf{u}} \\ \lambda_f \\ \lambda_e \\ \hat{\mathbf{p}}_\Gamma \\ \hat{\mathbf{u}}_\Gamma \end{bmatrix} = \begin{bmatrix} \hat{\mathbf{r}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.13}$$

Because the variables in a coupled system typically have vastly different scales, we use symmetric diagonal scaling to get the scaled system

$$\begin{bmatrix} \tilde{\mathbf{A}}_f & 0 & \tilde{\mathbf{B}}'_f & 0 & 0 & -\tilde{\mathbf{T}}\mathbf{J}_e \\ 0 & \tilde{\mathbf{A}}_e & 0 & \tilde{\mathbf{B}}'_e & -\tilde{\mathbf{T}}'\mathbf{J}_f & 0 \\ \tilde{\mathbf{B}}_f & 0 & 0 & 0 & 0 & 0 \\ 0 & \tilde{\mathbf{B}}_e & 0 & 0 & 0 & 0 \\ \mathbf{J}'_f & 0 & 0 & 0 & -\mathbf{I} & 0 \\ 0 & \mathbf{J}'_e & 0 & 0 & 0 & -\mathbf{I} \end{bmatrix} \begin{bmatrix} \tilde{\mathbf{p}} \\ \tilde{\mathbf{u}} \\ \tilde{\lambda}_f \\ \tilde{\lambda}_e \\ \tilde{\mathbf{p}}_\Gamma \\ \tilde{\mathbf{u}}_\Gamma \end{bmatrix} = \begin{bmatrix} \tilde{\mathbf{r}} \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \tag{3.14}$$

where the matrices and the vectors scale as $\tilde{\mathbf{A}}_f = \mathbf{D}_f\hat{\mathbf{A}}_f\mathbf{D}_f$, $\tilde{\mathbf{A}}_e = \omega^2\rho_f\mathbf{D}_e\hat{\mathbf{A}}_e\mathbf{D}_e$, $\tilde{\mathbf{T}} = \omega^2\rho_f\mathbf{D}_f\hat{\mathbf{T}}\mathbf{D}_e$, $\tilde{\mathbf{B}}_f = \mathbf{E}_f\mathbf{B}_f\mathbf{D}_f$, $\tilde{\mathbf{B}}_e = \mathbf{E}_e\mathbf{B}_e\mathbf{D}_e$, $\tilde{\mathbf{r}} = \mathbf{D}_f\hat{\mathbf{r}}$, $\hat{\mathbf{p}} = \mathbf{D}_f\tilde{\mathbf{p}}$, $\hat{\mathbf{u}} = \mathbf{D}_e\tilde{\mathbf{u}}$, $\lambda_f = \mathbf{D}_f\tilde{\lambda}_f$, $\lambda_e = \mathbf{D}_e\tilde{\lambda}_e$. The scaling matrices $\mathbf{D}_f$, $\mathbf{D}_e$, $\mathbf{E}_f$, and $\mathbf{E}_e$, are diagonal. We have chosen scaling matrices with positive diagonal entries such that the absolute values of the diagonal entries of $\tilde{\mathbf{A}}_f$ and $\tilde{\mathbf{A}}_e$ are one and the ℓ^2 norms of the columns of $\tilde{\mathbf{B}}_e$ and $\tilde{\mathbf{B}}_f$ are one.

Computing $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{u}}$ from the first two equations in (3.14) gives

$$\tilde{\mathbf{p}} = \tilde{\mathbf{A}}_f^{-1}(\tilde{\mathbf{r}} - \tilde{\mathbf{B}}'_f\tilde{\lambda}_f + \tilde{\mathbf{T}}\mathbf{J}_e\tilde{\mathbf{u}}_\Gamma) \tag{3.15}$$

$$\tilde{\mathbf{u}} = \tilde{\mathbf{A}}_e^{-1}(-\tilde{\mathbf{B}}'_e\tilde{\lambda}_e + \tilde{\mathbf{T}}'\mathbf{J}_f\tilde{\mathbf{p}}_\Gamma) \tag{3.16}$$

Substituting $\tilde{\mathbf{p}}$ and $\tilde{\mathbf{u}}$ from (3.15), (3.16) into the rest of the equations in (3.14), we obtain the reduced system

$$\mathbf{F}\mathbf{x} = \mathbf{b}, \tag{3.17}$$

where

$$\mathbf{F} = \begin{bmatrix} \tilde{\mathbf{B}}_f\tilde{\mathbf{A}}_f^{-1}\tilde{\mathbf{B}}'_f & 0 & 0 & -\tilde{\mathbf{B}}_f\tilde{\mathbf{A}}_f^{-1}\tilde{\mathbf{T}}\mathbf{J}_e \\ 0 & \tilde{\mathbf{B}}_e\tilde{\mathbf{A}}_e^{-1}\tilde{\mathbf{B}}_e & -\tilde{\mathbf{B}}_e\tilde{\mathbf{A}}_e^{-1}\tilde{\mathbf{T}}'\mathbf{J}_f & \\ -\mathbf{J}'_f\tilde{\mathbf{A}}_f^{-1}\tilde{\mathbf{B}}'_f & 0 & -\mathbf{I} & \mathbf{J}'_f\tilde{\mathbf{A}}_f^{-1}\tilde{\mathbf{T}}\mathbf{J}_e \\ 0 & -\mathbf{J}_e\tilde{\mathbf{A}}_e^{-1}\tilde{\mathbf{B}}_e & \mathbf{J}_e\tilde{\mathbf{A}}_e^{-1}\tilde{\mathbf{T}}'\mathbf{J}_f & -I \end{bmatrix}, \tag{3.18}$$

and

$$\mathbf{x} = \begin{bmatrix} \lambda_f \\ \lambda_e \\ \tilde{\mathbf{p}}_\Gamma \\ \tilde{\mathbf{u}}_\Gamma \end{bmatrix}, \qquad \mathbf{b} = \begin{bmatrix} \tilde{\mathbf{B}}_f\tilde{\mathbf{A}}_f^{-1}\tilde{\mathbf{r}} \\ 0 \\ -\mathbf{J}'_f\tilde{\mathbf{A}}_f^{-1}\tilde{\mathbf{r}} \\ 0 \end{bmatrix}.$$

In equation (3.18), the first diagonal block $\tilde{\mathbf{B}}_f\tilde{\mathbf{A}}_f^{-1}\tilde{\mathbf{B}}_f'$ is exactly same as in the FETI-H method for the Helmholtz equation. The second diagonal block $\tilde{\mathbf{B}}_f\tilde{\mathbf{A}}_f^{-1}\tilde{\mathbf{B}}_f'$ is the analogue of FETI-H for the elastodynamic problem.

Evaluating the matrix vector product $\mathbf{Fx}$ requires the solution of one independent problem per subdomain, because

$$\mathbf{F}\begin{bmatrix}\lambda_f\\ \lambda_e\\ \tilde{\mathbf{p}}_\Gamma\\ \tilde{\mathbf{u}}_\Gamma\end{bmatrix}=\begin{bmatrix}-\tilde{\mathbf{B}}_f\tilde{\mathbf{q}}\\ -\tilde{\mathbf{B}}_e\tilde{\mathbf{v}}\\ \mathbf{J}_f'\tilde{\mathbf{q}}-\tilde{\mathbf{p}}_\Gamma\\ \mathbf{J}_e'\tilde{\mathbf{v}}-\tilde{\mathbf{u}}_\Gamma\end{bmatrix},\quad\text{where}\quad\begin{cases}\tilde{\mathbf{q}}=\tilde{\mathbf{A}}_f^{-1}(-\tilde{\mathbf{B}}_f'\tilde{\lambda}_f+\tilde{\mathbf{T}}\mathbf{J}_e\tilde{\mathbf{u}}_\Gamma),\\ \tilde{\mathbf{v}}=\tilde{\mathbf{A}}_e^{-1}(-\tilde{\mathbf{B}}_e'\tilde{\lambda}_e+\tilde{\mathbf{T}}'\mathbf{J}_f\tilde{\mathbf{p}}_\Gamma).\end{cases}$$

The iterative method then consists of solving the linear system (3.17) by GMRES preconditioned by a subspace correction as follows. Let $\mathbf{Q}$ be a matrix with the same number of rows as $\mathbf{F}$ and linearly independent columns. The columns of $\mathbf{Q}$ form the basis of the *coarse space*. The orthogonality condition

$$\mathbf{Q}'(\mathbf{Fx}-\mathbf{b})=0, \tag{3.19}$$

is enforced through the iterations by adding a correction from the coarse space in each iteration. That is, GMRES is applied to the preconditioned system

$$\mathbf{PFx}=\mathbf{Pb}, \tag{3.20}$$

where $\mathbf{P}=(\mathbf{I}-\mathbf{Q}(\mathbf{Q}'\mathbf{FQ})^{-1}\mathbf{Q}'\mathbf{F})$ and the initial approximation $\mathbf{x}=\mathbf{Q}(\mathbf{Q}'\mathbf{FQ})^{-1}\mathbf{b}$ satisfies (3.19). Because the increments are in the range of $\mathbf{P}$ and $\mathbf{Q}'\mathbf{FP}=0$, all iterates satisfy (3.19).

We choose the matrix $\mathbf{Q}$ of the form

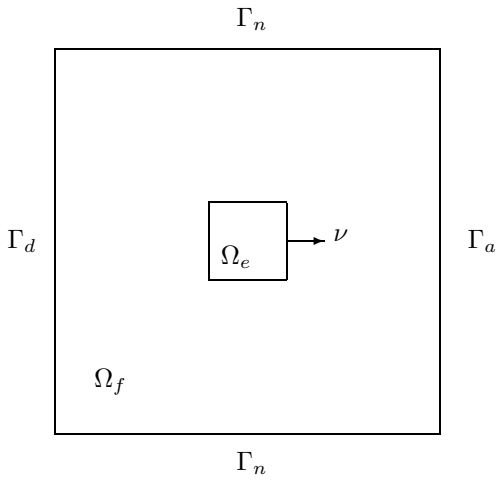
$$\begin{bmatrix}\mathbf{D}_f\mathbf{B}_f\mathrm{diag}(\mathbf{Y}_f^s)_s & 0 & 0 & 0\\ 0 & \mathbf{D}_e\mathbf{B}_e\mathrm{diag}(\mathbf{Y}_e^s)_s & 0 & 0\\ 0 & 0 & \mathbf{D}_f\mathbf{J}_f'\mathrm{diag}(\mathbf{Z}_f^s)_s & 0\\ 0 & 0 & 0 & \mathbf{D}_e\mathbf{J}_e'\mathrm{diag}(\mathbf{Z}_e^s)_s\end{bmatrix}. \tag{3.21}$$

and orthogonalize its columns by the QR algorithm. For a fluid subdomain Ω_f^s, we choose $\mathbf{Y}_f^s$ as the matrix of columns that are discrete representations of plane waves in a small number of equally distributed directions, or discrete representation of the constant function. For a solid subdomain Ω_e^s, the columns of $\mathbf{Y}_e^s$ are discrete representations of plane pressure and shear waves, or of rigid body motions. The matrices $\mathbf{Z}_f^s$ and $\mathbf{Z}_e^s$ are chosen in the same way as $\mathbf{Y}_f^s$ and $\mathbf{Y}_e^s$. See [9] for further details and a discussion why the method for the coupled problem can be expected to perform about as FETI-H for the fluid and the elastic parts separately.

4. Radiation-Like Condition on the Wet Interface. For some frequencies, the matrix $-\mathbf{K}_e+\omega^2\mathbf{M}_e$ in the coupled system (2.6) will be singular. The inverse of this matrix is required by the method if there is only one elastic subdomain. Therefore, in this case, we replace (2.6) by an equivalent system, obtained by adding to the second block of equations a linear combination of the first block in such a way that the term added to $-\mathbf{K}_e+\omega^2\mathbf{M}_e$ resembles a radiation condition:

$$\begin{bmatrix}-\mathbf{K}_f+k^2\mathbf{M}_f-ik\mathbf{G}_f & -\rho_f\omega^2\mathbf{T}\\ -\mathbf{T}'+i\frac{\beta}{-\rho_f\omega^2}\mathbf{T}'(-\mathbf{K}_f+k^2\mathbf{M}_f-ik\mathbf{G}_f) & -\mathbf{K}_e+\omega^2\mathbf{M}_e+i\beta\mathbf{T}'\mathbf{T}\end{bmatrix}\begin{bmatrix}\mathbf{p}\\ \mathbf{u}\end{bmatrix}=\mathbf{y}, \tag{4.1}$$

Figure 5.1: Model 2D Problem



where

$$\mathbf{y} = \begin{bmatrix} \mathbf{r} \\ i\frac{\beta}{-\rho_f\omega^2}\mathbf{T}'r \end{bmatrix}.$$

To obtain an added term with consistent physical units and similar to the artificial radiation condition (3.10), we choose

$$\beta = \beta_0 \frac{\omega\sqrt{\rho_e(\lambda+2\mu)}}{\|\mathbf{T}\|_1}. \tag{4.2}$$

In the case of more than several fluid subdomains and one elastic subdomain, this process is easily implemented using the local subdomain matrices for $\mathbf{T}$. For more than one elastic subdomain, computational experiments indicate that introducing an artificial radiation condition on the wet interface is not necessary.

5. Computational results. Computational results showing scalability of the method were presented in [9]. Here, we focus on the performance of the method when k equals or is close to a value that makes some of the subdomain matrices singular, and for different choices of the radiation-like condition between elastic subdomains.

We consider a model 2D problem with a scatterer in the center a waveguide, cf., Fig. 5.1. The fluid domain Ω_f is a square with side $1\,m$, filled with water with density $\rho_f = 1000\,kg\,m^{-3}$ and speed of sound $c_f = 1500\,m\,s^{-1}$. The scatterer is a square in the center of the fluid domain, consisting of aluminum with density $\rho_e = 2700\,kg\,m^{-3}$ and Lamé elasticity coefficients $\lambda = 5.5263.10^{10}\,N\,m^{-2}$, $\mu = 2.595.10^{10}\,N\,m^{-2}$. The domain is discretized with a mesh of 200 by 200 bilinear elements. The coarse space consists of 8 plane waves in the fluid subdomains and 4 plane pressure waves and 4 plane shear waves in the solid subdomains (the first two blocks of the matrix $\mathbf{Q}$ in (3.21)). The same number of coarse space functions is used for the coarse space for the wet interface (the last two blocks of the matrix $\mathbf{Q}$). The iterations are terminated when the relative residual reached 10^{-6}. Then the scaled residual in the original variables are checked,

$$Res = \max_i \frac{|\mathbf{d}_i - \sum_j \mathbf{K}_{ij}\mathbf{z}_j|}{\sum_j |\mathbf{K}_{ij}|\,|\mathbf{z}_j|}, \tag{5.1}$$

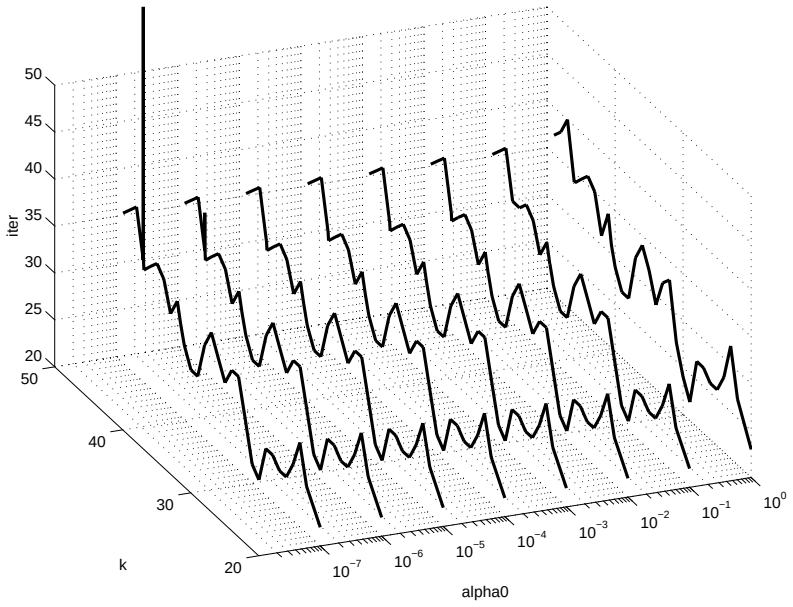


Figure 5.2: Number of iterations for different α_0 in artificial radiation condition (3.6) between elastic subdomains

where $\mathbf{z}$, $\mathbf{K}$, and $\mathbf{d}$ are the solution vector, the matrix, and the right-hand-side, respectively, of the coupled system (2.6). In all cases when the iterations converged, this scaled residual was of the order 10^{-6} to 10^{-7}.

Figure 5.2 shows the number of iterations for varying constant α_0 in the artificial radiation condition (3.6) between elastic subdomains. The scatter was size 0.4 by 0.4 and the fluid and the elastic domains were decomposed into 4 subdomains each. One can see that the number of iterations for $\alpha_0 = 1$ is slightly larger over all frequencies, while the iterations diverge for frequencies equal to or very close the resonance frequencies for $\alpha_0 \leq 10^{-4}$.

The number of iterations for the same test problem and the artifical radiation condition (3.9) was almost exactly same (not shown).

Figure 5.3 reports the number of iterations for the same problem with the artificial radiation condition (3.6) between elastic subdomains, but instead of orthogonalization of the rows of the matrices $\mathbf{B}$ and the columns of the matrix $\mathbf{Q}$, bases are selected as linearly independent subsets. There are more iterations required and divergence occurs for more frequencies and larger values of the parameter α_0.

Figure 5.4 reports the number of iterations for decreasing strength β_0 of the artificial radiation-like term on the wet interface. The scatterer was size 0.2 by 0.2, forming one elastic subdomain, and the fluid domain was decomposed along the midlines of the square into 4 subdomains. One can see that the choice $\beta_0 = 1$ increases the number of iterations significantly over all frequencies, while for $\beta_0 = 10^{-5}$, the iterations diverge for frequencies equal to or close to a resonance frequency. The elastic subdomain in this experiment is of the same size as the elastic subdomain for the examples in Figure 5.2, so the resonance frequencies are same.

6. Acknowledgements. This research was supported in part by the Office of Naval Research under grant N-00014-95-1-0663, and the National Science foundation under grant

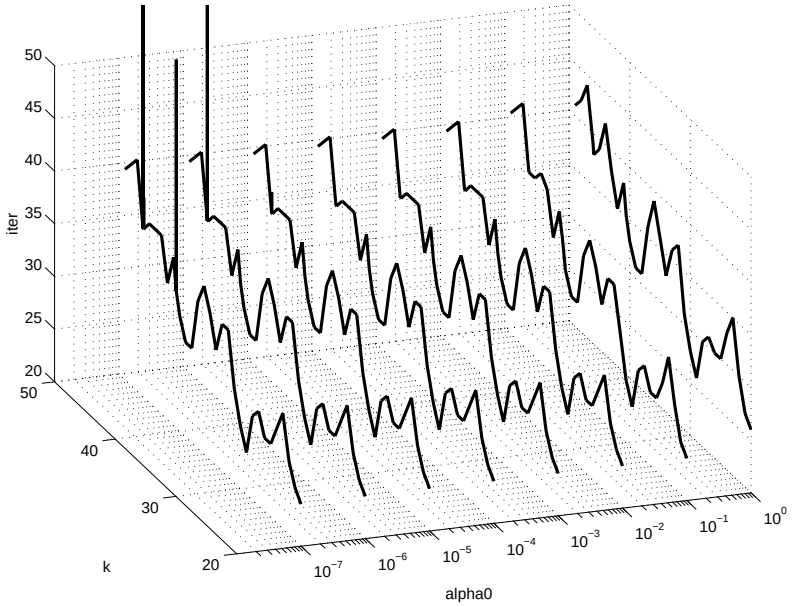


Figure 5.3: Number of iterations for different α_0 in artificial radiation condition (3.6) between elastic subdomains and selection of basis instead of orthogonalization of the rows of $\mathbf{B}$ and the columns of $\mathbf{Q}$

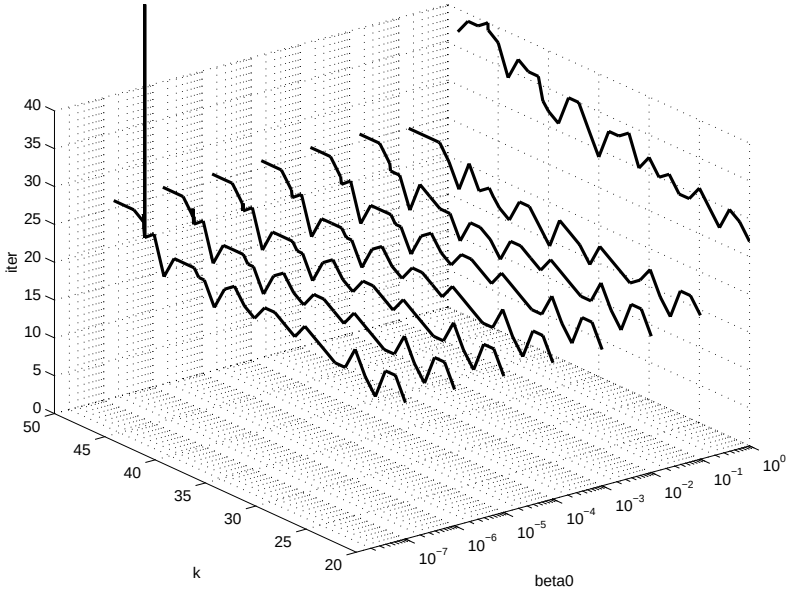


Figure 5.4: Number of iterations for different β_0 in artificial radiation on wet interface by (4.1) and (4.2)

DMS-007428. The author would like to thank Charbel Farhat, Rabia Djellouli, and Radek Tezaur for useful dicussions. Special thanks are due to Rabia Djellouli for pointing out that a radiation-like condition across the wet interface was missing in [9]. The prototype MATLAB code used in the experiments was partially based on a code written by Mirela Popa and the author [10].

REFERENCES

[1] A. I. Achil'diev and K. S. Nazhmidinov. An iterative method for solving a diffraction problem in an unbounded medium. *Dokl. Akad. Nauk Tadzhik. SSR*, 31(7):429–432, 1988.

[2] L. S. Bennethum and X. Feng. A domain decomposition method for solving a Helmholtz-like problem in elasticity based on the Wilson nonconforming element. *R.A.I.R.O., Anal. Numér.*, 31:1–25, 1997.

[3] P. Cummings and X. Feng. Domain decomposition methods for a system of coupled acoustic and elastic Helmholtz equations. In C.-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, editors, *Eleventh International Conference on Domain Decomposition Methods*, pages 203–210, Bergen, Norway, 1999. Domain Decomposition Press.

[4] A. de La Bourdonnaye, C. Farhat, A. Macedo, F. Magoulès, and F.-X. Roux. A non-overlapping domain decomposition method for exterior Helmholtz problems. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 42–66, Providence, RI, 1998. Amer. Math. Soc.

[5] B. Després. Domain decomposition method and the Helmholtz problem.II. In *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993)*, pages 197–206, Philadelphia, PA, 1993. SIAM.

[6] 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(2):283–303, 2000.

[7] C. Farhat, A. Macedo, and R. Tezaur. FETI-H: a scalable domain decomposition method for high frequency exterior Helmholtz problem. In C.-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, editors, *Eleventh International Conference on Domain Decomposition Method*, pages 231–241. DDM.ORG, 1999.

[8] C. Farhat and F. X. Roux. Implicit parallel processing in structural mechanics. *Comput. Mech. Adv.*, 2:1–124, 1994.

[9] J. Mandel. An iterative substructuring method for coupled fluid-solid acoustic problems. *Journal of Computational Physics*, 176:1–22, 2002.

[10] J. Mandel and M. Popa. A multigrid method for elastic scattering. UCD/CCM Report 109, Center for Computational Mathematics, University of Colorado at Denver, 1997.

[11] H. Morand and R. Ohayon. *Fluid Structure Interaction*. John Wiley and Sons, 1995.

[12] R. Tezaur, A. Macedo, and C. Farhat. Iterative solution of large-scale acoustic scattering problems with multiple right hand-sides by a domain decomposition method with Lagrange multipliers. *Internat. J. Numer. Methods Engrg.*, 51(10):1175–1193, 2001.

[13] V. K. Varadan and V. V. Varadan. Acoustic, electromagnetic, and elastodynamic fields. In V. K. Varadan, A. Lakhtakia, and V. V. Varadan, editors, *Field Representations and Introduction to Scattering*. North-Holland, Amsterdam, 1991.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

10. Direct simulation of the motion of settling ellipsoids in Newtonian fluid

T.-W. Pan[1], R. Glowinski[2], D.D. Joseph[3], R. Bai[4]

1. Introduction. In this article we first discuss the generalization of a Lagrange multiplier based fictitious domain method [7, 10] to the simulation of the motion of particles of general shape in a Newtonian fluid. Unlike the cases where the particles are spheres, we attach two points, besides the center of mass, to each particle of general shape and move them according to the rigid-body motion of the particle in order to track this motion. The equations describing the motion of these two points are solved by a distance preserving scheme so that rigidity can be maintained. We then apply it to simulate ellipsoids settling in a narrow channel filled with a Newtonian fluid. In the simulations, when there is only one ellipsoid it turns its broadside orthogonal to the stream as expected; for the two ellipsoid case they interact with each other as observed in experiments.

2. A model problem and fictitious domain formulation for three dimensional particulate flow. To perform the *direct numerical simulation* of the interaction between particles and fluid, we have developed a methodology which is a combination of a distributed Lagrange multiplier based fictitious domain (also called domain embedding) method and operator splitting methods [6, 8, 7, 9, 10], this approach (or closely related ones derived from it) has become the method of choice for other investigators around the world (refs., Baaijens in [2] and Wagner et al. in [21]). In the following we are going to recall the ideas at the basis of the above methodology, but with generalization to the motion of a single particle of general shape in a Newtonian viscous incompressible fluid (of density ρ_f and viscosity ν_f) under the effect of gravity. For the situation depicted in Figure 2.1 below, the flow is modeled by the Navier-Stokes equations, namely, (with obvious notation)

$$\rho_f\Big[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u}\Big] - \nu_f\Delta\mathbf{u} + \boldsymbol{\nabla}p = \rho_f\,\mathbf{g}\ in\ (\Omega\setminus\bar{B})\times(0,T), \tag{2.1}$$

$$\boldsymbol{\nabla}\cdot\mathbf{u} = 0\ in\ (\Omega\setminus\bar{B})\times(0,T). \tag{2.2}$$

$$\mathbf{u}(0) = \mathbf{u}_0(\mathbf{x}), (with\ \boldsymbol{\nabla}\cdot\mathbf{u}_0 = 0) \tag{2.3}$$

$$\mathbf{u} = \mathbf{g}_0\ on\ \Gamma\times(0,T),\ with\ \int_\Gamma \mathbf{g}_0\cdot\mathbf{n}\,d\Gamma = 0, \tag{2.4}$$

where $\Gamma = \partial\Omega$, $\mathbf{g}$ is gravity and $\mathbf{n}$ is the unit normal vector pointing outward to the flow region. We assume a *no-slip condition* on $\gamma(=\partial B)$ The motion of particle B satisfies the Euler-Newton's equations, namely

$$\mathbf{v}(\mathbf{x},t) = \mathbf{V}(t) + \boldsymbol{\omega}(t)\times\overrightarrow{\mathbf{G}(t)\mathbf{x}},\ \forall\{\mathbf{x},t\}\in\overline{B(t)}\times(0,T), \tag{2.5}$$

$$\frac{d\mathbf{G}}{dt} = \mathbf{V}, \tag{2.6}$$

$$M_p\frac{d\mathbf{V}}{dt} = M_p\,\mathbf{g} + \mathbf{F}_H + \mathbf{F}^r, \tag{2.7}$$

[1]University of Houston, Department of Mathematics, Houston, Texas 77204, USA, pan@math.uh.edu
[2]University of Houston, Department of Mathematics, Houston, Texas 77204, USA, roland@math.uh.edu
[3]Department of Aerospace Engineering & Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, USA, joseph@aem.umn.edu
[4]Department of Aerospace Engineering & Mechanics, University of Minnesota, Minneapolis, Minnesota 55455, USA, bai@aem.umn.edu

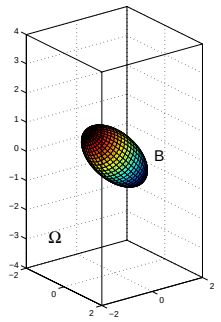


Figure 2.1: The flow region with one particle

$$\frac{d(\mathbf{I}_p\,\boldsymbol{\omega})}{dt} = \mathbf{T}_H + \overrightarrow{\mathbf{G}\mathbf{x}_r} \times \mathbf{F}^r, \tag{2.8}$$

with hydrodynamical forces and torques given by

$$\mathbf{F}_H = -\int_\gamma \boldsymbol{\sigma}\mathbf{n}\,d\gamma, \quad \mathbf{T}_H = -(\int_\gamma \overrightarrow{\mathbf{G}\mathbf{x}} \times \boldsymbol{\sigma}\mathbf{n}\,d\gamma), \tag{2.9}$$

completed by the following initial conditions,

$$\mathbf{G}(0) = \mathbf{G}_0,\ \mathbf{V}(0) = \mathbf{V}_0,\ \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0. \tag{2.10}$$

Above, M_p, $\mathbf{I}_p$, $\mathbf{G}$, $\mathbf{V}$ and $\boldsymbol{\omega}$ are the mass, inertia, center of mass, translation velocity of the center of mass and angular velocity of particle B, respectively. In (2.8) we found preferable to deal with the *kinematic angular momentum* $\mathbf{I}_p\,\boldsymbol{\omega}$ making the formulation more conservative. In order to avoid particle-particle and particle-wall penetration which can happen in the numerical simulation, we have introduced an artificial force $\mathbf{F}^r$ in (2.7) (for more details, see, e.g., [7] and [10]) and then a torque in (2.8) acting on the point $\mathbf{x}_r$ where $\mathbf{F}^r$ applies on B.

To solve system (2.1) – (2.10) we can use, for example, *Arbitrary Lagrange-Euler* (*ALE*) methods as in [12, 14, 17], or *fictitious domain methods*, which allow the flow calculation on a fixed grid, as in [6, 8, 7, 9, 10]. The fictitious domain methods that we advocate have some common features with the *immersed boundary method* of Ch. Peskin (see, e.g., refs. [18, 19]) but also some significant differences in the sense that we take systematically advantage of *distributed Lagrange multipliers* to force the rigid body motion inside the particle, which seems still to be a relatively novel approach in this context, and whose possibilities have not been fully explored yet. As with the methods in [18, 19], our approach takes advantage of the fact that the flow can be computed on a grid which does not have to vary in time, a substantial simplification indeed.

The principle of fictitious domain methods is simple. It consists of

- Filling the particles with a fluid having the same density and viscosity as the surrounding one.
- Compensating the above step by introducing, in some sense, an *anti-particle* of mass $(-1)M_p\,\rho_f/\rho_s$ and inertia $(-1)\mathbf{I}_p\,\rho_f/\rho_s$, taking into account the fact that any rigid body motion $\mathbf{v}(\mathbf{x},t)$ verifies $\boldsymbol{\nabla}\cdot\mathbf{v} = 0$ and $\mathbf{D}(\mathbf{v}) = \mathbf{0}$ (ρ_s : particle density).
- Finally, imposing the rigid body velocity on $\overline{B(t)}$, namely

$$\mathbf{v}(\mathbf{x},t) = \mathbf{V}(t) + \boldsymbol{\omega}(t) \times \overrightarrow{\mathbf{G}(t)\mathbf{x}}, \forall \mathbf{x} \in \overline{B(t)}, \forall t \in (0,T), \tag{2.11}$$

via a Lagrange multiplier $\boldsymbol{\lambda}$ supported by $\overline{B(t)}$. Vector $\boldsymbol{\lambda}$ forces rigidity in $B(t)$ in the same way that $\boldsymbol{\nabla} p$ forces $\boldsymbol{\nabla}\cdot\mathbf{v}=0$ for incompressible fluids.

We obtain then an equivalent formulation of (2.1)–(2.10) defined on the whole domain, namely

For a.e. $t>0$, *find* $\{\mathbf{u}(t),p(t),\mathbf{V}(t),\mathbf{G}(t),\boldsymbol{\omega}(t),\boldsymbol{\lambda}(t)\}$ *such that*

$$\mathbf{u}(t)\in\mathbf{W}_{\mathbf{g}_0}(t),\ p(t)\in L^2_0(\Omega),\ \mathbf{V}(t)\in\mathbb{R}^3,\ \mathbf{G}(t)\in\mathbb{R}^3,\ \boldsymbol{\omega}(t)\in\mathbb{R}^3,\ \boldsymbol{\lambda}(t)\in\Lambda(t) \tag{2.12}$$

and

$$\begin{cases}\rho_f\displaystyle\int_\Omega\left[\frac{\partial\mathbf{u}}{\partial t}+(\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u}\right]\cdot\mathbf{v}d\mathbf{x}-\int_\Omega p\boldsymbol{\nabla}\cdot\mathbf{v}d\mathbf{x}+\nu_f\int_\Omega\boldsymbol{\nabla}\mathbf{u}:\boldsymbol{\nabla}\mathbf{v}d\mathbf{x}\\ \quad+(1-\dfrac{\rho_f}{\rho_s})[M_p\dfrac{d\mathbf{V}}{dt}\cdot\mathbf{Y}+\dfrac{d(\mathbf{I}_p\,\boldsymbol{\omega})}{dt}\cdot\boldsymbol{\theta}]-\mathbf{F}^r\cdot\mathbf{Y}-\overrightarrow{\mathbf{G}\mathbf{x}_r}\times\mathbf{F}^r\cdot\boldsymbol{\theta}\\ =<\boldsymbol{\lambda},\mathbf{v}-\mathbf{Y}-\boldsymbol{\theta}\times\overrightarrow{\mathbf{G}\mathbf{x}}>_{\Lambda(t)}+(1-\dfrac{\rho_f}{\rho_s})M_p\ \mathbf{g}\cdot\mathbf{Y}+\rho_f\displaystyle\int_\Omega\mathbf{g}\cdot\mathbf{v}d\mathbf{x},\\ \forall\mathbf{v}\in(H^1_0(\Omega))^3,\ \forall\mathbf{Y}\in\mathbb{R}^3,\ \forall\boldsymbol{\theta}\in\mathbb{R}^3,\end{cases} \tag{2.13}$$

$$\int_\Omega q\boldsymbol{\nabla}\cdot\mathbf{u}(t)d\mathbf{x}=0,\ \forall q\in L^2(\Omega), \tag{2.14}$$

$$\frac{d\mathbf{G}}{dt}=\mathbf{V}, \tag{2.15}$$

$$<\boldsymbol{\mu},\mathbf{u}(t)-\mathbf{V}(t)-\boldsymbol{\omega}(t)\times\overrightarrow{\mathbf{G}(t)\mathbf{x}}>_{\Lambda(t)}=0,\ \forall\boldsymbol{\mu}\in\Lambda(t), \tag{2.16}$$

$$\mathbf{V}(0)=\mathbf{V}_0,\ \boldsymbol{\omega}(0)=\boldsymbol{\omega}_0,\ \mathbf{G}(0)=\mathbf{G}_0, \tag{2.17}$$

$$\mathbf{u}(\mathbf{x},0)=\tilde{\mathbf{u}}_0(\mathbf{x})=\begin{cases}\mathbf{u}_0(\mathbf{x}),\ \forall\mathbf{x}\in\Omega\backslash\overline{B(0)},\\ \mathbf{V}_0+\boldsymbol{\omega}_0\times\overrightarrow{\mathbf{G}_0\mathbf{x}},\ \forall\mathbf{x}\in\overline{B(0)},\end{cases} \tag{2.18}$$

with the following functional spaces

$$\mathbf{W}_{\mathbf{g}_0}(t)=\{\mathbf{v}|\mathbf{v}\in(H^1(\Omega))^3,\ \mathbf{v}=\mathbf{g}_0(t)\ on\ \Gamma\},$$
$$L^2_0(\Omega)=\{q|q\in L^2(\Omega),\ \int_\Omega q\,d\mathbf{x}=0\},\quad \Lambda(t)=(H^1(B(t)))^3.$$

In (2.12) – (2.18), only the center of mass, the translation velocity of the center of mass and the angular velocity of the particle are considered. Knowing these two velocities and the center of mass of the particle, one is able to translate and rotate the particle in space by tracking two extra points $\mathbf{x}_1$ and $\mathbf{x}_2$ in each particle, which follow the rigid body motion

$$\frac{d\mathbf{x}_i}{dt}=\mathbf{V}(t)+\boldsymbol{\omega}(t)\times\overrightarrow{\mathbf{G}(t)\mathbf{x}_i},\quad \mathbf{x}_i(0)=\mathbf{x}_{i,0},\ i=1,2. \tag{2.19}$$

In practice we shall track two orthogonal normalized vectors rigidly attached to the body B and originating from the center of mass $\mathbf{G}$.

3. Time and space discretization. For simplicity, we assume that $\Omega\subset\mathbb{R}^3$ is a rectangular parallelepiped. Concerning the *space approximation* of problem (2.12)–(2.19) by a *finite element method*, we have

$$\mathbf{W}_h=\{\mathbf{v}_h|\mathbf{v}_h\in(C^0(\overline{\Omega}))^3,\ \mathbf{v}_h|_T\in(P_1)^3,\ \forall T\in\mathcal{T}_h\}, \tag{3.1}$$

$$\mathbf{W}_{0h}=\{\mathbf{v}_h|\mathbf{v}_h\in\mathbf{W}_h,\ \mathbf{v}_h=\mathbf{0}\ on\ \Gamma\}, \tag{3.2}$$

$$L^2_h=\{q_h|q_h\in C^0(\overline{\Omega}),\ q_h|_T\in P_1,\ \forall T\in\mathcal{T}_{2h}\},\quad L^2_{0h}=\{q_h|q_h\in L^2_h,\ \int_\Omega q_h\,d\mathbf{x}=0\} \tag{3.3}$$

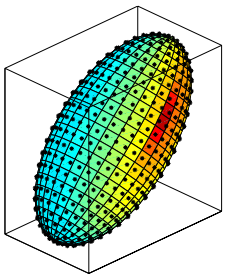


Figure 3.1: An example of grids covering the surface of $B(t)$

where $\mathcal{T}_h$ is a tetrahedrization of Ω, $\mathcal{T}_{2h}$ is twice coarser than $\mathcal{T}_h$, and P_1 is the space of the polynomials in three variables of degree ≤ 1. A finite dimensional space approximating $\Lambda(t)$ is as follows: let $\{\boldsymbol{\xi}_i\}_{i=1}^N$ be a set of points from $\overline{B(t)}$ which cover $\overline{B(t)}$ (uniformly, for example); we define then

$$\Lambda_h(t) = \{\boldsymbol{\mu}_h | \boldsymbol{\mu}_h = \sum\nolimits_{i=1}^N \boldsymbol{\mu}_i \delta(\mathbf{x} - \boldsymbol{\xi}_i),\ \boldsymbol{\mu}_i \in \mathbb{R}^3,\ \forall i = 1, ..., N\}, \tag{3.4}$$

where $\delta(\cdot)$ is the Dirac measure at $\mathbf{x} = \mathbf{0}$. Then we shall use $< \cdot, \cdot >_h$ defined by

$$< \boldsymbol{\mu}_h, \mathbf{v}_h >_h = \sum\nolimits_{i=1}^N \boldsymbol{\mu}_i \cdot \mathbf{v}_h(\boldsymbol{\xi}_i),\ \forall \boldsymbol{\mu}_h \in \Lambda_h(t),\ \mathbf{v}_h \in \mathbf{W}_h. \tag{3.5}$$

A typical choice of points for defining (3.4) is a collection of grid points for velocity field covered by the interior of the particle $B(t)$ and selected points from the surface of $B(t)$. An example of choice of surface points is shown in Figure 3.1

Using the above finite dimensional spaces leads to the following approximation for problem (2.12)–(2.19):

For a.e. $t > 0$, *find* $\mathbf{u}(t) \in \mathbf{W}_h(t)$, $p_h(t) \in L^2_{0h}(\Omega)$, $\mathbf{V}(t) \in \mathbb{R}^3$, $\mathbf{G}(t) \in \mathbb{R}^3$, $\boldsymbol{\omega}(t) \in \mathbb{R}^3$, $\boldsymbol{\lambda}_h(t) \in \Lambda_h(t)$ *such that*

$$\begin{cases} \rho_f \displaystyle\int_\Omega \left[\frac{\partial \mathbf{u}_h}{\partial t} + (\mathbf{u}_h \cdot \boldsymbol{\nabla})\mathbf{u}_h\right] \cdot \mathbf{v} d\mathbf{x} - \int_\Omega p_h \boldsymbol{\nabla} \cdot \mathbf{v} d\mathbf{x} + \nu_f \int_\Omega \boldsymbol{\nabla}\mathbf{u}_h : \boldsymbol{\nabla}\mathbf{v} d\mathbf{x} \\ \quad +(1 - \frac{\rho_f}{\rho_s})[M_p \frac{d\mathbf{V}}{dt} \cdot \mathbf{Y} + \frac{d(\mathbf{I}_p \boldsymbol{\omega})}{dt} \cdot \boldsymbol{\theta}] - \mathbf{F}^r \cdot \mathbf{Y} - \overrightarrow{\mathbf{G}\mathbf{x}_r} \times \mathbf{F}^r \cdot \boldsymbol{\theta} \\ =< \boldsymbol{\lambda}_h, \mathbf{v} - \mathbf{Y} - \boldsymbol{\theta} \times \overrightarrow{\mathbf{G}\mathbf{x}} >_h + (1 - \frac{\rho_f}{\rho_s}) M_p \mathbf{g} \cdot \mathbf{Y} + \rho_f \displaystyle\int_\Omega \mathbf{g} \cdot \mathbf{v} d\mathbf{x}, \\ \forall \mathbf{v} \in \mathbf{W}_{0h},\ \forall \mathbf{Y} \in \mathbb{R}^3,\ \forall \boldsymbol{\theta} \in \mathbb{R}^3, \end{cases} \tag{3.6}$$

$$\int_\Omega q \boldsymbol{\nabla} \cdot \mathbf{u}_h(t) d\mathbf{x} = 0,\ \forall q \in L^2_h, \tag{3.7}$$

$$\mathbf{u}_h = \mathbf{g}_{0h} \text{ on } \Gamma, \tag{3.8}$$

$$\frac{d\mathbf{G}}{dt} = \mathbf{V}, \tag{3.9}$$

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{V}(t) + \boldsymbol{\omega}(t) \times \overrightarrow{\mathbf{G}(t)\mathbf{x}_i},\quad \mathbf{x}_i(0) = \mathbf{x}_{i,0},\ i = 1, 2, \tag{3.10}$$

$$< \boldsymbol{\mu}, \mathbf{u}_h(t) - \mathbf{V}(t) - \boldsymbol{\omega}(t) \times \overrightarrow{\mathbf{G}(t)\mathbf{x}} >_h = 0,\ \forall \boldsymbol{\mu} \in \Lambda_h(t), \tag{3.11}$$

$$\mathbf{V}(0) = \mathbf{V}_0,\ \boldsymbol{\omega}(0) = \boldsymbol{\omega}_0,\ \mathbf{G}(0) = \mathbf{G}_0, \tag{3.12}$$

$$\mathbf{u}(\mathbf{x}, 0) = \tilde{\mathbf{u}}_{0h}(\mathbf{x}). \tag{3.13}$$

In (3.8), $\mathbf{g}_{0h}$ is an approximation of $\mathbf{g}_0$ belonging to $\gamma\mathbf{W}_h = \{\mathbf{z}_h|\mathbf{z}_h \in (C^0(\Gamma))^3,\ \mathbf{z}_h = \tilde{\mathbf{z}}_h|_\Gamma \ with\ \tilde{\mathbf{z}}_h \in \mathbf{W}_h\}$ and verifying $\int_\Gamma \mathbf{g}_{0h}\cdot\mathbf{n}d\Gamma = 0$.

3.1. An operator-splitting scheme à la Marchuk-Yanenko. Many operator-splitting schemes can be used to time-discretize (3.6)–(3.13). One of the advantage of operator-splitting schemes is that we can decouple difficulties like (i) the incompressibility condition, (ii) the nonlinear advection term, and (iii) a rigid-body-motion projection, so that each one of them can be handled separately, and in principle optimally. Let $\triangle t$ be a time discretization step and $t^{n+s} = (n+s)\triangle t$. By an operator-splitting scheme à la Marchuk–Yanenko as in [16], we have the following scheme after dropping some of the subscripts h (similar ones are discussed in [6, 8, 7, 9, 10]):

$$\mathbf{u}^0 = \tilde{\mathbf{u}}_0,\ \mathbf{G}^0 = \mathbf{G}_0,\ \mathbf{V}^0 = \mathbf{V}_0,\ \boldsymbol{\omega}^0 = \boldsymbol{\omega}_0,\ \mathbf{x}_1^0 = \mathbf{x}_{1,0},\ \mathbf{x}_2^0 = \mathbf{x}_{2,0}\ \ given; \tag{3.14}$$

for $n \geq 0, \mathbf{u}^n(\simeq \mathbf{u}(t^n))$, $\mathbf{G}^n$, $\mathbf{V}^n$, $\boldsymbol{\omega}^n$, $\mathbf{x}_1^n$ and $\mathbf{x}_2^n$ *being known, we compute* $\mathbf{u}^{n+1/5}$, $p^{n+1/5}$ *via the solution of*

$$\begin{cases} \rho_f\displaystyle\int_\Omega \frac{\mathbf{u}^{n+1/5}-\mathbf{u}^n}{\triangle t}\cdot\mathbf{v}d\mathbf{x} - \int_\Omega p^{n+1/5}\boldsymbol{\nabla}\cdot\mathbf{v}d\mathbf{x} = 0,\ \forall\mathbf{v}\in\mathbf{W}_{0h}, \\ \displaystyle\int_\Omega q\boldsymbol{\nabla}\cdot\mathbf{u}^{n+1/5}d\mathbf{x} = 0,\ \forall q\in L_h^2, \\ \mathbf{u}^{n+1/5}\in\mathbf{W}_h,\ \mathbf{u}^{n+1/5} = \mathbf{g}_{0h}^{n+1}\ on\ \Gamma,\ p^{n+1/5}\in L_{0h}^2. \end{cases} \tag{3.15}$$

Next, compute $\mathbf{u}^{n+2/5}$ *via the solution of*

$$\begin{cases} \displaystyle\int_\Omega \frac{\partial\mathbf{u}}{\partial t}\cdot\mathbf{v}d\mathbf{x} + \int_\Omega(\mathbf{u}^{n+1/5}\cdot\boldsymbol{\nabla})\mathbf{u}\cdot\mathbf{v}d\mathbf{x} = 0, \\ \forall\mathbf{v}\in\mathbf{W}_{0h}^{n+1,-},\ a.e.\ on\ (t^n,\ t^{n+1}), \\ \mathbf{u}(t^n) = \mathbf{u}^{n+1/5}, \\ \mathbf{u}(t)\in\mathbf{W}_h,\ \mathbf{u}(t) = \mathbf{g}_{0h}^{n+1}\ on\ \Gamma_-^{n+1}\times(t^n,\ t^{n+1}), \end{cases} \tag{3.16}$$

and set $\mathbf{u}^{n+2/5} = \mathbf{u}(t^{n+1})$.

Then, compute $\mathbf{u}^{n+3/5}$ *via the solution of*

$$\begin{cases} \rho_f\displaystyle\int_\Omega \frac{\mathbf{u}^{n+3/5}-\mathbf{u}^{n+2/5}}{\triangle t}\cdot\mathbf{v}d\mathbf{x} + \alpha\nu_f\int_\Omega \boldsymbol{\nabla}\mathbf{u}^{n+3/5}:\boldsymbol{\nabla}\mathbf{v}d\mathbf{x} = \rho_f\int_\Omega \mathbf{g}\cdot\mathbf{v}d\mathbf{x}, \\ \forall\mathbf{v}\in\mathbf{W}_{0h};\ \mathbf{u}^{n+3/5}\in\mathbf{W}_h,\ \mathbf{u}^{n+3/5} = \mathbf{g}_{0h}^{n+1}\ on\ \Gamma. \end{cases} \tag{3.17}$$

Now predict the motion of the center of mass and the angular velocity of the particle via

$$\frac{d\mathbf{G}}{dt} = \mathbf{V}(t), \tag{3.18}$$

$$\frac{d\mathbf{x}_i}{dt} = \mathbf{V}(t) + \boldsymbol{\omega}(t)\times\overrightarrow{\mathbf{G}(t)\mathbf{x}_i},\ for\ i = 1,2, \tag{3.19}$$

$$(1-\rho_f/\rho_s)M_p\frac{d\mathbf{V}}{dt} = (1-\rho_f/\rho_s)M_p\mathbf{g} + \mathbf{F}_r, \tag{3.20}$$

$$(1-\rho_f/\rho_s)\frac{d(\mathbf{I}_p\,\boldsymbol{\omega})}{dt} = \overrightarrow{\mathbf{G}\mathbf{x}_r}\times\mathbf{F}_r, \tag{3.21}$$

$$\mathbf{G}(t^n) = \mathbf{G}^n,\ \mathbf{V}(t^n) = \mathbf{V}^n,\ (\mathbf{I}_p\,\boldsymbol{\omega})^n = (\mathbf{I}_p\,\boldsymbol{\omega})(t^n), \tag{3.22}$$

$$\mathbf{x}_1(t^n) = \mathbf{x}_1^n,\ \mathbf{x}_2(t^n) = \mathbf{x}_2^n,$$

for $t^n < t < t^{n+1}$. *Then set* $\mathbf{G}^{n+4/5} = \mathbf{G}(t^{n+1})$, $\mathbf{V}^{n+4/5} = \mathbf{V}(t^{n+1})$, $(\mathbf{I}_p\,\boldsymbol{\omega})^{n+4/5} = (\mathbf{I}_p\,\boldsymbol{\omega})(t^{n+1})$, $\mathbf{x}_1^{n+4/5} = \mathbf{x}_1(t^{n+1})$, $\mathbf{x}_2^{n+4/5} = \mathbf{x}_2(t^{n+1})$, *and* $\mathbf{u}^{n+4/5} = \mathbf{u}^{n+3/5}$.
With the center $\mathbf{G}^{n+4/5}$, $\mathbf{x}_1^{n+4/5}$ *and* $\mathbf{x}_2^{n+4/5}$ *obtained at the above step, we enforce the rigid body motion in the region* $B(t^{n+4/5})$ *occupied by the particle*

$$\begin{cases} \rho_f \displaystyle\int_\Omega \frac{\mathbf{u}^{n+1}-\mathbf{u}^{n+4/5}}{\triangle t}\cdot\mathbf{v}\,d\mathbf{x} + \beta\nu_f\int_\Omega \boldsymbol{\nabla}\mathbf{u}^{n+1} : \boldsymbol{\nabla}\mathbf{v}\,d\mathbf{x} \\ +(1-\frac{\rho_f}{\rho_s})M_p\frac{\mathbf{V}^{n+1}-\mathbf{V}^{n+4/5}}{\triangle t}\cdot\mathbf{Y} + (1-\frac{\rho_f}{\rho_s})\frac{(\mathbf{I}_p\,\boldsymbol{\omega})^{n+1}-(\mathbf{I}_p\,\boldsymbol{\omega})^{n+4/5}}{\triangle t}\cdot\boldsymbol{\theta} \\ =< \boldsymbol{\lambda}^{n+4/5},\ \mathbf{v}-\mathbf{Y}-\boldsymbol{\theta}\times\overrightarrow{\mathbf{G}^{n+4/5}\mathbf{x}} >_h,\ \forall\mathbf{v}\in\mathbf{W}_{0h}, \mathbf{Y}\in\mathbb{R}^3,\ \boldsymbol{\theta}\in\mathbb{R}^3, \\ \mathbf{u}^{n+1}\in\mathbf{W}_h, \mathbf{u}^{n+1}=\mathbf{g}_{0h}^{n+1}\ on\ \Gamma,\ \boldsymbol{\lambda}^{n+4/5}\in\Lambda_h^{n+4/5}, \mathbf{V}^{n+1}\in\mathbb{R}^3, \boldsymbol{\omega}^{n+1}\in\mathbb{R}^3, \end{cases} \tag{3.23}$$

$$< \boldsymbol{\mu},\ \mathbf{u}^{n+1}-\mathbf{V}^{n+1}-\boldsymbol{\omega}^{n+1}\times\overrightarrow{\mathbf{G}^{n+4/5}\mathbf{x}} >_h = 0,\ \forall\boldsymbol{\mu}\in\Lambda_h^{n+4/5}. \tag{3.24}$$

In (3.14)–(3.24), $\Gamma_-^{n+1} = \{\mathbf{x}|\mathbf{x}\in\Gamma,\ \mathbf{g}_{0h}^{n+1}(\mathbf{x})\cdot\mathbf{n}(\mathbf{x}) < 0\}$ and $\mathbf{W}_{0h}^{n+1,-} = \{\mathbf{v}|\mathbf{v}\in\mathbf{W}_h,\ \mathbf{v} = \mathbf{0}\ on\ \Gamma_-^{n+1}\}$, $\Lambda_h^{n+s} = \Lambda_h(t^{n+s})$, and $\alpha+\beta = 1$. In the numerical simulation, we usually choose $\alpha = 1$ and $\beta = 0$.

3.2. On the solution of subproblems (3.15), (3.16), (3.17), (3.18)-(3.22), and (3.23)-(3.24). The degenerated quasi-Stokes problem (3.15) is solved by an Uzawa preconditioned conjugate gradient algorithm as in [11], where the discrete elliptic problems from the preconditioning are solved by a matrix-free fast solver from FISHPAK due to Adams et al. in [1]. The advection problem (3.16) for the velocity field is solved by a wave-like equation method as in [4, 5]. Problem (3.17) is a classical discrete elliptic problem which can be solved by the same matrix-free fast solver.

System (3.18)-(3.22) is a system of ordinary differential equations thanks to operator splitting. For its solution one can choose a time step smaller than $\triangle t$, (i.e., we can divide $\triangle t$ into smaller steps) to predict the translation velocity of the center of mass, the angular velocity of the particle, the position of the center of mass and the regions occupied by each particle so that the repulsion forces can be effective to prevent particle-particle and particle-wall overlapping. At each subcycling time step, keeping the distance as constant between the pair of points $\mathbf{x}_1$ and $\mathbf{x}_2$ in each particle is important since we are dealing with rigid particles. We have applied the following approach to satisfy the above constraint:

- Translate $\mathbf{x}_1$ and $\mathbf{x}_2$ according to the new position of the mass center at each subcycling time step.
- Rotate $\mathbf{Gx}_1$ and $\mathbf{Gx}_2$, the relative positions of $\mathbf{x}_1$ and $\mathbf{x}_2$ to the center of mass $\mathbf{G}$, by the following Crank-Nicolson scheme (a Runge-Kutta scheme of order 2, in fact):

$$\frac{\mathbf{Gx}_i^{new}-\mathbf{Gx}_i^{old}}{\tau} = \boldsymbol{\omega}\times\frac{\mathbf{Gx}_i^{new}+\mathbf{Gx}_i^{old}}{2} \tag{3.25}$$

  for $i = 1, 2$ with τ as a subcycling time step. By (3.25), we have $|\mathbf{Gx}_i^{new}|^2 = |\mathbf{Gx}_i^{old}|^2$ for $i = 1, 2$ and $|\mathbf{Gx}_2^{new}-\mathbf{Gx}_1^{new}|^2 = |\mathbf{Gx}_2^{old}-\mathbf{Gx}_1^{old}|^2$ (i.e., scheme (3.25) is distance and in fact shape preserving).

Remark 3.1 *In order to activate the short range repulsion force, we have to find the shortest distance between two ellipsoids. Unlike the cases for spheres, it is not trivial to locate the point from each surface of the ellipsoid where the distance is the shortest between two ellipsoids. There is no explicit formula for such distance. In practice, we first choose a set of points from the surface of each ellipsoid. Then we find the point among the chosen points from each*

surface at which the distance is the shortest We repeat this (kind of relaxation) process in the neighborhood of the newly located point on each surface of ellipsoid until convergence, usually obtained in very few iterations.

For the shortest distance between the wall and ellipsoid, there exists an explicit formula. To check whether two ellipsoids overlap each other, there exists an algorithm used by people working on computer graphics and in robotics (e.g., see, [20]).

After solving (3.18)-(3.22), the rigid body motion is enforced in $B(t^{n+4/5})$, via equation (3.24). At the same time those hydrodynamical forces acting on the particles are also taken into account in order to update the translation and angular velocities of the particles. To solve (3.23)-(3.24), we use a conjugate gradient algorithm as discussed in [7]. Since we take $\beta = 0$ in (3.23) for the simulation, we actually do not need to solve any non-trivial linear systems for the velocity field; this saves a lot of computing time. To get the angular velocity $\boldsymbol{\omega}^{n+1}$, computed via

$$\boldsymbol{\omega}^{n+1} = (\mathbf{I}_p^{n+4/5})^{-1}(\mathbf{I}_p\,\boldsymbol{\omega})^{n+1}, \tag{3.26}$$

we need to have $\mathbf{I}_p^{n+4/5}$, the inertia of the particle $B(t^{n+4/5})$. We first compute the inertia $\mathbf{I}_0$ in the coordinate system attached to the particle. Then via the center of mass $\mathbf{G}^{n+4/5}$ and points $\mathbf{x}_1^{n+4/5}$ and $\mathbf{x}_2^{n+4/5}$, we have the rotation transformation $\mathbf{Q}$ ($\mathbf{Q}\mathbf{Q}^T = \mathbf{Q}^T\mathbf{Q} = \mathbf{I}_d$, det$\mathbf{Q}$=1) which transforms vectors expressed in the particle frame to vectors in the flow domain coordinate system and $\mathbf{I}_p^{n+s} = \mathbf{Q}\mathbf{I}_0\mathbf{Q}^T$. Actually in order to update matrix $\mathbf{Q}$ we can also use *quaternion* techniques, as shown, in the review paper [3].

4. Numerical experiments.

4.1. One settling ellipsoid. The orientation of symmetric long body (loosely, a long body is a body where one dimension is much prevailing upon the other two) in liquids of different nature is a fundamental issue in many problems of practical interest (see [15], and references therein). In the first test case, we consider the simulation of the motion of a settling ellipsoid in a narrow channel of infinite length filled with a Newtonian fluid. The computational domain is $\Omega = (0,1) \times (0,0.25) \times (0,4)$ initially, then it moves down with the center of the ellipsoid (see, e.g., [13] for adjusting the computational domain according to the position of the particle). The fluid density is $\rho_f = 1$ and the fluid viscosity is $\nu_f = 0.01$. The flow field initial condition is $\mathbf{u} = \mathbf{0}$. The three semi-axes of the ellipsoid are 0.2, 0.1 and 0.1. The initial velocity and angular velocity of the ellipsoid are $\mathbf{0}$. The density of the ellipsoid is $\rho_s = 1.25$. Its vertical axis is the longest semi-axis (see Figure 4.1). The mesh size for the velocity field (resp., pressure) is $h_v = 1/80$ (resp., $h_p = 2h_v$). The time step is $\triangle t = 0.001$. The positions of the ellipsoid at different times in the channel are shown in Figure 4.1. (The computation was performed in a moving frame of reference, so the ellipsoid appears not moving downward.) The motion of the ellipsoid is very violent at the beginning, it moves very close to the side wall after release from its initial position. Later on the motion becomes periodic (see Figures 4.1 and 4.2). As expected, the ellipsoid turns its broadside to the stream while oscillating as shown in the last three snapshots of Figure 4.1. The averaged particle speed at the end of the simulation is about 4.256 so the particle Reynolds number with the long axis as characteristic length is 170.24.

4.2. Two ellipsoids sedimenting side-by-side. It had been observed experimentally by Joseph and Bai that when two ellipsoid-like long bodies sedimente side-by-side in a narrow channel filled with a Newtonian fluid, they interact each other periodically as shown in Figures 4.3. The particle Reynolds number is about 120. To reproduce this phenomenon, we consider the following test case. The computational domain is $\Omega = (0,1) \times (0,0.25) \times (0,4)$ initially, then it moves down with the lower center of two ellipsoids. The initial positions of the centers are $(0.22, 0.125, 0.75)$ and $(0.78, 0.125, 0.75)$, respectively. The frames rigidly

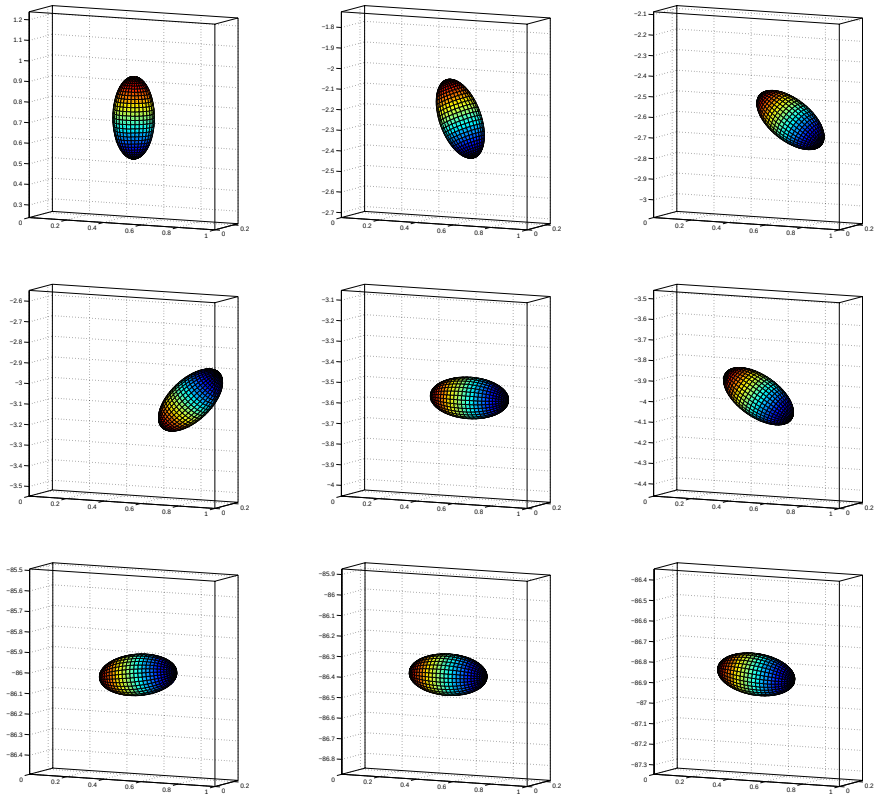


Figure 4.1: Position of the ellipsoid at $t = 0$, 0.41, 0.46, 0.56, 0.66, 0.75, 18.1, 18.18, and 18.28 (from left to right and from top to bottom).

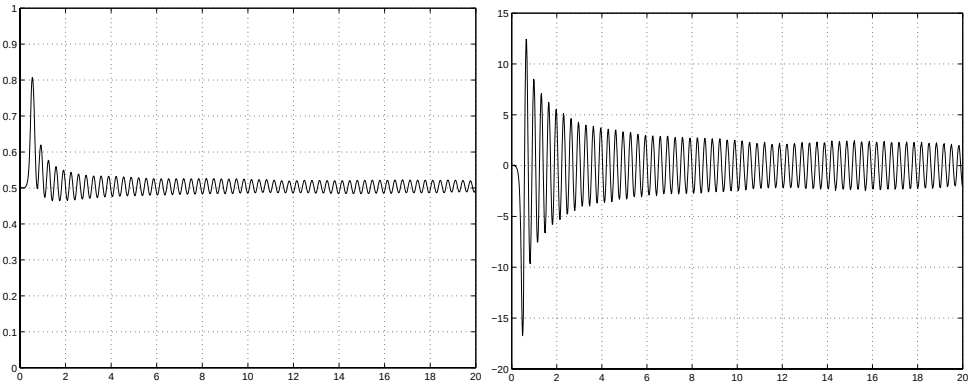


Figure 4.2: Histories of the x-coordinate of the center (left) and the y-component of the angular velocity of the ellipsoid (right) .

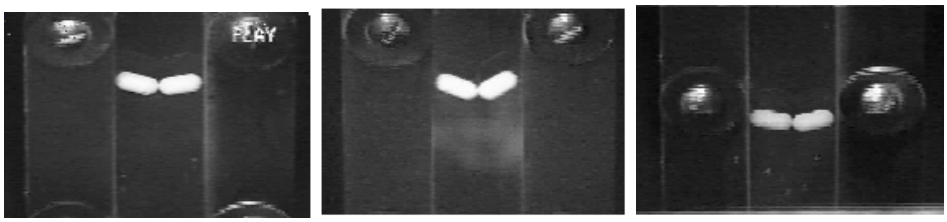


Figure 4.3: Snapshots of a period of the motion of two ellipsoid-like long bodies sedimenting in a narrow channel filled with a Newtonian fluid.

attached to the ellipsoids initially are $\{(\cos\pi/3, 0, \sin\pi/3),\ (0,1,0),\ (\cos 5\pi/6, 0, \sin 5\pi/6)\}$ and $\{(\cos(-\pi/3), 0, \sin(-\pi/3)),\ (0,1,0),\ (\cos\pi/6, 0, \sin\pi/6)\}$, respectively (see Figure 4.4). All others parameters are as in the previous case. Averaged terminal speed is about 2.497 obtained from last 300 time steps, so the averaged particle Reynolds number is 99.88 based on the length of the long axis (which is 0.4). In the simulation, we obtained result as seen in Figure 4.4 similar to the one in Figure 4.3 (the computation was performed in a moving frame of reference, so the ellipsoids appear not moving downward), which is in good agreement with experimental results qualitatively. In Figure 4.5, we can see very strong interaction between two ellipsoids of long axes 0.4. We also have tested the case with two ellipsoids of long axes 0.36 and found that they settle in the channel with very weak interaction between each other (see Figure 4.5).

Acknowledgments We acknowledge the helpful comments and suggestions of G.P. Galdi, J. He, and V.I. Paulsen. We acknowledge also the support of NSF (grants DMS-9973318, and CCR-9902035), Texas Board of Higher Education (ARP grant 003652-0383-1999), and DOE/LASCI (grant R71700K-292-000-99).

REFERENCES

[1] J. Adams, P. Swarztrauber, and R. Sweet. *FISHPAK: A package of Fortran subprograms for the solution of separable elliptic partial differential equations*. The National Center for Atmospheric Research, Boulder, CO, 1980.

[2] F. Baaijens. A fictitious domain/mortar element method for fluid-structure interaction. *Int. J. Numer. Meth. Fluids*, 35:743–761, 2001.

[3] J. C. K. Chou. Quaternion kinematic and dynamic differential equations. *IEEE transaction on robotics and automation*, 8:53–64, 1992.

[4] E. J. Dean and R. Glowinski. A wave equation approach to the numerical solution of the Navier-Stokes equations for incompressible viscous flow. *C.R. Acad. Sc. Paris*, 325(Serie 1):783–791, 1997.

[5] E. J. Dean, R. Glowinski, and T.-W. Pan. A wave equation approach to the numerical simulation of incompressible viscous fluid flow modeled by the Navier-Stokes equations. In J. D. Santo, editor, *Mathematical and Numerical Aspects of Wave Propagation*, pages 65–74, Philadelphia, PA, 1998. SIAM.

[6] R. Glowinski, T. I. Hesla, D. D. Joseph, T. W. Pan, and J. Periaux. Distributed Lagrange multiplier methods for particulate flows. In M. Bristeau, G. Etgen, W. Fitzgibbon, J. Lions, J. Periaux, and M. Wheeler, editors, *Computational Science for the 21st Century*, pages 270–279, Chichester, 1997. Wiley.

[7] R. Glowinski, T. Pan, T. Hesla, and D. Joseph. A distributed Lagrange multiplier/fictitious domain method for particulate flow. *International Journal of Multiphase Flow*, 25:755–794, 1999.

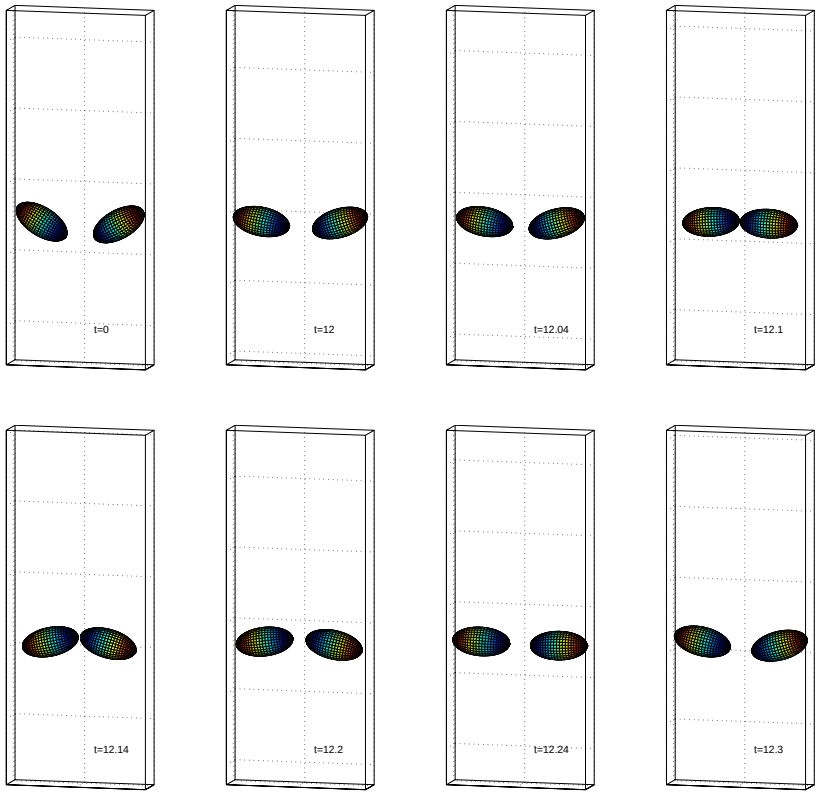


Figure 4.4: Position of ellipsoids at $t = 0$, 12, 12.04, 12.1, 12.14 12.2, 12.24, and 12.3 (from left to right and from top to bottom).

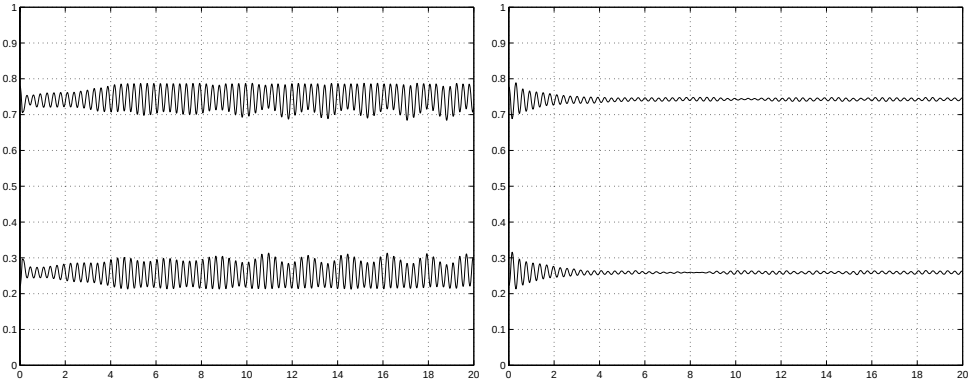


Figure 4.5: Histories of the x-coordinates of the centers of two ellipsoids of long axes 0.4 (left) and the x-coordinates of the centers of two ellipsoids of long axes 0.36 (right).

[8] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux. A fictitious domain method with distributed Lagrange multipliers for the numerical simulation of particulate flow. In J. Mandel, C. Farhat, , and X. Cai, editors, *Domain Decomposition Methods 10*, pages 121–137, Providence, RI, 1998. AMS.

[9] R. Glowinski, T. W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux. A distributed Lagrange multiplier/fictitious domain method for flows around moving rigid bodies: Application to particulate flow. *Int. J. Numer. Meth. Fluids*, 30:1043–1066, 1999.

[10] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *J. Comput. Phys.*, 169:363–426, 2001.

[11] R. Glowinski, T.-W. Pan, and J. Periaux. Distributed Lagrange multiplier methods for incompressible flow around moving rigid bodies. *Comput. Methods Appl. Mech. Engrg.*, 151:181–194, 1998.

[12] H. H. Hu. Direct simulation of flows of solid-liquid mixtures. *Int. J. Multiphase Flow*, 22:335–352, 1996.

[13] H. H. Hu, D. D. Joseph, and M. Crochet. Direct simulation of fluid particle motions. *Theoret. Comput. Fluid Dynamics*, 3:285–306, 1992.

[14] A. Johnson and T. Tezduyar. 3-d simulation of fluid-rigid body interactions with the number of rigid bodies reaching 100. *Comp. Meth. Appl. Mech. Eng.*, 145:301–321, 1997.

[15] Y. J. L. D. D. Joseph. Sedimentation of particles in polymer solutions. *J. Fluid Mech.*, 255:565–595, 1993.

[16] G. I. Marchuk. *Handbook of Numerical Analysis, Splitting and alternating direction methods*, volume I. Elsevier Science Publishers B.V., North-Holland, Amsterdam, 1990.

[17] B. Maury and R. Glowinski. Fluid-particle flow: a symmetric formulation. *C.R. Acad. Sc. Paris*, 324(Serie 1):1079–1084, 1997.

[18] C. S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220–252, 1977.

[19] C. S. Peskin and D. M. McQueen. Modeling prosthetic heart valves for numerical analysis of blood flow in the heart. *J. Comp. Phys.*, 37:113–132, 1980.

[20] E. Rimon and S. Boyd. Efficient distance computation using best ellipsoid fit. In *The IEEE International symposium on intelligent control, Glasgow, UK*, pages 360–365. IEEE, 1992.

[21] G. J. Wagner, N. Moes, W. K. Liu, and T. Belytschko. The extended finite element method for rigid particles in Stokes flow. *Int. J. Numer. Meth. Engng.*, 51:293–313, 2001.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

11. Domain Decomposition by Stochastic Methods

Éric PEIRANO and Denis TALAY [1]

1. Introduction: Monte Carlo methods for domain decomposition. As shown by P-L. Lions in [19], stochastic representations of solutions of linear and nonlinear partial differential equations are useful to analyze the convergence of the Schwarz alternating method and related decomposition methods. The aim of this note is to show that one can also deduce simulation algorithms from stochastic representations in view of decomposing domains. To summarize, the Monte Carlo method allows one to compute approximate Dirichlet conditions on the boundaries of the subdomains of the decomposition without approximating the solution in the whole domain. One can thus easily localize problems in bounded domains, or compute the solution outside sub–domains where the solution has strong variations or the viscosity is small or the coefficients are discontinuous, etc. An advantage of the Monte Carlo method is that it is extremely easy to program and the simulations can be made in parallel.

We start our discussion by recalling the use of a Monte Carlo method to approximate π. The method is a variant of Buffon's needle method. Draw a vertical line in the 2-D space. Attach the extremity of a needle with one unit length to the line and choose the angle θ between the needle and an horizontal line at random according to the uniform distribution on $\left[0, \frac{\pi}{2}\right]$.

One has

$$\mathbb{E}\,\cos(\theta) = \frac{2}{\pi}\int_0^{\frac{\pi}{2}} \cos(z)\; dz = \frac{2}{\pi}.$$

The Strong Law of Large Numbers implies that

$$\frac{\cos(\theta_1) + \ldots + \cos(\theta_N)}{N} \xrightarrow[N\to+\infty]{} \frac{2}{\pi} \text{ a.e.},$$

where the θ_k's are independent copies of θ, that is, they are independent random variables and have the same probability distribution as θ (in practice one throws the needle at random N times, or simulates that game on a computer by using random number generators).

Observe that, in order to construct our Monte Carlo method, we have written the quantity under consideration, namely $\frac{2}{\pi}$, as the expectation of a random variable whose law is explicitly known and easy to simulate. To solve partial differential equations the situation is usually much more complex. We start by considering a case where the probabilistic representation is simple, namely the heat equation in the whole space: consider a bounded continuous function f and

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) = \dfrac{1}{2}\Delta u(t,x),\ 0 < t,\ x \in \mathbb{R}^d, \\ u(0,x) = f(x). \end{cases}$$

Thus

$$u(t,x) = \mathcal{G}_t * f(x),$$

where $\mathcal{G}_t$ is the heat kernel

$$\mathcal{G}_t(y) := \frac{1}{(2\pi)^{d/2}}\ \exp\left(-\frac{|y|^2}{2t}\right).$$

Observe that $\mathcal{G}_t$ is the density function of the random vector $\sqrt{t}\,G$ where G is a d dimensional Gaussian vector with zero mean and unit variance. We therefore have

$$u(t,x) = \mathbb{E}\, f(x + \sqrt{t}\ G). \tag{1.1}$$

[1]INRIA, 2004 Route des Lucioles, B.P. 93, 06902 Sophia-Antipolis (France)

In view of the Strong Law of Large Numbers it comes

$$u(t,x) \simeq \frac{1}{N}\sum_{k=1}^{N} f(x+\sqrt{t}\ G_k),$$

where the G_k's are independent copies of G. The error corresponding to N trials is

$$u(t,x) - \frac{1}{N}\sum_{k=1}^{N} f(x+\sqrt{t}\ G_k),$$

and therefore is random. It may be large but with small probabilities only when N is large, as shown by the Central Limit Theorem or more precise results such as the Berry–Esseen theorem:

Theorem 1.1 (Berry–Esseen) *Let $(X_k)_{k\geq 1}$ be a sequence of independent and identically distributed random variables with zero mean. Denote by σ the common standard deviation. Suppose that*

$$\mathbb{E}\,|X_1|^3 < +\infty.$$

Then

$$\begin{aligned}\epsilon_N &:= \sup_{x\in\mathbb{R}}\left|\mathbb{P}\left(\frac{X_1+\cdots+X_N}{\sigma\sqrt{N}}\leq x\right) - \int_{-\infty}^{x} e^{-u^2/2}\frac{du}{\sqrt{2\pi}}\right| \\ &\leq \frac{C\ \mathbb{E}\,|X_1|^3}{\sigma^3\sqrt{N}}.\end{aligned}$$

In addition, one has $0.398 \leq C \leq 0.8$.

For a proof see, e.g., Shiryayev [24]. Using the preceding theorem one can easily estimate the minimal number N of simulations which allows one to get a prescribed accuracy ϵ with a probability larger than a prescribed confidence threshold $1-\delta$.

In order to approximate the solution of a general parabolic or elliptic equation by a Monte Carlo method we have to extend the probabilistic representation (1.1) to cases where the partial differential equation involves a differential operator different from the Laplace operator. Then the fundamental solution cannot be related to the law of random variables so simple as Gaussian laws, and we need to consider the class of stochastic processes which are solutions to stochastic differential equations. We now shortly introduce that difficult and widely studied subject.

2. Probabilistic representation of parabolic and elliptic equations. Let $b : \mathbb{R}^d \to \mathbb{R}^d$ and $\sigma_j : \mathbb{R}^d \to \mathbb{R}^d$ $(1 \leq j \leq r)$ be smooth vector fields. Denote by $\sigma(x)$ the matrix whose column vectors are the $\sigma_j(x)$'s. Consider the elliptic operator

$$L\psi(x) := \sum_{i=1}^{d} b^i(x)\ \partial_i\psi(x) + \frac{1}{2}\sum_{i,j=1}^{d} a^i_j(x)\ \partial_{ij}\psi(x),$$

where

$$a(x) := \sigma(x)\ \sigma(x)^t,$$

and the evolution problem

$$\begin{cases}\dfrac{\partial u}{\partial t}(t,x) &= Lu(t,x),\ t>0,\ x\in\mathbb{R}^d,\\ u(0,x) &= f(x),\ x\in\mathbb{R}^d.\end{cases} \tag{2.1}$$

Suppose that (2.1) a smooth solution with bounded derivatives on $[0,T]\times\mathbb{R}^d$. We aim to construct a probabilistic representation of that solution. To this end we introduce stochastic processes. A stochastic process is a family of random variables indexed by time. The time index may be (subsets of) $\mathbb{R}_+$ or $\mathbb{N}$.

Our basic stochastic process will be the one dimensional Brownian motion (W_t) which satisfies: for all integer $n>1$ and all times $0\le t_1<\ldots<t_n$, the random vector $(W_{t_1}-W_{t_0},,\ldots,W_{t_n}-W_{t_{n-1}})$ is Gaussian with zero mean and diagonal covariance matrix; the diagonal terms of the covariance matrix are

$$\mathbb{E}\,(W_{t_j}-W_{t_{j-1}})^2=t_j-t_{j-1}.$$

By definition a d dimensional Brownian motion is a process $(W_t^1,\ldots,W_t^d)$ whose components are independent one dimensional Brownian motions. Observe that we can rewrite (1.1) as

$$u(t,x)=\mathbb{E}\,f(W_t(x)),$$

where W is a d dimensional Brownian motion starting from x at time 0. When the differential operator L is not the Laplace operator, one needs to consider more complex processes, namely the solutions of stochastic differential equations. Unfortunately these objects cannot be rigorously introduced without the heavy machinery of stochastic calculus (see, e.g., the textbooks by Friedman [13] and Karatzas and Shreve [17]). To avoid too many complexities we limit ourselves to introduce discrete time processes which approximate the solutions of stochastic differential equations and, owing to elementary calculations, we establish their link with the smooth solutions of equations of the type (2.1). Let us thus onsider the Euler scheme defined as

$$\begin{cases} X_0^h(x) & =x,\\ X_{(p+1)h}^h(x) & =X_{ph}^h+b(X_{ph}^h(x))\,h+\sum_{j=1}^r\sigma_j(X_{ph}^h(x))\,\sqrt{h}G_{p+1}^j, \end{cases} \tag{2.2}$$

where $h:=\frac{T}{M}$ is a discretization step of the time interval $[0,T]$ and (G_p^j) is a family of real valued independent Gaussian random variables with zero mean and unit variance. As the function $u(t,x)$ is supposed smooth with bounded derivatives and as $u(0,x)=f(x)$ for all x a Taylor expansion leads to

$$\begin{aligned} \mathbb{E}\,f(X_T^h(x))-u(T,x) &= \sum_{p=0}^{M-1}\mathbb{E}\,\Big[u(T-(p+1)h,X_{(p+1)h}^h(x))-u(T-ph,X_{ph}^h(x))\Big]\\ &= \mathbb{E}\,\Big[u(T-(p+1)h,X_{ph}^h(x))-u(T-ph,X_{ph}^h(x))\Big]\\ &\quad+h\sum_{p=0}^{M-1}\mathbb{E}\,\Big[Lu(T-(p+1)h,X_{ph}^h(x))\Big]+\sum_{p=0}^{M-1}\mathcal{O}(h^2)\\ &= h\sum_{p=0}^{M-1}\mathbb{E}\,\Big[Lu(T-ph,X_{ph}^h(x))-\frac{\partial u}{\partial t}(T-ph,X_{ph}^h(x))\Big]\\ &\quad+\sum_{p=0}^{M-1}\mathcal{O}(h^2)\\ &= \sum_{p=0}^{M-1}\mathcal{O}(h^2)\\ &= \mathcal{O}(h). \end{aligned} \tag{2.3}$$

Thus

$$u(T,x)=\mathbb{E}\,f(X_T^h(x))+\mathcal{O}(h).$$

Remark 2.1 *The Euler scheme is easy to simulate since it requires Gaussian simulations only. Noticing that* $\sqrt{h}\ G_p^j$ *has the same Gaussian distribution function as* $W^j_{(p+1)h} - W^j_{ph}$*, one can think the Euler scheme as a time discretization of the stochastic differential equation*

$$X_t(x) = x + \int_0^t b(X_s(x))\ ds\ +\ \sum_{j=1}^{d} \int_0^t \sigma_j(X_s(x))\ dW_s^j,\ 0 \le t \le T, \tag{2.4}$$

where $\int_0^t \sigma_j(X_s(x))\ dW_s^j$ *denotes the 'stochastic integral of the process* $(\sigma_j(X_s(x)))$ *with respect to the Brownian motion* (W_s^j)*' whose construction requires long developments. Equation (2.4) is shown to have a unique solution (in the appropriate space of stochastic processes) when the vector fields* b *and* σ_j *are Lipschitz. One can prove that the exact probabilistic representation of (2.1) is*

$$u(T,x) = \mathbb{E}\, f(X_T(x)). \tag{2.5}$$

The key point of the proof is the Itô's formula which one needs to use instead of the above Taylor expansion: for all real valued function ϕ *of class* $\mathcal{C}^{1,2}([0,T] \times \mathbb{R}^d)$ *it holds that*

$$\phi(t, X_t(x)) = \phi(0,x) + \int_0^t L\phi(s, X_s(x))\ ds + \sum_{i=1}^{d}\sum_{j=1}^{r} \int_0^t \partial_i \phi(s, X_s)\ \sigma_j^i(s, X_s)\ dW_s^j.$$

Using that formula and deep notions of stochastic calculus, for a very large class of parabolic problems it can be shown that, if a smooth solution exists, then it verifies the equality (2.5). Stochastic calculus techniques may also be useful to prove the existence of smooth solutions: for example, when the coefficients b_i *and* σ_j^i *are smooth, then (2.4) defines a smooth stochastic flow of diffeomorphisms, so that the mapping* $x \mapsto X_t(x)$ *is almost surely differentiable; therefore, if the function* f *itself is smooth and its derivatives satisfy appropriate growth at infinity conditions, the mapping* $x \mapsto \mathbb{E}\, f(X_t(x))$ *also is differentiable: see, e.g., Kunita [18].*

From the preceding consideration one deduces that

$$u(T,x) \simeq \frac{1}{N} \sum_{k=1}^{N} \mathbb{E}\, f(X_T^{h,k}(x)), \tag{2.6}$$

where

$$(X_{ph}^{h,k}(x),\ p = 0, \dots, M,\ k = 0, \dots, N)$$

is a family of independent trajectories of the Euler scheme. Such trajectories can be simulated as follows: in view of (2.2), owing to $d \times M$ calls to the generator of Gaussian random variables one obtains $(X_{ph}^{h,1}(x),\ p = 0, \dots, M)$. Then time is reset to 0, and $d \times M$ new calls to the generator allow one to obtain the second trajectory, and so on. One finally computes the right hand side of (2.6) by averaging the end points of the N trajectories. Observe that the simulations can be done in parallel instead of sequentially if one can distribute the computations on a set of processors. The number of communications between the processors is extremely weak. In counterpart one has to ensure that the processors run independent sequences of random numbers, which may require a clever programming.

The global error of the Monte Carlo method (2.6) is

$$u(T,x) - \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\, f(X_T^{h,k}) = \underbrace{u(T,x) - \mathbb{E}\, f(X_T^h(x))}_{=:\epsilon_d(h)} + \underbrace{\mathbb{E}\, f(X_T^h(x)) - \frac{1}{N}\sum_{k=1}^{N} \mathbb{E}\, f(X_T^{h,k})}_{=:\epsilon_s(h,N)}.$$

The *discretization error* is described by the inequality (2.3) and even more accurate estimates. Indeed, under various sets of hypotheses including cases where f is supposed measurable only, one has (Talay and Tubaro [27], Bally and Talay [2])

$$e_d(h) = C_f(T,x)\ h + Q_h(f,T,x)\ h^2$$

and

$$|C_f(T,x)| + sup_h|Q_h(f,T,x)| \leq K(T)\|f\|_\infty \frac{1+\|x\|^Q}{T^q}$$

for some real number $C_f(T,x)$ which does not depend on the discretization step h. Thus Romberg extrapolation techniques are available: for example, simulations with the discretization steps h and $\frac{h}{2}$ lead to a second order accuracy in view of

$$u(T,x) - \Big(2\ \mathbb{E}\, f(X_T^{h/2}(x)) - \mathbb{E}\, f(X_T^h(x))\Big) = 2\ e_d(h/2) - e_d(h) = \mathcal{O}(h^2).$$

The *statistical error* $s(h,N)$ is described by the Berry–Esseen theorem 1.1 or its variants. Notice that (2.3) ensures that the standard deviation of $X_T^h(x)$ and $\mathbb{E}\,|X_T^h(x)|^3$ can be bounded from above by constants which do not depend on h, so that the time discretization step plays no role in the choice of the number of simulations corresponding to a desired accuracy and a prescribed confidence interval.

For parabolic and elliptic equations with Dirichlet boundary conditions (Neumann boundary conditions respectively) probabilistic interpretations, and therefore Monte Carlo methods, are also available (for various probabilistic interpretations we again refer to, e.g., the textbooks by Friedman [13] and Karatzas and Shreve [17]). We here give an example of a parabolic problem with Dirichlet boundary conditions.

Let D be a domain in $\mathbb{R}^d$, and consider

$$\begin{cases} \dfrac{\partial u}{\partial t}(t,x) & = Lu(t,x),\ t>0,\ x\in D, \\ u(0,x) & = f(x),\ x\in D, \\ u(t,x) & = g(x),\ t>0,\ x\in \partial D, \end{cases}$$

where $f(x) = g(x)$ on ∂D. Under various sets of hypotheses one has

$$u(T,x) = \mathbb{E}\,[f(X_T(x))\ \mathbb{I}_{T<\tau}] + \mathbb{E}\,[g(X_\tau(x))\ \mathbb{I}_{T\geq\tau}],$$

where τ is the 'first exit time of the domain', that is,

$$\tau := \inf\{0\leq t,\ X_t(x)\in\partial D\}. \tag{2.7}$$

In view of that formula it is natural to numerically approximate $u(T,x)$ by using the Euler scheme stopped at its 'first exit time of the domain' τ^h, that is,

$$\tau^h := \inf\{0\leq p\leq M,\ X_{ph}^h\in\partial D\}\times h. \tag{2.8}$$

If the boundary condition were of Neumann type then the solution $u(T,x)$ would have been expressed in terms of a process whose trajectories are reflected at the boundary of the domain and the Monte Carlo method would have involved the 'reflected Euler scheme'. For the error analysis of the stopped or reflected Euler scheme we refer to Gobet [14] and [15], Costantini, Pacchiarotti and Sartoretto [9]. It is worthy to notice that Gobet shows that, to preserve a rate of convergence of order $\mathcal{O}(h)$ one has to define the first exit time of the domain of the Euler scheme in a more clever way than (2.8), and to add the simulation of random times to the preceding algorithm; that additional simulation often has a low cost.

3. Application to Domain Decomposition. In view of the preceding stochastic representations it seems interesting to study the following domain decomposition technique: localize the problem by building artificial boundaries, and compute the solution along these boundaries by Monte Carlo simulations. More precisely, given points x_i on the artificial boundaries, one simulate N independent trajectories issued from each x_i and average the values of the $X_T^{h,k}(x_i)$'s. If the original problem is posed in a bounded domain with Dirichlet (Neumann respectively) boundary conditions, then the simulation needs to involve the stopped (reflected respectively) Euler scheme.

An important issue consists in estimating the error induced by the 'stochastic approximations along the artificial boundaries of the decomposed domain'. At the time being this question is widely open. For parabolic problems corresponding to European options, Crépey [10] has done a pionneering work. Berthelot [4] is studying the case of the variational inequalities corresponding to American options. To our knowledge no precise result is available on the convergence rate of the global error corresponding to the combination of the Monte Carlo method (along the artificial boundaries) and a classical deterministic method for the numerical resolution of the original problem in each one of the sub–domains with approximate Dirichlet conditions obtained from simulation. The Omega research group at Inria Sophia Antipolis has obtained only very preliminary results in that direction.

4. Stochastic particle methods for nonlinear equations. Stochastic numerical methods have been developed for nonlinear equations. The structure of such methods is much more complex than for linear problems: for variational inequalities (particularly those which describe American option prices in finance) one has to consider backward stochastic differential equations (see, e.g., the review papers by El Karoui, Quenez and Pardoux [11], Pardoux [21]); for Burgers equation, McKean–Vlasov–Fokker–Planck and Boltzman equations one has to consider interacting stochastic particle systems and their limits in the 'propagation of chaos' sense (see, e.g., Sznitman [25], Bossy and Talay [7], Jourdain [16], Méléard [20], Fournier and Méléard [12]). For estimates on the numerical methods deduced from such stochastic representations, see, e.g., Chevance [8], Bally and Pagès [1] for backward stochastic differential equations; Bossy [5], Bossy and Jourdain [6], Talay [26] for interacting stochastic particle methods (the reference [6] being a pionneering work in the analysis of the convergence rate of stochastic particle methods for problems with boundary conditions).

We now give the example of an extension of the Sherman and Peskin [23] method for convection-reaction–diffusion equations

$$\left.\begin{array}{r} \dfrac{\partial V}{\partial t}(t,x) = LV(t,x) + f \circ V(t,x), \\ V(0,x) = V_0(x), \end{array}\right\} \tag{4.1}$$

where L is defined as

$$L\psi(x) := b(x)\ \psi'(x) + \frac{1}{2}\sigma(x)^2\ \psi''(x),$$

V_0 is a distribution function, and f is a smooth function such that $V(t,\cdot)$ is a distribution function for all $t > 0$. Set $u(t,x) := \dfrac{\partial V}{\partial x}(t,x)$. It solves

$$\left.\begin{array}{r} \dfrac{\partial u}{\partial t}(t,x) = \dfrac{1}{2}\,\sigma^2(x)\,\dfrac{\partial^2 u}{\partial x^2}(t,x) + [b(x) + \sigma(x)\sigma'(x)]\dfrac{\partial u}{\partial x}(t,x) + b'(x)u(t,x) \\ \qquad\qquad + f'\Big(\displaystyle\int_{-\infty}^{x} u(t,y)\ dy\Big)u(t,x), \\ u(0,x) = V_0'(x), \end{array}\right\} \tag{4.2}$$

with, for example (Fisher equation), $f(u) := u(u-1)$. The numerical method described below is based on the representation of the measure $u(T,x)\ dx$ in terms of the limit of the

empirical distribution of the living particles at time T of a branching interacting particles system.

The algorithm is as follows.

(i) At time 0, N particles with mass $1/N$ are located at points $V_0^{-1}\big(\frac{i}{N}\big)$, $i = 1, \ldots, N-1$, and at $V_0^{-1}\big(1 - \frac{1}{2N}\big)$.

(ii) Let h be the time discretization step; between times kh and $(k+1)h$ each particle living at time kh moves independently of the other particles; its position at time $(k+1)h$ is

$$\begin{aligned}\overline{Y}_{(k+1)h} = \overline{Y}_{kh} + \left\{\sigma\left(\overline{Y}_{kh}\right)\sigma'\left(\overline{Y}_{kh}\right) - b\left(\overline{Y}_{kh}\right)\right\}h + \sigma\left(\overline{Y}_{kh}\right)\left(W_{(k+1)h} - W_h\right)\\ + \frac{1}{2}\sigma\left(\overline{Y}_{kh}\right)\sigma'\left(\overline{Y}_{kh}\right)\left\{\left(W_{(k+1)h} - W_h\right)^2 - h\right\}.\end{aligned}$$

(iii) At each time step one creates and deletes particles according to the following rule. Let $\overline{\mathcal{N}}^N_{(k+1)h}$ denote the number of particles living at time $(k+1)h$, and

$$\overline{V}^N((k+1)h, x) := \frac{1}{N}\sum_{j=1}^{\overline{\mathcal{N}}^N_{(k+1)h}} H(x - \overline{y}^j_{(k+1)h}),$$

where $\{\overline{y}^j_{(k+1)h}\}$ is the set of the simulated particles which are living at time $(k+1)h$ and H is the Heaviside function. The particle numbered j dies with probability $h\big|f' \circ \overline{V}^N((k+1)h, \overline{y}^j_{(k+1)h})\big|$. If $f' \circ \overline{V}^N((k+1)h, \overline{y}^j_{(k+1)h}) \geq 0$, it gives birth to two particles.

Finally, the function $\overline{V}^N$ is our approximation of $V(T, x)$. The corresponding statistical error is $\mathcal{O}\frac{1}{\sqrt{N}}$ and discretization error is $\mathcal{O}\sqrt{h} + \mathcal{O}\frac{1}{\sqrt{N}}$: see Régnier and Talay [22].

As the Monte Carlo methods considered in the preceding section for linear problems, the stochastic particle methods may be used to approximate the solutions to nonlinear problems on artificial boundaries. However a lot of work still needs to be done in that area, either to construct probabilistic interpretations or to develop numerical methods based on the probabilistic interpretations. To illustrate that point, consider the 2D Navier-Stokes equation

$$\begin{cases}\dfrac{\partial u}{\partial t}(t,x) = \nu\Delta u(t,x) - (u(t,x)\cdot\nabla)u(t,x) - \nabla p(t,x),\ 0 < t,\ x \in \partial D,\\ \text{div } u(t,x) = 0,\ 0 < t,\ x \in \partial D,\\ u(t,x) = 0,\ 0 < t,\ x \in \partial D,\\ u(0,x) = f(x),\ x \in D.\end{cases}$$

Set $\omega := \text{rot } u$. Benachour, Roynette and Vallois [3] have shown that

$$\int_{\bar{D}} \omega(t,x)\ f(x)\ \text{Leb}(dx) = \mathbb{E}_{\omega_0}\int_{\bar{D}} f(x)\ dY_t \text{ for all } f \in \mathcal{C}^\infty(D),$$

where (Y_t) is a measure valued branching process with reflected paths. Unfortunately the law of the branching times is quite complex (much more complex than in the case of the Sherman and Peskin algorithm discussed above) and, at the time being, there is no efficient way of simulating the process (Y_t).

5. A first numerical illustration: an elliptic problem. One seeks the numerical approximation of $u(x,y)$ which satisfies

$$\begin{cases} L\,u & = f \quad \text{in} \quad D, \quad \text{with} \quad D = [0,2]\times[0,1], \\ u & = g \quad \text{on} \quad \partial D. \end{cases} \tag{5.1}$$

The elliptic operator is defined as

$$L = B^x(x,y)\frac{\partial}{\partial x} + B^y(x,y)\frac{\partial}{\partial x} + \frac{1}{2}a(x,y)\frac{\partial^2}{\partial x^2} + c(x,y)\frac{\partial^2}{\partial x \partial y} + \frac{1}{2}b(x,y)\frac{\partial^2}{\partial y^2}. \tag{5.2}$$

5.1. Problem definition. The diffusion matrix is defined by the following coefficients: $a(x,y)$, $b(x,y)$ and $c(x,y)$. For $b(x,y)$ one has

$$b(x,y) = (C_B + 1) + (x-1)^2,$$

and for $a(x,y)$, if $x \leq 1$,

$$a(x,y) = \left\{ \frac{C_B - C_A}{\pi} \arctan[-C(x-0.8)] + \frac{C_B + C_A}{2} \right\} b(x,y), \tag{5.3}$$

else,

$$a(x,y) = \left\{ \frac{C_B - C_A}{\pi} \arctan[C(x-1.2)] + \frac{C_B + C_A}{2} \right\} b(x,y), \tag{5.4}$$

with $C_B = (K/(k\,\pi))^2$ and $C_A = (K/(l\,\pi))^2$. The last coefficient, $c(x,y)$, is defined as

$$c(x,y) = 0.1\,(x-1)^2 + 0.005. \tag{5.5}$$

The values of the different constants k, l, K and C are listed in Table 5.1.

Table 5.1: Numerical values for the constants in Eqs (5.3) and (5.4).

k	l	K	C
5	20	1	1

The drift vector is defined by its two components $B^x(x,y)$ and $B^y(x,y)$, that is,

$$\begin{cases} B^x(x,y) & = K\,c(x,y), \\ B^y(x,y) & = 0.1. \end{cases} \tag{5.6}$$

The function $f(x,y)$ reads

$$\begin{aligned} f(x,y) = {} & C_E(x)\sin(C_E(x)\pi x)\exp(-Ky)\,[c(x,y)K - B^x(x,y)] \\ & + \cos(C_E(x)\pi x)\exp(-Ky) \\ & \quad \left[-KB^y(x,y) - \frac{1}{2}(C_E(x)\pi)^2 a(x,y) + \frac{1}{2}K^2 b(x,y) \right], \end{aligned}$$

with $C_E(x) = k$ for $x \in [0;0.8] \cup [1.2;2]$ and $C_E(x) = l$ for $x \in]0.8;1.2[$. The analytical solution to system (5.1) is

$$u(x,y) = 2 + \cos(C_E(x)\pi x)\exp(-Ky). \tag{5.7}$$

Figure 5.1 shows the shape of both the function $f(x,y)$ and the analytical solution $u(x,y)$ for the numerical values indicated in Table 5.1.

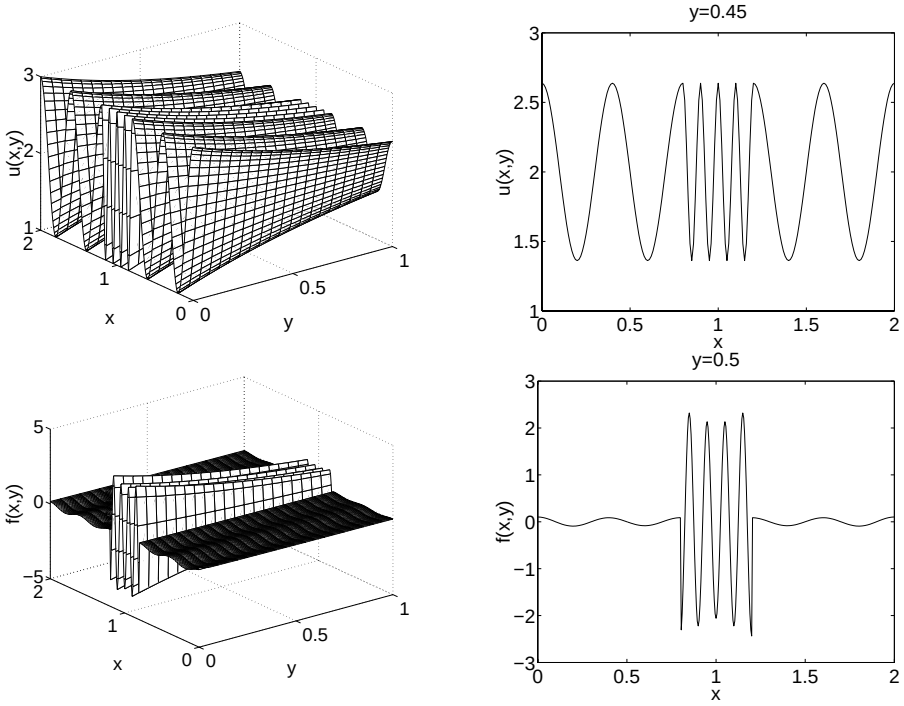


Figure 5.1: Top: shape of the analytical solution $u(x, y)$. Bottom: shape of $f(x, y)$. The results are given for the numerical values indicated in Table 5.1.

5.2. Deterministic method. Here, the elliptic problem is solved by classical discretisation techniques such as finite difference methods. For example, the second order derivative is approximated by (second order scheme)

$$\frac{\partial^2 u}{\partial x^2} = \frac{1}{(\Delta x)^2}[u(x+\Delta x, y) - 2u(x,y) + u(x-\Delta x, y)] + \mathrm{o}(\Delta x)^2. \tag{5.8}$$

All other derivatives are also computed with centered schemes. The computational domain is discretized with a uniform Cartesian mesh, that is

$$\begin{cases} x_i &= x_m + (i-1)(x_M - x_m)/(N_x - 1), \\ y_j &= y_m + (j-1)(y_M - y_m)/(N_y - 1). \end{cases} \tag{5.9}$$

The domain is defined by $[x_m, x_M] \times [y_m, y_M]$ and N_x and N_y represent the total number of discrete points in the x and y directions, respectively.

Eq. (5.1) can be written in its discretized form as $(u(x_i, y_j) = u_{i,j})$

$$\begin{aligned} f_{i,j} = {} & P_{imjp}\, u_{i-1,j+1} + P_{ijp}\, u_{i,j+1} + P_{ipjp}\, u_{i+1,j+1} \\ & + P_{imj}\, u_{i-1,j} + P_{ij}\, u_{i,j} + P_{ipj}\, u_{i+1,j} \\ & + P_{imjm}\, u_{i-1,j-1} + P_{ijm}\, u_{i,j-1} + P_{ipjm}\, u_{i+1,j-1}, \end{aligned} \tag{5.10}$$

where

$$\begin{aligned} & P_{imjm} = \frac{c_{i,j}}{4\,\Delta x\,\Delta y}, \quad P_{ijm} = \frac{b_{i,j}}{2\,\Delta x\,\Delta y} - \frac{B^y_{i,j}}{2\,\Delta y}, \quad P_{ipjm} = -P_{imjm}, \\ & P_{imj} = \frac{a_{i,j}}{2(\Delta x)^2} - \frac{B^x_{i,j}}{2\,\Delta x}, \quad P_{ij} = -\frac{a_{i,j}}{(\Delta x)^2} - \frac{b_{i,j}}{(\Delta y)^2}, \quad P_{ipj} = \frac{a_{i,j}}{2(\Delta x)^2} + \frac{B^x_{i,j}}{2\Delta x}, \\ & P_{imjp} = -P_{imjm}, \quad P_{ijp} = \frac{b_{i,j}}{2\,\Delta x\,\Delta y} + \frac{B^y_{i,j}}{2\Delta y}, \quad P_{ipjp} = P_{imjm}. \end{aligned}$$

The linear non-symmetric $(N_x - 2) \times (N_y - 2)$ system is solved with a preconditioned conjugate gradient method (NAG library, routine name: F11DEF).

The numerical error (relative error) is computed as follows (where $\mathcal{O}u(x,y)$ is the approximated value of $u(x,y)$)

$$e(x,y) = |u(x,y) - \mathcal{O}u(x,y)|, \tag{5.11}$$

which gives for the maximum error

$$e_{max} = \sup_{(x,y)\in D} e(x,y). \tag{5.12}$$

The maximum error is given, for different resolutions, in Table 5.2. The results show that (i) only the resolution along the x axis is important, (ii) the numerical scheme is, in reality, of first order in space. The CPU time is given for further comparison in execution time with alternative numerical methods (Monte-Carlo). The computer used to perform the simulations is a SUN Ultra 5/10 with a 440 MHz sparcv8plus+vis processor.

The shape of the numerical error along the x axis is displayed in Figure 5.2 for two different resolutions. It is seen, as expected, that the maximum error is obtained in the regions of steepest gradients.

Table 5.2: Numerical parameters and results for the deterministic method: spatial resolution, maximum error and CPU time (in seconds).

N_x	41	401	401	4001
N_y	21	201	21	201
Δx	0.05	0.005	0.005	0.0005
Δy	0.05	0.005	0.05	0.005
e_{max}	0.4164	0.0519	0.0519	0.0052
CPU (s)	0.1	33	0.8	744

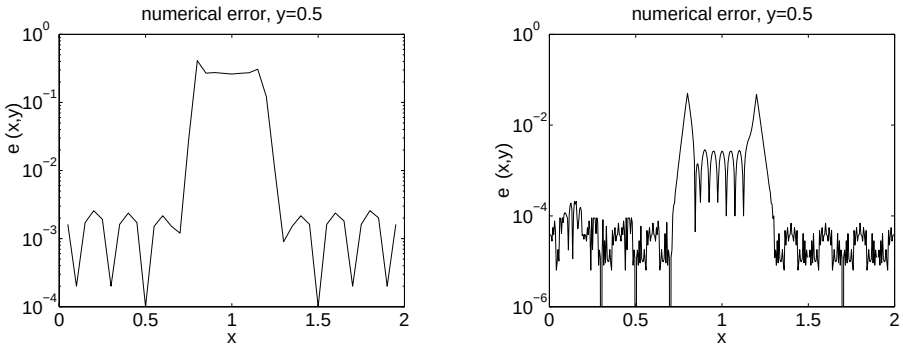


Figure 5.2: Numerical error along the x axis for $y = 0.5$. Two different resolutions are displayed: $(N_x, N_y) = (41, 21)$ (left) and $(N_x, N_y) = (401, 201)$ (right).

5.3. Probabilistic method. The probabilistic interpretation of the solution of (5.1) is

$$u(x,y) = -\mathbb{E}\left[\int_0^\tau f(X_t(x,y))\,dt\right] + \mathbb{E}\left[g(X_\tau(x,y))\right],$$

where the underlying stochastic process solves

$$\begin{cases} X_t^1(x,y) &= x + \int_0^t B^x(X_s(x,y))\,ds + \int_0^t \sigma_1^1(X_s(x,y))\,dW_s^1 + \int_0^t \sigma_2^1(X_s(x,y))\,dW_s^2, \\ X_t^2(x,y) &= y + \int_0^t B^y(X_s(x,y))\,dt + \int_0^t \sigma_1^2(X_s(x,y))\,dW_s^1 + \int_0^t \sigma_2^2(X_s(x,y))\,dW_s^2. \end{cases}$$

Here (W_t^1, W_t^2) are two independent Wiener processes, σ is a matrix valued function such that

$$\sigma(x,y)\sigma(x,y)^t = \begin{pmatrix} a(x,y) & c(x,y) \\ c(x,y) & b(x,y) \end{pmatrix},$$

and τ is the first exit time of the domain D as defined in Section 2.

The Euler scheme reads

$$\begin{aligned} {X^h}^1_{(p+1)h}(x,y) &= {X^h}^1_{ph}(x,y) + B^x(X^h_{ph}(x,y))h + \sigma_1^1(X^h_{ph}(x,y))\,(W^1_{(p+1)h} - W^1_{ph}) \\ &\quad + \sigma_2^1(X^h_{ph}(x,y))\,(W^2_{(p+1)h} - W^2_{ph}) \\ {X^h}^2_{(p+1)h}(x,y) &= {X^h}^2_{ph}(x,y) + B^y(X^h(x,y))h + \sigma_1^2(X^h_{ph}(x,y))\,(W^1_{(p+1)h} - W^1_{ph}) \\ &\quad + \sigma_2^2(X^h_{ph}(x,y))\,(W^2_{(p+1)h} - W^2_{ph}). \end{aligned}$$

The corresponding Monte Carlo approximation is defined as

$$\mathcal{O}u(x,y) = \frac{1}{N}\sum_{k=1}^{N}\left[-h\sum_{p=0}^{\tau^h-1} f(X_{ph}^{h,k}(x,y)) + g(X_{\tau^h}^{h,k}(x,y))\right].$$

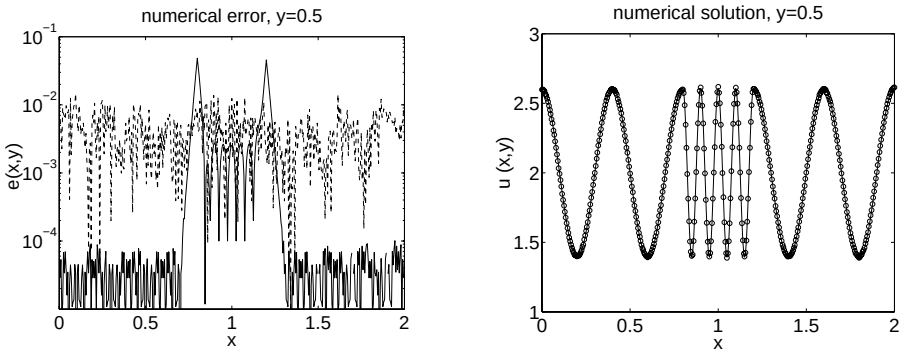


Figure 5.3: Numerical results for a Monte-Carlo simulation with $N = 10^4$ and $h = 10^{-4}$ s. The resolution is given by $(N_x, N_y) = (401, 21)$ but the results are presented along the x axis for $y = 0.5$. Left: numerical error (dashed line for the Monte-Carlo simulation and continuous line for the deterministic method). Right: numerical solution ($\circ$ for the Monte-Carlo simulation and continuous line for the exact solution).

Different numerical parameters (time steps and number of trajectories) and numerical procedures (Romberg extrapolation and treatment of the killed diffusion with Brownian bridges as proposed by Gobet [14] and [15]) were employed. In the computations presented here, $h = 10^{-4}$ and $N = 10^4$. The Euler scheme has been used with no specific treatment (the simulation is stopped at the time step where the point leaves the domain). The numerical error and the numerical solution are displayed in Figure 5.3. It can be observed that the numerical error is not too sensitive to the gradients. However, each Monte-Carlo point takes approximately 120 s of computer time.

Domain decomposition can now be performed. For example, one can consider the following domain: $D_1 = [0, 0.8] \times [0, 1]$. The computations can be done with the deterministic method by using the results of the Monte-Carlo simulation as boundary conditions (for $x = 0.8$). Figure 5.4 displays the results of such a computation.

The computation is performed for $(Nx, Ny) = (81, 21)$ on D_1. The Monte-Carlo points used as boundary conditions are obtained from the previous results presented in Figure 5.3.

It can be concluded that, even though the domain decomposition is technically feasible, there is no improvement for the CPU time. Indeed, for a similar precision in the domain of steep gradients ($x \in [0.8, 1.2]$), the CPU time for one Monte-Carlo point is roughly equal to the whole computation with the deterministic method and this for roughly 10^3 points.

6. A second numerical illustration: a parabolic problem. One seeks the numerical approximation of $u(x, y)$ which satisfies

$$\begin{cases} \dfrac{\partial u}{\partial t} + L\,u = 0 \quad \text{in} \quad D := [0,T] \times \mathbb{R}, \\ u(T,x) = f(x), \end{cases} \tag{6.1}$$

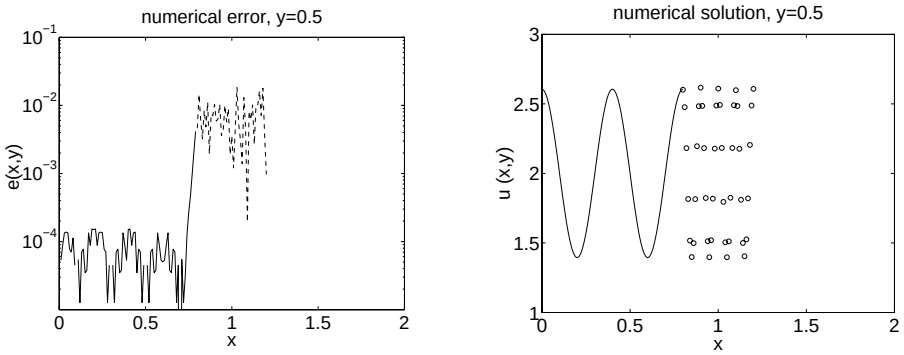


Figure 5.4: Numerical results for a deterministic/Monte-Carlo simulation. For the deterministic computation in D_1, one has $(N_x, N_y) = (81, 21)$. For the Monte-Carlo simulation $N = 10^4$ and $h = 10^{-4}$ s. The results are presented along the x axis for $y = 0.5$. Left: numerical error (dashed line for the Monte-Carlo simulation and continuous line for the deterministic method). Right: numerical solution ($\circ$ for the Monte-Carlo simulation and continuous line for the deterministic method).

where the elliptic operator L is defined as

$$L = D(t,x)\frac{\partial}{\partial x} + \frac{1}{2}B(t,x)\frac{\partial^2}{\partial x^2}. \tag{6.2}$$

The functions $D(t,x)$ and $B(t,x)$ are given by

$$D(t,x) = \cos(x)\,\sin(x)\left\{\lambda\cos(\lambda t) - [\cos(x)\exp(a(t))]^2\right\}, \tag{6.3}$$

and

$$B(t,x) = \left[\cos^2(x)\exp[a(t)]\right]^2, \tag{6.4}$$

respectively. The function $a(t)$ is defined as $a(t) = \sin(\lambda t)$ where λ is a constant (a real positive number). The final condition, $u(T,x) = f(x)$ is defined as

$$f(x) = \exp\left[\cos\left(\frac{1}{0.1+\lambda x^2}\right)\right] + \exp\left[\sin\left(\frac{1}{0.1+\mu x^2}\right)\right], \tag{6.5}$$

where $\mu \in \mathbb{R}^+$.

It can be shown that the analytical solution to system (6.1) is

$$u(t,x) = \int_{-\infty}^{+\infty} f\left(\arctan\left\{\exp[a(T)-a(t)]\tan(x) + y\,\exp[a(T)]\sqrt{T-t}\right\}\right) p(y)\,dy, \tag{6.6}$$

where $p(y)$ is the normal centered Gaussian law,

$$p(y) = \frac{1}{\sqrt{2\pi}}\exp(-y^2/2). \tag{6.7}$$

The solution can be computed numerically by resorting to a proper numerical procedure (here a NAG routine, D01AHF, is used). Figure 6.1 shows the shape of $u(t,x)$ for $[0,3]\times[1.0,1.4]$ ($f(x)$ is also shown).

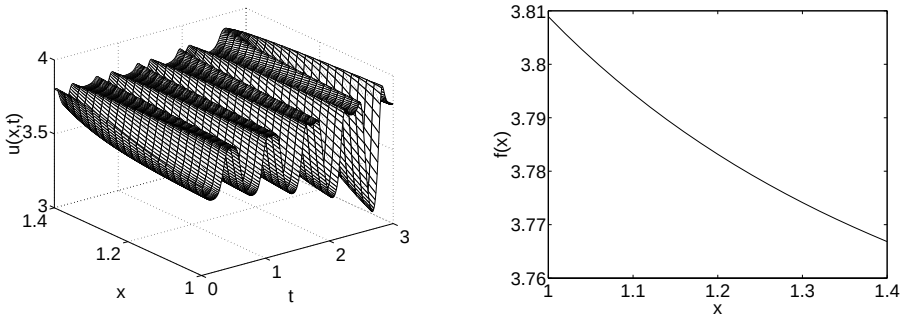


Figure 6.1: Left: approximated solution of system (6.1)on $[0,3]\times[1.0,1.4]$ by numerical integration of Eq. (6.6). Right: shape of the final condition, $f(x)$.

6.1. Deterministic method. System (6.1) can be solved by a simple deterministic method. A finite difference method is adopted where a first order approximation is used for the time derivative and a second order approximation for the space derivatives, that is, for example,

$$\frac{\partial u}{\partial x} = \frac{1}{2\Delta x}[u(x+h,x) - u(x-h,x)] + \mathrm{o}(\Delta x)^2,$$
$$\frac{\partial u}{\partial t} = \frac{1}{h}[u(t,x+h) - u(t,x)] + \mathrm{o}(h).$$

The scheme is explicit and it reads

$$u_i^{k+1} = -\frac{h}{2\Delta x}\left(D_i^k + \frac{B_i^k}{\Delta x}\right) u_{i+1}^k + \left(1 + \frac{B_i^k h}{2\,\Delta x^2}\right) u_i^k - \frac{\Delta t}{2\Delta x}\left(\frac{B_i^k}{\Delta x} - D_i^k\right) u_{i-1}^k,$$

where u_i^k is the approximation of $u(t,x)$ for $x = i\,\Delta x$ and $t = k\,h$ (Δx is the space resolution and h is the time step). Figure 6.2 shows the result of a computation on $[0,T]\times[A,B]$ with $\Delta x = 2.10^{-2}$ and $h = 10^{-3}$ (it has been checked by Von Neumann analysis that the scheme is stable for these values of Δx and h). The values of the different constants μ, λ, A, B and T are listed in Table 6.1. As in the previous examples, the maximum numerical error is obtained in the regions of steepest gradients, Figure 6.2.

Table 6.1: Numerical values for the numerical solution of system (6.1) on $[0,T]\times[A,B]$.

μ	λ	A	B	T
10	10	1.0	1.4	3

6.2. Probabilistic method. The preceding deterministic method requires to know exact values, or at least good approximations, of the solution $u(t,x)$ along the artificial boundaries $x = A$ and $x = B$. The Monte Carlo method allows one to get good approximations for all choice of the pair (A,B).

For all $0 \le t \le T$ the probabilistic interpretation of the solution to the system (6.1) is

$$u(t,x) = \mathbb{E}\left[f(X_T^{t,x})\right] \tag{6.8}$$

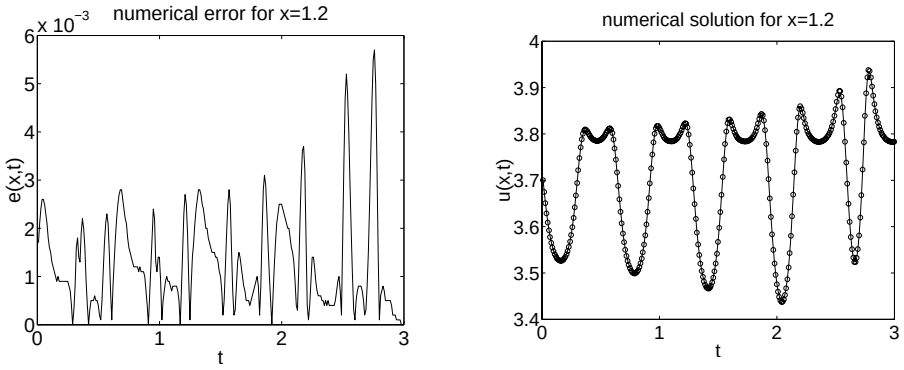


Figure 6.2: Numerical results for a deterministic simulation of system (6.1) on $[0,3] \times [1.0, 1.4]$. The time step is $h = 10^{-3}$ and the space resolution is $\Delta x = 2.10^{-2}$. The results are presented along the t axis for $x = 1.2$. Left: numerical error. Right: numerical solution ($\circ$ for the simulation and continuous line for the 'exact' solution).

where the underlying stochastic process is the solution to

$$\begin{aligned} X_\theta^{t,x} = x &+ \int_t^\theta \cos(X_s^{t,x}) \sin(X_s^{t,x}) \left\{ a'(s) - \left[\cos(X_s^{t,x}) b(s)\right]^2 \right\} ds \\ &+ \int_t^\theta \cos^2(X_s^{t,x})\, b(s)\, dW(s),\ t \le \theta \le T. \end{aligned} \tag{6.9}$$

Here, $a'(t)$ is the time derivative of $a(t)$ and $b(t) = \exp[a(t)]$. Notice that the solution $u(t,x)$ is expressed in terms of a process starting at time t, whereas in (2.5) the solution $u(T,x)$ is expressed in terms of a process observed at time T: that difference is due to the fact that, in (6.1), the initial condition is fixed at time T instead of 0 and one integrates backward in time instead of forward in time, which leads to a more convenient probabilistic interpretation when the coefficients of L are time dependent.

The numerical solution is obtained by a Monte-Carlo simulation as done in Section 5.3. The trajectories of (X_t) are simulated by applying the Euler scheme to Eq. (6.9). Figure 6.3 shows the result of a Monte-Carlo computation for $N = 10^4$ and $h = 10^{-4}$, which gives a numerical precision of the same order of magnitude as the computation performed with the deterministic method, see Figure 6.2. However, for such a simulation, each Monte-Carlo point takes approximately 30 s of computer time whereas the deterministic method requires 2 s for 6000 nodes (the value of 30 s for a Monte-Carlo point is an expected value for several points in the domain since the computations are faster for points near the boundary), see Table 6.2.

Table 6.2: CPU time for the deterministic and the Monte-Carlo method.

	CPU time: 1 point	CPU time: 6000 nodes
Monte-Carlo method	30 s	
Deterministic method		2 s

7. Conclusion. It is possible to use Monte-Carlo methods for domain decomposition problems when solving PDEs with deterministic techniques. However, the CPU time required

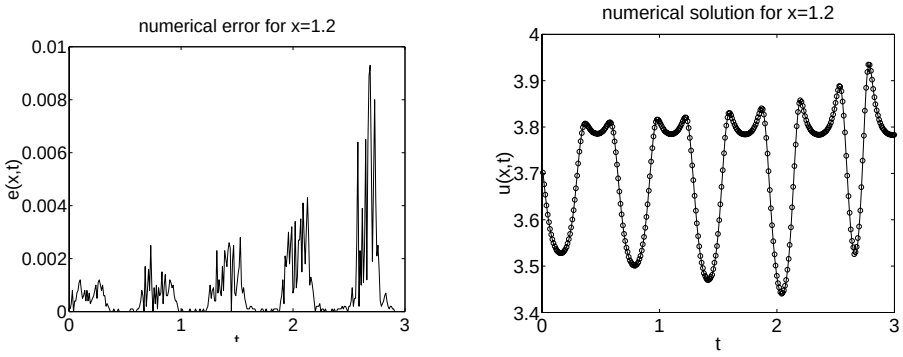


Figure 6.3: Numerical results for a Monte-Carlo simulation with $N = 10^4$ and $h = 10^{-4}$ s. The results are presented along the t axis for $x = 1.2$. Left: numerical error. Right: numerical solution ($\circ$ for the Monte-Carlo simulation and continuous line for the 'exact' solution).

by the Monte-Carlo methods does not make, in this sense, any improvement compared to full deterministic methods. Monte-Carlo methods are, however, interesting in cases where they are the only alternative (unknown boundary conditions or high dimensional problems, for example).

A widely still open problem is to find optimal estimates for the global error of algorithms combining deterministic methods in sub–domains and Monte Carlo methods to approximate the solutions along the artificial boundaries produced by a domain decomposition.

REFERENCES

[1] V. Bally and G. Pagès. A quantization algorithm for solving multi-dimensional optimal stopping problems. Submitted for publication, 2002.

[2] V. Bally and D. Talay. The law of the Euler scheme for stochastic differential equations (I) : convergence rate of the distribution function. *Probability Theory and Related Fields*, 104(1), 1996.

[3] S. Benachour, B. Roynette, and P. Vallois. Branching process associated with 2D Navier Stokes equation. *Revista Matematica Iberoamericana*, 2002 (to appear).

[4] C. Berthelot. PhD thesis in preparation.

[5] M. Bossy. Optimal rate of convergence of a stochastic particle method to solutions of 1D scalar conservation laws. Submitted for publication, 2002.

[6] M. Bossy and B. Jourdain. Rate of convergence of a stochastic particle method to solutions of 1D scalar conservation laws in a bounded interval. *Annals Prob.*, 2002. To appear.

[7] M. Bossy and D. Talay. A stochastic particle method for the McKean-Vlasov and the Burgers equation. *Math. Comp.*, 66(217):157–192, 1997.

[8] D. Chevance. Numerical methods for backward stochastic differential equations. In L. Rogers and D. Talay, editors, *Numerical Methods in Finance*, Publications of the Newton Institute. Cambridge University Press, 1997.

[9] C. Costantini, B. Pacchiarotti, and F. Sartoretto. Numerical approximation for functionals of reflecting diffusion processes. *SIAM J. Appl. Math.*, 58(1):73–102, 1998.

[10] S. Crépey. *Contribution des Méthodes Numériques Appliquées la Finance et aux Jeux*. PhD thesis, École Polytechnique, 2000.

[11] N. El Karoui, E. Pardoux, and M-C. Quenez. Reflected backward stochastic differential equations and American options. In L. Rogers and D. Talay, editors, *Numerical Methods in Finance*, Publications of the Newton Institute, pages 215–231. Cambridge University Press, 1997.

[12] N. Fournier and S. Méléard. A stochastic particle numerical method for 3D Boltzmann equations without cutoff. *Math. Comp.*, 71:583–604, 2002.

[13] A. Friedman. *Stochastic Differential Equations and Applications*, volume 1. Academic Press, New York, 1975.

[14] E. Gobet. Weak approximation of killed diffusion using Euler schemes. *Stoch. Proc. Appl.*, 87:167–197, 2000.

[15] E. Gobet. Euler schemes and half-space approximation for the simulation of diffusion in a domain. *ESAIM Probability and Statistics*, 5:261–297, 2001.

[16] B. Jourdain. Probabilistic characteristics method for a 1D scalar conservation law. *Annals Appl. Prob.*, 12(1):334–360, 2002.

[17] I. Karatzas and S. Shreve. *Brownian Motion and Stochastic Calculus*. Springer-Verlag, New York, 1988.

[18] H. Kunita. Stochastic differential equations and stochastic flows of diffeomorphisms. In *Ecole d'Été de Saint-Flour XII*, volume 1097 of *Lecture Notes in Mathematics*. Springer, 1984.

[19] P.-L. Lions. On the Schwarz alternating method. II. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods*, pages 47–70, Philadelphia, PA, 1989. SIAM.

[20] S. Méléard. Asymptotic behaviour of some interacting particle systems; McKean–Vlasov and Boltzmann models. In D. Talay and L. Tubaro, editors, *Probabilistic Models for Nonlinear PDE's and Numerical Applications*, Lecture Notes in Math., Berlin, Heidelberg, New York, 1996. Springer Verlag.

[21] E. Pardoux. Backward stochastic differential equations and viscosity solutions of systems of semilinear parabolic and elliptic PDEs of second order. In L. Decreusefond, J. Gjerde, B. Oksendal, and A. Ustünel, editors, *Stochastic Analysis and Related Topics : The Geilo Workshop*, pages 79–127. Birkhäuser, 1998.

[22] H. Régnier and D. Talay. Vitesse de convergence d'une méthode particulaire stochastique avec branchements. *Note au C.R.A.S.*, t. 332(Série I):933–938, 2001.

[23] A. Sherman and C. Peskin. A Monte Carlo method for scalar reaction-diffusion equations. *SIAM J. Sci. Statist. Comput.*, 7(4):1360–1372, 1986.

[24] A. Shiryayev. *Probability*. Springer-Verlag New-York, 1984.

[25] A. Sznitman. Topics in propagation of chaos. In P. Hennequin, editor, *Ecole d'Eté de Probabilités de Saint-Flour XIX - 1989*, volume 1464 of *Lecture Notes in Math.*, pages 165–251, Berlin, Heidelberg, New York, 1991. Springer-Verlag.

[26] D. Talay. Probabilistic numerical methods for partial differential equations: elements of analysis. In D. Talay and L. Tubaro, editors, *Probabilistic Models for Nonlinear Partial Differential Equations*, volume 1627 of *Lecture Notes in Mathematics*, pages 148–196. Springer-Verlag, 1996.

[27] D. Talay and L. Tubaro. Expansion of the global error for numerical schemes solving stochastic differential equations. *Stoch. Anal. Appl.*, 8(4):94–120, 1990.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

12. Partition of Unity Coarse Spaces: Enhanced Versions, Discontinuous Coefficients and Applications to Elasticity

M. Sarkis [1] [2]

1. Introduction. In this paper, we consider overlapping Schwarz methods for finite element discretizations for certain elliptic problems. In order to make Schwarz methods scalable with respect to the number of subdomains, we add an appropriate coarse problem to the algorithms. The main purpose of this paper is to introduce new coarse spaces for overlapping Schwarz methods. The proposed coarse spaces are based on partitions of unity and on local functions of low energy. The set of local functions must contain the kernel of the discrete operator when restricted to the overlapping subdomains. For instance, for linear elasticity, it must include the local rigid body motions, and for Poisson equation it should include the constant function. We consider several classes of choices of partition of unity. We consider PU based on the kind of partition of unity used in the theoretical analysis of standard additive Schwarz methods, as well as PU based on the class of additive Schwarz methods based on harmonic extensions. The condition number of the algorithms grows only linearly or quasi-linearly with respect to the relative size of the overlap. And for certain choices of partition of unity, the methods are robust also with respect to jumps of coefficients.

Work on two-level methods on unstructured meshes is not new. Several different approaches have been introduced and some can be found in [4, 5, 7, 9, 8, 11, 12, 14, 17, 16, 1, 18, 22, 21] and papers cited therein. Related works to ours [6, 20, 19], based on two-level agglomeration techniques, can be found in [4, 14, 15]. Their algorithms and analysis use a class of partition of unity coarse space based on agglomeration smoothing techniques. In this paper, we consider coarse spaces that combine the partition of unity and low energy functions associated to the (overlapping) subdomains in order to design a new coarse spaces for elliptic problems. The partition of unity is used: 1) to localize the coarse basis functions to the subdomains, and 2) to force the coarse basis functions to have a smooth decaying to zero near the boundary of the subdomains. The low energy local functions associated to the subdomains allow us to have good approximation properties for the coarse space. The proposed coarse spaces, have several advantages over traditional coarse spaces: 1) it is applicable to discretizations on unstructured meshes, 2) it is algebraic (see below), 3) the associated algorithms do not require that the subdomains be connected or that the boundary of the subdomains be smooth, 4) the coarse basis functions of the PU coarse space are constructed explicitly and without the use of exact local solvers, 5) the stencil of the coarse matrix is sparser than the traditional ones, and 6) the support of the coarse basis functions is localized on the subdomains, and therefore easy in communication if implemented in a distributed memory parallel machine.

The preconditioners to be considered here are algebraic in the sense that the preconditioners are built in terms of the graph of the sparse matrix and the mesh partition. In this paper we provide some unified mathematical analysis to the finite element problems considered in this paper.

2. Elliptic Problems and Discretization. In this paper we consider two problems: the two-dimensional linear elasticity and the scalar transmission problem.

2.1. Linear Elasticity. We consider an isotropic elastic material in the configuration region $\Omega \subset \Re^2$. Let us denote $u^* = (u_1^*, u_2^*)^t$ to be the displacement and $f = (f_1, f_2)^t$ the body

[1]Instituto de Matemática Pura e Aplicada, Est. Dona Castorina, 110, Rio de Janeiro, RJ, CEP 22420-320, Brazil, msarkis@impa.edu

[2]Mathematical Sciences Department, Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01609, USA

force. Let the region Σ be Ω or a subregion of Ω, and the spaces $\vec{H}^1(\Sigma)$, $\vec{L}_2(\Sigma)$, and $\vec{H}^1_0(\Sigma)$ to be the spaces $(H^1(\Sigma), H^1(\Sigma))^t$, $(L_2(\Sigma), L_2(\Sigma))^t$, and $(H^1_0(\Sigma), H^1_0(\Sigma))^t$, respectively. The weak formulation of the static theory of linear elasticity with zero boundary displacement condition is given as follows:

Find $u^* \in \vec{H}^1_0(\Omega)$ such that

$$a(u^*, v) = f(v), \quad \forall v \in \vec{H}^1_0(\Omega), \tag{2.1}$$

where

$$a(u^*, v) = \int_\Omega \left(\mu E(u^*) : E(v) + \lambda \mathrm{div}(u^*)\, \mathrm{div}(v)\right)\, dx\,,$$

$$f(v) = \int_\Omega f \cdot v\, dx \quad \text{for} \quad f \in \vec{L}^2(\Omega),$$

and

$$E(v) = \frac{1}{2}\left(\nabla v + (\nabla v)^t\right).$$

The positive constants μ and λ are called the Lamé constants. It is well-known that $a(\cdot,\cdot)$ is elliptic and bounded, and therefore the system (1.1) is well posed [2, 3].

For simplicity, let Ω be a bounded polygonal region in $\Re^2$ with a diameter of size $O(1)$. The extension of the results to $\Re^3$ can also be carried out using similar ideas. Let $T^h(\Omega)$ be a shape regular, quasi-uniform triangulation of grid size $O(h)$ of Ω, and $V \subset \vec{H}^1_0(\Omega)$ be the finite element space consisting of continuous piecewise linear functions associated with the triangulation $T^h(\Omega)$ and zero Dirichlet boundary condition. The extension of the results for the case of local quasi-uniform triangulation is also straightforward.

We are interested in solving the discrete problem associated to (1.1): Find $u \in V$ such that

$$a(u, v) = f(v), \quad \forall\ v \in V. \tag{2.2}$$

Since $V \subset \vec{H}^1_0(\Omega)$, the discrete version is also well-posed.

2.2. Transmission Problem. We also consider a finite element problem, the scalar transmission problem with zero Dirichlet boundary condition. Find $u^* \in H^1_0(\Omega)$, such that

$$a(u^*, v) = f(v), \quad \forall\ v \in H^1_0(\Omega), \tag{2.3}$$

where now

$$a(u^*, v) = \int_\Omega \rho(x) \nabla u \cdot \nabla v\, dx \quad \text{and} \quad f(v) = \int_\Omega f v\, dx \quad \text{for} \quad f \in L^2(\Omega).$$

We assume the coefficient ρ satisfy $0 < \rho_{\min} \le \rho(x) \le \rho_{\max}$ and is constant and equal to ρ_i inside each substructure Ω_i. We allow the coefficient ρ to have highly discontinuity across substructures. Here, we let $V \subset H^1_0(\Omega)$ be the finite element space consisting of continuous piecewise linear functions associated with the triangulation $\mathcal{T}_h(\Omega)$ with zero Dirichlet boundary condition. We introduce the discrete problem (2.2) with the $a(\cdot,\cdot)$ and V given in this subsection.

Throughout this paper, C is positive generic constant that do not depend of any of the mesh parameters, the number of subdomains, and the parameters λ and μ. All the domains and subdomains are assumed to be open; i.e., boundaries are not included in their definitions. We will use a unified notation for both problems since the techniques used to design and analyze the algorithms are essentially the same.

3. Algebraic Subregions. Given the domain Ω and the triangulation $T^h(\Omega)$, we assume that a domain partition has been applied and resulted in N substructures (non-overlapping subregions) $\Omega_i, i = 1, \ldots N$, of size $O(H)$, such that

$$\overline{\Omega} = \cup_{i=1}^{N} \overline{\Omega}_i, \qquad \Omega_i \cap \Omega_j = \emptyset, \quad \text{for} \quad j \neq i.$$

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 the immediate neighboring elements $\tau_h \in T^h(\Omega)$ of Ω_i such that $\overline{\tau}_h \cap \overline{\Omega}_i \neq \emptyset$. Using the idea recursively, we can define a δ-extension overlapping subdomains Ω_i^δ

$$\Omega_i \subset \Omega_i^1 \subset \cdots \Omega_i^\delta.$$

Here the integer $\delta \geq 1$ indicates the level of element extension and δh is the approximate length of the extension. We note that this extension can be coded easily through the knowledge of the adjacent matrix associated to the mesh and the partition.

We want to design coarse spaces based on a class of partition of unity for which has been used as a very powerful tool in the theoretical analysis of Schwarz type domain decomposition methods. We note however that the partition of unity functions on this class do not necessarily vanish on $\partial\Omega$. Hence, they cannot be used straightforwardly as coarse basis functions since they should satisfy zero Dirichlet boundary conditions for Dirichlet type boundary problems. Hence, for the coarse basis functions that touch $\partial\Omega$, we modify them so that they have a controlled decaying to zero near $\partial\Omega$. To obtain such coarse basis functions, we next introduce a Dirichlet boundary treatment. Let Ω_B^1 be one layer of elements near the Dirichlet boundary $\partial\Omega$ and then define recursively,

$$\Omega_B^1 \subset \Omega_B^2 \cdots \Omega_B^\delta$$

with δ levels of extension by adding recursively neighboring elements.

To define and analyze the new methods, we introduce some notations. We subdivide Ω_i^δ as follow: Let $\gamma_i^\delta = \partial\Omega_i^\delta \backslash \partial\Omega, i = 1, \cdots, N$; i.e., the part of the boundary of Ω_i^δ that does not belong to the physical boundary of Ω, and let $\gamma_B^\delta = \partial\Omega_B^\delta \backslash \partial\Omega$. We define the interface overlapping boundary Γ^δ as the union of all the γ_i^δ and γ_B^δ; i.e., $\Gamma^\delta = \cup_{i=B,1}^{N} \gamma_i^\delta$. We also need the following subsets of Ω_i^δ

- $\Gamma_i^\delta = \Gamma^\delta \cap \Omega_i^\delta$ (local interface)
- $N_i^\delta = \Omega_i^\delta \backslash (\cup_{j \neq i} \Omega_j^\delta \cup \Omega_B^\delta \cup \Gamma_i^\delta)$ (non-overlapping region)
- $O_i^\delta = \Omega_i^\delta \backslash (N_i^\delta \cup \Gamma_i^\delta)$ (overlapping region)

In order to consider discontinuous coefficients for the scalar transmission problem, we introduce the following notations. Let the interior local interface $\Gamma_i^0 = \Gamma_i^\delta \cap \Omega_i$ be the part of Γ_i^δ which is inside of substructure Ω_i. We also introduce complementary local interface $\Gamma_i^c = \Gamma_i^\delta \backslash \overline{\Gamma}_i^0$. We note for later use that nodes on $\overline{\Gamma}_i^0 \cap \overline{\Gamma}_i^c$ also belong to $\partial\Omega_i \backslash \partial\Omega$. We subdivide the regions O_i^δ into subregions $O_{ij}^\delta = O_i^\delta \cap \Omega_j$ where the substructures Ω_j are neighbors of Ω_i. We note that the coefficient ρ is constant inside each of the subregion O_{ij}^δ and N_i^δ.

4. Additive Overlapping Schwarz Methods. We next describe the PU coarse spaces and introduce the corresponding overlapping additive Schwarz method and a hybrid Schwarz method. We first consider the local problems.

4.1. The Local Problems. We introduce local spaces as

$$V_i^\delta = V \cap \vec{H}_0^1(\Omega_i^\delta) \quad \text{or} \quad (V_i^\delta = V \cap H_0^1(\Omega_i^\delta)) \qquad i = 1, 2, \cdots, N,$$

extended by zero to $\Omega \backslash \Omega_i^\delta$. It is easy to verify that

$$V = V_1^\delta + V_2^\delta + \cdots + V_N^\delta. \tag{4.1}$$

The property (4.1) gives robustness for the preconditioners defined in this paper in the sense that a convergence is always attained independently of the quality of the partitioning. The coarse space is only introduced to accelerate the convergence of the iterative method. This is an advantage over some iterative substructuring methods in which are based on the requirement that all the substructures Ω_i must be connected. We point out that the space decomposition given by (4.1) is not a direct sum if $\delta > 1$. This increases robustness for the methods when the Ω_i have rough (*zigzag*) boundaries. By extending the substructures to Ω_i^δ, we allow the possibility of decomposing a function of V as a sum of functions of V_i^δ without the *zigzag* behavior. So it is possible to obtain low energy decompositions, and hence better lower bounds for the condition number of the preconditioners.

We define the local projections (local problems) $P_i^\delta : V \to V_i^\delta$ as follows:

$$a(P_i^\delta u, v) = a(u, v), \quad \forall v \in V_i^\delta. \tag{4.2}$$

We next introduce the PU coarse space V_0^δ for the linear elasticity and for the transmission problem.

4.2. Partition of Unity. We next construct a partition of unity θ_i^δ such that $\theta_i^\delta \in V_i^\delta$, $0 \le \theta_i^\delta(x) \le 1$, $|\nabla \theta_i^\delta(x)| \le C/(\delta h)$ in the interior of the elements, and $\sum_{i=B,1}^N \theta_i^\delta \equiv 1$. Such construction is natural, algebraic and easy to implement. We first construct the function $\hat{\theta}_B^\delta \in V_i^\delta$ as follows. We let $\hat{\theta}_B^\delta(x) = 1$ for nodes x on $\partial\Omega$. For the first layer of neighboring nodes x of $\partial\Omega$ we let $\hat{\theta}_B^\delta(x) = (\delta - 1)/\delta$. For the second layer of neighboring nodes x of $\partial\Omega$ we let $\hat{\theta}_B^\delta(x) = (\delta - 2)/\delta$, and recursively until $k = \delta - 1$, we let $\hat{\theta}_B^\delta(x) = (\delta - k)/\delta$ for the $(k)st$ layer of neighboring nodes x of $\partial\Omega$. For the reminaing nodes x of $\overline{\Omega}$ we let $\hat{\theta}_B^\delta(x) = 0$. Similarly, for $i = 1, \cdots, N$, we let $\hat{\theta}_i^\delta(x) = 1$ for nodes x of $\overline{\Omega}_i \backslash \Omega_B^\delta$. For the first layer of neighboring nodes x of $\overline{\Omega}_i \backslash \Omega_B^\delta$ we let $\hat{\theta}_i^\delta(x) = (\delta - 1)/\delta$, and recursively until $k = \delta - 1$, we let $\hat{\theta}_i^\delta(x) = (\delta - k)/\delta$ for the $(k)st$ layer of neighboring nodes x of $\overline{\Omega}_i \backslash \Omega_B^\delta$. For the remaining nodes x of $\overline{\Omega}$ we let $\hat{\theta}_i^\delta(x) = 0$. It is easy to verify that $0 \le \hat{\theta}_i^\delta(x) \le 1$, and for quasi-uniform triangulation $|\nabla \hat{\theta}_i^\delta(x)| \le C/(\delta h)$ in the interior of the elements. The partition of unity θ_i^δ is defined as

$$\theta_i^\delta = I_h\left(\frac{\hat{\theta}_i^\delta}{\sum_{j=B,1}^N \hat{\theta}_j^\delta}\right).$$

Here I_h is the regular pointwise linear interpolation operator from the continuous functions to piecewise linear and continuous functions. It is easy to verify that $\sum_{i=B,1}^N \theta_i^\delta(x) = 1$, $0 \le \theta_i^\delta(x) \le 1$, and $|\nabla \theta_i^\delta(x)| \le C/(\delta h), \forall x \in \overline{\Omega}$.

4.3. PU Coarse Space for Linear Elasticity. We next consider the key ingredient for designing the coarse space for linear elasticity: the local rigid body motions. We let

$$R\vec{M}(\Sigma) = \{v \in \vec{L}_2(\Sigma) : v = c + b(x_2, -x_1)^t, c \in \Re^2, b \in \Re\}$$

be the space of rigid body motions functions on Σ. An important property of the space $R\vec{M}(\Sigma)$, and which plays an important role in the design and analysis of the algorithms, is described as follows. If $v \in R\vec{M}(\Sigma)$ then $a_\Sigma(v, v) \equiv 0$. In addition, in certain extent the

converse is also true; if $a_\Sigma(v,v) \equiv 0$ and Σ is connected, then $v \in R\vec{M}(\Sigma)$. Here, the bilinear form $a_\Sigma(\cdot,\cdot)$ on $\vec{H}^1(\Sigma) \times \vec{H}^1(\Sigma)$ is given by

$$a_\Sigma(u,v) = \int_\Sigma \left(2\mu E(u) : E(v) + \lambda \mathrm{div}(u)\, \mathrm{div}(v)\right)\, dx.$$

A PU coarse space V_0^δ is defined as

$$V_0^\delta = \{\sum_{i=1}^N \vec{I}_h \left(R\vec{M}(\Omega_i^\delta)\theta_i^\delta\right)\} = \{\sum_{i=1}^N \vec{I}_h \left([c_i + b_i(x_2,-x_1)^t]\theta_i^\delta\right), \forall c_i \in \Re^2, \forall b_i \in \Re\}. \quad (4.3)$$

Here, the interpolator $\vec{I}_h = (I_h, I_h)^t$ is the regular componentwise pointwise linear interpolation operator. We note that we do not include the θ_B^δ in the sum of (4.3), and therefore the number of degrees of freedom of V_0^δ is $3N$. The function θ_B^δ is needed only to define the others functions θ_i^δ, $i = 1, \cdots, N$.

We define the global projection (global problem) $P_0^\delta : V \to V_0^\delta$ as follows:

$$a(P_0^\delta u, v) = a(u,v), \quad \forall v \in V_0^\delta. \quad (4.4)$$

4.4. PU Coarse Space for the Scalar Transmission Equation. The PU coarse space V_0^δ for the transmission problem is defined as

$$V_0^\delta = \{\sum_{i=1}^N I_h(c_i\theta_i^\delta), \forall c_i \in \Re\},$$

and the global projection P_0^δ by (4.4), where of course V and $a(\cdot,\cdot)$ are the ones related to the transmission. We note here also as in the elasticity case, the constant functions c_i are the kernel of the operator $a_{\Omega_i^\delta}(\cdot,\cdot)$.

4.5. Enhanced PU Coarse Spaces. We can also make richer the coarse spaces V_0^δ. We do this by redefining V_0^δ as

$$V_0^\delta = \{\sum_{i=1}^N \vec{I}_h \left([c_i + b_i(x_2,-x_1)^t + f_i(x)]\theta_i^\delta\right), \forall c_i \in \Re^2, \forall b_i \in \Re, f_i \in V_i^E(\Omega_i^\delta)\},$$

for the finite elasticity problem, and

$$V_0^\delta = \{\sum_{i=1}^N I_h \left([c_i + f_i(x)]\theta_i^\delta\right), \forall c_i \in \Re, \forall f_i \in V_i^E\}$$

for the scalar transmission problem. For each subdomain Ω_i^δ, we let the space $V_i^E(\Omega_i^\delta)$ be defined as the vector space generated by few lowest finite element eigenmodes associated to operator $a_{\Omega_i^\delta}(\cdot,\cdot)$ without assuming Dirichlet boundary condition on $\partial\Omega_i^\delta$. Another possibility and also cheaper to construct is to choose $V_i^E(\Omega_i^\delta)$ as the vector space of polynomial functions of small degrees.

4.6. Preconditioners. We consider two preconditioners:

- The two-level overlapping additive Schwarz operator [10] given by

$$P_{as}^\delta = \sum_{i=0}^N P_i^\delta,$$

- The hybrid Schwarz operator [16, 13] given by

$$P_{hyb}^\delta = P_0^\delta + (I - P_0^\delta)(\sum_{i=1}^N P_i^\delta)(I - P_0^\delta).$$

4.7. Condition Number. It is possible to show that the solution of (2.2) is the solution of the preconditioned system $P_{as}^{\delta}u = g_{as}$ ($P_{hyb}^{\delta}u = g_{hyb}$), for an appropriate right hand side g_{as} (g_{hyb}); see [13]. These preconditioned systems are typically solved by the conjugate gradient method, without further preconditioning, using $a(\cdot,\cdot)$ as the inner product. The preconditioned systems presented in this paper are applicable to any unstructured mesh and partitioning. The notions of subdomains, the classification of the regions O_i^{δ} and N_i^{δ} and the interfaces Γ_i^{δ}, etc., can all be defined in terms of the graph of the sparse matrix. The two algorithms (preconditioners) will converge even if the substructures Ω_i are not connected. For the next theoretical result [19, 20], we assume that the substructures Ω_i have nice aspect ratios and are connected.

Theorem 4.1 *There exists a constant $C > 0$ such that*

- *Linear Elasticity*
$$\kappa(P_{hyb}^{\delta}) \leq \kappa(P_{as}^{\delta}) \leq C(1+\lambda/\mu)(1+\frac{H}{\delta h}). \tag{4.5}$$
- *Transmission Problem*
$$\kappa(P_{hyb}^{\delta}) \leq \kappa(P_{as}^{\delta}) \leq Cc(\rho)(1+\frac{H}{\delta h}). \tag{4.6}$$

The constant C does not depend on h, δ, H, λ, and μ. The constant $c(\rho) \leq C \max_{ij} \frac{\rho_i}{\rho_j}$, where the pairs ij run over all ij combinations such that $\overline{\Omega}_i \cap \overline{\Omega}_j \neq \emptyset$.

We note that the discretization considered in this paper gives satisfactory (second order accurate) convergent finite element approximation to the elasticity problem when λ/μ is not large. It can be shown [2, 3] that the a priori error estimate of this finite element method deteriorates as $\lambda \gg \mu$; this phenomenon is called *locking effect* or *volume locking*. We note that the upper bound estimate of the preconditioners presented here also follows similar patterns. Here also, we cannot remove the λ/μ dependence on the upper bound estimates for the conditioning number of the preconditioned systems. To see this we use the following arguments: If $\mathrm{div}(u) = 0$ and λ is close to ∞, the only way to obtain a decomposition stable with respect to λ is to have all the $\mathrm{div}(u_i) = 0$. However, it is easy to see that $\mathrm{div}(u_0) = 0$ implies that u_0 vanishes. Hence, there is no global communication and therefore the condition number must have a H dependence on the upper bound estimation. Hence fortunately, the preconditioners considered here in this paper are effective exactly when the discretization is accurate. For incompressible ($\lambda = \infty$) or almost incompressible materials, other discretizations based on hybrid or non-conforming finite elements approximations [2, 3] are more appropriate and they will not be considered here.

For the transmission problem, the upper bound (4.6) is satisfactory if the jumps on the coefficient ρ are moderate. Later in the paper we design better coarse spaces for highly discontinuity in the coefficients.

5. AS Methods with Harmonic Overlap (ASHO). We next introduce the PU coarse spaces for the ASHO methods.

5.1. Local Problems for the ASHO Methods. We define $\widetilde{V}_i^{\delta}$ as the subspace of V_i^{δ} consisting of functions that are discrete harmonic at all nodes interior to O_i^{δ}, i.e. $u \in \widetilde{V}_i^{\delta}$, if for all nodes $x_k \in O_i^{\delta}$,
$$a(u, \phi_{x_k}) = 0.$$
Here, $\phi_{x_k} \in V$ is the regular componentwise finite element basis function associated with a node x_k.

We define $\widetilde{V}^{\delta}$ as a subspace of V defined as
$$\widetilde{V}^{\delta} = \widetilde{V}_1^{\delta} + \widetilde{V}_2^{\delta} + \cdots + \widetilde{V}_N^{\delta}.$$

We note that the above sum is not a direct sum and $\widetilde{V}_i^\delta \subset V_i^\delta$. We define $\widetilde{P}_i^\delta : \widetilde{V}^\delta \to \widetilde{V}_i^\delta$ to be the projection operators such that, for any $u \in \widetilde{V}^\delta$

$$a(\widetilde{P}_i^\delta u, v) = a(u, v), \quad \forall v \in \widetilde{V}_i^\delta.$$

We next introduce a PU coarse space $\widetilde{V}_0^\delta$ for the ASHO method.

5.2. A PU Coarse Space for ASHO Methods. For the finite elasticity, we define the PU coarse space $\widetilde{V}_0^\delta \subset \widetilde{V}^\delta$, by simply modifying the basis functions

$$\varphi_i^\delta = \vec{I}_h \left([c_i + b_i(x_2, -x_1)^t] \theta_i^\delta \right) \quad \text{or} \quad (\varphi_i^\delta = c_i \theta_i^\delta)$$

to $\widetilde{\varphi}_i^\delta$. The $\widetilde{\varphi}_i^\delta$ are defined to be equal to the φ_i^δ except on O_i^δ. On O_i^δ we make the $\widetilde{\varphi}_i^\delta$ discrete harmonic in the $a(\cdot,\cdot)$ inner product. The PU coarse space $\widetilde{V}_0^\delta$ is defined as the linear combination of the coarse basis functions $\widetilde{\varphi}_i^\delta, i = 1, \cdots, N$. We introduce $\widetilde{P}_0 : \widetilde{V}^\delta \to \widetilde{V}_0^\delta$ as the operator such that, for any $u \in \widetilde{V}^\delta$,

$$a(\widetilde{P}_0^\delta u, v) = a(u, v), \quad \forall v \in \widetilde{V}_0^\delta. \tag{5.1}$$

Then, the two-level additive and hybrid ASHO with the PU coarse problem $\widetilde{P}_0^\delta$ are defined as

$$\widetilde{P}_{as}^\delta = \sum_{i=0}^{N} \widetilde{P}_i^\delta, \quad \text{and} \quad \widetilde{P}_{hyb}^\delta = \widetilde{P}_0^\delta + (I - \widetilde{P}_0^\delta)(\sum_{i=1}^{N} \widetilde{P}_i^\delta)(I - \widetilde{P}_0^\delta).$$

The following bounds can be obtained [19].

Theorem 5.1 *On the space $\widetilde{V}_\delta$, we have*

$$\kappa(\widetilde{P}_{hyb}^\delta) \leq \kappa(\widetilde{P}_{as}^\delta) \leq \kappa(P_{as}^\delta)$$

5.3. A Robust PU Coarse Space for ASHO Methods. We next construct the coarse basis functions $\widetilde{\varphi}_i^\delta$ that make the ASHO methods robust with respect to the jumps of the coefficients ρ.

We redefine $\widetilde{\varphi}_i^\delta \in \widetilde{V}_i^\delta$ as follows. Nodes x_k on $(\Gamma_i^0 \cup N_i^\delta$, we define $\widetilde{\varphi}_i^\delta(x_k) = 1$. Nodes x_k on Γ_i^c, we let $\widetilde{\varphi}_i^\delta(x_k) = 0$. A node x_k on $\overline{\Gamma}_i^0 \cap \overline{\Gamma}_i^c$ also belongs to the $\partial\Omega_i \backslash \partial\Omega$. Hence, x_k belongs to $\overline{\Omega}_i$ and to some neighboring substructures $\overline{\Omega}_j$ and we define

$$\widetilde{\varphi}_i^\delta(x_k) = \frac{\rho_i^\beta}{\rho_i^\beta + \sum_j \rho_j^\beta},$$

where $\beta \geq 1/2$. Nodes x_k on $\overline{\Omega} \backslash \Omega_i^\delta$ we let $\widetilde{\varphi}_i^\delta(x_k) = 0$. It remains only to define $\widetilde{\varphi}_i^\delta(x_k)$ at nodes in O_i^δ. There, we make $\widetilde{\varphi}_i^\delta$ to be discrete harmonic on the $a(\cdot,\cdot)$ inner product.

Theorem 5.2 *On the space $\widetilde{V}^\delta$ we have*

$$\kappa(\widetilde{P}_{hyb}^\delta) \leq \kappa(\widetilde{P}_{as}^\delta) \leq C \left(1 + \frac{H}{\delta h} + \log(\frac{H}{\delta h}) \log(\delta) \right).$$

The constant C does not depend on h, δ, H, and ρ.

Proof. We here give a sketch of the proof. We define $\hat{\varphi}_i^\delta \in V_i^\delta$ as follows. Nodes x_k on $\Gamma_i^0 \cup N_i^\delta$, we define $\hat{\varphi}_i^\delta(x_k) = 1$. Nodes x_k on Γ_i^c, we let $\hat{\varphi}_i^\delta(x_k) = 0$. Nodes x_k on $\partial\Omega_i \backslash \partial\Omega$ we define

$$\hat{\varphi}_i^\delta(x_k) = \frac{\rho_i^\beta}{\rho_i^\beta + \sum_j \rho_j^\beta},$$

where $\beta \geq 1/2$, where the indices $j \neq i$ are the domains Ω_j for which $x_k \in (\partial\Omega_j \backslash \partial\Omega)$. Nodes x_k on $\overline{\Omega}\backslash\Omega_i^\delta$ we let $\hat{\varphi}_i^\delta(x_k) = 0$. It remains only to define $\hat{\varphi}_i^\delta(x_k)$ at nodes on the O_{ij}^δ. There, we make $\hat{\varphi}_i^\delta$ to be discrete harmonic.

There is an important distinction between the functions $\hat{\varphi}_i^\delta$ and $\widetilde{\varphi}_i^\delta$. The function $\hat{\varphi}_i^\delta$ is discrete harmonic on the regions O_{ij}^δ while the function $\widetilde{\varphi}_i^\delta$ is discrete harmonic (in the $a(\cdot,\cdot)$ inner product) on the region O_i^δ. We note that in each region O_{ij}^δ, the coefficient ρ is constant and therefore $\hat{\varphi}_i^\delta$ is discrete harmonic (in the H_1-seminorm) on the regions O_{ij}^δ. Because ρ is constant on the regions O_{ij}^δ we can borrow several previous results developed for RASHO [6] and for elliptic problems with discontinuous coefficients [9, 17, 12, 18, 8] to obtain

$$\kappa(\hat{P}_{as}) \leq C\left(1 + \frac{H}{\delta h} + \log(\frac{H}{\delta h})\log(\delta)\right),$$

where

$$\hat{P}_{as} = \hat{P}_0^\delta + \sum_{i=1}^{N} P_i^\delta,$$

and the global projection $\hat{P}_0^\delta : V \to \hat{V}_0^\delta$ is defined as

$$a(\hat{P}_0^\delta u, v) = a(u, v), \quad \forall v \in \hat{V}_0^\delta.$$

Here, the coarse space $\hat{V}_0^\delta$ is the space generated by the coarse basis functions $\hat{\varphi}_i^\delta$. Finaly, we use similar arguments as in [20], where we use that a function on $\widetilde{V}_i^\delta$ has smaller or equal $a(\cdot,\cdot)$ norm than a function on V_i^δ with the same values on Γ_i^δ, to obtain

$$\kappa(\widetilde{P}_{as}) \leq \kappa(\hat{P}_{as}).$$

■

6. Remarks about ASHO Methods. We next show that the explicit elimination of the variables associated with the overlapping nodes is not needed in order to apply $\widetilde{P}^\delta$ to any given vector $v \in \widetilde{V}^\delta$.

Lemma 6.1 *For any $u \in \widetilde{V}^\delta$, we have*

$$\widetilde{P}_i^\delta u = P_i^\delta u, \quad i = 1, \cdots, N.$$

Proof. If $u \in \widetilde{V}^\delta$ then

$$a(P_i^\delta u, \phi_{x_k}) = a(u, \phi_{x_k}) = 0, \quad \forall x_k \in O_i^\delta.$$

Hence, $P_i^\delta u \in \widetilde{V}_i^\delta$. Here, $\phi_{x_k} \in V_i^\delta$ are the regular basis functions associated to the nodes x_k. To complete the proof of the lemma, we just need to verify that

$$a(P_i^\delta u, v) = a(u, v), \quad \forall v \in \widetilde{V}_i^\delta. \tag{6.1}$$

To verify (6.1), we use the definition of P_i^δ (4.2) and that $\widetilde{V}_i^\delta$ is a subset of V^δ. ■

We note that the solution u of (2.2) is not in the subspace $\widetilde{V}^\delta$, therefore, the operators $\widetilde{P}_{as}^\delta$ and $\widetilde{P}_{hyb}^\delta$ cannot be used to solve the linear system (2.2) directly. We will need to modify the right-hand side of this system. A reformulated problem will be presented in Lemma 6.2 below. Using the matrix notations, the next lemma shows how to modify the system (2.2) so that its solution belongs to $\widetilde{V}^\delta$. Let $O^\delta = \cup_i O_i^\delta$. Let W_O^δ be the set of nodes associated

to the degree of freedom of V^δ in O^δ. We define the restriction operator, or a matrix, R_{O^δ}: $W \to W$ as follows

$$(R_{O^\delta} v)(x_k) = \begin{cases} v_k & \text{if } x_k \in W_{O^\delta} \\ 0 & \text{otherwise.} \end{cases}$$

The matrix representation of R_O^δ is given by a diagonal matrix with 1 for nodal points in the interior of O^δ and zero for the remaining nodal points. We denote by A the matrix associated to the problem (2.2). Using the restriction operator R_O^δ, we define the subdomain stiffness matrix as

$$A_{O^\delta} = R_{O^\delta}\ A\ R_{O^\delta}^T,$$

which can also be obtained by the discretization of the original finite element problem on O^δ with zero Dirichlet data on ∂O^δ and extended by zero outside of O^δ. We remark that O is a disconnected region where $\partial O = \Gamma_i^\delta \cup \partial\Omega$. Therefore, $A_{O^\delta} w = f$ can be solved locally and inexpensively.

It is easy to see that the following lemma holds; see [6].

Lemma 6.2 *Let u and f be the exact solution and the right-hand side of (2.2), and*

$$w = R_{O^\delta}^T A_{O^\delta}^+ R_{O^\delta} f. \tag{6.2}$$

Then $\widetilde{u} = u - w \in \widetilde{V}^\delta$ and satisfies the following modified linear system of equations

$$A\widetilde{u}^* = f - Aw = \widetilde{f}.$$

Acknowledgements: The author thanks local organizers of the conference for the support. The work was supported in part also by the NSF grant CCR-9984404 and PRH-ANP/MME/MCT 32.

REFERENCES

[1] P. E. Bjørstad, M. Dryja, and E. Vainikko. Additive Schwarz methods without subdomain overlap and with new coarse spaces. In R. Glowinski, J. Périaux, Z. Shi, and O. B. Widlund, editors, *Domain Decomposition Methods in Sciences and Engineering*. John Wiley & Sons, 1997. Proceedings from the Eight International Conference on Domain Decomposition Metods, May 1995, Beijing.

[2] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, Cambridge, 2001. Second Edition.

[3] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1994.

[4] M. Brezina and P. Vanek. A black-box iterative solvers based on a two-level Schwarz method. *Computing*, 63:233–363, 1999.

[5] X.-C. Cai. An optimal two-level overlapping domain decomposition method for elliptic problems in two and three dimensions. *SIAM J. Sci. Comp.*, 14:239–247, January 1993.

[6] 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. Comp.*, 2002. Submitted.

[7] T. F. Chan and B. F. Smith. Multigrid and domain decomposition on unstructured grids. In D. F. Keyes, , and J. Xu, editors, *Seventh International Conference of Domain Decomposition Methods in Scientific and Engineering Computing*, Providence, RI, 1995. AMS. Has also appeared in ETNA.

[8] M. Dryja, M. V. Sarkis, and O. B. Widlund. Multilevel Schwarz methods for elliptic problems with discontinuous coefficients in three dimensions. *Numer. Math.*, 72(3):313–348, 1996.

[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(6):1662–1694, December 1994.

[10] M. Dryja and O. B. Widlund. An additive variant of the Schwarz alternating method for the case of many subregions. Technical Report 339, also Ultracomputer Note 131, Department of Computer Science, Courant Institute, 1987.

[11] M. Dryja and O. B. Widlund. Domain decomposition algorithms with small overlap. *SIAM J. Sci.Comput.*, 15(3):604–620, May 1994.

[12] M. Dryja and O. B. Widlund. Schwarz methods of Neumann-Neumann type for three-dimensional elliptic finite element problems. *Comm. Pure Appl. Math.*, 48(2):121–155, February 1995.

[13] W. D. Gropp and B. F. Smith. Experiences with domain decomposition in three dimensions: Overlapping Schwarz methods. In A. Quarteroni, Y. A. Kuznetsov, J. Périaux, and O. B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, volume 157 of *Contemporary Mathematics*, pages 323–334. AMS, 1994. Held in Como, Italy, June 15–19,1992.

[14] E. Jenkins, C. Kees, C. Kelley, and C. Miller. An aggregation-based domain decomposition preconditioner for groundwater flow. *SIAM J. Sci. Comp*, 25:430–441, 2001.

[15] C. Lasser and A. Toselli. Convergence of some two-level overlapping domain decomposition preconditioners with smoothed aggregation coarse space. In L. F. Pavarino and A. Tosell, editors, *Recent Developments in Domain Decomposition Method*, pages 93–116. Springer-Verlag, 2002.

[16] J. Mandel. Hybrid domain decomposition with unstructured subdomains. In A. Quarteroni, Y. A. Kuznetsov, J. Périaux, and O. B. Widlund, editors, *Domain Decomposition Methods in Science and Engineering: The Sixth International Conference on Domain Decomposition*, volume 157 of *Contemporary Mathematics*, pages 103–112. AMS, 1994. Held in Como, Italy, June 15–19,1992.

[17] J. Mandel and M. Bresina. Balancing domain decomposition for problems with large jumps in coefficients. *Math.Comp.*, 65(216):1387–1401, 1996.

[18] M. Sarkis. Nonstandard coarse spaces and Schwarz methods for elliptic problems with discontinuous coefficients using non-conforming element. *Numerische Mathematik*, 77:383–406, 1997.

[19] M. Sarkis. A coarse space for elasticity: Partition of unity rigid body motions coarse space. In Z. D. et. al., editor, *Proceedings of the Applied Mathematics and Scientific Computing Dubrovnik, Croacia, June, 2001.* Kluwer Academic Press, 2002. To appear.

[20] M. Sarkis. Partition of unity coarse space and Schwarz methods with harmonic overlap. In L. F. Pavarino and A. Tosell, editors, *Recent Developments in Domain Decomposition Method*, pages 75–92. Springer-Verlag, 2002.

[21] R. Tezaur, P. Vanek, and M. Brezina. wo-level method for solids on unstructured meshes. Center for Computational Mathematics Report CCM TR 73, University of Colorado at Denver, 1995.

[22] P. Vaněk, J. Mandel, and M. Brezina. Algebraic multigrid by smooth aggregation for second and fourth order elliptic problems. *Computing*, 56:179–196, 1996.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

13. Algorithms and arteries: Multi-domain spectral/*hp* methods for vascular flow modelling

S.J. Sherwin[1] and J. Peiró[2]

1. Introduction. The growing interest in the mathematical and numerical modelling of biomedical systems and, in particular, the human cardiovascular system is supported by the numerous works which have appeared on the subject in recent years, for example [2, 9, 16, 21] and the references therein. Traditionally there has been a strong focus on low dimensional models [13]. However the association of vascular disease, such as atherosclerosis, with arterial branching has promoted an interest in the application of computational fluid dynamics (CFD) to vascular flow modelling. Nevertheless the nature of the flow presents a variety of challenges. Firstly, the flow is pulsatile and in a Reynolds number regime where the viscous and inertial effects are both significant. Secondly, the geometric characteristics of the vascular system are very intricate. Finally, blood is a non-Newtonian fluid and arterial walls are distensible.

A particular focus of the CFD modelling has been to determine the wall shear stress distribution of the unsteady flow at arterial junctions and bypass grafts. The sensitivity of the wall shear stress to surface curvature therefore make the geometric representation an important factor. The flows in and around regions of stagnation and separation are also of physiological interest. In an incompressible flow we know that the normal derivative of the wall normal flow is zero near the wall and therefore requires at least a second order approximation to be resolved. Both these factors and the requirement to reproduce the unsteady flow and its derivatives in complex geometries make high-order algorithms, such as the unstructured spectral/*hp* element method, suitable from the point of view of attaining a specified error at a lower computational cost. However the problem still poses many numerical challenges which have motivated a range of developments in spectral/*hp* element methods that we shall discuss in this review article.

Furthermore we cannot completely decouple the local branching flow at an arterial junction from the full vascular system. The flow waveform observed at a given location in the vascular tree is the result of changes in sectional area of the compliant vessels to accommodate the incompressible flow of blood as it is pumped from the heart. Starting at the heart, the arterial waves are propagated and reflected at each arterial branch [25] leading to a complex waveform which changes at different locations. Although the wavelengths of these waves are much larger than the length of local arterial branches, the flow waveform can be altered by the presence of disease or surgical intervention. Therefore there is an inherent need to include a multiscale modelling to the localised CFD as discussed in [16]. Within this context, the application of simplified models has been shown to provide useful information for practitioners at a reasonable computational cost [7].

In this paper we will briefly review three topics related to the application of spectral/*hp* discretisation to vascular flow modelling. In section 2 we review the work in [19] and discuss the one-dimensional full vascular tree modelling using one-dimensional equations. In section 3 we discuss the problem of generating high-order meshes to consistently model the arterial geometries based on the work in [22, 14]. Finally in section 4 we overview a recent development in elliptic preconditioning for unstructured model spectral/*hp* methods based on a low energy numerical basis which relates to the work of [1, 18].

2. Reduced 1D modelling of the human circulation. In this section we focus on the application of a one-dimensional model of blood flow in compliant vessels to study wave propagation in the arterial tree as previously detail in [19].

[1]Department of Aeronautics, Imperial College, London, U.K., s.sherwin@ic.ac.uk
[2]Department of Aeronautics, Imperial College, London, U.K., j.peiro@ic.ac.uk

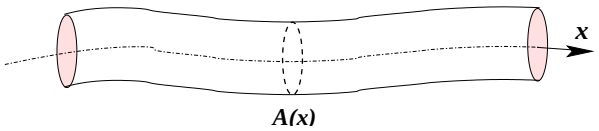


Figure 2.1: Simple compliant tube.

2.1. Governing equations. We consider a simple compliant tube, illustrated in figure 2.1, as a model of the artery. Following Brook et al. [3] we write the system of equations representing continuity of mass and momentum, for $a \leq x \leq b$ and $t > 0$, as

$$\frac{\partial \boldsymbol{U}}{\partial t} + \frac{\partial \boldsymbol{F}}{\partial x}(\boldsymbol{U}) = \frac{\partial}{\partial t}\begin{bmatrix} A \\ u \end{bmatrix} + \frac{\partial}{\partial x}\begin{bmatrix} Au \\ \dfrac{u^2}{2} + \dfrac{p}{\rho} \end{bmatrix} = \begin{bmatrix} 0 \\ -K_R u \end{bmatrix} \tag{2.1}$$

where the x is the axial direction, $A = A(x,t) = \int_S d\sigma$ is the area of a cross section S, ρ is the density of the blood which is taken to be constant, p is the internal pressure and $u(x,t)$ denotes the velocity of the fluid averaged across the section. The term K_R is a strictly positive quantity which represents the viscous resistance of the flow per unit length of tube. The unknowns in this system are p, A and u. Their number exceeds the number of equations and a common way to close the system is to explicitly provide an algebraic relationship between the pressure of the vessel p and the vessel area A. For example, by assuming *static equilibrium* in the radial direction of a cylindrical tube, one can derive a pressure relationship of the form

$$p = p_{ext} + \beta(\sqrt{A} - \sqrt{A_0}), \tag{2.2}$$

where

$$\beta = \frac{\sqrt{\pi} h_0 E}{(1-\nu^2)A_0}.$$

Here h_0 and $A_0 = A_0(x)$ denote the vessel thickness and sectional area, respectively, at the equilibrium state $(p,u) = (p_{ext}, 0)$, $E = E(x)$ is the Young modulus, p_{ext} is the external pressure, assumed constant and ν is the Poisson ratio. This ratio is typically taken to be $\nu = 1/2$ since biological tissue is practically incompressible.

2.2. Discontinuous Galerkin method. The wave propagation speeds in the large arteries are typically an order of magnitude higher than the average flow speeds. The characteristic speed of the system is also inherently subcritical and does not produce shock under physiological conditions. Therefore the numerical challenge is to propagate waves for many periods without suffering from excessive dispersion and diffusion errors. If the solution remains smooth then high-order methods are particularly attractive due to the fast convergence of the phase and diffusion properties with order of the scheme [17].

Following the work of Cockburn and Shu [4] we initially consider the one-dimensional hyperbolic system in conservative form (2.1) and assume that $R_K = 0$. To solve this system in a region $\Omega = [a,b]$ discretised into a mesh of N_{el} elemental non-overlapping regions $\Omega_e = [x_e^l, x_e^u]$, such that $x_e^u = x_{e+1}^l$ for $e = 1, \ldots, N_{el}$. We then proceed by constructing the weak form of (2.1), i.e.

$$\left(\frac{\partial \boldsymbol{U}}{\partial t}, \boldsymbol{\psi}\right)_\Omega + \left(\frac{\partial \boldsymbol{F}}{\partial x}, \boldsymbol{\psi}\right)_\Omega = 0 \qquad i = 1,2 \tag{2.3}$$

where $(\mathbf{u}, \mathbf{v})_\Omega = \int_\Omega \mathbf{u}\ \mathbf{v}\ dx$ is the standard $\mathbf{L}^2(\Omega)$ inner product. Decomposing the integral into elemental regions we obtain

$$\sum_{e=1}^{N_{el}} \left[\left(\frac{\partial \boldsymbol{U}}{\partial t}, \boldsymbol{\psi}\right)_{\Omega_e} + \left(\frac{\partial \boldsymbol{F}}{\partial x}, \boldsymbol{\psi}\right)_{\Omega_e} \right] = 0. \tag{2.4}$$

Integrating the second term in (2.4) by parts leads to

$$\sum_{e=1}^{N_{el}} \left(\frac{\partial \boldsymbol{U}}{\partial t}, \boldsymbol{\psi} \right)_{\Omega_e} - \left(\boldsymbol{F}, \frac{d\boldsymbol{\psi}}{dx} \right)_{\Omega_e} + [\boldsymbol{\psi} \cdot \boldsymbol{F}]_{x_e^l}^{x_e^u} = 0 \tag{2.5}$$

To form the discrete approximation of our problem we choose $\boldsymbol{U}$ to be in the finite space of $\mathbf{L}^2(\Omega)$ functions which are polynomials of degree P on each element. Furthermore we indicate an element of such space using the superscript δ. To attain a global solution in the domain Ω we need to allow information to propagate between the elemental regions. Information is propagated between elements by upwinding the boundary flux in the third term of equation (2.5). Denoting the upwinded flux as $\boldsymbol{F}^u$, the discrete weak formulation can now be written as

$$\sum_{e=1}^{N_{el}} \left(\frac{\partial \boldsymbol{U}^\delta}{\partial t}, \boldsymbol{\psi}^\delta \right)_{\Omega_e} - \left(\boldsymbol{F}(\boldsymbol{U}^\delta), \frac{d\boldsymbol{\psi}^\delta}{dx} \right)_{\Omega_e} + \left[\boldsymbol{\psi}^\delta \cdot \boldsymbol{F}^u \right]_{x_e^l}^{x_e^u} = 0, \tag{2.6}$$

Following the traditional Galerkin approach, we choose the test function within each element to be in the same discrete space as the numerical solution $\boldsymbol{U}^\delta$. At this point if we defined our polynomial basis and choose an appropriate quadrature rule we would now have a semi-discrete scheme. However, from an implementation point of view, the calculation of the second term can be inconvenient and consequently we choose to integrate this term by parts once more to obtain

$$\sum_{e=1}^{N_{el}} \left(\frac{\partial \boldsymbol{U}^\delta}{\partial t}, \boldsymbol{\psi}^\delta \right)_{\Omega_e} + \left(\frac{\partial \boldsymbol{F}(\boldsymbol{U}^\delta)}{\partial x}, \boldsymbol{\psi}^\delta \right)_{\Omega_e} + \left[\boldsymbol{\psi}^\delta \cdot \left[\boldsymbol{F}^u - \boldsymbol{F}(\boldsymbol{U}^\delta) \right] \right]_{x_e^l}^{x_e^u} = 0. \tag{2.7}$$

We note that the information between elements is transmitted by the third boundary term as the difference between the upwinded and the local fluxes, $\left[\boldsymbol{\psi}^\delta \cdot \left[\boldsymbol{F}^u - \boldsymbol{F}(\boldsymbol{U}^\delta) \right] \right]_{x_e^l}^{x_e^u}$.

To complete the discretisation we also require a time integration scheme and in the current implementation we have adopted a second order Adams-Bashforth scheme. The upwind flux is calculated using a straightforward upwinding of the characteristic variables as discussed in [19]. This type of upwinding process is used to impose the characteristic boundary conditions through the flux at the ends of the global domain Ω.

The 1D model of the compliant tube can be extended to handle the arterial tree by imposing suitable interface conditions at the bifurcations or branching points of the tree. At a bifurcation we have six degrees of freedom corresponding to the area and velocity conditions within each vessel. Therefore we require six equations to determine a unique solution. Applying the subsonic flow assumption we can determine the three characteristics entering the junction providing three equations. Finally, continuity of mass flux and total pressure at the bifurcation provide the three conditions required to close the system, see [19] for details.

2.3. Simulation of wave propagation in the arterial network. We have adopted the modifications proposed in [25] to the published models [26, 24] to compute the pulsatile one-dimensional blood flow through the arterial system using the discontinuous Galerkin method. The numerical values of the parameters of the arterial tree formed by the 55 main arteries is given in [19]. Figure 2.2 shows the connectivity of the arteries used in our model of the arterial network. The flow in the 55 arteries is assumed initially to be at rest. A periodic half sine wave is imposed as an input wave form at the ascending aorta (artery 1). Figure 2.2 also shows the inflow boundary conditions imposed at the ascending aorta and the time history graphs over a single cycle for three different arteries in the network: ascending aorta (artery 1), femoral artery (artery 46) and anterior tibial (artery 49).

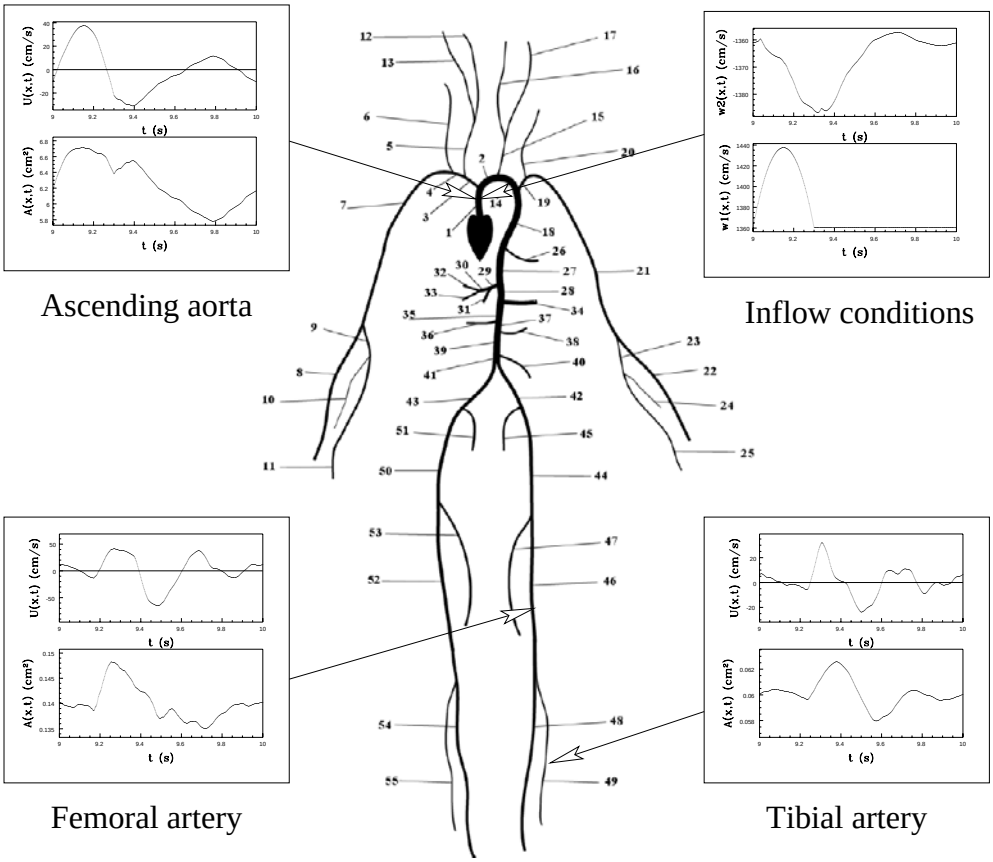


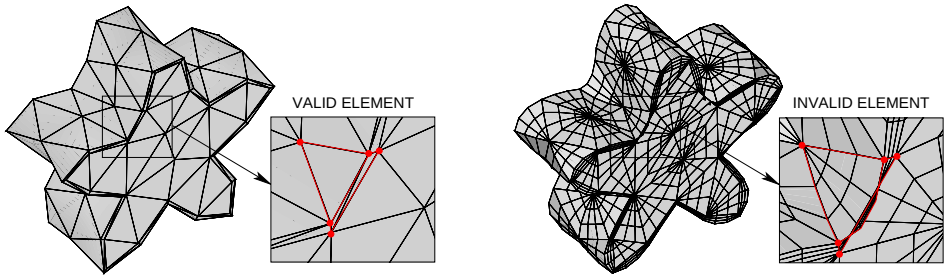
Figure 2.2: Connectivity of the 55 main arteries in the human arterial system.

The inclusion of resistance in the terminal arteries increases the number of waves in the system due to forward travelling waves being reflected at the terminal vessels and introduces backward travelling waves which are re-reflected at the bifurcations, hence a complex pattern of waves occurs in the network. Terminal resistance also creates regions of flow reversal due to the reflected velocity wave and increases in area as a result of the re-enforcing effect of the reflected pressure wave. It has also produced a waveform which includes a diacrotic notch in the ascending aorta (artery 1). These results are qualitatively similar to what we would expect to see from in-vivo measurements in the human body.

3. Geometric modelling of arterial branching. The ability to construct suitable computational meshes is currently a significant limiting factor in the development of high-order algorithms in very complex geometries. In this section we will address the issues encountered in applying the high-order finite element type approach to vascular flow modelling as previously discussed in [22, 14].

3.1. Mesh generation of high-order elements. The extension of standard unstructured mesh generation technology to high-order algorithms is a not trivial exercise. Complications arise due to the conflicting requirements to generate coarse meshes whilst maintaining good elemental properties in regions of high curvature. This is shown in figure

3.1 where we illustrate the type of invalid elements which can arise.



(a) Linear elements (b) High-order elements

Figure 3.1: The subdivision of a valid mesh of linear elements (a) to generate a high-order tetrahedral mesh (b) might lead to elemental regions with singular Jacobian mappings.

In our approach, the generation of an unstructured mesh of high-order spectral/*hp* elements is accomplished through the subdivision of a coarse mesh of linear elements. Given a surface representation in terms of cubic splines, the surface is initially discretised into a coarse distribution of linear surface elements. The local topology of these linear elements is influenced by the desire to include a boundary layer region or by taking into account surface curvature as described in section 3.3. The mesh generation then proceeds in a manner consistent with standard linear mesh generation process. Our current approach is based upon the method of advancing layers described in [15] but alternative mesh generation techniques can also be used. In this method the vertices of the original linear triangulation in the near-wall regions are assigned a direction and new interior vertices are created in successive layers up to a prescribed boundary layer thickness. These points are then linked to form a mesh of tetrahedral or prismatic elements, known as the boundary layer mesh. The rest of the domain is finally filled with a mesh of linear tetrahedra which, in our case, is generated by means of the advancing front technique.

A high-order surface discretisation is generated by following a "bottom-up" procedure where initially the triangular edges are discretised into $P+1$ points for a P^{th} order polynomial mesh. Subsequently the $(P-3)(P-2)/2$ points internal to the triangular faces are generated to complete the polynomial representation. The high-order point generation is typically performed in the parametric space of the surface splines which may have a non-isometric mapping to the physical space. In order to optimise the high-order element point distribution, a non-linear minimisation procedure is adopted, as discussed in [22], which generates the edge and face points as geodesics of the surface with a view to minimising the variation in the surface Jacobian.

3.2. Optimizing surface representation. To address the problem of obtaining an optimal distribution of points, consider a quadrature with N integration points and associated normalised weights z_i; $i = 1, \ldots, N$ $(-1 \leq z_i \leq 1)$ in a 1D interval $a \leq x \leq b$. It is known that the optimal positions x_i; $i = 1, \ldots, N$ of the points are given by

$$x_i = \frac{a}{2}(1 - z_i) + \frac{b}{2}(1 + z_i) \quad i = 1, \ldots, N. \tag{3.1}$$

This leads to an isometric mapping and therefore a constant Jacobian. The extension to elements with straight sides and faces is straightforward. However, a different strategy for

curved edges and faces is required to account for the distortion introduced by surface curvature.

A parametric surface is a mapping between a 2D parametric space and the 3D space. Mesh generation is considerably simplified if performed in the parametric space. However, approximating the element edges using the point distribution (3.1) along straight lines in the parametric plane might lead to highly deformed or invalid elemental regions if the surface mapping induces severe distortion.

An "optimal" point distribution can be obtained by recasting the problem as that of minimising the potential energy of a set of springs linking adjacent points. It is easily shown that the optimal distribution (3.1) is a minimum of the potential energy of such system of springs given by

$$\mathcal{J}(x_2, ..., x_{N-1}) = \sum_{i=1}^{N-1} \frac{(x_{i+1} - x_i)^2}{z_{i+1} - z_i} = \sum_{i=1}^{N-1} \frac{\delta_i^2}{z_{i+1} - z_i}. \tag{3.2}$$

This approach, unlike (3.1), is directly applicable to curved edge and faces on surfaces. A more detailed description of the procedure could be consulted in [22].

The high-order surface definition implies that the elements adjacent to a deformed wall will also have curved internal faces. These are constructed as a blend, consistent with the spectral/*hp* element expansion, between the internal straight edges and the deformed surface edge (see [11] for more details). In general, high-order elements allow for all internal face and edges to be deformed which, as discussed in the work of Dey *et al.* [5], may be necessary in very curved domains.

3.3. Curvature based mesh refinement for high-order elements. Mesh refinement as a function of the curvature has been proposed by several authors [8, 12] as a way to obtain an accurate piecewise linear approximation of a curved surface. In [22] we have shown that the use of curvature based refinement enhances the quality of the high-order meshes generated from linear tetrahedral and prismatic meshes. However, this criterion on its own is not sufficient to guarantee validity of all high-order elements as it does not account for the possible intersection of the boundary sides and faces with those on the interior. In [14] we have proposed an alternative method more suitable for the discretisation of boundary layers which we detail below.

A curve is locally approximated by a circle of radius R, the radius of curvature. We assume that the mesh spacing can be represented by a chord of length c in the circle and a spacing δ in the normal direction. In the modelling of viscous flows, the value of δ is usually prescribed to achieve a certain boundary layer resolution. The value of c is therefore chosen to guarantee that the osculating circle representing the curve does not intersect the interior sides of the elements, i.e. $\theta \geq 90^o$ for the triangular element. The value of c, which should be considered as a maximum mesh spacing, can now be obtained as a function of R and δ. Its value c_t for triangular elements is

$$c_t \leq R\sqrt{\frac{2\delta}{R+\delta}}. \tag{3.3}$$

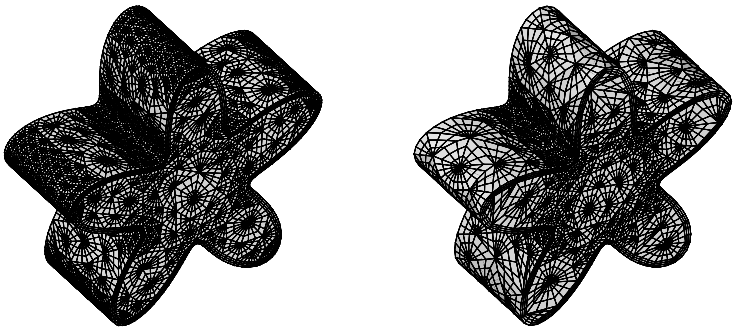
The corresponding value c_q for quadrilateral elements is

$$c_q \leq \frac{2R\delta}{R+\delta}\sqrt{1+\frac{2R}{\delta}}, \tag{3.4}$$

where the boundary displacement is assumed to be the same on either side of the rectangle. It is interesting to notice that, for a given δ, the quadratic element allows for a mesh spacing c_q which is about twice the value of spacing c_t for the triangular element.

The extension of this method to surfaces is straightforward. The refinement criterion given by formulas (3.3) and (3.4) is used for the two principal directions and the corresponding mesh spacings, c_1 and c_2, are calculated from the values of the principal curvatures $k_{1,2} = 1/R_{1,2}$.

An example of a hybrid mesh generated for the geometry previously considered in figure 3.1 and using the criterion (3.4), is shown in figure 3.2(a). This high-order mesh does not contain singular elements. However, the refinement applied here does not account for the sign of the surface curvature and the use of criterion (3.4) to ensure element validity is too restrictive in those regions where the domain is locally convex. For a convex region, the less restrictive criterion $\delta < R$ suffices to guarantee element validity. This is highlighted in figure 3.2(b) where the refinement criterion (3.4) has been selectively applied to concave regions only. The result is a valid mesh with fewer elements.



(a) Curvature refinement (b) Selective refinement

Figure 3.2: Curvature based mesh refinement for prismatic elements: (a) Refinement according to equation (3.4), (b) Refinement is applied to concave regions only.

As previously discussed our area of interest is the surgical intervention required when an artery becomes blocked, typically due to vascular disease, and the blockage is circumvented by an anastomosis. This procedure typically requires the construction of an alternative path normally using an autologous vein. A high percentage of long term failures of arterial bypass grafts are observed at the downstream, or distal, end of the bypass loop. Understanding the nature of this failure has made the geometric features of the bypass junction a particular focus of three-dimensional computational modelling. An example of a high-order mesh for an anatomically realistic geometry is shown in figure 3.3. This mesh has 1624 prismatic elements and 3545 tetrahedral elements. Also shown is the distribution of wall shear stresses calculated using a fourth-order polynomial approximation.

4. Low energy preconditioning for spectral/*hp* discretisations. To solve the fluid flow problem at arterial branches, as shown in figure 3.3 we have applied a three-dimensional unstructured spectral/*hp* element solver [20] with a high order splitting scheme of the Navier-Stokes equations [10]. The splitting scheme requires the solution of a Pressure Poisson equation and three Helmholtz problems. The iterative inversion of the discrete elliptic problems is currently the limiting factor in computational speed.

Building on the work of Bica [1], we have developed an efficient preconditioning strategy for substructured solvers based on a transformation of the closed form expansion basis to a "low energy" basis [18]. Following this approach, the strong coupling in the matrix system between two different degrees of freedom of the original basis is significantly reduced by

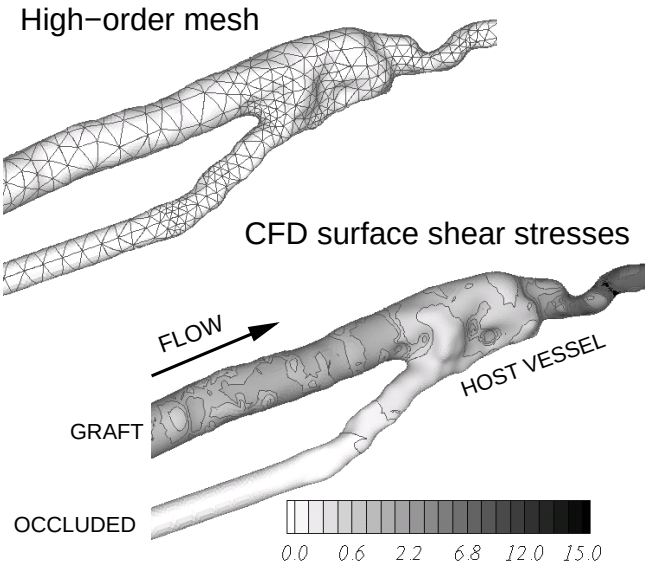


Figure 3.3: High-order mesh and distribution of surface shear stresses obtained using a fourth-order polynomial approximation in the spectral/*hp* CFD solver. The values of the shear stress have been normalized so that the inflow wall shear stress (Hagen-Poiseuille flow) is 1.

introducing a degree of orthogonality between degrees of freedom. The transformed matrix system is then amenable to block diagonal preconditioning.

The efficiency of the preconditioner is maintained by developing a new low energy basis on a symmetric reference element and ignoring, in the preconditioning step, the role of the Jacobian of the mapping from the reference to a global element. By applying an additive Schwarz block preconditioner to the low energy basis combined with a coarse space linear vertex solver we have observed up to six fold reductions in execution time for our complex geometry Navier-Stokes solver.

4.1. Overview. In this section we outline the key concepts behind the preconditioner. Full details of the formulation can be found in [18]. A representative elliptic boundary value problem is

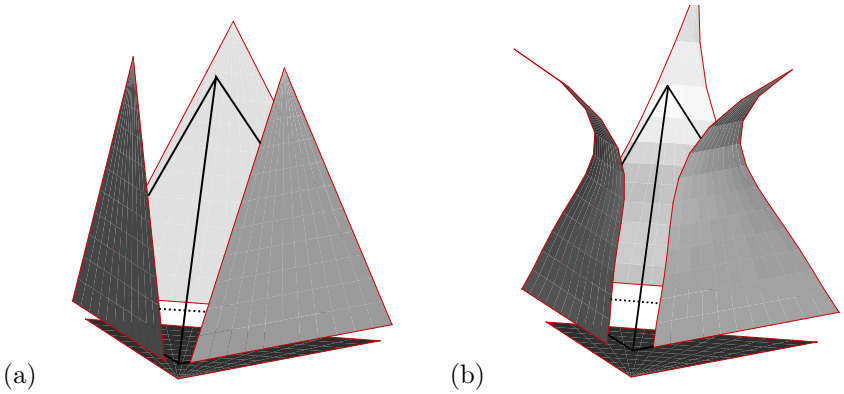
$$\nabla^2 u(x,y,z) + \lambda u(x,y,z) = f(x,y,z) \tag{4.1}$$

which is discretised into spectral/*hp* elements by decomposing the solution domain into non-overlapping subdomains within which a polynomial expansion is applied [11]. The Galerkin formulation of equation (4.1) leads us to a matrix problem of the form

$$\boldsymbol{H}\hat{\boldsymbol{u}} = \boldsymbol{f}$$

where $\boldsymbol{H}$ is the weak Helmholtz operator, $\boldsymbol{f}$ is the inner product of the forcing term and $\hat{\boldsymbol{u}}$ represent the expansion coefficients of the original closed form basis. In a spectral/*hp* element approach the expansion basis is normally decomposed into interior and boundary modes where the interior modes have zero support on the element boundaries and the boundary modes make the expansion complete. Such a decomposition lends itself to substructuring [23] where we construct the boundary degrees of freedom Schur complement $\boldsymbol{S}$ of $\boldsymbol{H}$. This is

essentially an orthogonalisation of the boundary degrees of freedom from the interior degrees of freedom and may also be considered as a basis transformation. This is attractive because it leaves a block diagonal matrix corresponding to the interior modes which is easily invertible. At this stage we still need to invert the positive definite Schur Complement $\boldsymbol{S}$ and this can be achieved using a preconditioned conjugate gradient technique.



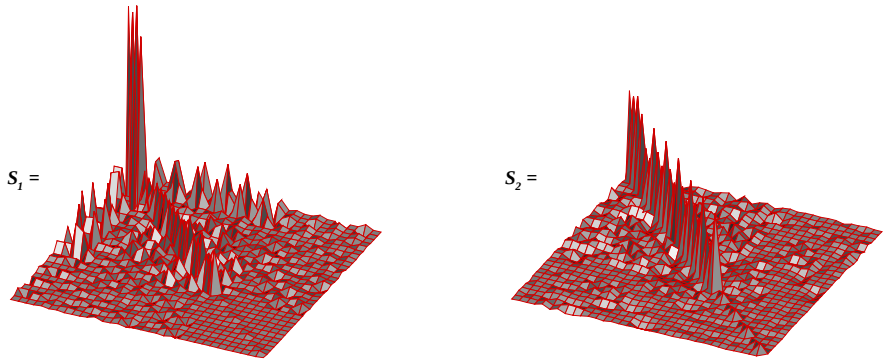
(a) (b)

Figure 4.1: Projected mode shape of vertex 4 (a) original and (b) low energy basis. The polynomial order was $P = 5$.

The choice of the preconditioner therefore defines the efficiency of the numerical algorithm. For two-dimensional hierarchical spectral/*hp* type discretisations the block diagonal preconditioner proposed by Dryja et al [6] leads to the attractive property of polylogarithmic conditioning. However for a three-dimensional hierarchical expansion this approach is not so effective [1, 18]. A significant factor is the coupling between the face expansions modes (i.e. the modes which have zero support on all edges and vertices) with the "wire-basket" space containing expansion modes which have support along the edges and at the vertices. The low energy preconditioning strategy transforms the original closed form bases to a numerically defined basis which decouples the degrees of freedom associated with each face from the vertex and edge degrees of freedom. In doing so the new basis has low energy in the sense that the inner product in the bilinear energy norm of two boundary modes is small or at least significantly reduced.

The formal details of transforming the basis are dealt with in [18]. However to illustrate the concepts we consider the shape of a vertex mode in the original and low energy basis at a polynomial order of $P = 5$ as shown in figure 4.1. The closed form original vertex mode is identical to the standard linear finite element mode and it can be appreciated that the energy associated with the inner product of this mode with any other mode in the energy norm will be reasonably high due to its high magnitude throughout the subdomain. Not too surprisingly, the shape of the low energy vertex modes decays rapid away from the vertex where it is required to have the same magnitude as the original basis. The rapid decay is consistent with the concept of low energy in the energy norms.

From an implementation point of view the numerical orthogonalisation of each of the face boundary modes from the wire-basket modes would be as difficult as inverting the full matrix. Nevertheless the important feature of the low energy basis can be captured by defining the new basis on a rotationally symmetric region. This inherently ignores the mapping from the symmetric region to the local element within the computational domain but maintains the computational efficiency of the standard implementation.



(a) (b)

Figure 4.2: Scatter plot of Schur complement matrices of a $P = 5$ polynomial expansion: (a) Original basis (b) Low energy basis (scaled by a factor of 4).

We conclude the summary by considering a scatter plot of the Schur complement systems arising from the original and low energy basis disretisation of a Poisson equation as shown in figure 4.2. In this figure the boundary modes were ordered so that the vertex modes were followed by the edges modes which, in turn, were followed by the face boundary modes. From this plot we see that original basis has a high magnitude/energy in the vertex modes even in the off-diagonal component. There is also a significant energy between the edges and vertices. Furthermore, we see that the coupling between the face and wire-basket modes is larger than the coupling between the face modes with themselves. The low energy basis on the other hand has a more diagonally dominated structure which makes it suitable for block diagonal preconditioning.

4.2. Result. Tests of regular elements [18], where the effect of ignoring the mapping of the elements is not significant, have demonstrated that a polylogarithmic scaling of the condition number is recovered when using the low energy basis preconditioner.

In figure 4.3(a) we shown a geometrically complex computational domain of practical interest. This problems originated from the reconstruction of the downstream junction of a porcine arterial bypass [14]. The domain consists of an unstructured triangular surface discretisation from which prismatic elements are constructed by extruding the triangular surface elements in the surface normal direction. The interior region is then discretised using tetrahedral subdomains. The discretisation shown in figure 4.3 consists of 749 prismatic and 1720 tetrahedral elements.

In this domain, we have solved a Poisson equation with Dirichlet boundary conditions corresponding to the solution $u(x, y, z) = \sin x \sin y \sin z$. The condition number of the diagonal and low energy preconditioned systems are shown in figure 4.3(b). This improvement in the condition number also reflected in the speed up of the back solve of the low energy preconditioner over the diagonal preconditioner. We have observed speed-ups of approximately 6 at a polynomial order of $P = 8$ and the break-even polynomial order was approximately $P = 3$.

5. Acknowledgments. This work was partially supported by the Smiths' Charity, and the Clothworkers' Foundation. The Imperial College centres for Biomedical Visualization and Parallel Computing provided computational resources.

The authors would like to thank Prof. Kim Parker of the Department of Bioengineering

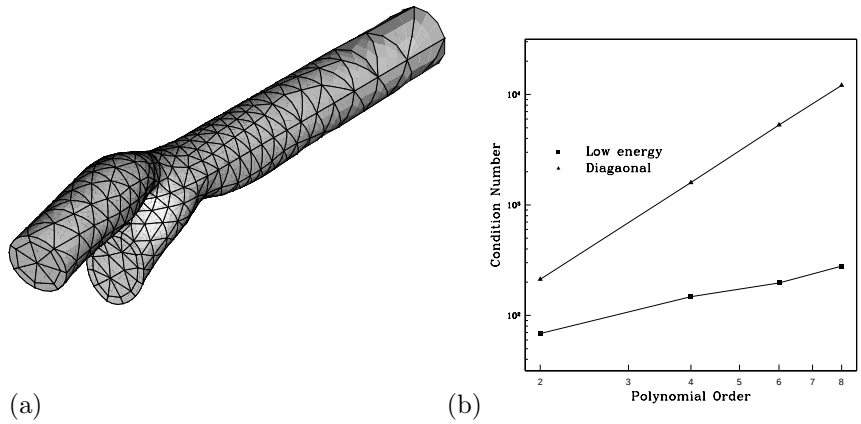


Figure 4.3: (a) Hybrid domain of a downstream arterial bypass graft. (b) Condition number as a function of polynomial order of the diagonal and low energy basis for a Poisson problem.

at Imperial College (London) and Prof. Luca Formaggia of EPFL (Lausanne) for many fruitful discussions on the modelling of wave propagation in the vascular system. The work described here has benefitted from the collaboration with Victoria Franke, Sergio Giordana and Dr. Denis Doorly of the Department of Aeronautics at Imperial College.

We also would like to acknowledge Mario Casarin for his direct contribution in the development of the low energy preconditioning methods.

REFERENCES

[1] I. Bica. *Iterative substructuring algorithms for the p-version finite element method for elliptic problems*. PhD thesis, Courant Institute, NYU, 1997.

[2] R. Botnar, G. Rappitsch, M. Scheidegger, D. Liepsch, K. Perktold, and P. Boesiger. Hemodynamics in the carotid artery bifurcation: A comparison between numerical simulations and in-vitro measurements. *J. of Biomech.*, 33:137–144, 2000.

[3] B. Brook, S. Falle, and T. Pedley. Numerical solution for unsteady gravity-driven flows in collapsible tubes: evolution and roll-wave instability of a steady state. *J. Fluid Mech.*, 396:223–256, 1999.

[4] B. Cockburn and C. Shu. TVB Runge-Kutta projection discontinous Galerkin finite element methods for conservation laws II general framework. *Math. Comm.*, 52:411–435, 1989.

[5] S. Dey, M. S. Shephard, and J. E. Flaherty. Geometry representation issues associated with p-version finite element computations. *Comp. Meth. Appl. Mech. Engng.*, 150:39–55, 1997.

[6] M. Dryja, B. F. Smith, and O. B. Widlund. Schwarz analysis of iterative substructuring algorithms for elliptic problems in three dimensions. *SIAM J. Numer. Anal.*, 31(6):1662–1694, December 1994.

[7] L. Formaggia, F. Nobile, A. Quarteroni, and A. Veneziani. Multiscale modelling of the circolatory system: a preliminary analysis. *Computing and Visualisation in Science*, 2:75–83, 1999.

[8] P. Frey and P.-L. George. *Maillages*. Editions Hermes, 1999.

[9] T. Hughes, C. Taylor, and C. Zarins. Finite element modeling of blood flow in arteries. *Comp. Meth. Appl. Mech. Eng.*, 158:155–196, 1998.

[10] G. Karniadakis, M. Israeli, and S. Orszag. High-order splitting methods for incompressible Navier-Stokes equations. *J. Comp. Phys.*, 97:414, 1991.

[11] G. E. Karniadakis and S. Sherwin. *Spectral/hp Element Methods for CFD*. Oxford University Press, 1999.

[12] C. K. Lee. On curvature element-size control in metric surface generation. *Int. J. Numer. Meth. Engng.*, 50:787–807, 2001.

[13] T. J. Pedley. *The fluid mechanics of large blood vessels*. Cambridge monographs on mechanics and applied mathematics. Cambridge University Press, 1980.

[14] J. Peiró, S. Giordana, C. Griffith, and S. Sherwin. High-order algorithms for vascular flow modelling. *Int. J. Num. Meth. Fluids*, 39, 2002.

[15] J. Peiró and A. I. Sayma. A 3-D unstructured multigrid Navier-Stokes solver. In K. W. Morton and M. J. Baines, editors, *Numerical Methods for Fluid Dynamics V*. Oxford University Press, 1995.

[16] A. Quarteroni, M. Tuveri, and A. Veneziani. Computational vascular fluid dynamics: Problems, models and methods. *Computing and Visualisation in Science*, 2:163–197, 2000.

[17] S. Sherwin. Dispersion analysis of the continuous and discontinuous Galerkin formulations. In *International Symposium on Discontinuous Galerkin Methods*, 1999. Newport, RI.

[18] S. Sherwin and M. Casarin. Low-energy basis preconditioning for elliptic substructured solvers based on unstructured spectral/*hp* element discretization. *J. Comp. Phys*, 171:394–417, 2001.

[19] S. Sherwin, L. Formaggia, J. Peiró, and V. Franke. Computational modelling of 1D blood flow with variable mechanical properties and its application to the simulation of wave propagation in the human arterial system. *Int J. Num. Meth. Fluids*, 2002. under review.

[20] S. Sherwin and G. Karniadakis. Tetrahedral *hp* finite elements: Algorithms and flow simulations. *J. Comp. Phys.*, 124:14–45, 1996.

[21] S. Sherwin, O. Shah, D. Doorly, J. Peiró, Y. Papaharilaou, N. Watkins, C. Caro, and C. Dumoulin. The inflence of out-of-plane geometry on the flow within a distal end-to-side anastomosis. *ASME J. Biomech.*, 122:1–10, 2000.

[22] S. J. Sherwin and J. Peiró. Mesh generation in curvilinear domains using high-order elements. *Int. J. Numer. Meth. Engng.*, 53:207–223, 2002.

[23] B. Smith, P. Bjorstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[24] N. Stergiopulos and D. Young. Computer simulation of arterial flow with applications to arterial and aortic stenoses. *J. Biomech.*, 25(12):1477–1488, 1992.

[25] J. Wang and K. Parker. Wave propagation in a model of the arterial circulation. Submitted to J. Biomech., 2002.

[26] N. Westerhof, F. Bosman, C. D. Vries, and A. Noordergraaf. Analog studies of the human systemic arterial tree. *J. Biomech.*, 2:121–143, 1969.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

14. Wave Propagation Analysis of Multigrid Methods for Convection Dominated Problems

W.L. Wan[1], T.F. Chan[2],

1. Introduction. The basic multigrid principle is that the smoother damps the oscillatory high frequency errors whereas the coarse grid correction reduces the smooth low frequency errors. However, this principle may not hold for convection dominated problems since the success of the standard techniques often rely on the intrinsic properties of elliptic PDEs, for instance, symmetry and positive definiteness, which are not generally true for convection dominated problems.

Several smoothing techniques have been proposed for convection dominated problems. One approach is to apply Gauss-Seidel with the so-called downwind ordering [3, 1, 6, 11, 16]. The idea is that the linear system given by the upwind discretization can be well-approximated by the lower triangular part if the unknowns are ordered according to the flow direction. Another approach is to use time-stepping methods as smoothers [7, 8, 10, 13]. The idea is that this class of smoothers do not just reduce the high frequency errors, but more importantly, also propagate the errors along the flow directions. Thus, the multigrid process can be interpreted as speeding up the error propagation by taking larger time step sizes on the coarse grids.

To analyze the efficiency of multigrid methods, one must then take into account the wave propagation property. In the classical Fourier-based analysis of multigrid methods [17], only the magnitude of the Fourier error components are considered, thus ignoring completely the phase angles which account for the wave propagation [15]. Gustafsson and Lötstedt [4, 12] first analyze the phase speed of this multigrid approach, and prove that a speedup of $2^K - 1$ is obtained using K grids for smooth errors. In this paper, we present a more refined phase velocity analysis which is able to explain the dispersive behavior of multigrid process that turns out to have significant influence on the convergence rate.

Phase velocity analysis is not just useful for analyzing the wave propagation multigrid approach but also applicable to explain the efficiency of other coarse grid correction methods as well. One common coarse grid correction approach is to use the discretization matrices as the coarse grid operators together with an exact coarse grid solve. It has been shown by Brandt and Yavneh [3] that the resulting coarse grid correction is not accurate for the Fourier components in the characteristic direction. Our phase velocity analysis not only recovers the same result, but also proves that coarse grid correction is only first order accurate for components in the cross-characteristic direction due to the phase shift error caused by the discretization coarse grid operators. Another approach is to use Galerkin coarsening [14, 19]. It turns out that its phase error is minimal, resulting in more accurate coarse grid correction.

In Section 3, explicit analytic formulae for the asymptotic expansion of the phase velocity of the different coarse grid correction approaches are established in one dimension. In Section 4, similar results in two dimensions are presented with the emphasis on Fourier components in the characteristic and cross characteristic directions. Numerical results are given in Section 5 to compare how these coarse grid correction approaches affect the actual multigrid convergence. Finally, concluding remarks are given in Section 6.

2. Model problem. The model problem we are interested in is the steady state solution of the convection-diffusion equation:

$$u_t - \epsilon \Delta u + w \cdot \nabla u = f \qquad x \in \Omega,$$

[1]University of Waterloo, jwlwan@math.uwaterloo.ca
[2]University of California, Los Angeles, chan@math.ucla.edu

subject to appropriate boundary conditions, and Ω is a d-dimensional unit cube. Here, we assume $\epsilon \ll 1$ and hence the equation is convection dominated. Discretizing the equation by finite difference methods on a standard uniform fine grid Ω^h with mesh size h results in a linear system

$$L^h u^h = f^h.$$

We consider solving the discrete problem using K grids, $\{\Omega^l\}_{l=0}^{K-1}$.

For pure hyperbolic equations, it is well-known that dissipation and dispersion are two fundamental quantities for analyzing numerical methods. Consider the 1D wave equation

$$u_t + a u_x = 0. \tag{2.1}$$

Given a finite difference scheme, suppose the Fourier transform of the numerical solution at time step $n+1$ can be written as

$$\hat{u}^{n+1}(\mu) = g(\mu)\hat{u}^n(\mu),$$

where $g(\mu)$ is the amplification factor. The scheme is dissipative if $|g(\mu)| < 1$, and it is dispersive if the phase speed [15], $\kappa(\mu)$, defined as,

$$\kappa(\mu) \equiv -\frac{\arg(g(\mu))}{\mu\pi\Delta t},$$

is different for different Fourier modes μ.

Thus, the classical multigrid analysis using Fourier analysis is deemed to be inadequate since it only considers the dissipation property. To give a more precise account of the wave propagation property of multigrid V-cycles, Gustafsson and Lötstedt [4, 12] analyzed the phase velocity of a two-grid iteration matrix M. Let $\hat{M}$ be its Fourier transform. It is well-known [5, 17] that $\hat{M}$ is block diagonal with 2×2 subblocks $\hat{M}_\mu$ where $\mu = 0, \ldots, N-1$.

Theorem 2.1 *Let λ_1 be the first eigenvalue of $\hat{M}_\mu$. For frequency $\mu \approx 0$,*

$$\lambda_1(\mu) = 1 - (\Delta t_h + \Delta t_H) i\mu\pi + O(\mu^2).$$

Consequently, the phase velocity of a two-grid method is

$$\kappa(\mu) = -\frac{\arg(\lambda_1(\mu))}{\mu\pi\Delta t_h} = 1 + \frac{\Delta t_H}{\Delta t_h} = 3.$$

The result can be generalized to K-level multigrid, in which case, $\kappa(\mu) = 2^K - 1$.

We remark that their analysis focuses primarily on the leading order terms of the asymptotic expansion of λ_1. If the initial wave consists of nonnegligible higher frequency modes, the effective speed of wave propagation is much slower than the analysis predicts. Figure 2.1 shows the propagation of a square wave by a three-level multigrid V-cycle on a grid with 128 grid points. It should have converged in $128/(2^3 - 1) \approx 36$ iterations; but instead, it takes more than 100 iterations due to numerical oscillations generated.

In the next section, we give a more detail analysis to explain the oscillation phenomenon. Furthermore, we consider two other coarse grid correction approaches and study their phase error behaviors for convection dominated problems in one and two dimensions. For all these approaches, we show that the convergence behavior of multigrid can be precisely described by the phase velocity analysis of the coarse grid correction matrix.

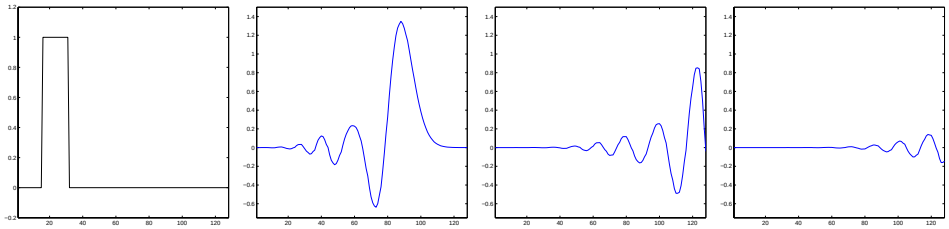


Figure 2.1: The numerical solutions given by a 3-level multigrid V-cycle at (a) iteration = 0, (b) iteration = 20, (c) iteration = 40, (d) iteration = 60.

3. One dimension. We start with 1D in which explicit formulae for the asymptotic expansion of the phase velocity can be established. The model problem becomes (assume ϵ is negligibly small):

$$u_x = f(x) \qquad -1 < x < 1,$$

with periodic boundary condition: $u(-1) = u(1)$. Without loss of generosity, we assume that $f^h \equiv 0$. Thus, we are interested in how the iteration error converges to zero. We shall consider three types of coarse grid correction approaches commonly used in the literature.

3.1. Inexact coarse grid correction. The coarse grid problem is solved inexactly by a few smoothing steps and the coarse grid operator is obtained by direct discretization. This is also the same approach considered by Gustafsson and Lötstedt [4], and others [7, 13]. Here, we extend the phase velocity analysis of Gustafsson and Lötstedt to include also the first correction term in the asymptotic expansions.

In a two-grid method consists of one pre-smoothing (one step of m-stage Runge-Kutta on the fine grid) followed by the coarse grid correction (one step of m-stage of Runge-Kutta on the coarse grid), the iteration matrix M of the two-grid method can be written as $M = CS$, where the coarse grid correction matrix C and smoothing matrix S are

$$\begin{aligned} C &= I + \sum_{j=1}^{m} \Delta t_H^j \, p \, (L^H)^{j-1} \prod_{k=m-j+1}^{m} (-\alpha_k) r L^h \\ S &= I + \sum_{j=1}^{m} \Delta t_h^j \, (L^h)^j \prod_{k=m-j+1}^{m} (-\alpha_k), \end{aligned}$$

where p is the linear interpolation and $r = \frac{1}{2} p^T$ its transpose.

Let $\hat{M} = \hat{C}\hat{S}$ be the Fourier transform of M. In two-grid analysis, it is customary to reorder the rows and columns of $\hat{M}$ ($\hat{C}$, $\hat{S}$) such that the low and high frequency modes are paired up; see [5, 17] for details. As a result, $\hat{M}$ ($\hat{C}$, $\hat{S}$) is block diagonal with 2×2 subblocks, $\hat{M}_\mu$, indexed by the wave numbers: $\mu = -N/2, \ldots, N/2 - 1$ corresponding to smooth or less oscillatory waves.

For easy exposition, we assume $m = 1$, and the coarse grid time step size, $\Delta t_H = \lambda H$. Then the coarse grid correction matrix can be simplified as:

$$C = I - \lambda H p r L^h,$$

Hence, the 2×2 subblocks of the Fourier transform of C are given by

$$\begin{aligned} \hat{C}_\mu &= I - \lambda H \hat{p}_\mu \hat{r}_\mu \hat{L}_\mu^h \\ &= I - \lambda H \begin{bmatrix} c_\mu^2 \\ -s_\mu^2 \end{bmatrix} \begin{bmatrix} c_\mu^2 & -s_\mu^2 \end{bmatrix} \frac{1}{h} \begin{bmatrix} 1 - e^{-\mu\pi h i} & 0 \\ 0 & 1 + e^{-\mu\pi h i} \end{bmatrix}. \end{aligned} \tag{3.1}$$

Since the high frequency errors have been reduced by the smoothing process, we are more interested in the "low-low" interaction, i.e. how the smooth waves are changed by the coarse grid correction. Hence, we focus just on the (1,1) entry of $\hat{C}_\mu$. By (3.1),

$$\hat{C}_\mu(1,1) = 1 - 2\lambda c_\mu^4(1 - e^{-\mu\pi hi}) \equiv |\hat{C}_\mu(1,1)|e^{-i\kappa(\mu)\mu\pi h\lambda}.$$

Here is our result on the dispersion of $\hat{C}_\mu(1,1)$ which is not considered explicitly by Gustafsson and Lötstedt [4, 12].

Theorem 3.1 *The dissipation and phase velocity of $\hat{C}_\mu$ are given, respectively, by*

$$\begin{aligned} |\hat{C}_\mu(1,1)| &\leq 1 \qquad \text{if and only if} \qquad 0 < \lambda \leq \frac{1}{2}, \\ \kappa(\mu) &= 2 + \frac{8\lambda - 15}{12}(\mu\pi h)^2 + O(\mu\pi h)^4. \end{aligned}$$

Proof. Express $\hat{C}_\mu(1,1)$ in terms of $s_\mu \equiv \sin(\mu\pi h)$ and $c_\mu \equiv \cos(\mu\pi h)$, and use the Taylor expansions of s_μ, c_μ and arctan. ∎

Remarks: (1) The coarse grid correction is dissipative. Moreover, the CFL condition on λ is more restrictive than the standard upwinding. (2) While the leading order term indicates propagation speed of 2 on the coarse grid, the negative second term shows that it is dispersive, which accounts for the oscillations observed in Figure 2.1.

Based on our more refined phase velocity analysis, convergence will be slowed down by oscillations unless the the smoother is extremely effective in damping most of the high frequency modes, for instance, by the use of artificial viscosity [13], or modified Runge-Kutta methods [7]. Otherwise, a fundamental change in the algorithm is needed to obtain a nonoscillatory multigrid method [9].

3.2. Exact coarse grid correction. In the second coarse grid correction approach, we consider exact coarse grid solve instead. Thus the coarse grid correction matrix becomes:

$$C = I - p(L^H)^{-1} r L^h.$$

As in the previous approach, we are interested in the low-low interaction, i.e. the (1,1) entry of $\hat{C}_\mu$.

Theorem 3.2 *The coarse grid correction of smooth waves given by the exact coarse grid solve together with linear interpolation is only first order accurate, i.e.*

$$|\hat{C}_\mu(1,1)| = \frac{\mu\pi h}{2} + O(\mu\pi h)^2.$$

Proof. By direct calculation,

$$\hat{C}_\mu(1,1) = 1 - c_\mu^3 e^{\mu\pi hi/2},$$

and the result follows by Taylor expansion. ∎

In the expression of $\hat{C}_\mu(1,1)$, the second term shows that after the coarse grid solve, the error is damped by c_μ^3, and more importantly, it has a phase error of $\mu\pi h/2$, implying that the coarse grid error is shifted precisely by 1/2 grid point (to the left). This shift arises from the discretization of the first order PDE on two different mesh sizes and consequently, leads to only first order accuracy in the coarse grid correction.

3.3. Galerkin coarse grid correction. Thirdly, we consider the use of Galerkin approach to form the coarse grid correction operator

$$G^H = rL^h p,$$

together with exact coarse grid solve. Then, the Fourier transform of C is given by

$$\hat{C}_\mu = I - \hat{p}(\hat{G}_\mu^H)^{-1}\hat{r}\hat{L}^h.$$

Theorem 3.3 *The coarse grid correction of smooth waves given by the exact coarse grid solve together with Galerkin coarse grid operator is third order accurate, i.e.*

$$|\hat{C}_\mu(1,1)| = \frac{1}{8}(\mu\pi h)^3 + O(\mu\pi h)^5.$$

Proof. Using the above formula, we obtain $\hat{C}_\mu(1,1) = (s_\mu^3 - ic_\mu^3)s_\mu^3/(c_\mu^6 + s_\mu^6)$. By Taylor expansion, we have

$$|\hat{C}_\mu(1,1)| = \frac{s_\mu^3}{\sqrt{c_\mu^6 + s_\mu^6}} = \frac{1}{8}(\mu\pi h)^3 + O(\mu\pi h)^5.$$

■

By a similar argument as before, it is easy to see from the expression of $\hat{C}_\mu(1,1)$ that the phase error $= O(\mu\pi h)^3$, which is negligibly small, and hence the coarse grid correction is much more accurate.

4. Two dimensions. The phase velocity analysis of Section 3.1 can be extended to 2D. Consider the convection dominated problem on a unit square:

$$-\epsilon\Delta u + a(x,y)u_x + b(x,y)u_y = f \qquad x \in \Omega = (-1,1)\times(-1,1),$$

with periodic boundary condition. In particular, we focus on two model problems:

(1) Entering flow (constant coefficient):

$$a(x,y) \equiv a, \quad b(x,y) \equiv b.$$

(2) Recirculating flow (variable coefficient):

$$a(x,y) = 4x(x-1)(1-2y), \quad b(x,y) = -4y(y-1)(1-2x).$$

We discretize the equation using the first order upwind scheme for the convection terms and center differencing for the Laplacian [3]. Our primary focus is on the limit $\epsilon \to 0$. We remark that some of the proves of the results in this section are similar to the 1D case and hence they are omitted.

4.1. Inexact coarse grid correction. The Fourier transform of the coarse grid correction matrix C is given by

$$\hat{C}_{\mu,\nu} \quad = \quad I - \lambda H \hat{p}_{\mu,\nu}\hat{r}_{\mu,\nu}\hat{L}^h_{\mu,\nu}.$$

where p is bilinear interpolation and r is full-weighting restriction. The (1,1) entry of $\hat{C}_{\mu,\nu}$ is then given by

$$\hat{C}_{\mu,\nu}(1,1) = 1 - 2\lambda h c_\mu^4 c_\nu^4 [\frac{a}{h}(1 - e^{-\mu\pi h i}) + \frac{b}{h}(1 - e^{-\nu\pi h i})].$$

To get more insight into the formula of $\hat{C}_{\mu,\nu}(1,1)$, we consider the special case where $a = b = 1$ and frequencies in the characteristic direction, i.e. $\nu = \mu$.

Theorem 4.1 *Assume $a = b = 1$. In the characteristic direction, i.e. $\nu = \mu$, the coarse grid correction is dissipative for $0 < \lambda \leq 1/4$, and dispersive, i.e.*

$$\begin{aligned} |\hat{C}_\mu(1,1)| &\leq 1 \qquad \text{if and only if} \qquad 0 < \lambda \leq \frac{1}{4}, \\ \kappa(\mu) &= 2 + (2\lambda - \frac{9}{4})(\mu\pi h)^2 + O(\mu\pi h)^4. \end{aligned}$$

Remark: the 2D approach is also dispersive, consistent with the 1D result.

As an example, we solve the model entering flow problem by multigrid, and snap shots of the errors in the first 15 V-cycles are shown in Figure 4.1. The mesh size is $h = 1/32$, and $\lambda = 0.25$. We observe that oscillations are generated at the tail as the square wave propagates from (-1,-1) to (1,1), which is justified by our phase velocity analysis. For the recirculating flow problem, Fourier analysis is not feasible, and yet we still observe a similar wave propagation phenomenon as in the entering flow problem.

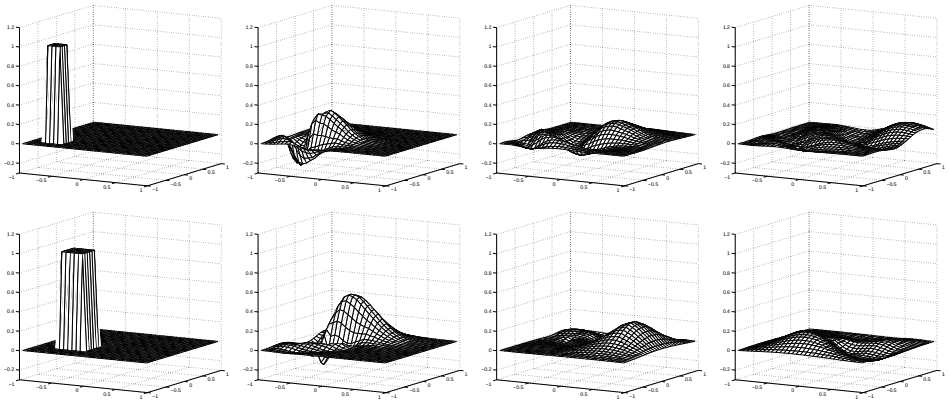


Figure 4.1: Numerical solutions given by a 3-level multigrid for (top row) the entering flow problem, and (bottom row) recirculating flow problem at iteration = 0, 5, 10, 15.

4.2. Exact coarse grid correction. With exact coarse grid correction and direct discretization for the coarse grid operator, the Fourier transform of the coarse grid correction matrix is

$$\hat{C}_{\mu,\nu} = I - \hat{p}_{\mu,\nu}(\hat{L}^H_{\mu,\nu})^{-1}\hat{r}_{\mu,\nu}\hat{L}^h_{\mu,\nu}.$$

Therefore,

$$\hat{C}_{\mu,\nu}(1,1) = 1 - c_\mu^4 c_\nu^4 \frac{\frac{a}{h}(1 - e^{-\mu\pi hi}) + \frac{b}{h}(1 - e^{-\nu\pi hi})}{\frac{a}{2h}(1 - e^{-\mu\pi 2hi}) + \frac{b}{2h}(1 - e^{-\nu\pi 2hi})}.$$

To facilitate understanding, we consider two special and yet important cases: frequency components in the characteristic direction, i.e. (μ, ν) such that

$$b\mu - a\nu = 0,$$

and, cross-characteristic direction [2, 3, 18], i.e. (μ, ν) such that

$$a\mu + b\nu = 0.$$

Theorem 4.2 *For the components in the characteristic direction and assuming $a = b$,*

$$|\hat{C}_{\mu,\nu}(1,1)| = \frac{\mu\pi h}{2} + O(\mu\pi h)^2.$$

For the components in the cross-characteristic direction and general a, b,

$$\lim_{\mu\to 0} \hat{C}_{\mu,\nu}(1,1) = \frac{1}{2}.$$

In particular, for $a = b$, then $\hat{C}_{\mu,\nu}(1,1) = 1 - c_\mu^6/2$.

Proof. In the characteristic direction, and $a = b$, then

$$\hat{C}_{\mu,\nu}(1,1) = 1 - c_\mu^7 e^{\frac{\mu\pi h i}{2}},$$

and hence

$$|\hat{C}_{\mu,\nu}(1,1)| = \frac{\mu\pi h}{2} + O(\mu\pi h)^2.$$

In the cross-characteristic direction, results follows from l'Hospital's rule. ■

We note that our analysis for the cross-characteristic direction is consistent with the result of Brandt and Yavneh [3] in which they consider the special case $b = 0$, and they point out that the coarse grid error is not a good approximation to the fine grid error for components in the cross-characteristic directions.

However, in both [3, 18], phase errors are not discussed. In the characteristic direction, the magnitude of the coarse grid error is in fact accurate: $|c_\mu^7 e^{\frac{\mu\pi h i}{2}}| = c_\mu^7$, but it has a phase error of $\mu\pi h/2$, just like the 1D case. Qualitatively speaking, the coarse grid error is shifted by $h/2$ in the characteristic direction, leading to the first order accuracy of $\hat{C}_{\mu,\nu}(1,1)$.

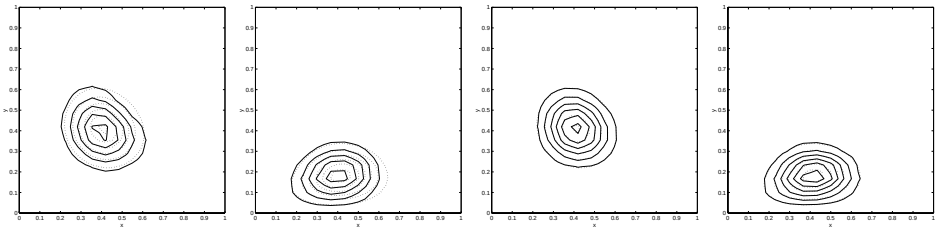


Figure 4.2: Contour plots of the fine grid error (dashed line) and the interpolated coarse grid error (solid line) for (a) the entering flow (exact coarse grid solve), (b) the recirculating flow (exact coarse grid solve), (c) the entering flow (Galerkin), (d) the recirculating flow (Galerkin).

Figure 4.2(a) and (b) show the contour plots of the fine grid error (dashed line) and the interpolated coarse grid error (solid line) for the entering flow and recirculating flow, respectively. Both results agree with the phase analysis that the interpolated coarse grid errors are shifted behind the directions of the flow.

4.3. Galerkin coarse grid correction. The Fourier transform is given by

$$\hat{C}_{\mu,\nu} = I - \hat{p}_{\mu,\nu}(\hat{r}_{\mu,\nu}\hat{L}^h_{\mu,\nu}\hat{p}_{\mu,nu})^{-1}\hat{r}_{\mu,\nu}\hat{L}^h_{\mu,\nu}.$$

We again consider the characteristic and cross-characteristic components.

Theorem 4.3 *For the components in the characteristic direction and assuming $a = b$,*

$$|\hat{C}_{\mu,\nu}(1,1)| = \frac{(\mu\pi h)^3}{8} + O(\mu\pi h)^5.$$

For the components in the cross-characteristic direction and general a, b,

$$\lim_{\mu\to 0} \hat{C}_{\mu,\nu}(1,1) = 0.$$

In particular, if $a = b$, then

$$|\hat{C}_{\mu,\nu}(1,1)| = \frac{(\mu\pi h)^2}{4} + O(\mu\pi h)^4.$$

In the Galerkin approach, the phase error in both directions is negligibly small as opposed to the exact coarse grid correction approach; see Figure 4.2(c) and (d). As a result, the coarse grid correction is second and third order accurate in the characteristic and cross-characteristic components, respectively.

5. Numerical results. In practice, the inexact coarse grid approach is appealing since it is simple and the same smoothing method can be used on all the coarse grids. However, such coarse grid correction is dispersive and oscillations generated delay multigrid convergence. In the exact coarse grid correction approach, the same smoothing method can also be used on all the coarse grids. Moreover, with exact coarse grid solve, the dispersive effect is much improved. However, the coarse grid correction is only first order accurate due to phase error, resulting in slower convergence. For the Galerkin approach, the coarse grid correction is more accurate, and hence the resulting multigrid convergence should be like the elliptic case.

We note that although our analysis suggests that the Galerkin approach has the least phase error, in practice, however, there are several drawbacks. It has been observed that the Galerkin coarse grid operator on the coarse grids become more and more like the central finite difference discretization. Operator-dependent interpolations may be needed to remedy the problems [19]. Another issue is extra storage for the coarse grid operators.

In the following, we compare the effects on the convergence of multigrid V-cycle by the inexact, nonGalerkin and Galerkin coarse grid correction approaches. The first example is the steady state solution of the one-dimensional linear wave equation:

$$u_t + u_x = 0.$$

First order Runge-Kutta method is used as the smoother for all the approaches with CFL number $\lambda = 0.5$. Linear interpolation and full weighting restriction are used between grids. The multigrid V-cycle iterations stop when the relative residual norm is less than 10^{-6}.

The number of multigrid V-cycles are shown in Table 5.1. To verify the results of the previous sections, we use two multigrid levels and consider a smooth initial guess and a square wave initial guess (in parenthesis). The results show that the number of multigrid V-cycles taken by the inexact coarse grid correction increases as mesh size decreases; thus we do not observe the classical mesh-independent convergence. Moreover, the convergence is slow due to the dispersion. Both exact and Galerkin coarse grid correction approaches, which use

h	Inexact	Exact	Galerkin
1/32	35 (31)	14 (13)	11 (8)
1/64	52 (44)	14 (9)	12 (5)
1/128	83 (73)	14 (6)	12 (3)
1/256	144 (141)	14 (5)	12 (3)

Table 5.1: Number of two-grid V-cycles for the 1D linear wave equation using inexact, exact, and Galerkin coarse grid corrections.

	Inexact					Exact				
h	2	3	4	5	6	2	3	4	5	6
1/32	31	35	39			13	25	34		
1/64	44	43	45	51		9	22	37	46	
1/128	73	52	58	61	61	6	13	31	49	55
1/256	141	83	64	72	72	5	9	19	40	59

Table 5.2: Number of multigrid V-cycles for the 1D linear wave equation using inexact and exact coarse grid corrections.

exact coarse grid solve, show much better convergence. Because of the shifting of the coarse grid error, the exact approach is not as efficient as the Galerkin approach.

In Table 5.2, it shows the multilevel results of the inexact and nonGalerkin coarse grid correction approach. The Galerkin approach requires different smoothing parameters on the coarse grids and hence it is not tested in this case. For the inexact coarse grid correction approach, the convergence should, in principle, have been improved by using more coarse grids based on the result of Gustafsson and Lötstedt (cf. Theorem 2.1). It is true when the mesh size is very small ($h = 1/256$) and hence the small wave number components are more dominant in the initial guess. But when the coarse grid gets smaller, the convergence starts to deteriorate. For the exact coarse grid correction approach, the multigrid convergence also starts to deteriorate on the coarser grids due to the phase shift of the coarse grid errors which is more serious with larger mesh size.

h	Inexact	Exact	Galerkin
1/32	28 (29)	13 (14)	7 (7)
1/64	41 (45)	13 (14)	5 (8)
1/128	70 (77)	11 (14)	5 (9)

Table 5.3: Number of two-grid V-cycles for the 2D entering flow problem using inexact, exact, and Galerkin coarse grid corrections.

We next consider the model entering flow and recirculating flow problems in two dimensions (cf. Section 4). Similar multigrid setting as in the one-dimensional case: Euler's smoothing, linear interpolation and full weighting restriction. Since the CFL number $\lambda = 0.25$ in two dimension, we use 2 presmoothing and postsmoothing steps instead.

The two-grid results are shown in Table 5.3 with a smooth initial guess and a square wave initial guess (in parenthesis). As in the 1D case, the multigrid convergence of the inexact coarse grid solve depends on the mesh size whereas the other two do not. Also, the Galerkin approach is more efficient than the exact coarse grid correction approach.

6. Conclusions. We have demonstrated that phase velocity analysis is a useful tool to analyze multigrid methods for convection dominated problems, and brings more insight into the efficiency of different coarse grid correction approaches.

For inexact coarse grid correction, the propagation of smooth waves is accelerated by using coarse grids. However, dispersion occurs in the coarse grid correction process which slows down substantially the multigrid convergence. The exact coarse grid correction approach does not rely on wave propagation and hence dispersion is not an issue. However, due to the use of the discretization matrix as the coarse grid operator, there is a phase error in the coarse grid solve which deteriorates the multigrid convergence. The Galerkin approach has the advantage of maintaining small phase shift error in the coarse grid correction. However, one needs to form the coarse grid operators on every grids, and hence to determine new sets of parameters, e.g. time-step size, for the smoother to obtain good smoothing efficiency.

We have addressed the issue of phase velocity analysis of multigrid methods for convection dominated problems. However, the design of new multigrid methods which possess good phase velocity property requires further investigation.

REFERENCES

[1] J. Bey and G. Wittum. Downwind numbering: Robust multigrid for convection-diffusion problems. *Appl. Numer. Math.*, 23:177–192, 1997.

[2] A. Brandt. Multi-level adaptive solutions to boundary value problems. *Numer. Math.*, 31:333–390, 1977.

[3] A. Brandt and I. Yavneh. Accelerated multigrid convergence and high-Reynolds recirculating flows. *SIAM J. Sci. Comput.*, 14:607–626, 1993.

[4] B. Gustafsson and P. Lotstedt. Analysis of the multigrid method applied to first order systems. In J. Mandel, editor, *Proceedings of the Fourth Copper Mountain Conference on Multigrid Methods*, pages 181–233, Philadelphia, 1989. SIAM.

[5] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, Berlin – Heidelberg – New York, 1985.

[6] W. Hackbusch and T. Probst. Downwind Gauss-Seidel smoothing for convection dominated problems. *Numer. Lin. Alg. Appl.*, 4:85–102, 1997.

[7] A. Jameson. Solution of the Euler equation for two dimensional transonic flow by a multigrid method. *Appl. Math. Comp.*, 13:327–355, 1983.

[8] A. Jameson. Computational transonics. *Comm. Pure Appl. Math.*, 41:507–549, 1988.

[9] A. Jameson and W. L. Wan. Monotonicity preserving and total variation diminishing multigrid time stepping methods. Technical Report CS-2001-11, Department of Computer Science, University of Waterloo, April 2001.

[10] D. Jespersen. A time-accurate multiple-grid algorithm. *AIAA paper 85-1493-CP*, 1985.

[11] K. Johannsen. Robust smoothers for convection-diffusion problems. Technical report, Institute for Computer Applications, University of Stuggart, 1999.

[12] P. Lotstedt and B. Gustafsson. Fourier analysis fo multigrid methods for general systems of pdes. *Math. Comp.*, 60:473–493, 1993.

[13] R. H. Ni. A multiple-grid scheme for solving the Euler equations. *AIAA*, 20:1565–1571, 1982.

[14] A. Reusken. Multigrid with matrix-dependent transfer operators for a singular perturbation problem. *Computing*, 50:199–211, 1993.

[15] R. Vichnevetsky and J. B. Bowles. *Fourier Analysis of Numerical Approximations of Hyperbolic Equations*. SIAM, Philadelphia, 1982.

[16] F. Wang and J. Xu. A crosswind block iterative method for convection dominated problems. *SIAM J. Sci. Comput.*, 21:620–645, 1999.

[17] P. Wesseling. *An Introduction To Multigrid Methods*. Wiley, New York, 1992.

[18] I. Yavneh. Coarse-grid correction for nonelliptic and singular perturbation problems. *SIAM J. Sci. Comput.*, 19:1682–1699, 1998.

[19] P. D. Zeeuw. Matrix-dependent prolongations and restrictions in a blackbox multigrid solver. *J. Comp. Appl. Math.*, 33:1–27, 1990.

Part II

Mini Symposium: Distributed Lagrange Multipliers for Domain Decomposition and Fictitious Domains

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

15. Numerical Simulation of The Motion of Pendula in an Incompressible Viscous Fluid by Lagrange Multiplier/Fictitious Domain Methods

L. H. Juárez[1], R. Glowinski[2]

1. Introduction. Lagrange Multiplier/Fictitious Domain Methods have proved to be effective in the direct numerical simulation of the motion of rigid bodies in incompressible viscous fluids [7]. In this work we discuss the application of this methodology, combined with finite element approximations and operator splitting, to the numerical simulation of the motion of pendula in a Newtonian incompressible viscous fluid. The pendula are circular cylinders constrained to move in a circular trajectory. The motion of the cylinders are driven only by the hydrodynamical forces and gravity.

In the present calculations we allowed solid surfaces to touch and penetrate, contrary to what was done in previous work where these methods were applied [7]. In fact, a good feature of this methodology is that the numerical solution does not break down when the rigid bodies overlap. On the other hand, numerical methods in which the computational domain is remeshed may break down when collision occurs, because this would break the lattice modeling of the fluid [8]. Hence a repulsive force between the particles need to be incorporated when they are close to each other to prevent contact between surfaces. In the numerical simulations in this work we did not introduce these artificial repulsive forces, in part because we wanted to investigate the solutions when the rigid bodies are near collision or when they actually collide and overlap. The mechanics of how solid particles in viscous liquids stick or rebound has not been fully understood and is still subject of current research. It has been demonstrated theoretically that when a perfect rigid sphere approaches a rigid wall its kinetic energy is dissipated by non-conservative viscous forces. The rate of close approach is asymptotically slow and the sphere do not deform or rebound [2]. By simultaneously accounting for elastic deformation of the body and viscous fluid forces, Davis *et al.* [3] showed that part of the incoming particle kinetic energy is dissipated by fluid forces and internal solid friction, and the rest is transformed into elastic-strain energy of deformation. Depending on the fraction of the kinetic energy that becomes stored as elastic-strain energy, the deformation of the spheres may be significant and rebound may occur. The relevant parameter for the bouncing transition, which is often obtained experimentally [9], is the *Stokes number*, which characterize the particle inertia relative to viscous forces. Numerical results of colliding bodies in viscous fluids may help to understand the mechanics of individual collisions in solid-liquid flows, which is an important issue in particulate multi-phase flow modeling and in the actual numerical computations of these flows. The numerical experiments in this work include the motion of a single pendulum, and the motion of two pendula. The two pendula case include different numerical experiments where the disks may have different densities and initial positions. An interesting study of pendula in viscous fluids with some applications can be found in [12] and references therein.

In Section 2 we describe the model for a single pendulum. A Lagrange Multiplier/Fictitious Domain equivalent formulation is presented in Section 3. The discretization of the resulting problem is discused in Sections 4 and 5. Numerical results and conclusions are given in Section 6.

2. Fluid-Rigid Body Interaction Model. We describe the model for the case of one pendulum in a viscous fluid. Its generalization to several pendula is straightforward. Let $\Omega \subset \mathbb{R}^2$ be a space region with boundary Γ, filled with an *incompressible viscous fluid*,

[1]University of Houston and UAM-I, hector@math.uh.edu, and hect@xanum.uam.mx
[2]University of Houston, roland@math.uh.edu

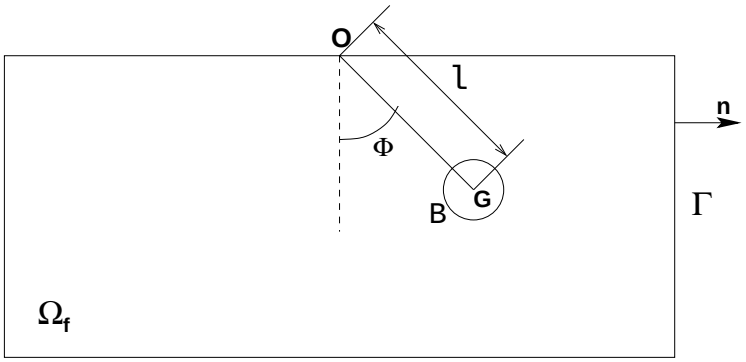


Figure 2.1: Pendulum in a viscous fluid

of density ρ_f, and that contains a rigid body B, with center of mass $\mathbf{G}$. The rigid body is constrained to move in a circular trajectory around the axis of rotation defined by a point $\mathbf{O}$, as shown in Figure 2.1. The position of the rigid body is known at any time t through the angle $\Phi = \Phi(t)$. We denote by $\mathbf{n}$ the unit normal vector pointing outward to the flow region $\Omega_f(t) = \Omega \setminus \overline{B}(t)$. Assuming that the only external force acting on the mixture is *gravity* (denoted by $\mathbf{g}$), in the vertical negative direction, the fluid flow is modeled by the *Navier-Stokes equations*

$$\rho_f \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u} \cdot \boldsymbol{\nabla})\mathbf{u} \right] = \rho_f \mathbf{g} + \boldsymbol{\nabla} \cdot \boldsymbol{\sigma} \ in \ \Omega_f(t), \tag{2.1}$$

$$\boldsymbol{\nabla} \cdot \mathbf{u} = 0 \ in \ \Omega_f(t), \tag{2.2}$$

where $\mathbf{u}$ denotes the velocity of the fluid, p is the fluid pressure, and $\boldsymbol{\sigma} = \boldsymbol{\tau} - p\mathbf{I}$ is the *stress-tensor*, with $\boldsymbol{\tau} = \mu_f(\boldsymbol{\nabla}\mathbf{u} + \boldsymbol{\nabla}\mathbf{u}^t)$ for a *Newtonian fluid* with viscosity μ_f. These equations are completed by some *initial conditions* and by the following *no-slip boundary conditions*: $\mathbf{u} = \mathbf{0}$ *on* Γ, and $\mathbf{u}(\mathbf{x}, t) = \mathbf{V}(t) + \boldsymbol{\omega}(t) \times \overrightarrow{\mathbf{G}(t)\mathbf{x}}$, $\forall \mathbf{x} \in \partial B(t)$. Here $\mathbf{V}(t)$ and $\boldsymbol{\omega}(t)$ are the translational and angular velocities of the rigid body, respectively. The motion of the rigid body B is modeled by the *Newton-Euler* equations:

$$M\frac{d\mathbf{V}}{dt} = M\mathbf{g} + \mathbf{F}, \tag{2.3}$$

$$\mathbf{I}\frac{d\boldsymbol{\omega}}{dt} = \mathbf{T}, \tag{2.4}$$

where M and $\mathbf{I}$ are the *mass* and *inertia tensor* of the rigid body, respectively. $\mathbf{F}$ is the resultant of the *hydrodynamical forces* acting on B, and $\mathbf{T}$ is the *torque* at $\mathbf{G}$ of the above hydrodynamical forces acting on B. The previous equations are completed by the kinematic equations $d\mathbf{G}/dt = \mathbf{V}$, $d\boldsymbol{\Phi}/dt = \boldsymbol{\omega}$, and by imposing initial conditions on $\mathbf{G}$, $\boldsymbol{\Phi}$, $\mathbf{V}$, and $\boldsymbol{\omega}$. Here we use the notation $\boldsymbol{\omega} = (0, 0, \omega)$, and $\boldsymbol{\Phi} = (0, 0, \phi)$. Finally, the above equations are simplified by using the constraint relation $\mathbf{V} = \boldsymbol{\omega} \times \overrightarrow{\mathbf{GO}} = l\,\omega\,(cos\phi,\ sin\phi)$.

3. Fictitious Domain Formulation with Distributed Lagrange Multipliers. To obtain this formulation we fill the rigid bodies with the surrounding fluid, and compensate the above step by introducing an "antiparticle" of mass $-M\rho_f/\rho_B$ and inertia $-I\rho_f/\rho_B$. Finally we impose the rigid body motion on $\overline{B}(t)$ via a Lagrange multiplier $\boldsymbol{\lambda}$ supported by $\overline{B}(t)$. We obtain, then, a flow problem over the entire region Ω, for which the

global variational formulation is: *For $t > 0$, find* $\mathbf{u}(t) \in (H_0^1(\Omega))^2$, $p(t) \in L^2(\Omega)$, $\omega(t) \in R$, $\phi(t) \in R$, $\boldsymbol{\lambda}(t) \in \Lambda(t) = (H^1(B(t)))^2$ *such that*

$$\begin{cases} \rho_f \int_\Omega \left[\frac{\partial \mathbf{u}}{\partial t} + (\mathbf{u}\cdot\boldsymbol{\nabla})\mathbf{u}\right]\cdot \mathbf{v} d\mathbf{x} - \int_\Omega p\boldsymbol{\nabla}\cdot\mathbf{v} d\mathbf{x} + \mu_f \int_\Omega \boldsymbol{\nabla}\mathbf{u} : \boldsymbol{\nabla}\mathbf{v} d\mathbf{x} + \\ (1-\frac{\rho_f}{\rho_B})(Ml^2+I)\frac{d\omega}{dt}\theta - \langle \boldsymbol{\lambda}, \mathbf{v} - \boldsymbol{\theta}\times\overrightarrow{\mathbf{Ox}}\rangle_{\Lambda(t)} = \\ \rho_f \int_\Omega \mathbf{g}\cdot\mathbf{v}\, d\mathbf{x} - (1-\frac{\rho_f}{\rho_B})M\mathbf{g}\, l\, sin\phi\, \theta,\ \forall \mathbf{v} \in (H_0^1(\Omega))^2,\ \ \forall \theta \in R, \end{cases} \tag{3.1}$$

$$\int_\Omega q\boldsymbol{\nabla}\cdot\mathbf{u}(t) d\mathbf{x} = 0,\ \forall q \in L^2(\Omega), \tag{3.2}$$

$$\langle \boldsymbol{\mu}, \mathbf{u}(t) - \boldsymbol{\omega}(t)\times\overrightarrow{\mathbf{Ox}}\rangle_{\Lambda(t)} = 0,\ \forall \boldsymbol{\mu} \in \Lambda(t), \tag{3.3}$$

$$\frac{d\phi}{dt} = \omega, \tag{3.4}$$

$$\omega(0) = \omega^0,\ \phi(0) = \phi^0, \tag{3.5}$$

$$\mathbf{u}(\mathbf{x}, 0) = \mathbf{u}_0(\mathbf{x})\ in\ \Omega,\ with\ \mathbf{u}_0(\mathbf{x}) = \boldsymbol{\omega}^0 \times \overrightarrow{\mathbf{Ox}}\ in \overline{B(0)}, \tag{3.6}$$

A natural choice for $< \cdot, \cdot >$ is defined by

$$< \boldsymbol{\mu}, \mathbf{v} >= \int_{B(t)} (\boldsymbol{\mu}\cdot\mathbf{v} + \delta^2 \boldsymbol{\nabla}\boldsymbol{\mu} : \boldsymbol{\nabla}\mathbf{v}) d\mathbf{x},\ \forall \boldsymbol{\mu},\ \mathbf{v} \in \Lambda(t), \tag{3.7}$$

with δ a *characteristic length* (the diameter of B, for example).

4. The Finite Element Discretization. We assume that Ω is a polygonal domain in R^2. Let $h(= h_\Omega)$ be a *space discretization step*, $\mathcal{T}_h$ a finite element triangulation of $\overline{\Omega}$, and P_s the space of polynomials in two variables of degree $\leq s$. The functional spaces $(H^1(\Omega))^2$ for velocity, and $L^2(\Omega)$ for pressure, are approximated by the following finite dimensional spaces

$$V_h = \{\mathbf{v}_h \in (C^0(\overline{\Omega}))^2 :\ \mathbf{v}_h|_T \in P_2 \times P_2,\ \forall T \in \mathcal{T}_h\}, \tag{4.1}$$

$$L_h^2 = \{q_h \in C^0(\overline{\Omega}) :\ q_h|_T \in P_1,\ \forall T \in \mathcal{T}_h\}, \tag{4.2}$$

respectively. The space $(H_0^1(\Omega))^2$ is then approximated by $V_{0h} = \{\mathbf{v}_h \in V_h : \mathbf{v}_h = \mathbf{0}\, on\, \Gamma\}$. This is the *Taylor-Hood* finite element approximation [13]. For the discretization of the Lagrange multipliers $\boldsymbol{\lambda}(t)$, we can approximate the functional spaces $\Lambda(t) = (H^1(B(t)))^2$ by a finite element on a grid defined on the rigid body $B(t)$. However we prefer the following alternative that is easier to implement: let $\{\mathbf{x}_i\}_{i=1}^N$ be a set of points from $\overline{B(t)}$ which cover $\overline{B(t)}$. We define

$$\Lambda_h(t) = \{\boldsymbol{\mu}_h : \boldsymbol{\mu}_h = \sum\nolimits_{i=1}^N \boldsymbol{\mu}_i \delta(\mathbf{x} - \mathbf{x}_i),\ \boldsymbol{\mu}_i \in \mathbb{R}^2,\ \forall i = 1, ..., N\}, \tag{4.3}$$

where $\delta(\cdot)$ is the Dirac measure at $\mathbf{x} = \mathbf{0}$. Then, instead of the scalar product of $(H^1(B_h(t)))^2$ we use $< \cdot, \cdot >_h$ defined by

$$< \boldsymbol{\mu}_h, \mathbf{v}_h >_h = \sum\nolimits_{i=1}^N \boldsymbol{\mu}_i \cdot \mathbf{v}_h(\mathbf{x}_i),\ \forall \boldsymbol{\mu}_h \in \Lambda_h(t),\ \mathbf{v}_h \in V_h. \tag{4.4}$$

This approach makes little sense for the continuous problem, but is meaningful for the discrete problem; it amounts to forcing the rigid body motion of $B(t)$ via a *collocation method*. A similar technique has been used to enforce Dirichlet boundary conditions by Bertrand *et al.* [1].

5. Time Discretization by Operator Splitting. After space discretization of (3.1)–(3.6) by the finite element method, we obtain an initial value problem of the form

$$\frac{d\varphi}{dt} + \sum_{i=1}^{4} A_i(\varphi, t) = f, \ \varphi(0) = \varphi_0, \tag{5.1}$$

where the operators A_i can be *multivalued*, and are associated to each of the following numerical difficulties: (i) the incompressibility condition and the related unknown pressure, (ii) an advection term, (iii) a diffusion term, (iv) the rigid body motion of the $B_h(t)$ and the related multipliers $\boldsymbol{\lambda}_h(t)$. The following fractional step method à la Marchuk-Yanenko [10] is used to solve this problem: *Given* $\varphi^0 = \varphi_0$, *for* $n \geq 0$, *compute* φ^{n+1} *from* φ^n *via*

$$\frac{\varphi^{n+i/4} - \varphi^{n+(i-1)/4}}{\Delta t} + A_i(\varphi^{n+i/4}, (n+1)\Delta t) = f_i^{n+1}, \quad i = 1, ..., 4, \tag{5.2}$$

with $\sum_{i=1}^{4} f_i^{n+1} = f^{n+1}$, and Δt a time discretization step. An application of this scheme to the finite element formulation of (3.1)–(3.6) results in the following equations: *Given* $\mathbf{u}^0 = \mathbf{u}_{0h}$, ϕ^0, ω^0, B^0, *for* $n \geq 0$, *knowing* $\mathbf{u}^n$, ϕ^n, ω^n, B^n, *compute* $\mathbf{u}^{n+1/4} \in V_{0h}$, *and* $p^{n+1/4} \in L^2_{0h}$ *via the solution of*

$$\begin{cases} \rho_f \displaystyle\int_\Omega \frac{\mathbf{u}^{n+1/4} - \mathbf{u}^n}{\Delta t} \cdot \mathbf{v} d\mathbf{x} - \int_\Omega p^{n+1/4} \boldsymbol{\nabla} \cdot \mathbf{v} d\mathbf{x} = 0, \ \forall \mathbf{v} \in V_{0h}, \\ \displaystyle\int_\Omega q \boldsymbol{\nabla} \cdot \mathbf{u}^{n+1/4} d\mathbf{x} = 0, \ \forall q \in L^2_h. \end{cases} \tag{5.3}$$

Compute $\mathbf{u}^{n+2/4} = \mathbf{u}(t^{n+1}) \in V_{0h}$, *where* $\mathbf{u}(t)$ *is the discrete solution of the following pure advection problem on* (t^n, t^{n+1})

$$\begin{cases} \displaystyle\int_\Omega \frac{\partial \mathbf{u}}{\partial t} \cdot \mathbf{v} d\mathbf{x} + \int_\Omega (\mathbf{u}^{n+1/4} \cdot \boldsymbol{\nabla}) \mathbf{u} \cdot \mathbf{v} d\mathbf{x} = 0, \ \forall \mathbf{v} \in V_{0h}, \\ \mathbf{u}(t^n) = \mathbf{u}^{n+1/4}. \end{cases} \tag{5.4}$$

Next, find $\mathbf{u}^{n+3/4} \in V_{0h}$ *by solving the diffusion problem*

$$\rho_f \int_\Omega \frac{\mathbf{u}^{n+3/4} - \mathbf{u}^{n+2/4}}{\Delta t} \cdot \mathbf{v} d\mathbf{x} + \mu_f \int_\Omega \boldsymbol{\nabla} \mathbf{u}^{n+3/4} : \boldsymbol{\nabla} \mathbf{v} d\mathbf{x} = \rho_f \int_\Omega \mathbf{g} \cdot \mathbf{v} d\mathbf{x}, \ \forall \mathbf{v} \in V_{0h}. \tag{5.5}$$

Now, predict the position and velocity of the rigid body by solving

$$\frac{d\omega}{dt} = -\frac{M\, g\, l\, sin\phi}{(M\, l^2 + I)}, \quad and \quad \frac{d\phi}{dt} = \omega, \tag{5.6}$$

on $t^n < t < t^{n+1}$, *with* $\phi(t^n) = \phi^n$, *and* $\omega(t^n) = \omega^n$. *Then set* $\phi^{n+3/4} = \phi(t^{n+1})$, *and* $\omega^{n+3/4} = \omega(t^{n+1})$. *Finally, we enforce the rigid body motion in the region* $B(t^{n+3/4})$ *by solving for* $\mathbf{u}^{n+1}$, $\boldsymbol{\lambda}^{n+1}$, *and* ω^{n+1} *the following equation*

$$\begin{cases} \rho_f \displaystyle\int_\Omega \frac{\mathbf{u}^{n+1} - \mathbf{u}^{n+\frac{3}{4}}}{\Delta t} \cdot \mathbf{v} d\mathbf{x} + (1 - \frac{\rho_f}{\rho_B})(M\, l^2 + I) \frac{\omega^{n+1} - \omega^{n+\frac{3}{4}}}{\Delta t} \theta = \\ < \boldsymbol{\lambda}^{n+1}, \mathbf{v} - \boldsymbol{\theta} \times \overrightarrow{\mathbf{Ox}}^{n+\frac{3}{4}} > \quad \forall \mathbf{v} \in V_{0h}, \ \forall \theta \in \mathbb{R}, \\ < \boldsymbol{\mu}_j, \ \mathbf{u}^{n+1} - \boldsymbol{\omega}^{n+1} \times \overrightarrow{\mathbf{Ox}}^{n+\frac{3}{4}} >= 0, \quad \forall \boldsymbol{\mu}_j \in \Lambda_h^{n+\frac{3}{4}}. \end{cases} \tag{5.7}$$

Problems (5.3) and (5.7) are finite dimensional linear saddle-point problems which are solved by an *Uzawa/conjugate gradient algorithm* [6]. The pure advection problem (5.4) is solved by the wave-like equation method discussed in Dean *et al.* [4] and [5]. Problem (5.5) is a discrete elliptic system whose iterative or direct solution is a quite classical problem. In this work all the linear systems are solved by a sparse matrix algorithm based on Markowitz' method [11]

6. Numerical Experiments and Conclusions. We consider a two-dimensional rectangular domain $\Omega = (-3, 3) \times (-1, 1)$ filled with a viscous fluid of density $\rho_f = 1$. The axis of rotation of the pendula is fixed at $\mathbf{O} = (0, 1)$, and the diameter of the circular rigid bodies is 0.25 in all cases bellow.

As a test case we consider one pendulum with a circular rigid body of density $\rho_B = 3$ released from rest at $\phi^0 = 1.4$ radians in a liquid of viscosity $\mu_f = 0.005$. We solved this problem using two meshes: an unstructured mesh (Fig. 6.1) which takes advantage that we know in advance the possible trajectory of the rigid body, and a uniform mesh with space discretization step $h = 1/64$. Figure 6.2 shows the comparison of the time history of the angle and of the angular velocity obtained with the two meshes. The agreement is satisfactory. As expected, the pendulum exhibits damped oscillations around the vertical position, and it goes to a steady position as time increases. The maximum Reynolds number obtained (based on the maximum falling velocity and diameter of the circular rigid body) was 835. Since the unstructured mesh has much less velocity degrees of freedom than the regular mesh (13823 versus 49665), we used the unstructured mesh in the subsequent calculations.

As a second example we consider two pendula. One pendulum with a circular rigid body of density $\rho_1 = 1.1$ is initially hold in the vertical position $\phi_1^0 = 0$, and the other pendulum with density $\rho_2 = 5$ is released from rest at $\phi_2^0 = 1.4$ radians in a liquid of viscosity $\mu_f = 0.005$. Figure 6.3 shows that, after a short time, the heavier cylinder collides with the lighter fixed body. After collision the two bodies move together as a single body all the time. This is more evident in in Figure 6.4 where the time history of the angle, angular velocity, and separation distance is shown. The maximum Reynolds number in this case was 1,085. We expected the two bodies to separate after they reach the maximum negative angle since the heavier rigid body is below to the lighter one at that position, and the action of gravity is stronger on the heavier body. However they never separate after collision. The only forces in our model problem that can prevent separation after collision are the viscous forces which in this case seem to dominate. To corroborate this strong dependence from viscous effects, we reduced μ_f from 0.005 to 0.001 and repeated the numerical calculation. Figure 6.5 shows that this time, after the two bodies collide, they stick together until they reach the maximum negative angle (where the angular velocity is close to zero), and then separate when they start to move in the counterclockwise direction by the action of gravity. This is clearly shown in Figure 6.6 where we plot the time history of angle, angular velocity, and separation distance. The maximum Reynolds number this time was 5,800. It is evident that a more detailed study of this and related phenomena is needed in order to better understand the mechanics of particle collision in viscous liquids and to generate models that simulate more accurately solid-liquid particulate flows which are very important in applications.

REFERENCES

[1] F. Bertand, P. A. Tanguy, and F. Thibault. A three-dimensional fictitious domain method for incompressible fluid flow problems. *Int. J. Num. Meth. Fluids*, 1997.

[2] R. E. Cox and H. Brenner. The slow motion of a sphere through a viscous fluid towards a plane surface–II. Small gap widths including inertial effects. *Chem. Engng. Sci.*, 22:1753, 1967.

[3] R. A. Davis, J. M. Serayssol, and E. J. Hinch. The elastohydrodynamic collision of two spheres. *J. Fluid Mech.*, 163:479–497, 1986.

[4] E. J. Dean and R. Glowinski. A wave equation approach to the numerical solution of the Navier-Stokes equations for incompressible viscous flow. *C.R. Acad. Sc. Paris*, 325(Serie 1):783–791, 1997.

[5] E. J. Dean, R. Glowinski, and T. W. Pan. A wave equation approach to the numerical simulation of incompressible viscous flows modeled by the Navier-Stokes equations. In J. A. D. Santo, editor, *Mathematical and Numerical Aspects of Wave Propagation*, pages 65–74, Philadelphia, PA, 1998. SIAM.

[6] R. Glowinski and P. Le Tallec. Augmented Lagrangian and operator splitting methods in nonlinear mechanics. In T. F. Chan, R. Glowinski, J. Périaux, and O. B. Widlund, editors, *Proc. 2nd International Symposium on Domain Decomposition Methods*, Philadelphia, 1989. SIAM.

[7] R. Glowinski, T.-W. Pan, T. I. Hesla, D. D. Joseph, and J. Periaux. A fictitious domain approach to the direct numerical simulation of incompressible viscous flow past moving rigid bodies: Application to particulate flow. *J. Comput. Phys.*, 169:363–426, 2001.

[8] H. H. Hu, D. D. Joseph, and M. Crochet. Direct simulation of fluid particle motions. *Theoret. Comput. Fluid Dynamics*, 3:285–306, 1992.

[9] G. Joseph, R. Zenit, M. Hunt, and A. Rosenwinkel. Particle-wall collision in a viscous fluid. *J. Fluid Mech.*, 433:329–346, 2001.

[10] G. I. Marchuk. *Handbook of Numerical Analysis, Splitting and alternating direction methods*, volume I. Elsevier Science Publishers B.V., North-Holland, Amsterdam, 1990.

[11] S. Pissanetzky. *Sparse Matrix Technology.* Academic Press, 1984.

[12] Y. Roux, E. Rivoalen, and B. Marichal. Vibrations induites d'un pendule hydrodynamique. *C. R. Acad. Sci. Paris*, 328(Série II b):479–497, 2000.

[13] C. Taylor and P. Hood. A numerical solution of the Navier-Stokes equations using the finite element method. *Comput. & Fluids*, 1:73–100, 1973.

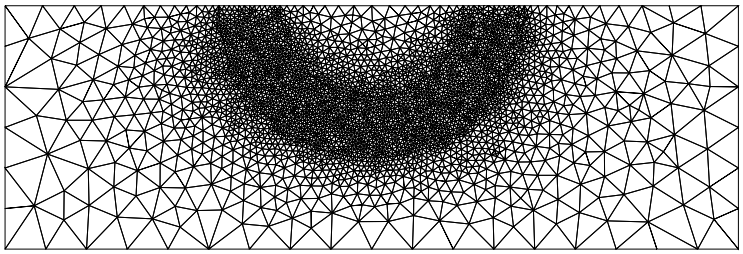


Figure 6.1: The unstructured mesh

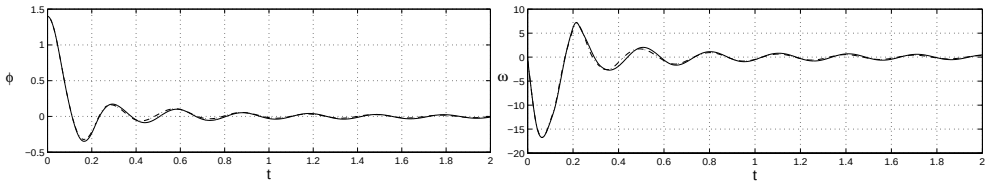


Figure 6.2: Comparison of the time history of the angle (left) and of the angular velocity (rigth) obtained with the unstructured mesh (dashed line) and the regular mesh (continuous line) for one pendulum

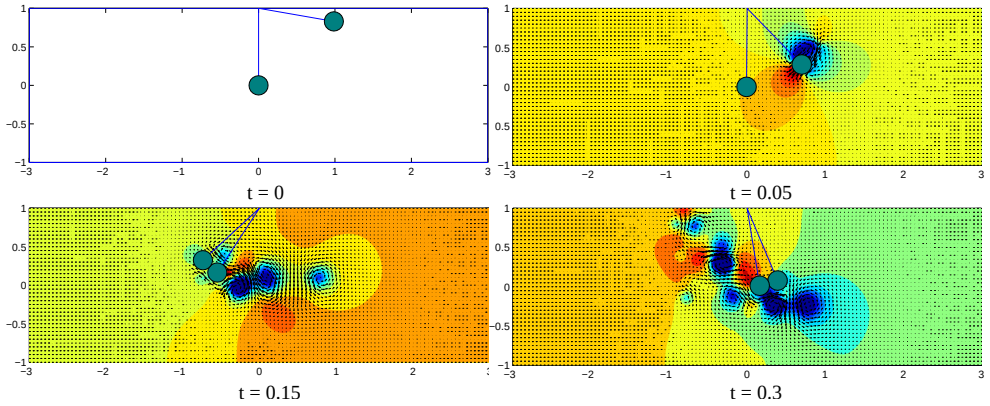


Figure 6.3: Velocity vector field and pressure at different times for the two pendula with $\mu_f = 0.005$, $\rho_1 = 1.1$, and $\rho_2 = 5$.

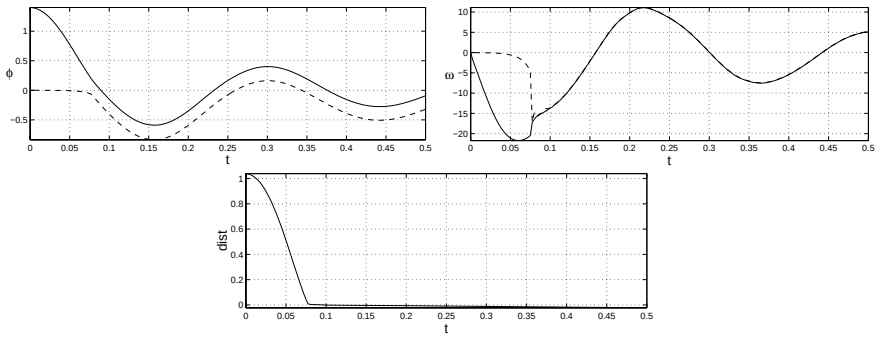


Figure 6.4: Time history of the angle (top left), angular velocity (top right), and separation distance (bottom) of the two pendula with $\mu_f = 0.005$, $\rho_1 = 1.1$, and $\rho_2 = 5$.

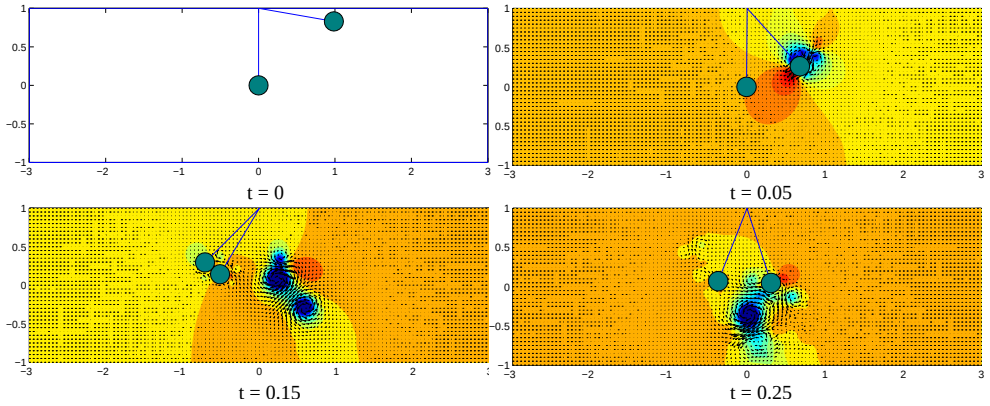


Figure 6.5: Velocity vector field and pressure at different times for the two pendula with $\mu_f = 0.001$, $\rho_1 = 1.1$, and $\rho_2 = 5$.

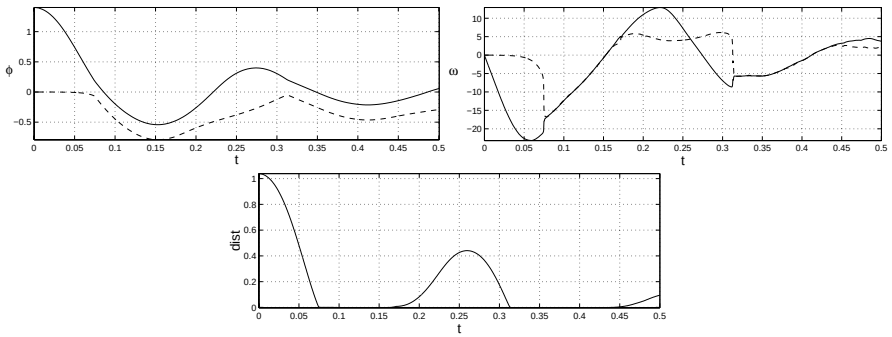


Figure 6.6: Time history of the angle (top left), angular velocity (top right), and separation distance (bottom) of the two pendula with $\mu_f = 0.001$, $\rho_1 = 1.1$, and $\rho_2 = 5$.

Part III

Mini Symposium: On FETI and Related Algorithms

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

16. Modifications to Graph Partitioning Tools for use with FETI methods

M.K. Bhardwaj[1], D.M. Day[2]

1. Introduction and Summary. Engineering solutions are presented for certain massively parallel implementation issues associated with FETI domain decomposition methods [2]. A wrapper around a graph partitioner is defined so that a computational domain is decomposed into subdomains that may be used with FETI methods. The techniques described here may find use with other domain decomposition methods for structural dyamics in which the subdomain matrices are factored. Our solution methodology is imperfect, but it is the most robust way known to the authors to use an off the shelf graph partitioner with FETI methods.

A unique aspect of finite element methods in structual dynamics is the variety of elements combined in a model. FETI methods employ partitions of the dual or element connectivity graph. This article contributes a set of weights depending on the element type that improve load balance with FETI methods.

A serious problem with FETI methods is that incompletely connected subdomains result in subdomain mechanisms that are difficult to characterize geometrically. A general definition of element connectivity is given, and used in a post process of the partition that further decomposes each subdomain into its connected components.

The discussion is organized as follows. The remainder of this section reviews certain relevant aspects of structural dynamics and describes the model problems. Section two concerns element weights. The resulting load imbalance is presented for the more problematic of the two models. The third section addresses subdomain mechanisms and connectivity. Numerical results and conclusions are presented in section four. Numerical examples are also integrated into the exposition.

The United States Department of Energy (DOE) has supported work at Sandia National Laboratory on full systems analysis. The goal is to simulate designs of applications of interest to DOE using massively parallel platforms. The development of hardware, software and algorithms for these tasks is challenging.

A design cycle of component models culminates in an evaluation based on an analysis of a few hundred of the smallest eigenvalues and eigenvectors. The first step in our design cycle is to generate a conforming mesh for the full system using a commercial mesh generation package such as Patran or Sandia's Cubit framework. Second the mesh is partitioned using Chaco. Parallel graph partitioning packages such as including METIS [4] and Zoltan are available. Graph partitioners have also been developed specifically for use with FETI methods (e.g. TopDomDec). Our comments apply to all of these tools. A finite element code is used to build matrices, such as Salinas. The inverted generalized symmetric semi-definite eigenvalue problem is solved using PARPACK. FETI methods are used to solve the resulting sequence of linear systems [1].

Graph partitioning software packages routinely determine partitions in which processor loads vary by less than one tenth of one percent. Domain decomposition algorithms have more specific requirements on a partition than are addressed by graph partitioners. For FETI methods the weight of a subdomain depends primarily on the number of nonzeros in the Cholesky factor of the stiffness matrix, a nonlinear objective function. For FETI-DP methods, another critical variable is the size of the resulting coarse grid linear system. The techniques described here significantly reduce the size of the FETI-DP coarse grid linear

[1]Sandia National Labs, mkbhard@sandia.gov
[2]Sandia National Labs dmday@sandia.gov

system. The interface size and roughness are secondary contributions.

Another set of problems stem from the linear elasticity equation. In the dual formulation, FETI-1 ([3]), the subdomain stiffness matrices are singular. A six dimensional null space may be determined from the geometry (coordinates) for each face connected component (defined in section 3) of the subdomain. In a subdomain that is not face connected, it is possible for the Cholesky factorization routine to incorrectly reveal the null space. The problem persists in dual-primal methods, and is addressed through sophisticated corner node selection methods. As the number of processors increases, the probability that an off the shelf graph partitioning tool will introduce mechanisms also increases.

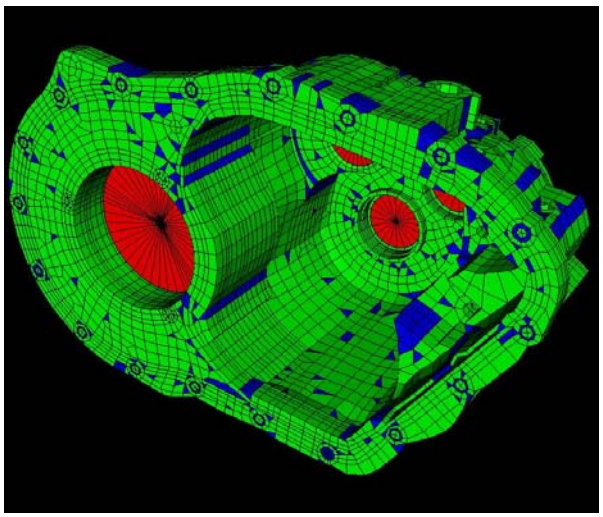


Figure 1.1: Engine model

Model problem one is an engine manifold (see Figure 1.1) and model problem two is the electronics package from a structure of interest at Sandia (see Figure 1.2. For both problems PARPACK needs to solve 31 linear systems in order to approximate the ten smallest modes. The engine model has 203894 nodes (three unknowns per node) and 193960 elements. Most of the elements are eight node hexagons, and the other elements are six node wedge elements and six node triangles. The computations on the engine model were performed on the ASCI Red platform (see http://www.sandia.gov/ASCI/Red/). The component model has 248226 nodes (three unknowns per node) and 167928 elements. The elements are six node triangles and ten node tetrahedrons. Computations with the component model were performed on the CPLANT platform (see http://www.cs.sandia.gov/cplant/), the worlds fastest Linux cluster. The CPLANT platform is composed of 1536 Compag DS10L 1U servers connected via Myrinet networking hardware.

2. Elements and Weights. Finite element models of aerospace structures routinely contain many different element types. The ratio of unknowns to elements is asymptotically constant on homogeneous submeshes with simple topologies. Unfortunately the load balance problem for FETI methods is not linear, depending on the number of nonzeros in the Cholesky factor of each subdomain matrix.

The nonlinear load balance problem is addressed by the selection of element weights. Initially a nearby linear problem is solved. The asymptotic ratio of unknowns to elements for a regular mesh is used as an initial guess for the element weight. The weights were then

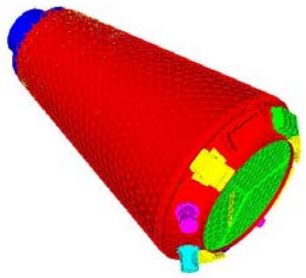


Figure 1.2: Component model

calibrated on a few model problems.

Structural dynamics models make use of a one, two and three dimensional elements. One dimensional elements, including truss and bar elements, are all converted to beam elements in our finite element code. The shell elements are quadrilateral and triangular. Beam and shell elements have six unknowns per node. The additional drilling degrees of freedom at each node are essential for maintaining subdomain connectivity. The solid elements are hexagonal, prismatic or tetrahedral, and all have three unknowns per node. An element may have nodes only at the vertices (linear shape functions) or nodes at both the vertices and the midpoints of the edges (quadratic shape functions).

Element weights balancing the number of subdomain unknowns for models with regular meshes are known. The element weight is the ratio of the number of unknowns to the number of elements. In two dimensions the number of knowns is asymptotically equal to the sum of the unknowns per node and three times the number of nodes per edge (see §1.9 of [5]). There are similar formulas for solid elements. These element weights vary by an order of magnitude.

Balancing the unknowns per subdomain does not solve the load balance problem for FETI methods. Subdomains consisting of irregular solid elements may have Cholesky factors with relatively large numbers of nonzeros. An example of such a subdomain is presented in Figure 2.1. Furthermore a subdomain that consists entirely of shell elements usually comes from a two dimensional subcomponent (e.g. an aero-shell); such a subdomain has a relatively sparse Cholesky factorization and a one dimensional boundaries.

The weights of the solid elements have been experimentally increased to decrease the load imbalance. One set of sub-optimal weights are used for all models. Reports of load imbalance problems ceased once the graph partitioner was modified to use the weights listed in 2.1.

The load balance for the component model partitioned into 540 subdomains using the weights is depicted in Figure 2.2. The data for partitions into 137 and 277 is similar. In each case the ratio of the maximum to the average for *both* unknowns and nonzeros is 3/2;

The subdomain stiffness matrices with the most nonzeros still correspond to irregularly meshed solid elements. The large spread in the number of nonzeros in the Cholesky factorization represented how inexactly the nonlinear load balance problem is solved. The processors

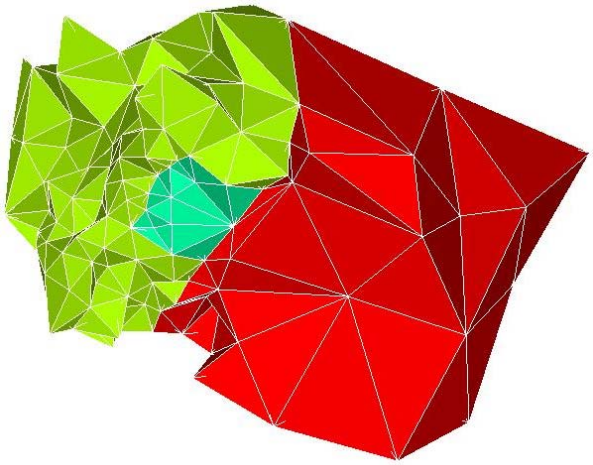


Figure 2.1: The subdomain with largest Cholesky factor in the 137 subdomain partition of the component is shown

with large numbers of unknowns are subdomains of the aero-shell. The source of the extremely small subdomains will be explained in the next section.

3. Mechanisms and Connectivity. A feature of FETI-1 methods is that singular matrices must be accurately factored. This is possible if the subdomain is face connected and the nodes are ordered so that either the last three nodes are 3 unknowns per node nodes and the nodes are not collinear, or the node is a 6 unknown per node node. One of the three main properties of FETI-DP methods is that only nonsingular matrices are factored. The other two nice properties of FETI-DP are that the coarse problem is sparse and that fewer iterations are required for convergence. For an arbitrary partition the null space of the subdomain stiffness matrix could be anything.

FETI-DP is more reliable than FETI-1, but is still sensitive to the partition. If a subdomain is not face connected, the corner nodes may not eliminate the entire null space. A feature of FETI-DP is that the coarse grid problem is approximately three times larger. Evidence will be presented in the next section that the load balance techniques developed here usually result in smaller coarse grid problems.

Figure 3.1 depicts the subgraph assigned to one processor. If we define two elements that share a node to be connected (nodal connectivity), then the subgraph has four connected components. Note that there is a triangular element that shares a node but not an edge with it's neighbors.

Here a more restrictive definition of connectivity is used, face connectivity. Two solid elements (or a solid and a shell) are face connected if they share a face. A shell element and a solid element that share three or more nodes are face connected. Two shell elements are face connected if they share an edge. A beam element is face connected if it shares a node with another beam element or a shell element. The subgraph in Figure 3.1 has five face connected components. In the remainder of this work connectivity always refers to face connectivity.

Connectivity is ensured by assigning each extra connected component of each subdomain to an additional processor. For the models considered, this results in a modest increase in the number of processors (see Figure 3.2).

Element	*Number of Nodes*	*Weight*
Wedge	6	2
Wedge	15	12
Tet	4	1
Tet	10	3
Hex	8	3
Hex	20	12
Tri	3	3
Tri	6	12
Quad	4	6
Quad	8	12
Beam	2	1

Table 2.1: graph weights for elements with linear (e.g. Tet4) or quadratic (c.f. Tet10) shape functions.

The number of face connected components of a partition may be acceptably large. Examples of such meshes have come to the attention of the authors in which beam elements have been painstakingly used to sow together nonconformal solid meshes.

4. Results and Conclusions. Results are presented for the engine and component models.

The engine model problem is solved efficiently on 2^7 or 2^8 processors. Though the engine model consists mostly of hexagonal elements, the modified graph partitioner is noticably more efficient. On approximately 2^8 processors the improved partitions reduce the time required to solve 31 linear systems from 261 seconds to 172 seconds.

The 2^9 processor runs illustrate different failure modes with two different corner selection strategies. For one corner selection strategy, with the improved partition the factorization of the coarse grid nonetheless erroneously detects zero pivots, and slow convergence results. It is noteworthy that our framework is not yet robust. For the other corner selection strategy, the improved partition is much more efficient due to the reduction in the size of the coarse grid problem.

The component model contains many triangular elements, and better illustrates the improvements in the partitions. Only results for the component model with the improved partition are presented. For the standard partition, initially FETI-DP broke down due to singular subdomain matrices across the processor range. Singular subdomain matrices were avoided by solving the shifted problem ($K + M10^5$). Unfortunately memory is insufficient to factor the shifted stiffness matrices on 128, 256 or 512 processors using a serial or a parallel linear solver for the coarse grid.

For the improved partitions and using the serial coarse grid solver FETI-DP is successful on 137 or 277 subdomain partitions, but on the 540 subdomain partition, memory was exhausted. For the 540 subdomain partition, FETI-DP succeeded using the DSCPACK parallel coarse grid linear solver.

In summary a technique for improving the partitions determined by an off-the-shelf graph partitioner have been presented. A carefully calibrated set of element weights is used to maintain load balance. Furthermore extra subdomains are added to ensure the face connectivity of the subdomains. The technique also results in smaller coarse grid problems for FETI-DP.

REFERENCES

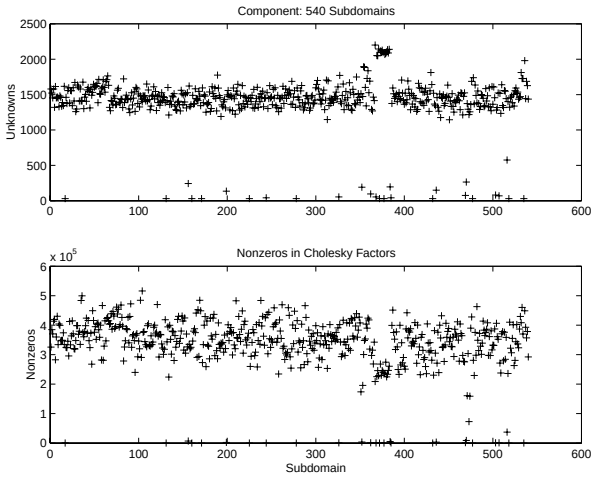


Figure 2.2: The processor loads for the component model partitioned into 540 subdomains is displayed. For both the number of unknowns and the number of nonzeros, the ratio to the maximum to the average is 3/2. Similar results are observed for partitions into 137 and 277 subdomains.

[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 one thousand processors and discussion of highly heterogeneous problems. *Int. J. Numer. Meth. Engrg.*, 47:513–535, 2000.

[2] 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.*, 50:1523–1544, 2001.

[3] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115:367–388, 1994.

[4] G. Karypis and V. Kumar. Metis, unstructured graph partitioning and sparse matrix ordering system. version 2.0. Technical report, University of Minnesota, Department of Computer Science, Minneapolis, MN 55455, August 1995.

[5] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, N.J., 1973.

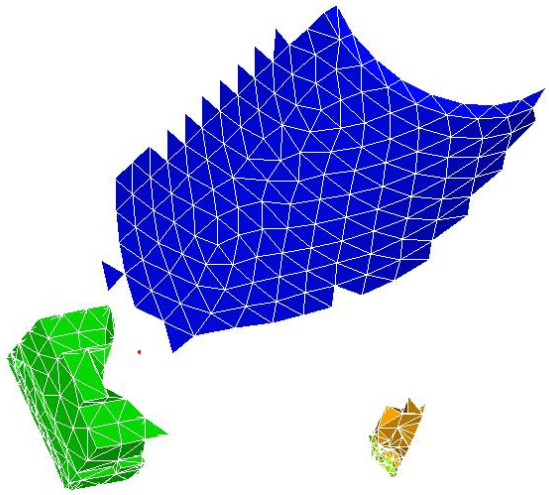


Figure 3.1: A disconnected subdomain computed by a graph partitioner for a 128 subdomain partition of the component model is depicted. The subdomain has four node-connected components, and five face-connected components.

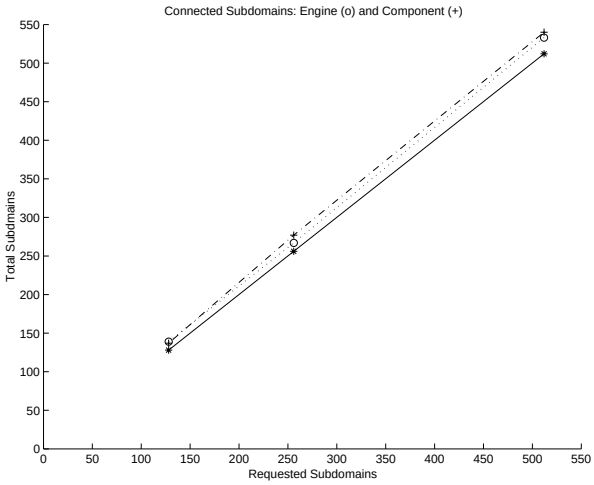


Figure 3.2: The figure displays the number of subdomains determined if 128, 256 and 512 subdomain partitions are requested for both the engine and the component model. The solid * line depicts no extra subdomains. The dash dot + line corresponds to the component model, and the dotted o line corresponds to the engine model. In the latter two cases each extra connected component of each subdomain is assigned to an additional subdomain.

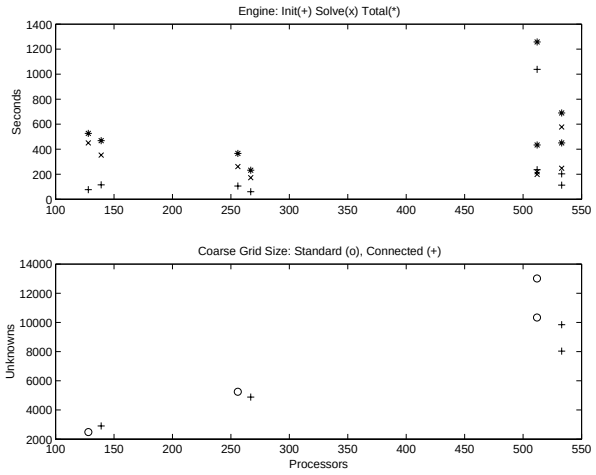


Figure 4.1: The figure displays the results for the engine model. The upper figure shows the initialization time (+), solve time (gap between (x) and (+)) and total time to compute the ten lowest modes on different numbers of processors. The lower figure shows the corresponding number of coarse grid unknowns for FETI-DP with the standard partition (o) and with the weighted partition maintaining connectivity (+). The 2^9 processor runs were run twice with different corner selection strategies.

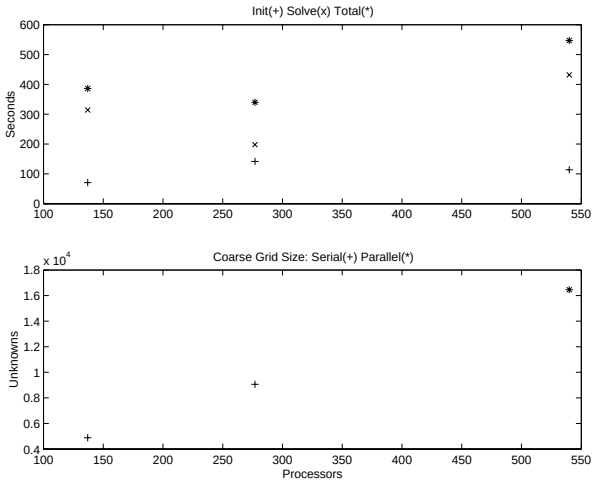


Figure 4.2: The results for the component with the weighted partition maintaining connectivity. No data is shown with the standard partitions due to break downs.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

17. Regularized formulations of FETI

Pavel Bochev[1], R.B. Lehoucq[2]

1. Introduction. Our report introduces two regularized formulations of the FETI-1 [2, 3] algorithm. These formulations provide an alternative way for handling the rigid body modes (RBM) associated with floating subdomains. Both formulations start with the FETI-1 Lagrangian but differ in the treatment of the RBMs. They provide coercive bilinear forms on the floating subdomains resulting in symmetric, positive definite finite element linear systems and so pseudoinverse computations can be avoided.

Our report is organized as follows. Section 2 formulates a consistently stabilized variant of FETI-1. This is accomplished by augmenting the FETI-1 Lagrangian with a redundant term that uses a suitable set of solution moments. Section 3 also employs the FETI-1 Lagrangian and the same set of moments but uses them to induce a splitting of the Sobolev space for the floating subdomain. A brief summary of the relevant moments and their properties is given in Section 4.

We quickly review our use of standard notation. Let Ω be a bounded domain in $\mathbb{R}^d$ where $d = 2, 3$ with Lipschitz boundary $\partial\Omega$ and so let $H^1(\Omega)$ denote a Sobolev space of order 1; $H^1(\Omega, \partial\Omega_D)$ denote a subspace of $H^1(\Omega)$ consisting of functions that vanish on $\partial\Omega_D \subset \partial\Omega$. We further suppose that Ω is partitioned into two nonoverlapping subdomains Ω_1 and Ω_2 with interface Γ; let $H^{1/2}(\Gamma)$ denote the trace space of $H^1(\Omega_i)$ on Γ; and let the dual spaces of $H^1(\Omega, \partial\Omega_D)$ and $H^{1/2}(\Gamma)$ be denoted by $H^{-1}(\Omega, \partial\Omega_D)$ and $H^{-1/2}(\Gamma)$, respectively. Let the norms and inner products on $H^1(\Omega)$ be given by $\|\cdot\|_1$ and $(\cdot,\cdot)_1$, respectively; and let $\langle\cdot,\cdot\rangle$ denote the duality pairing between a space and its dual.

Finally, we define the moments $c(\cdot) : H^1(\Omega, \partial\Omega_D) \mapsto \mathbb{R}^p$ for some positive integer p.

2. FETI-CS: A consistently stabilized FETI-1 algorithm. We consider the problem

$$\inf_{v \in H^1(\Omega, \partial\Omega_D)} \frac{1}{2} a(v, v) - \langle f, v\rangle_\Omega \tag{2.1}$$

where $a(v, v)$ is a coercive symmetric bilinear form and $f \in H^{-1}(\Omega, \partial\Omega_D)$. For example, the bilinear form could represent a scalar Poisson or linear elasticity equation in the plane or space. Equivalently, the minimization problem (2.1) may be posed over the subdomains Ω_1 and Ω_2 and recast as: Find a saddle-point $(u_1, u_2, \lambda, \tau, \mu) \in H^1(\Omega_1, \partial\Omega_1) \times H^1(\Omega_2) \times H^{1/2}(\Gamma) \times \mathbb{R}^p \times \mathbb{R}^p$ for the Lagrangian

$$\mathcal{L}(\hat{u}_1, \hat{u}_2, \hat{\lambda}, \hat{\tau}, \hat{\mu}) = \sum_{i=1}^{2} \big(\frac{1}{2} a(\hat{u}_i, \hat{u}_i)_{\Omega_i} - \langle f, \hat{u}_i\rangle_{\Omega_i}\big) + \langle \hat{\lambda}, \hat{u}_1 - \hat{u}_2\rangle_\Gamma + \hat{\tau}^T (c(\hat{u}_2) - \hat{\mu}). \tag{2.2}$$

The last term introduces a Lagrange multiplier $\hat{\tau}$ for the difference of the moments of the Lagrange multiplier $\hat{\mu}$ representing the (unknown) moment of the minimizer of (2.1) on subdomain Ω_2. Without this term (2.2) is simply the FETI-1 Lagrangian.

The optimality system for (2.2) is: Find $(u_1, u_2, \lambda, \tau, \mu) \in H^1(\Omega_1, \partial\Omega_1) \times H^1(\Omega_2) \times$

[1]Sandia National Labs, pbboche@sandia.gov. 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.

[2]Sandia National Labs, rblehou@sandia.gov. 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.

$H^{1/2}(\Gamma) \times \mathbb{R}^p \times \mathbb{R}^p$

$$\begin{array}{rcll} a(\hat{u}_1, u_1)_{\Omega_1} + \langle \hat{u}_1, \lambda \rangle_\Gamma & = & \langle f, \hat{u}_1 \rangle_{\Omega_1} & \forall \hat{u}_1 \in H^1(\Omega_1, \partial\Omega_1) \\ a(\hat{u}_2, u_2)_{\Omega_2} - \langle \hat{u}_2, \lambda \rangle_\Gamma + c(\hat{u}_2)^T \tau & = & \langle f, \hat{u}_2 \rangle_{\Omega_2} & \forall \hat{u}_2 \in H^1(\Omega_2) \\ \langle \hat{\lambda}, u_1 - u_2 \rangle_\Gamma & = & 0 & \forall \hat{\lambda} \in H^{1/2}(\Gamma) \\ \hat{\tau}^T (c(u_2) - \mu) & = & 0 & \forall \hat{\tau} \in \mathbb{R}^p \\ \hat{\mu}^T \tau & = & 0 & \forall \hat{\mu} \in \mathbb{R}^p. \end{array} \tag{2.3}$$

The last two equations imply that $\tau = \mathbf{0}$ and $c(u_2) - \mu = \mathbf{0}$. Therefore the last term of the Lagrangian (2.3) is a redundant constraint and we recover the FETI-1 optimality system

$$\begin{array}{rcll} a(\hat{u}_1, u_1)_{\Omega_1} + \langle \hat{u}_1, \lambda \rangle_\Gamma & = & \langle f, \hat{u}_1 \rangle_{\Omega_1} & \forall \hat{u}_1 \in H^1(\Omega_1, \partial\Omega_1) \\ a(\hat{u}_2, u_2)_{\Omega_2} - \langle \hat{u}_2, \lambda \rangle_\Gamma & = & \langle f, \hat{u}_2 \rangle_{\Omega_2} & \forall \hat{u}_2 \in H^1(\Omega_2) \\ \langle \hat{\lambda}, u_1 - u_2 \rangle_\Gamma & = & 0 & \forall \hat{\lambda} \in H^{1/2}(\Gamma). \end{array} \tag{2.4}$$

However, instead of using (2.3) directly, we stabilize the second and third constraints of (2.3) as

$$\begin{array}{ll} \hat{\tau}^T (c(u_2) - \mu) = \hat{\tau}^T \Upsilon^{-1} \tau & \forall \hat{\tau} \in \mathbb{R}^p \\ \hat{\mu}^T \tau = \hat{\mu}^T \Upsilon (c(u_2) - \mu) & \forall \hat{\mu} \in \mathbb{R}^p \end{array} \tag{2.5}$$

where Υ is a diagonal matrix of order p with positive diagonal elements. We can now eliminate τ from (2.3). Equation (2.5) implies

$$c(\hat{u}_2)^T \tau = c(\hat{u}_2)^T \Upsilon (c(u_2) - \mu). \tag{2.6}$$

With these relations, we obtain the optimality system: Find $(u_1, u_2, \lambda, \mu) \in H^1(\Omega_1, \partial\Omega_1) \times H^1(\Omega_2) \times H^{1/2}(\Gamma) \times \mathbb{R}^p$

$$\begin{array}{rcll} a(\hat{u}_1, u_1)_{\Omega_1} + \langle \hat{u}_1, \lambda \rangle_\Gamma & = & \langle f, \hat{u}_1 \rangle_{\Omega_1} & \forall \hat{u}_1 \in H^1(\Omega_1, \partial\Omega_1) \\ \tilde{a}(\hat{u}_2, u_2)_{\Omega_2} - \langle \hat{u}_2, \lambda \rangle_\Gamma - c(\hat{u}_2)^T \Upsilon \mu & = & \langle f, \hat{u}_2 \rangle_{\Omega_2} & \forall \hat{u}_2 \in H^1(\Omega_2) \\ \langle \hat{\lambda}, u_1 - u_2 \rangle_\Gamma & = & 0 & \forall \hat{\lambda} \in H^{1/2}(\Gamma) \\ -\hat{\mu}^T \Upsilon c(u_2) + \hat{\mu}^T \Upsilon \mu & = & 0 & \forall \hat{\mu} \in \mathbb{R}^p \end{array} \tag{2.7}$$

where $\tilde{a}(\cdot,\cdot)_{\Omega_2} \equiv a(\cdot,\cdot)_{\Omega_2} + c(\cdot)^T \Upsilon c(\cdot)$. We remark that this optimality system can also be derived by penalizing a FETI-1 Lagrangian by

$$\frac{1}{2} \| c(\hat{u}_2) - \hat{\mu} \|^2$$

or, equivalently, by replacing the last term of (2.2) with the above least-squares term. In either case, we have the following two results.

Lemma 2.1 *The symmetric bilinear form $\tilde{a}(\cdot,\cdot)_{\Omega_2}$ is coercive on $H^1(\Omega_2) \times H^1(\Omega_2)$.*

Proof. See Bochev and Lehoucq [1] for the proof. ∎

Theorem 2.1 *(u_1, u_2, λ) solves (2.4) if and only if $(u_1, u_2, \lambda, \mu = c(u_2))$ solves (2.7).*

Proof. The theorem is easily established by using the stabilized constraints (2.5) and recalling that $\tau = \mathbf{0}$. ∎

The theorem demonstrates that (2.5) represents a consistent stabilization. The impact of this innocuous sleight of hand is that the resulting coarse grid problem is equivalently stabilized. We now demonstrate this.

A conforming FEM for (2.7) results in the discrete optimality system

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \mathbf{B}_1^T & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{K}}_2 & -\mathbf{B}_2^T & -\mathbf{C}_2^T\Upsilon \\ \mathbf{B}_1 & -\mathbf{B}_2 & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\Upsilon\mathbf{C}_2 & \mathbf{0} & \Upsilon \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{0} \\ \mathbf{0} \end{bmatrix} \tag{2.8}$$

where $\tilde{\mathbf{K}}_2 \equiv \mathbf{K}_2 + \mathbf{C}_2^T\Upsilon\mathbf{C}_2$.

Elimination of the primal variables in (2.8) results in the coarse grid problem

$$\begin{bmatrix} \mathbf{B}_1\mathbf{K}_1^{-1}\mathbf{B}_1^T + \mathbf{B}_2\tilde{\mathbf{K}}_2^{-1}\mathbf{B}_2^T & \mathbf{B}_2\tilde{\mathbf{K}}_2^{-1}\mathbf{C}_2^T\Upsilon \\ \Upsilon\mathbf{C}_2\tilde{\mathbf{K}}_2^{-1}\mathbf{B}_2^T & \Upsilon\mathbf{C}_2\tilde{\mathbf{K}}_2^{-1}\mathbf{C}_2^T\Upsilon - \Upsilon \end{bmatrix} \begin{bmatrix} \lambda \\ \mu \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_1 \end{bmatrix} \tag{2.9}$$

where

$$\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1\mathbf{K}_1^{-1}\mathbf{f}_1 - \mathbf{B}_2\tilde{\mathbf{K}}_2^{-1}\mathbf{f}_2 \\ -\Upsilon\mathbf{C}_2\tilde{\mathbf{K}}_2^{-1}\mathbf{f}_2 \end{bmatrix}.$$

As compared with FETI-1, the columns of $\tilde{\mathbf{K}}_2^{-1}\mathbf{C}_2^T\Upsilon$ are approximating a basis for the rigid body modes associated with Ω_2, and $\tilde{\mathbf{K}}_2^{-1}$ is an approximation to the pseudoinverse of $\mathbf{K}_2$. Inserting the solution of the coarse grid problem (2.9) into (2.8) results in

$$\mathbf{K}_1\mathbf{u}_1 = \mathbf{f}_1 - \mathbf{B}_1^T\lambda \tag{2.10}$$

$$\tilde{\mathbf{K}}_2\mathbf{u}_2 = \mathbf{f}_2 + \mathbf{B}_2^T\lambda + \mathbf{C}_2^T\Upsilon\mu. \tag{2.11}$$

These two linear systems have symmetric positive definite coefficient matrices and can be solved in parallel.

We remark that (2.11) corresponds to the minimization problem

$$\inf_{v\in H^1(\Omega_2)} \frac{1}{2}\tilde{a}(v,v) - \langle \tilde{f}, v\rangle_{\Omega_2}$$

where $\tilde{f}$ is the continuous load associated with the discrete load of (2.11).

3. FETI-SS: Regularization by space splitting. In this section we introduce a modification of FETI-1 that allows for a wider choice of well-posed primal problems for these domains. In particular, our approach results in nonsingular linear systems with properties that can be easily controlled.

Our starting point is the splitting of $H^1(\Omega_2)$ into the direct sum

$$H^1(\Omega_2) = H_c^1(\Omega_2) \oplus \mathcal{N}_2$$

where $\mathcal{N}_2$ is the RBM space for Ω_2 and

$$H_c^1(\Omega_2) = \{u \in H^1(\Omega_2) \,|\, c_2(u) = 0\},$$

is the complement space with respect to the moments c_2. The report [1] demonstrates that such a splitting exists for any non-degenerate set of moments. As a result, any $u_2 \in H^1(\Omega_2)$ can be uniquely written as $u_{2c} + \mathbf{R}_2\alpha$ where $\mathbf{R}_2$ is a basis for $\mathcal{N}_2$ and $\alpha \in \mathbb{R}^p$. To solve (2.1) we consider the problem of finding the saddle-point $(u_1, u_{2c}, \alpha, \lambda) \in H^1(\Omega_1, \partial\Omega_1) \times H_c^1(\Omega_2) \times \mathbb{R}^p \times H^{1/2}(\Gamma)$ of the Lagrangian

$$\mathcal{L}(\hat{u}_1, \hat{u}_{2c}, \hat{\alpha}, \hat{\lambda}) = \sum_{i=1}^{2} \big(\frac{1}{2}a(\hat{u}_i, \hat{u}_i)_{\Omega_i} - \langle f, \hat{u}_i\rangle_{\Omega_i}\big) + \langle \hat{\lambda}, \hat{u}_1 - (\hat{u}_{2c} + \mathbf{R}_2\hat{\alpha})\rangle_\Gamma. \tag{3.1}$$

This Lagrangian only differs from the FETI-1 Lagrangian by explicitly specifying a particular solution on the floating subdomain. The optimality system for (3.1) is to seek $(u_1, u_{2c}, \alpha, \lambda) \in H^1(\Omega_1, \partial\Omega_1) \times H^1_c(\Omega_2) \times \mathbb{R}^p \times H^{1/2}(\Gamma)$ such that

$$\begin{array}{rcll} a(\hat{u}_1, u_1)_{\Omega_1} + \langle \hat{u}_1, \lambda\rangle_\Gamma &=& \langle f, \hat{u}_1\rangle_{\Omega_1} & \forall \hat{u}_1 \in H^1(\Omega_1, \partial\Omega_1) \\ a(\hat{u}_{2c}, u_{2c})_{\Omega_2} - \langle \hat{u}_{2c}, \lambda\rangle_\Gamma &=& \langle f, \hat{u}_{2c}\rangle_{\Omega_2} & \forall \hat{u}_{2c} \in H^1_c(\Omega_2) \\ -\langle \mathbf{R}_2\hat{\alpha}, \lambda\rangle_\Gamma &=& \langle f, \mathbf{R}_2\hat{\alpha}\rangle_{\Omega_2} & \forall \hat{\alpha} \in \mathbb{R}^p \\ \langle \hat{\lambda}, u_1 - (u_{2c} + \mathbf{R}_2\alpha)\rangle_\Gamma &=& 0 & \forall \hat{\lambda} \in H^{1/2}(\Gamma). \end{array} \tag{3.2}$$

Note that in (3.2) the floating subdomain problem is restricted to finding a particular solution out of the complement space $H^1_c(\Omega_2)$ rather than the space $H^1(\Omega_2)$. This seemingly minor change makes the floating subdomain problem *uniquely solvable*. Therefore, its conforming discretization, that is restriction to a finite element subspace of $H^1_c(\Omega_2)$, would engender a non-singular linear system. However, building a finite element subspace of $H^1_c(\Omega_2)$ may not be a simple matter and discretization by standard finite element subspaces of $H^1(\Omega_2)$ is preferred.

To enable the use of standard finite elements the floating subdomain equation is further replaced by a regularized problem in which the bilinear form $a(\cdot,\cdot)_{\Omega_2}$ is augmented by the term $c_2(\hat{u}_2)^T\Upsilon c_2(u_2)$. The regularized optimality system is to seek $(u_1, u_2, \alpha, \lambda) \in H^1(\Omega_1, \partial\Omega_1) \times H^1(\Omega_2) \times \mathbb{R}^p \times H^{1/2}(\Gamma)$ such that

$$\begin{array}{rcll} a(\hat{u}_1, u_1)_{\Omega_1} + \langle \hat{u}_1, \lambda\rangle_\Gamma &=& \langle f, \hat{u}_1\rangle_{\Omega_1} & \forall \hat{u}_1 \in H^1(\Omega_1, \partial\Omega_1) \\ a(\hat{u}_2, u_2)_{\Omega_2} + c_2(\hat{u}_2)^T\Upsilon c_2(u_2) - \langle \hat{u}_2, \lambda\rangle_\Gamma &=& \langle f, \hat{u}_2\rangle_{\Omega_2} & \forall \hat{u}_2 \in H^1(\Omega_2) \\ -\langle \mathbf{R}_2\hat{\alpha}, \lambda\rangle_\Gamma &=& \langle f, \mathbf{R}_2\hat{\alpha}\rangle_{\Omega_2} & \forall \hat{\alpha} \in \mathbb{R}^p \\ \langle \hat{\lambda}, u_1 - (u_2 + \mathbf{R}\alpha)\rangle_\Gamma &=& 0 & \forall \hat{\lambda} \in H^{1/2}(\Gamma). \end{array} \tag{3.3}$$

Theorem 3.1 *Problems (3.2) and (3.3) are equivalent.*

Proof. The only point that needs to be verified is that a solution $(u_1, u_2, \alpha, \lambda)$ of (3.3) has its second component in the complement space $H^1_c(\Omega_2)$. Choosing $\hat{u}_2 = \mathbf{R}_2\hat{\alpha}$ in the second equation in (3.3) combined with the third equation gives

$$c_2(\mathbf{R}_2\hat{\alpha})^T\Upsilon c_2(u_2) = \langle \mathbf{R}_2\hat{\alpha}, \lambda\rangle_\Gamma + \langle f, \mathbf{R}_2\hat{\alpha}\rangle_{\Omega_2} \equiv 0$$

for any $\hat{\alpha} \in \mathbb{R}^p$. Therefore, $c_2(u_2) = 0$ and $u_2 \in H^1_c(\Omega_2)$. ∎

A conforming FEM for (3.3) results in the linear system

$$\begin{bmatrix} \mathbf{K}_1 & \mathbf{0} & \mathbf{B}_1^T & \mathbf{0} \\ \mathbf{0} & \tilde{\mathbf{K}}_2 & -\mathbf{B}_2^T & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -(\mathbf{B}_2\mathbf{R}_2)^T & \mathbf{0} \\ \mathbf{B}_1 & -\mathbf{B}_2 & \mathbf{0} & -\mathbf{B}_2\mathbf{R}_2 \end{bmatrix} \begin{bmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{f}_2 \\ \mathbf{R}_2^T\mathbf{f}_2 \\ \mathbf{0} \end{bmatrix} \tag{3.4}$$

where $\tilde{\mathbf{K}}_2$ is the same matrix as in (2.8) and we redundantly use $\mathbf{R}_2$ to denote the coefficients associated with the finite element approximants for the RBMs.

We note the close similarity between (3.4) and a FETI-1 discrete problem. In both cases a particular solution for the floating subdomain is generated and a component in $\mathcal{N}_2$ is added to satisfy the interface continuity condition. However, in contrast to a FETI-1, in (3.4) the floating subdomain matrix is non-singular and we have complete control over the choice of the particular solution by virtue of the moments c_2. These moments can be further selected so as to optimize the nonsingular matrix $\tilde{\mathbf{K}}_2$ with respect to a particular solver.

Elimination of the primal variables in (3.4) results in the coarse grid problem

$$\begin{bmatrix} \mathbf{B}_1\mathbf{K}_1^{-1}\mathbf{B}_1^T + \mathbf{B}_2\tilde{\mathbf{K}}_2^{-1}\mathbf{B}_2^T & -\mathbf{B}_2\mathbf{R} \\ -(\mathbf{B}_2\mathbf{R})^T & \mathbf{0} \end{bmatrix} \begin{bmatrix} \lambda \\ \alpha \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_1 \end{bmatrix} \tag{3.5}$$

where

$$\begin{bmatrix} \mathbf{d}_1 \\ \mathbf{d}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{B}_1\mathbf{K}_1^{-1}\mathbf{f}_1 - \mathbf{B}_2\tilde{\mathbf{K}}_2^{-1}\mathbf{f}_2 \\ \mathbf{R}^T\mathbf{f}_2 \end{bmatrix}.$$

Inserting the solution of the coarse grid problem (3.5) into (3.4) results in

$$\mathbf{K}_1\mathbf{u}_1 = \mathbf{f}_1 - \mathbf{B}_1^T\lambda \tag{3.6}$$

$$\tilde{\mathbf{K}}_2\mathbf{u}_2 = \mathbf{f}_2 + \mathbf{B}_2^T\lambda. \tag{3.7}$$

This primal system and the FETI-1 primal system only differ in the coefficient matrix for $\mathbf{u}_2$. Here $\tilde{\mathbf{K}}_2$ is symmetric positive definite whereas FETI-1 uses the singular $\mathbf{K}_2$. Therefore a computation of a pseudoinverse is avoided.

4. The moments $c(\cdot)$. Suppose that we have a floating subdomain Ω, a RBM subspace $\mathcal{N}$ and resulting basis $\mathbf{R}$ (discrete or continuous). The moments $c(\cdot)$ play a central role in our regularization strategy. Both of the FETI formulations introduced in this report rely upon these moments to regularize the floating subdomain problems. The purpose of the moments is to provide an "energy" measure for the RBMs that otherwise have zero strain energy $a(\cdot,\cdot)$.

Therefore, the guiding principle in their choice is to ensure that they form a non-degenerate set. By non-degenerate here we mean that the matrix $c(\mathbf{R})$ of order p is non-singular. For linear elasticity [1] one such set of moments is given by the functional

$$c(v) \equiv \begin{bmatrix} \displaystyle\int_\Omega \Theta_1 v \\ \displaystyle\int_\Omega \Theta_2 \nabla \times v \end{bmatrix} \tag{4.1}$$

where the diagonal elements of

$$\Theta_1 = \mathrm{diag}(\theta_{1,1}, \theta_{1,2}, \theta_{1,3}) \quad \text{and} \quad \Theta_2 = \mathrm{diag}(\theta_{2,1}, \theta_{2,2}, \theta_{2,3}) \tag{4.2}$$

are elements of $H^{-1}(\Omega)$ satisfying the hypothesis

$$\int_\Omega \theta_{1,i} \neq 0 \quad \text{and} \quad \int_\Omega \theta_{2,i} \neq 0$$

for $i = 1, 2, 3$. These dual functions serve the useful purpose of allowing us to enforce the mean and mean of the curl of the displacement along a portion of Ω.

When the moments (4.1) are restricted to finite element subspaces they generate full rank matrices with p columns, where p is the dimension of $\mathcal{N}$. The regularizing term added to the singular stiffness matrix on a floating subdomain is simply a rank-p correction to this matrix. When the dual functions in (4.2) have small supports the rank-p correction is a sparse matrix and the regularized problem is amenable to a direct solver methods. Larger supports generally improve the condition number of the regularized matrix but they also lead to formally dense systems. Therefore, regularization via moments is useful for iterative solution methods where it is only necessary to compute the product of the rank-p correction matrix with a direction vector.

5. Conclusions. Our report introduced two regularized formulations of the FETI- 1 [2, 3] algorithm. These formulations provide an alternative way for handling the rigid body modes (RBM) associated with floating subdomains. Both formulations start with the FETI-1 Lagrangian but differ in the treatment of the RBMs. They provide coercive bilinear forms on the floating subdomains resulting in symmetric, positive definite finite element linear systems and so pseudoinverse computations can be avoided.

REFERENCES

[1] P. Bochev and R. B. Lehoucq. Energy principles and finite element methods for pure traction linear elasticity. Technical Report SAND2002-0500J, Sandia National Laboratories, New Mexico, USA, 2002.

[2] C. Farhat and F.-X. Roux. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *Int. J. Numer. Meth. Engrg.*, 32:1205–1227, 1991.

[3] 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.*, 13:379–396, 1992.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

18. Balancing Neumann-Neumann for (In)Compressible Linear Elasticity and (Generalized) Stokes — Parallel Implementation

P. Goldfeld[1]

1. Introduction. In this paper, an extension of the Balancing Neumann-Neumann method for a class of symmetric, indefinite problems is presented, with an emphasis on implementational and algorithmic aspects; theoretical results on a bound for the condition number of the relevant operator are also stated, without proof, and results of large-scale numerical experiments are reported. For a development of the theory see [7], [4].

The Balancing Neumann-Neumann domain decomposition technique (see, e.g., Mandel [5] or Mandel and Brezina [6]) has recently been extended to a class of saddle-point problems, including the Stokes Equation (see Pavarino and Widlund [7]) and the mixed formulation of linear elasticity (see Goldfeld, Pavarino and Widlund [3], [4]).

In this algorithm, after decomposing the original domain into nonoverlapping subdomains, the interior velocity/displacement and all but the subdomain-wise constant pressure unknowns are eliminated. A preconditioner for the resulting saddle-point Schur complement problem is constructed based on the solution of a coarse problem, with one pressure and a few velocity/displacement unknowns per subdomain, and on the solution of local problems with mixed or natural boundary conditions. Local Dirichlet problems must also be solved in order to compute the action of the Schur complement operator. The quality of this preconditioner can be shown to be independent of the number of subdomains and to depend only polylogarithmically on the size of the local problems, when the coefficients are constant. Numerical experiments indicate that this is still the case when there are arbitrary jumps on the coefficients.

This paper is organized as follows. In Section 2, we briefly describe the class of problems considered and their mixed finite element discretizations. The substructuring process is explained in Section 3, where we also include some remarks on the practical implementation of the Schur complement operator. In Section 4, the Balancing Neumann-Neumann preconditioner is introduced. In Section 5, the theoretical results on the quality of the preconditioner are stated and, finally, numerical experiments are reported in Section 6.

2. Problems and Discretizations. We consider the problems of linear elasticity with a mixed formulation (compressible, incompressible or almost incompressible cases), Stokes' equations and generalized Stokes' equations (with compressibility). All of them have a variational formulation of the following form: *For* $\Omega \subset \mathbb{R}^d$*, a polyhedral domain, given* $\mathbf{f} \in \left(H^{-1}(\Omega)\right)^d$, $\mathbf{g} \in \left(H^{1/2}(\partial\Omega)\right)^d$ *and* $h \in L^2(\Omega)$*, find* $(\mathbf{u}, p) \in \left(\tilde{\mathbf{g}} + \left(H_0^1(\Omega)\right)^d\right) \times L^2(\Omega)$ *satisfying*

$$\left\{\begin{array}{rcrcll} a(\mathbf{u},\mathbf{v}) & + & b(\mathbf{v},p) & = & \langle \mathbf{f},\mathbf{v}\rangle & \forall \mathbf{v} \in \left(H_0^1(\Omega)\right)^d \\ b(\mathbf{u},q) & - & c(p,q) & = & \langle h,q\rangle & \forall q \in L^2(\Omega) \\ & & \mathbf{u}|_{\partial\Omega} & = & \mathbf{g} & \end{array}\right. . \tag{2.1}$$

Here $\tilde{\mathbf{g}}$ is any function in $\left(H^1(\Omega)\right)^d$ such that $\tilde{\mathbf{g}}|_{\partial\Omega} = \mathbf{g}$. The choice of the bilinear forms a, b and c depends upon the problem we are solving:

[1]New York University, paulo.goldfeld@pobox.com

	$a(\mathbf{u},\mathbf{v})$	$b(\mathbf{v},q)$	$c(p,q)$
compressible elasticity	$2\mu\int_\Omega \varepsilon(\mathbf{u}):\varepsilon(\mathbf{v})$	$-\int_\Omega q\nabla\cdot\mathbf{v}$	$\frac{1}{\lambda}\int_\Omega pq$
incompressible elasticity	$2\mu\int_\Omega \varepsilon(\mathbf{u}):\varepsilon(\mathbf{v})$	$-\int_\Omega q\nabla\cdot\mathbf{v}$	0
Stokes	$\nu\int_\Omega \nabla\mathbf{u}:\nabla\mathbf{v}$	$-\int_\Omega q\nabla\cdot\mathbf{v}$	0
generalized Stokes	$\nu\int_\Omega \nabla\mathbf{u}:\nabla\mathbf{v}$	$-\int_\Omega q\nabla\cdot\mathbf{v}$	$\frac{1}{\lambda}\int_\Omega pq$

Here $\varepsilon(\mathbf{u}):\varepsilon(\mathbf{v}) = \frac{1}{4}\sum_{i,j=1}^{d}\left(\frac{\partial u_i}{\partial x_j}+\frac{\partial u_j}{\partial x_i}\right)\left(\frac{\partial v_i}{\partial x_j}+\frac{\partial v_j}{\partial x_i}\right)$ and $\nabla\mathbf{u}:\nabla\mathbf{v} = \sum_{i,j=1}^{d}\frac{\partial u_i}{\partial x_j}\frac{\partial v_i}{\partial x_j}$.

To fix ideas, we will focus, in the remainder of this paper, on the elasticity problem. Therefore, $\mathbf{u}$ will be the displacement vector and the relevant coefficients will be the Lamé parameters μ and λ.

A conforming mixed finite-element discretization of (2.1) yields a linear system of the form

$$K\underline{\mathbf{u}} = K\begin{bmatrix}\mathbf{u}\\ p\end{bmatrix} = \begin{bmatrix}A & B^T\\ B & -C\end{bmatrix}\begin{bmatrix}\mathbf{u}\\ p\end{bmatrix} = \underline{\mathbf{f}} = \begin{bmatrix}\mathbf{f}\\ h\end{bmatrix}.$$

We select an inf-sup stable pair of finite element spaces for pressure and displacement. As will become evident in the next subsection, our method requires the pressure space to be discontinuous, at least across the interface.

Note that although this paper is written in the language of finite elements only, the method here presented is equally suitable for spectral element discretizations (see [7], [3], [4]).

3. Substructuring. The domain Ω is decomposed into N nonoverlapping subdomains, $\{\Omega_i\}_{i=1,2,\ldots,N}$, the boundaries of which do not cut through any element. Denote by Γ_h the set of nodes on the interface between subdomains, i.e., the nodes belonging to more than one subdomain. As usual, K and $\underline{\mathbf{f}}$ can be generated by subassembly:

$$\begin{aligned} K &= \sum_{i=1}^{N} {R^{(i)}}^T K^{(i)} R^{(i)} = \sum_{i=1}^{N} {R^{(i)}}^T \begin{bmatrix} A^{(i)} & {B^{(i)}}^T \\ B^{(i)} & -C^{(i)}\end{bmatrix} R^{(i)}, \qquad (3.1)\\ \underline{\mathbf{f}} &= \sum_{i=1}^{N} {R^{(i)}}^T \begin{bmatrix}\mathbf{f}^{(i)}\\ h^{(i)}\end{bmatrix}, \end{aligned}$$

where the restriction matrix $R^{(i)}$ is a matrix of zeros and ones which translates global indices of the nodes into local numbering.

Assume that the basis for the pressure space can be split as follows:

- there are N *coarse* pressures, $\{\psi_{0,i}\}_{i=1,2,\ldots,N}$, defined by $\psi_{0,i} = \chi_{\Omega_i}$, where χ_{Ω_i} is the characteristic function of the set Ω_i. We also refer to these functions as the *constant* or *interface* pressures;

- the remaining, *interior* pressures, $\{\psi_{I,j_i}\}_{j_i\in J_i}$, have zero average, $\int_\Omega \psi_{I,j_i} = 0$, and are local, in the sense that $\mathrm{supp}\,(\psi_{I,j_i}) \subset \Omega_i$.

After reordering unknowns and equations, the vectors $\underline{\mathbf{u}}$ and $\underline{\mathbf{f}}$, and the stiffness matrix K are expressed as

$$\begin{aligned}
\underline{\mathbf{u}} &= \left[\begin{array}{c} \underline{\mathbf{u}}_I \\ \hline \underline{\mathbf{u}}_\Gamma \end{array}\right] = \left[\begin{array}{c} \mathbf{u}_I \\ p_I \\ \hline \mathbf{u}_\Gamma \\ p_0 \end{array}\right], \quad \underline{\mathbf{f}} = \left[\begin{array}{c} \underline{\mathbf{f}}_I \\ \hline \underline{\mathbf{f}}_\Gamma \end{array}\right] = \left[\begin{array}{c} \mathbf{f}_I \\ h_I \\ \hline \mathbf{f}_\Gamma \\ h_0 \end{array}\right], \\
K &= \left[\begin{array}{c|c} K_{II} & K_{I\Gamma} \\ \hline K_{\Gamma I} & K_{\Gamma\Gamma} \end{array}\right] = \left[\begin{array}{cc|cc} A_{II} & B_{II}^T & A_{I\Gamma} & 0 \\ B_{II} & -C_{II} & B_{I\Gamma} & 0 \\ \hline A_{\Gamma I} & B_{I\Gamma}^T & A_{\Gamma\Gamma} & B_{0\Gamma}^T \\ 0 & 0 & B_{0\Gamma} & -C_{00} \end{array}\right].
\end{aligned} \tag{3.2}$$

The (1,1)-block K_{II} is invertible, even when K is not (i.e., when the problem is incompressible and therefore C is zero and the solution is only defined up to a constant pressure.) We can eliminate the interior variables and define a Schur complement problem,

$$S\underline{\mathbf{u}}_\Gamma = \tilde{\underline{\mathbf{f}}}_\Gamma, \tag{3.3}$$

where

$$\begin{aligned}
S &= K_{\Gamma\Gamma} - K_{\Gamma I}K_{II}^{-1}K_{I\Gamma} = \left[\begin{array}{cc} S_\Gamma & B_{0\Gamma}^T \\ B_{0\Gamma} & -C_{00} \end{array}\right] \quad \text{and} \\
\tilde{\underline{\mathbf{f}}}_\Gamma &= \underline{\mathbf{f}}_\Gamma - K_{\Gamma I}K_{II}^{-1}\underline{\mathbf{f}}_I,
\end{aligned} \tag{3.4}$$

with

$$S_\Gamma = A_{\Gamma\Gamma} - \left[\begin{array}{cc} A_{\Gamma I} & B_{I\Gamma}^T \end{array}\right] \left[\begin{array}{cc} A_{II} & B_{II}^T \\ B_{II} & -C_{II} \end{array}\right]^{-1} \left[\begin{array}{c} A_{I\Gamma} \\ B_{I\Gamma} \end{array}\right]. \tag{3.5}$$

We note that K_{II} is block-diagonal, which allows us to generate S by subassembly, by means of restriction matrices $R_\Gamma^{(i)}$:

$$S = \sum_{i=1}^{N} {R_\Gamma^{(i)}}^T S^{(i)} R_\Gamma^{(i)} = \sum_{i=1}^{N} {R_\Gamma^{(i)}}^T \left(K_{\Gamma\Gamma}^{(i)} - K_{\Gamma I}^{(i)} {K_{II}^{(i)}}^{-1} K_{I\Gamma}^{(i)}\right) R_\Gamma^{(i)}. \tag{3.6}$$

We present a preconditioner for the operator S. Once the system $S\underline{\mathbf{u}}_\Gamma = \tilde{\underline{\mathbf{f}}}_\Gamma$ is solved, the computations required to obtain $\underline{\mathbf{u}}_I$ are completely local.

3.1. Implementing S. Before we describe the Neumann-Neumann preconditioner, we discuss how to compute the action of the operator S on a given vector.

We have assumed that the basis functions for the pressure degrees of freedom can be divided into two groups: zero-average functions and constant functions. We now show how S can be implemented using a standard basis for the pressure, as long as the pressure space *admits* a basis of that special form.

In our actual implementation we generate, instead of the stiffness matrix in (3.2), a stiffness matrix $\tilde{K}$ using a standard nodal basis and introduce a Lagrange multiplier to enforce the zero average of the pressure. Furthermore, we never assemble the entire matrix $\tilde{K}$, but rather work with the local stiffness matrices $\tilde{K}^{(i)}$:

$$\tilde{K} = \sum_{i=1}^{N} {\tilde{R}^{(i)}}^T \tilde{K}^{(i)} \tilde{R}^{(i)}, \quad \text{where} \quad \tilde{K}^{(i)} = \left[\begin{array}{ccc} A^{(i)} & {\tilde{B}^{(i)}}^T & 0 \\ \tilde{B}^{(i)} & -\tilde{C}^{(i)} & w^{(i)} \\ 0 & {w^{(i)}}^T & 0 \end{array}\right]. \tag{3.7}$$

Note that, since a different basis has been used, $\tilde{K}^{(i)}$, $\tilde{B}^{(i)}$, and $\tilde{C}^{(i)}$ are different from $K^{(i)}$, $B^{(i)}$, and $C^{(i)}$ in equation (3.1). The entries of the vector $w^{(i)}$ are the integrals of the pressure basis functions over Ω.

In each of the local matrices $\tilde{K}^{(i)}$, we eliminate the interior velocities, *all* the pressures and the Lagrange multiplier. This corresponds to taking the Schur complement with respect to the (2,2)-block in the following matrix, which is a reordering of (3.7):

$$\left[\begin{array}{ccc|c} A_{II}^{(i)} & \tilde{B}_{I}^{(i)^T} & 0 & A_{I\Gamma}^{(i)} \\ \tilde{B}_{I}^{(i)} & -\tilde{C}^{(i)} & w^{(i)} & \tilde{B}_{\Gamma}^{(i)} \\ 0 & w^{(i)^T} & 0 & 0 \\ \hline A_{\Gamma I}^{(i)} & \tilde{B}_{\Gamma}^{(i)^T} & 0 & A_{\Gamma\Gamma}^{(i)} \end{array}\right].$$

We can show that the result of this static condensation is precisely $S_{\Gamma}^{(i)}$, the $(1,1)$-block of $S^{(i)}$ (see (3.4), (3.5), (3.6)). The remainder of the matrix $S^{(i)}$, namely the vector $B_{0\Gamma}^{(i)^T}$ and the scalar $C_{00}^{(i)}$, can be computed by means of the formula:

$$\left[\begin{array}{cc} A_{\Gamma\Gamma}^{(i)} & B_{0\Gamma}^{(i)^T} \\ B_{0\Gamma}^{(i)} & C_{00}^{(i)} \end{array}\right] = \left[\begin{array}{c} I \\ e^{(i)^T} \end{array}\right] \left[\begin{array}{cc} A_{\Gamma\Gamma}^{(i)} & \tilde{B}_{\Gamma}^{(i)^T} \\ \tilde{B}_{\Gamma}^{(i)} & -\tilde{C}^{(i)} \end{array}\right] \left[\begin{array}{cc} I & e^{(i)} \end{array}\right].$$

Here the matrix at the right side of the equation is a submatrix of (3.7) and the entries of the vector $e^{(i)}$ are the coefficients that express the constant pressure on subdomain Ω_i in terms of the regular basis functions:

$$\sum_{k=1}^{\tilde{n}_p} \left(e^{(i)}\right)_k \tilde{\psi}_k = \chi_{\Omega_i},$$

where $\left\{\tilde{\psi}_k\right\}_{k=1,\ldots,\tilde{n}_p}$ is the basis for the pressure space.

4. Preconditioner. The Balancing Neumann-Neumann preconditioner is of the form:

$$Q = Q_0 + (I - Q_0 S)\, Q_{loc}\, (I - S Q_0)\,,$$

where Q_0 is the coarse-level part of the preconditioner and Q_{loc} the local-level part.

4.1. Local Level. The local part of the preconditioner basically involves the solution of local problems with natural or mixed boundary conditions (for floating and non-floating subdomains, respectively). Q_{loc} is defined by

$$Q_{loc} = \sum_{i=1}^{N} R_{\Gamma}^{(i)^T} \left[\begin{array}{cc} D^{(i)^{-1}} & 0 \\ 0 & 0 \end{array}\right] S^{(i)^{\dagger}} \left[\begin{array}{cc} D^{(i)^{-1}} & 0 \\ 0 & 0 \end{array}\right] R_{\Gamma}^{(i)}.$$

The dagger (†) above indicates a pseudo-inverse, since $S^{(i)}$ is singular on a floating subdomain (the nullspace being constant velocities for Stokes' equation and rigid-body displacements for elasticity). The matrices $D^{(i)^{-1}}$ are diagonal and determine a partition of unity on Γ. A proper choice of this partition is necessary for the method to be insensitive to jumps in the coefficients:

$$\left(D^{(i)^{-1}}\right)_{jj} = \frac{\mu_i^{\gamma}}{\displaystyle\sum_{x_j \in \partial\Omega_k} \mu_k^{\gamma}}\,, \quad \gamma \geq \frac{1}{2}.$$

In computing the action of Q_{loc}, it is useful to remember that

$$\left[\begin{array}{ccc} I & 0 \end{array}\right] S^{(i)\dagger} \left[\begin{array}{c} I \\ 0 \end{array}\right] = \left[\begin{array}{ccc} 0 & 0 & I \end{array}\right] \left[\begin{array}{ccc} A_{II}^{(i)} & \tilde{B}_I^{(i)^T} & A_{I\Gamma}^{(i)} \\ \tilde{B}_I^{(i)} & -\tilde{C}^{(i)} & \tilde{B}_\Gamma^{(i)^T} \\ A_{\Gamma I}^{(i)} & \tilde{B}_\Gamma^{(i)} & A_{\Gamma\Gamma}^{(i)} \end{array}\right]^\dagger \left[\begin{array}{c} 0 \\ 0 \\ I \end{array}\right].$$

Note that the matrix in the right side of the equation is a submatrix of (3.7). The coarse step preceding the local step ensures that the right-hand sides are consistent and in this case a good approximation for the pseudo-inverse can be obtained by perturbing the original system, replacing the original $A^{(i)}$ by $A^{(i)} + \varepsilon I$ or $A^{(i)} + \varepsilon M^{(i)}$, where $M^{(i)}$ is the local mass matrix for the displacement variables and ε is a small positive constant.

4.2. Coarse Level. The application of the coarse term Q_0 amounts to the solution of a coarse, global problem:

$$Q_0 \quad = \quad R_0^T (R_0 S R_0^T)^\dagger R_0,$$

where

$$R_0 = \left[\begin{array}{cc} L^T & 0 \\ 0 & I \end{array}\right].$$

The columns of the matrix R_0^T span the coarse space: the identity block corresponds to the coarse pressures, one per subdomain; the displacement coarse space is determined by the columns of the matrix L. In order to ensure solvability of the local problems with natural boundary conditions, L must contain the traces of the elements of a basis of the nullspace of $A^{(i)}$ scaled by $D^{(i)^{-1}}$ for all i corresponding to the floating subdomains (cf. subsection 4.1). These scaled rigid-body displacements can also be added for non-floating subdomains, as long as care is taken to avoid linearly dependence; this can be accomplished by dropping the contribution of one non-floating subdomain.

In order to obtain an inf-sup stable coarse space, we need to enrich L further. Two alternatives are: adding the traces of either the coarse bi/tri-linear functions (the space Q_1^H) or the quadratic coarse edge/face bubble functions for the normal directions.

Remark We can show that QS is positive-definite on range $(I - Q_0 S)$. If an initial guess is chosen such that the initial error is in range $(I - Q_0 S)$, then the error on every step of a Krylov method will also be restricted to range $(I - Q_0 S)$, since $Q_0 S$ is a projection. The importance of this observation is that it allow us to use the preconditioned conjugate gradient method as our iterative solver, even though our original operator is indefinite.

5. Theoretical Bound. A theoretical bound for the condition number of the preconditioned operator QS restricted to the appropriate subspace to which the iterates are confined is proved in [7], [4], for the constant coefficient case:

$$\kappa \leq C \left(1 + \log\left(\frac{H}{h}\right)\right)^2.$$

We note that κ does not depend on the number of subdomains and depends only polylogarithmically on the size of the subdomain problems. The constant C depends, in the incompressible or quasi-incompressible cases, on the inf-sup constants of the original and coarse spaces. This is the reason why we enrich the displacement coarse space to achieve inf-sup stability.

Figure 6.1: material properties of a heterogeneous problem.

s	r	s	r	$\cdots$	s	r
r	a	r	a	$\cdots$	r	a
s	r	s	r	$\cdots$	s	r
r	a	r	a	$\cdots$	r	a
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\vdots$
s	r	s	r	$\cdots$	s	r
r	a	r	a	$\cdots$	r	a

$s-$ steel-like
$a-$ aluminium-like
$r-$ rubber-like

$\mu_s = 8.20 \quad \lambda_s = 10.00 \quad \nu_s = 0.275$
$\mu_a = 2.60 \quad \lambda_a = 5.60 \quad \nu_a = 0.341$
$\mu_r = 0.01 \quad \lambda_r = 0.99 \quad \nu_r = 0.495$

6. Numerical Experiments. Our algorithm has been implemented in C, using the PETSc library (see [1], [2]). Parallel numerical experiments were run on the Linux cluster Chiba City at Argonne National Laboratory (with 256 Dual Pentium III processors with 512MB of local RAM). We report on results for compressible/almost-incompressible elasticity only, although similar results have been obtained for incompressible elasticity, Stokes and generalized Stokes equations.

We consider an elasticity problem defined on a square heterogeneous domain, which is composed of an arrangement of three different materials in the pattern depicted in figure 6.1. Note that the material r is almost incompressible, with a Poisson ratio close to 0.5. The problem is discretized with $Q_2 - Q_0$ finite elements and the domain Ω divided into $\sqrt{N} \times \sqrt{N}$ square subdomains, each of them composed of a single material. The saddle point Schur complement (3.3) is solved iteratively by PCG with our balancing Neumann-Neumann preconditioner and the coarse space $V_0 = \{\text{scaled rigid body motions}\} + Q_1^H$. The initial guess is a random vector modified so that the initial error is in the range of $(I - Q_0 S)$, the right hand side is a random, uniformly distributed vector, and the stopping criterion is $\|r_k\|_2 / \|r_0\|_2 \leq 10^{-6}$, where r_k is the residual at the $k-$th iterate.

In the lower half of Table 6.1, we show the results for increasing mesh sizes, always with 64 subdomains. The condition number and the iteration count grow weakly as we increase the size of the local problems, as can also be observed in the left part of Figure 6.2.

The last two columns of this table display CPU-times for these runs. The last column gives the total time for the code to run, while the column labeled "fact." gives the time spent on LU factorizations; there are three of them: two local, namely Dirichlet and Neumann subdomain-level problems, and one global coarse problem. We note that the cost of the factorizations grows rapidly and dominates the cost of the computation. The upper part of Table 6.1 shows results for an increasing number of subdomains of fixed size (about 58,000 degrees of freedom). The corresponding graph, on the right in Figure 6.2, shows an almost horizontal tail, indicating independence of the condition number and the iteration count on the number of subdomains. This is numerical evidence that our result in section 5 remains valid in the case of discontinuous coefficients. The fact that the factorization time remains constant for the entire range of problem sizes tested (from 16 to 169 subdomains) indicates that the cost associated with the factorization of the coarse problem is still tiny compared with that of the local problems. One can expect this scenario to change if the number of subdomains increases significantly.

REFERENCES

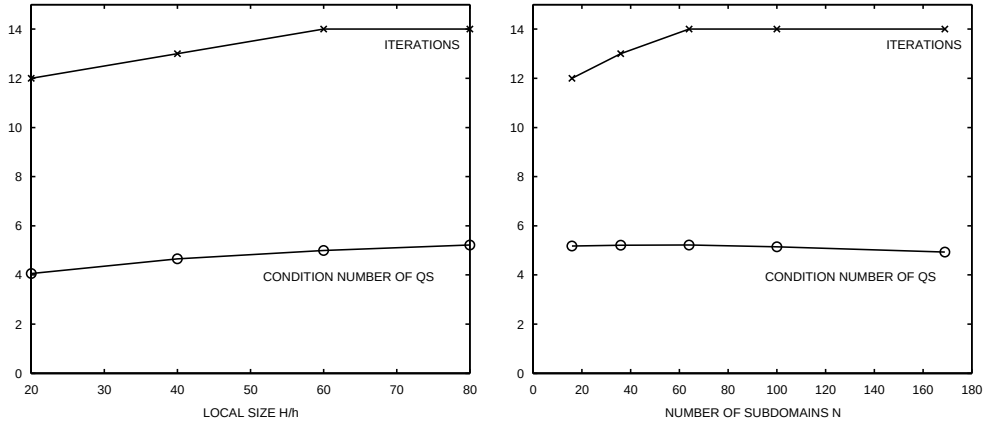
[1] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, L. C. McInnes, and B. F. Smith. PETSc home page. http://www.mcs.anl.gov/petsc, 2001.

[2] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. PETSc users manual. Technical Report ANL-95/11 - Revision 2.1.1, Argonne National Laboratory, 2001.

Table 6.1: results for elasticity problem in heterogeneous medium with $Q_2 - Q_0$ finite elements and $V_0 = (\text{scaled rigid-body motions}) + \left(Q_1^H\right)^2$. The iterative method is PCG and the termination criterion is $\|r_{\text{final}}\| \leq 10^{-6} \|r_0\|$. The initial guess and right-hand side are randomly generated. The ordering of the displacement variables is determined by quasi-minimal degree.

fixed H/h, local problem with 80×80 elements (58,242 dof)						
grid size	# subd.	dof	iter.	cond.	time (sec.)	
(# elem.)		($\times 10^6$)			fact.	other
320×320	4×4	0.92	12	5.14	258.0	63.4
480×480	6×6	2.08	13	5.12	253.7	63.7
640×640	8×8	3.69	14	5.13	260.8	84.5
800×800	10×10	5.77	14	5.06	262.8	93.9
1040×1040	13×13	9.74	14	4.87	261.2	102.7
fixed number of subdomains $N = 8 \times 8$						
grid size	loc. dof	dof	iter.	cond.	time (sec.)	
(# elem.)	($\times 10^3$)	($\times 10^6$)			fact.	other
160×160	3.8	0.23	12	4.00	1.4	16.7
320×320	14.7	0.92	13	4.57	18.2	22.7
480×480	32.9	2.08	14	4.91	84.2	42.1
640×640	58.2	3.69	14	5.13	260.8	84.5

Figure 6.2: results for elasticity problem in heterogeneous medium with $Q_2 - Q_0$ finite elements: PCG iteration count and condition number of QS vs. local size H/h (left) and number of subdomains N (right), from Table 6.1.



[3] P. Goldfeld, L. F. Pavarino, and O. B. Widlund. Balancing Neumann-Neumann methods for mixed approximations of linear elasticity. In L. F. Pavarino and A. Toselli, editors, *Recent Developments in Domain Decomposition Methods*, pages 53–76. Springer-Verlag, 2002.

[4] P. Goldfeld, L. F. Pavarino, and O. B. Widlund. Balancing Neumann-Neumann preconditioners for mixed approximations of heterogeneous problems in linear elasticity. Technical Report 825, Courant Institute of Mathematical Sciences, Computer Science Department, April 2002.

[5] J. Mandel. Balancing domain decomposition. *Comm. Numer. Meth. Engrg.*, 9:233–241, 1993.

[6] J. Mandel and M. Brezina. Balancing Domain Decomposition for Problems with Large Jumps in Coefficients. *Math. Comp.*, 65:1387–1401, 1996.

[7] L. F. Pavarino and O. B. Widlund. Balancing Neumann-Neumann methods for incompressible Stokes equations. *Comm. Pure Appl. Math.*, 55(3):302–335, 2002.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

19. A FETI-DP Corner Selection Algorithm for three-dimensional problems

M. Lesoinne[1]

1. Introduction. The FETI-DP algorithm is a numerically scalable iterative domain decomposition method for static and dynamic problems. It was first derived as an alternative to the two-level FETI method for fourth order problems [1] and later extended to three dimensional second order problems [5, 2]. Later, several authors have showed that FETI-DP is scalable for scalar and mechanical problems [6] even in the presence of heterogeneities [4].

As it is derived from the two-level FETI method for fourth order problems, the choice of corner in such problems has to follow the same rules [3], however, for second order, three dimensional problems, the FETI-DP implementations remain flexible on the choice of corners. However a few constraints have to be placed on their choices, so that the resulting subdomain matrices and the resulting coarse problem is non-singular.

This article describes a robust algorithm for the selection of corners for three-dimensional problems that guarantees that none of the matrices involved in the FETI operator will be singular.

2. The Dual-Primal FETI Method. Let Ω be partitioned into a set of N_s, non-overlapping subdomains (or substructures) Ω^s. Select a set of points called corner points on which the degrees of freedom will remain primal variable. The mechanical interpretation of this particular method of mesh splitting can be viewed as making incisions into the mesh but leaving the corner points attached. This is analogous to the "tearing" stage of FETI. The "interconnecting" stage occurs only on the subdomain interfaces which now excludes the corner points (see Figure 2.1). By splitting, u^s into two sub-vectors such that:

$$u = \begin{bmatrix} u_r \\ u_c \end{bmatrix} = \begin{bmatrix} u_r^1 \\ \vdots \\ u_r^{N_s} \\ u_c \end{bmatrix} \tag{2.1}$$

where u_r^s is the remaining subdomain solution vector and u_c is a global/primal solution vector over all defined corner degrees of freedom. The solution at the corner points is continuous by definition when the solution vector is constructed in this manner. Using this notation, we can split the subdomain stiffness matrix into:

$$K^s = \begin{bmatrix} K_{rr}^s & K_{rc}^s \\ K_{rc}^{s^T} & K_{cc}^s \end{bmatrix} \tag{2.2}$$

[1]Assistant Professor, Department of Aerospace Engineering and Sciences and Center for Aerospace Structures University of Colorado at Boulder Boulder, CO 80309-0429, U.S.A. Email: Michel@Colorado.EDU

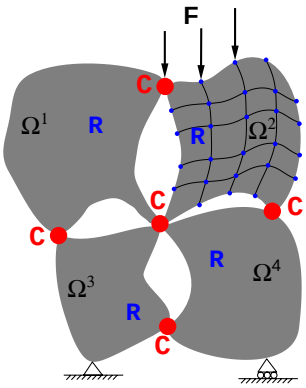


Figure 2.1: Dual-Primal Mesh Partitions

Then the FETI-DP equilibrium equations can be written using the following matrix partitioning where the subscripts c and r denote the corner and the remainder degrees of freedom.

$$\begin{bmatrix} K_{rr}^1 & \dots & 0 & K_{rc}^1 B_c^1 \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \dots & K_{rr}^{N_s} & K_{rc}^{N_s} B_c^{N_s} \\ B_c^{1^T} K_{rc}^{1^T} & \dots & B_c^{N_s{}^T} K_{rc}^{N_s{}^T} & \sum_{s=1}^{N_s} B_c^{s^T} K_{cc}^s B_c^s \end{bmatrix} \begin{bmatrix} u_r^1 \\ \vdots \\ u_r^{N_s} \\ u_c \end{bmatrix} = \begin{bmatrix} f_r^1 - B_r^{1^T} \lambda \\ \vdots \\ f_r^{N_s} - B_r^{N_s{}^T} \lambda \\ \sum_{s=1}^{N_s} B_c^{s^T} f_c^s \end{bmatrix} \tag{2.3}$$

While the compatibility equations of interface displacements take the form:

$$\sum_{s=1}^{N_s} B_r^s u_r^s = 0 \tag{2.4}$$

In the preceeding, the corner stiffness matrix, $K_{cc} = \sum_{s=1}^{N_s} B_c^{s^T} K_{cc}^s B_c^s$ is a global stiffness quantity, B_c^s maps the local corner equation numbering to global corner equation numbering, f_r^s is the external force applied on the r degrees of freedom, $B_r^{s^T}$ is a boolean matrix that extracts the interface of a subdomain, and λ are the Lagrange multipliers.

Let K_{rr} denote the block diagonal subdomain stiffness matrix restricted to the remaining, r, points, K_{rc} the block column vector of *r-c* coupling stiffness matrices, f_r the block column vector of subdomain force vectors, K_{cc} the global corner stiffness matrix and using the "rc" notation, we can rewrite the equilibrium compatibility equations in the more compact form:

$$\begin{bmatrix} K_{rr} & K_{rc} & B_r^T \\ K_{rc}^T & K_{cc} & 0 \\ B_r & 0 & 0 \end{bmatrix} \begin{bmatrix} u_r \\ u_c \\ \lambda \end{bmatrix} = \begin{bmatrix} f_r \\ f_c \\ 0 \end{bmatrix} \tag{2.5}$$

In this formulation, the FETI-DP operator is a schur-complement obtained by eliminating the u_r and u_c degrees of freedom. The elimination of the u_r degrees of freedom is a subdomain per subdomain operation, while the elimination of the u_c degrees of freedom is a global operation

that provides the FETI-DP operator with a coarse problem, coupling all the subdomains together.

Though this approach is scalable for two-dimensional problems and for plates and shells, it was shown that for second order three-dimensional problems, an augmented coarse problem is necessary.

The augmented FETI-DP system is obtained by adding new coarse degrees of freedom in the form of new Lagrange multipliers μ that are used to guarantee that at each iteration, the residual is orthogonal to a subspace Q:

$$Q^T B u_r = 0 \tag{2.6}$$

Thus leading to the system of equations:

$$\left(\begin{array}{c|cc|c} K_{rr} & K_{rc} & B^tQ & B^t \\ \hline K_{cr} & K_{cc} & 0 & 0 \\ Q^tB & 0 & 0 & 0 \\ \hline B & 0 & 0 & 0 \end{array}\right) \left(\begin{array}{c} u_r \\ \hline u_c \\ \mu \\ \hline \lambda \end{array}\right) = \left(\begin{array}{c} f_r \\ \hline f_c \\ 0 \\ \hline 0 \end{array}\right) \tag{2.7}$$

In this set of equations, the first line is the set of subdomain by subdomain equations while the second and third lines represent the coarse problem which is global. By doing the Schur complement of the u_r equations, we obtain the coarse matrix:

$$\widetilde{K}_{cc} = \begin{pmatrix} K_{cc} - K_{cr}K_{rr}^{-1}K_{rc} & -K_{cr}K_{rr}^{-1}B^tQ \\ -Q^tBK_{rr}^{-1}K_{rc} & -Q^tBK_{rr}^{-1}B^tQ \end{pmatrix} \tag{2.8}$$

3. Preliminary Observations. There are two essential conditions that the corner selection should satisfy:

1. Each subdomain stiffness matrix should be non singular.
2. The resulting coarse problem matrix should be non singular.

Additionally, as they do not contribute significantly to the convergence rate, keeping the number of corner nodes low reduces the overall cost of the computation and improves its scalability.

3.1. Non-Singular $K_{rr}^{(s)}$. The non singularity of each subdomain $K_{rr}^{(s)}$ can be guaranteed simply by making sure that every subdomain has either 3 non-colinear corner nodes in 3 dimensions or 2 non-coincidental corner nodes in 2 dimensions.

3.2. Non-Singular $\widetilde{K}_{cc}$ and Pivoting. As presented here, the FETI-DP method only requires that $\widetilde{K}_{cc}$ be non singular for the corner degrees of freedom. However this matrix is not positive and without pivoting, zero diagonal terms could appear during the factorization on one of the corner degree of freedom. It is to be noticed that a singularity on one of the augmentation degree of freedom can be dealt with simply by eliminating the augmentation degree of freedom. Such an occurrence only affects the convergence rate but does not otherwise adversely affect the method. However it is imperative that no singularity appears on the corner degrees of freedom.

We note that is is always possible to deal with the occurrence of a zero pivot in the factorization of K_{cc}^*, the corner node portion of $\widetilde{K}_{cc}$ by using pivoting if we assume the global coarse matrix to be non-singular. However pivoting solver are generally complex and usually have a slightly lower performance when compared with non-pivoting solver. Moreover, guaranteeing that the augmentation correctly addresses the singularity in K_{cc}^* is by no means a trivial task.

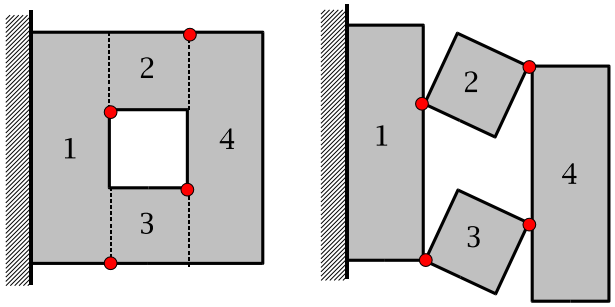


Figure 4.1: Coarse Problem Mechanism

Fortunately, it can be guaranteed that no zero pivot will appear on the corner degree of freedoms if K^*_{cc} is non singular. Because of this simple remark, we propose to build our corner selection algorithm to guarantee that K^*_{cc} is non singular. This choice will make pivoting unnecessary and consequently simplify the implementation and improve the performance of the code.

3.3. Subdomains as Super-Elements. In order to facilitate the discussion of the non-singularity of K^*_{cc}, we first notice that

$$K^*_{cc} = \sum_{s=1}^{N_{sub}} K^{(s)}_{cc} - K^{(s)}_{cr} K^{-1}_{rr(s)} K^{(s)}_{rc} \tag{3.1}$$

is an assembly of subdomain as Super-Elements where only the corner nodes are kept for attaching subdomains together. We will assume in what follows that every subdomain created by the decomposer is free of any internal mechanism. This is to say that in three dimensions, each subdomain, before the application of any boundary condition has exactly 6 rigid body modes, while in two dimensions, each subdomain has exactly 3 rigid body modes.

4. An Ad-Hoc Algorithm. In our early implementation of the FETI-DP algorithm, we extended the two-dimensional view of corners to three-dimensions by using the following algorithm:

1. Pick nodes with more than 4 neighbors as corner nodes
2. Post-guarantee the non singularity of K_{rr}

Unfortunately this algorithm generally leads to a large number of corners and more importantly, it does not offer any guarantee as to the non-singularity of the $\widetilde{K}_{cc}$ matrix. Figure 4.1 shows a two dimensional example. In this problem there are no points where three or more subdomains meet. Therefore, the corners have been chosen to guarantee the non singularity of all the subdomains – i.e. in this case, at least two corner nodes per subdomain. It can be seen that the resulting system has a spurious mechanism.

5. A Robust Algorithm. To keep the following discussion clear, let us introduce two important definitions:

Mechanism-Free entity: a set of elements such that when combined together, there is no mechanism between any part of the set

Subdomain to Subdomain Face: the set of nodes shared by two given subdomains

Under our assumption about the decomposer, every subdomain is a *Mechanism-Free entity*.

It is easy to check that, using only corner nodes to attach subdomains together, two *Mechanism-Free entities* can be combined into a single composite *Mechanism-Free entity* if they share at least 3 non colinear corner nodes in three dimensions or 2 non colocated nodes in two dimensions.

We also note that when a subdomain is merged with any *Mechanism-Free entity*, it will guarantee that its local $K_{rr}^{(s)}$ matrix is non singular.

Thus, by recursively combining pairs of *Mechanism-Free entities* until the whole set of subdomains has been merged into a single entity, we can attain our goals, thus leading to:

Corner Selection Algorithm

1. Mark Corner Candidates on Each Subdomain Face
2. Declare Each Subdomain a Mechanism-Free Entity
3. Iterate Until all Subdomain are Assembled into a Single Entity:
   (a) For each Entity, Choose 2 Preferred Neighboring Entities by:
      i. Favoring Already Picked Corners.
      ii. Maximizing the Area Formed by the Corner Nodes Joining the 2 Entities.
   (b) Check if Previous Choices of Corner Create a Tie between Entities
   (c) Merge Entities, Favoring Pre-Existing ties, then Paring

In the first step of the algorithm, we pick candidate corner nodes from which all corner nodes will be chosen. In most three-dimensional problems, the faces between neighboring subdomain are two-dimensional and therefore if we have at least three non-colinear corner nodes on such faces, we can guarantee that we can tie each subdomain to a neighbor as a Mechanism-Free Entity, guaranteeing by the same operation that the subdomain $K_{rr}^{(s)}$ will be non-singular.

We will note that there are some special cases to deal with. It is possible to have a structure in which subdomains are attached by faces that are all two dimensions lower than the dimension of the problem –i.e. single nodes in 2 dimensions. This means that even when use all the potential corner nodes, it is not possible to tie two subdomains into a single *Mechanism-Free entity* just by one face. In such a case, the algorithm may end up with a final set of *Mechanism-Free entities* that it cannot guarantee can be tied into a single one. In such a case, we will take all the remaining corner candidates shared by at least two entities as corners. Assuming the global problem was mechanism-free, the resulting choice will guarantee the non singularity of K_{cc}^{*}. If the resulting K_{cc}^{*} remains singular however, we will conclude that the global problem was singular and an error can be generated for the user.

6. Numerical Results. We present two numerical examples of large three-dimensional structures. The first model is of a car engine component and has 985,340 degrees of freedom. It is made of four noded tetrahedra (see Figure 6). The second model, illustrated in Figures 6 has 2,437,104 degrees of freedom. Both models were run first with the Ad-Hoc algorithm and secondly with new algorithm. The results show the number of corners generated, the total number of degrees of freedom of the coarse problem, the memory used by the $\widetilde{K}_{cc}$ matrix, the number of iterations and the total elapsed time for the solution.

Figure 6.1: Engine Gas Collector Geometry

Algorithm	Number of Corners	Coarse Pb Size	Memory Usage	Iteration Count	Solution Time
Old	2,285	15,692	60MB	38	501s
New	1,571	13,679	44MB	41	490s

Table 6.1: Results for the Engine Gas Collector

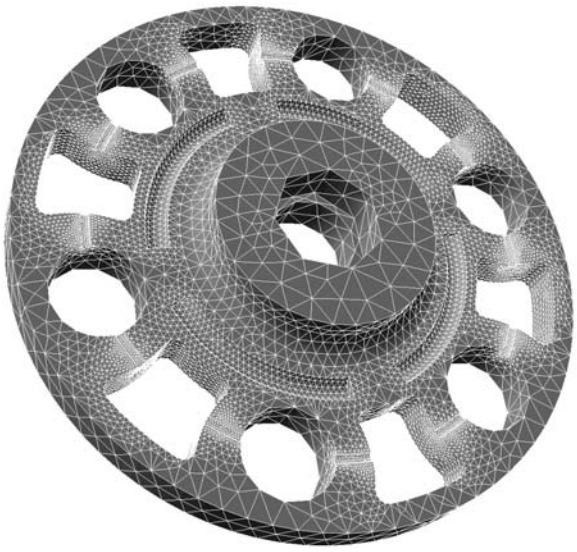


Figure 6.2: Wheel Carrier Geometry

Algorithm	Number of Corners	Coarse Pb Size	Memory Usage	Iteration Count	Solution Time
Old	3,163	28,572	154MB	104	1272s
New	2,210	25,095	118MB	104	1218s

Table 6.2: Results for the Wheel Carrier Problem

The engine component was run using 8 CPUs on an SGI Origin 2000 machine. The results show that the number of corner nodes was reduced by roughly 30% while the total number of degree of freedom in the coarse problem is reduced by 12.5% This reduction leads to a saving of memory of 27% We observe a slight but increase in the number of iterations to reach the solution, however the smaller coarse problem leads to a lower cost per iteration and a shorter factorization of the coarse matrix. As a result, the overall timing is about 2% faster.

The wheel carrier shows similar effects. The reduction in number of corner is again roughly one third while the reduction in number of coarse degrees of freedom is lower. In this case, the number of iterations remains unaffected by the number of corners and the overall execution time is faster with the smaller coarse problem.

7. Conclusions. We have presented an algorithm for the selection of corner nodes for three-dimensional problems for the FETI-DP algorithm. This algorithm offers the benefit over the previous Ad-Hoc algorithm of guaranteeing that no zero pivot will appear during the factorization of the coarse problem.

With two large examples of three-dimensional problems, it was shown that the improved algorithm leads to a reduction of number of corners by roughly one third and is accompanied by a very small decrease of convergence rate. However, the coarse problem matrix is smaller and the resulting reduction in factorization cost as well as a reduction in the cost for the solution of the coarse problem at each iteration results in a slight reduction of the overall solution time.

REFERENCES

[1] P. L. K. P. C. Farhat, M. Lesoinne and D. Rixen. Feti-dp: A dual-primal unified feti method - part i: A faster alternative to the two-level feti method. *International Journal for Numerical Methods in Engineering*, 2000. in press.

[2] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.

[3] C. Farhat and J. Mandel. The two-level feti method for static and dynamic plate problems - part i: an optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering*, 155:129–152, 1998.

[4] A. Klawonn, O. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional elliptic problems with heterogeneous coefficients. Technical Report 815, Courant Institute of Mathematical Sciences, Department of Computer Science, April 2001. to appear in SIAM J. Numer. Anal.

[5] M. Lesoinne and K. Pierson. Feti-dp: An efficient, scalable, and unified dual-primal feti method. In T. Chan, T. Kako, H. Kawarada, and O. Pironneau, editors, *Domain Decomposition Methods in Sciences and Engineering*, pages 421–428. DDM.org, 1999.

[6] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

20. A Dual-Primal FETI Method for solving Stokes/Navier-Stokes Equations

Jing Li[1]

1. Introduction. The Dual-Primal Finite Element Tearing and Interconnecting (FETI-DP) methods were first proposed by Farhat et al [3] for elliptic partial differential equations. In this method, the spatial domain is decomposed into non-overlapping subdomains, and the interior subdomain variables are eliminated to form a Schur complement problem for the interface variables. Lagrange multipliers are then introduced to enforce continuity across the interface, except at the subdomain vertices where continuity is enforced directly, i.e., the neighboring subdomains share the degrees of freedom at the subdomain vertices. A symmetric positive semi-definite linear system for the Lagrange multipliers is solved by using the preconditioned conjugate gradient (PCG) method. FETI-DP methods have been shown to be numerically scalable for second order elliptic problems. Thus, Mandel and Tezaur [6] have proved that the condition number grows at most as $C(1+log(H/h))^2$ in two dimensions, where H is the subdomain diameter and h is the element size. Klawonn et al [4] proposed new preconditioners of this type and proved that the condition numbers are bounded from above by $C(1+log(H/h))^2$ in three dimensions; these bounds are also independent of possible jumps of the coefficients of the elliptic problem. In [5], we developed a dual-primal FETI method for the two-dimensional incompressible Stokes problem and proved that the condition number is bounded from above by $C(1+log(H/h))^2$. In this paper, we will extend this algorithm to solving three-dimensional incompressible Stokes problem, give the same condition number bound and an inf-sup stability result for the coarse level saddle point problem, which appeared as an assumption in [5]. We will also extend this dual-primal FETI algorithm to solving nonlinear Navier-Stokes equations by using a Picard iteration, where in each iteration step, we will solve a non-symmetric linearized incompressible Navier-Stokes equation. Illustrative numerical results are presented by solving lid driven cavity problems.

2. FETI-DP algorithm for Stokes problem. We will consider the following Stokes problem on a three-dimensional, bounded, polyhedral domain Ω,

$$\left\{\begin{array}{rcll} -\Delta\mathbf{u}+\nabla p &=& \mathbf{f}, & \text{in } \Omega \\ -\nabla\cdot\mathbf{u} &=& 0, & \text{in } \Omega \\ \mathbf{u} &=& \mathbf{g}, & \text{on } \partial\Omega\ , \end{array}\right. \tag{1}$$

where the boundary velocity $\mathbf{g}$ satisfies the compatibility condition $\int_{\partial\Omega}\mathbf{g}\cdot\mathbf{n}=0$. The domain Ω is decomposed into N non-overlapping polyhedral subdomains Ω^i of characteristic size H. The interface is defined as $\Gamma=(\cup\partial\Omega^i)\backslash\partial\Omega$ and $\Gamma^{ij}=\partial\Omega^i\cap\partial\Omega^j$ is the interface between two neighboring subdomains Ω^i and Ω^j. We will consider subdomain incompressible Stokes problems,

$$\left\{\begin{array}{rcll} -\Delta\mathbf{u}^i+\nabla p^i &=& \mathbf{f}^i, & \text{in } \Omega^i \\ -\nabla\cdot\mathbf{u}^i &=& 0, & \text{in } \Omega^i \\ \mathbf{u}^i &=& \mathbf{g}^i, & \text{on } \partial\Omega\cap\partial\Omega^i \\ \frac{\partial\mathbf{u}^i}{\partial\mathbf{n}^i}-p^i\mathbf{n}^i &=& \lambda^i, & \text{on } \Gamma^{ij}\ , \end{array}\right.$$

$$\left\{\begin{array}{rcll} -\Delta\mathbf{u}^j+\nabla p^j &=& \mathbf{f}^j, & \text{in } \Omega^j \\ -\nabla\cdot\mathbf{u}^j &=& 0, & \text{in } \Omega^j \\ \mathbf{u}^j &=& \mathbf{g}^j, & \text{on } \partial\Omega\cap\partial\Omega^j \\ \frac{\partial\mathbf{u}^j}{\partial\mathbf{n}^j}-p^j\mathbf{n}^j &=& \lambda^j, & \text{on } \Gamma^{ij}\ , \end{array}\right.$$

[1]Courant Institute of Mathematical Sciences, lijing@cims.nyu.edu. This work was supported in part by the U.S. Department of Energy under contract DE-FG02-92ER25127.

where $\lambda^i + \lambda^j = 0$. We first form subdomain discrete problems by using an inf-sup stable mixed finite element method on each subdomain. We denote the discrete finite element space for the pressures inside the subdomain Ω^i by Π_I^i, and the subdomain constant pressure space by Π_0. We denote the discrete finite element space for the velocity components on Ω^i by $\mathbf{W}^h(\Omega^i)$, which is decomposed as $\mathbf{W}^h(\Omega^i) = \mathbf{W}_I^i \oplus \mathbf{W}_\Gamma^i$, with $\mathbf{W}_I^i$ the interior velocity part and $\mathbf{W}_\Gamma^i$ the subdomain boundary velocity part. Let $\Pi_I = \prod_{i=1}^N \Pi_I^i$, $\mathbf{W}_I = \prod_{i=1}^N \mathbf{W}_I^i$, and $\mathbf{W}_\Gamma = \prod_{i=1}^N \mathbf{W}_\Gamma^i$ be the corresponding product spaces. $\widetilde{\mathbf{W}}_\Gamma$ is a subspace of $\mathbf{W}_\Gamma$ and is given by

$$\widetilde{\mathbf{W}}_\Gamma = \mathbf{W}_\Pi \oplus \mathbf{W}_\Delta,$$

where the primal subspace $\mathbf{W}_\Pi$ consists of two parts. The first is the subdomain corner velocity part, which is spanned by the nodal finite element basis function $\theta_{\mathcal{V}^{il}}$ of the subdomain corners. The other part corresponds to the integrals of the velocity over each subdomain interface, and it is spanned by the pseudoinverse $\mu_{\mathcal{F}^{ij}}^\dagger$ of the counting functions $\mu_{\mathcal{F}^{ij}}$ corresponding to each face $\mathcal{F}^{ij}$ of the subdomain Ω^i: $\mu_{\mathcal{F}^{ij}}$ is 0 at the interface nodes outside $\bar{\mathcal{F}}^{ij}$ while its value at any node on $\mathcal{F}^{ij}$ equals the number of subdomains shared by that node. Its pseudoinverse $\mu_{\mathcal{F}^{ij}}^\dagger$ is the function $1/\mu_{\mathcal{F}^{ij}}(x)$ for all interface nodes where $\mu_{\mathcal{F}^{ij}}(x) \neq 0$, and it vanishes at all other points. We also note that, we make both $\mu_{\mathcal{F}^{ij}}$ and $\mu_{\mathcal{F}^{ij}}^\dagger$ vanish at the subdomain corners. $\mathbf{W}_\Delta$ is the dual part, which is the direct sum of the local subspaces $\mathbf{W}_\Delta^i$. In the 3D case,

$$\mathbf{W}_\Delta^i := \{\mathbf{w} \in \mathbf{W}_\Gamma^i : \mathbf{w}(\mathcal{V}^{il}) = 0; \bar{\mathbf{w}}_{\mathcal{F}^{ij}} = 0,\ \forall \mathcal{V}^{il}, \mathcal{F}^{ij} \subset \partial\Omega^i\},$$

with $\bar{\mathbf{w}}_{\mathcal{F}^{ij}}$ defined by

$$\bar{\mathbf{w}}_{\mathcal{F}^{ij}} = \frac{\int_{\mathcal{F}^{ij}} \mathbf{w} d\mathbf{x}}{\int_{\mathcal{F}^{ij}} d\mathbf{x}}.$$

With these notations, we can decompose the discrete velocity and pressure space of the original problem (1) as follows

$$\mathbf{W} = \mathbf{W}_I \oplus \mathbf{W}_\Pi \oplus \mathbf{W}_\Delta,$$

and

$$\Pi = \Pi_I \bigoplus \Pi_0.$$

If we further introduce a Lagrange multiplier space Λ to enforce the continuity of the velocities across the subdomain interfaces, then we have the following discrete problem: find a vector $(\mathbf{u}_I, p_I, \mathbf{u}_\Pi, p_0, \mathbf{u}_\Delta, \lambda) \in (\mathbf{W}_I, \Pi_I, \mathbf{W}_\Pi, \Pi_0, \mathbf{W}_\Delta, \Lambda)$ such that

$$\begin{pmatrix} A_{II} & B_{II}^T & A_{\Pi I}^T & 0 & A_{\Delta I}^T & 0 \\ B_{II} & 0 & B_{\Pi I} & 0 & B_{\Delta I} & 0 \\ A_{\Pi I} & B_{\Pi I}^T & A_{\Pi\Pi} & B_{\Pi 0}^T & A_{\Delta\Pi}^T & 0 \\ 0 & 0 & B_{\Pi 0} & 0 & 0 & 0 \\ A_{\Delta I} & B_{\Delta I}^T & A_{\Delta\Pi} & 0 & A_{\Delta\Delta} & B_\Delta^T \\ 0 & 0 & 0 & 0 & B_\Delta & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ p_I \\ \mathbf{u}_\Pi \\ p_0 \\ \mathbf{u}_\Delta \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f}_I \\ 0 \\ \mathbf{f}_\Pi \\ 0 \\ \mathbf{f}_\Delta \\ 0 \end{pmatrix}. \tag{2}$$

It is important to note that the B_Δ matrix here is a scaled matrix with elements given by $\{0, \pm\sqrt{\mu_{\mathcal{F}^{ij}}^\dagger}\}$ placing different weights on the face and edge nodes, unlike in the two-dimensional case where B_Δ is constructed from $\{0, \pm 1\}$. It follows immediately from the definition of B_Δ that, on each subdomain interface $\mathcal{F}^{ij}$,

$$(B_\Delta^T B_\Delta \mathbf{w})^i|_{\mathcal{F}^{ij}} = \pm(\mu_{\mathcal{F}^{ij}}^\dagger(\mathbf{w}^i - \mathbf{w}^j))|_{\mathcal{F}^{ij}}\ , \forall \mathbf{w} \in \mathbf{W}_\Gamma. \tag{3}$$

Also note that we are not requiring the pressure to be continuous across the subdomain interfaces in our algorithm. In fact, we consider only mixed methods with discontinuous pressure spaces. By defining a Schur complement operator $\tilde{S}$ as

$$\begin{pmatrix} A_{II} & B_{II}^T & A_{\Pi I}^T & 0 & A_{\Delta I}^T \\ B_{II} & 0 & B_{\Pi I} & 0 & B_{\Delta I} \\ A_{\Pi I} & B_{\Pi I}^T & A_{\Pi\Pi} & B_{\Pi 0}^T & A_{\Delta\Pi}^T \\ 0 & 0 & B_{\Pi 0} & 0 & 0 \\ A_{\Delta I} & B_{\Delta I}^T & A_{\Delta\Pi} & 0 & A_{\Delta\Delta} \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ p_I \\ \mathbf{u}_\Pi \\ p_0 \\ \mathbf{u}_\Delta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \tilde{S}\mathbf{u}_\Delta \end{pmatrix}, \tag{4}$$

solving the linear system (2) is reduced to solving the following linear system

$$\begin{pmatrix} \tilde{S} & B_\Delta^T \\ B_\Delta & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_\Delta \\ \lambda \end{pmatrix} = \begin{pmatrix} \mathbf{f}_\Delta^* \\ 0 \end{pmatrix} . \tag{5}$$

By using a further Schur complement procedure, the problem is finally reduced to solving the following linear system with the Lagrange multipliers λ as the remaining variable:

$$B_\Delta \tilde{S}^{-1} B_\Delta^T \lambda = B_\Delta \tilde{S}^{-1} \mathbf{f}_\Delta^*, \tag{6}$$

Our preconditioner is the standard Dirichlet preconditioner, $B_\Delta S_\Delta B_\Delta^T$, with S_Δ defined as

$$\begin{pmatrix} A_{II} & B_{II}^T & A_{\Delta I}^T \\ B_{II} & 0 & B_{\Delta I} \\ A_{\Delta I} & B_{\Delta I}^T & A_{\Delta\Delta} \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ p_I \\ \mathbf{u}_\Delta \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ S_\Delta \mathbf{u}_\Delta \end{pmatrix} . \tag{7}$$

We have now formed the preconditioned linear system

$$B_\Delta S_\Delta B_\Delta^T B_\Delta \tilde{S}^{-1} B_\Delta^T \lambda = B_\Delta S_\Delta B_\Delta^T B_\Delta \tilde{S}^{-1} f_\Delta^* , \tag{8}$$

which is our FETI-DP algorithm to solve the incompressible Stokes problem (1). In [5], we show that both S_Δ and $\tilde{S}^{-1}$ are symmetric, positive definite on the space $\mathbf{W}_\Delta$. Therefore a preconditioned conjugate gradient method, as well as GMRES, can be used to solve equation (8). We note that we need to apply both S_Δ and $\tilde{S}^{-1}$ to a vector in each iteration step. Multiplying S_Δ by a vector requires solving subdomain incompressible Stokes problems with Dirichlet boundary conditions, and multiplying $\tilde{S}^{-1}$ by a vector requires solving a coarse level saddle point problem, as well as subdomain problems. In [5], we made an assumption about the inf-sup stability condition of the coarse level saddle point problem. In the next section we will give an inf-sup stability estimate as well as a condition number bound of the preconditioned linear system (8).

3. Inf-sup stability of the coarse saddle point problem and a condition number estimate. We know, from the definition (4), that to find a vector $\mathbf{u}_\Delta = \tilde{S}^{-1} \cdot \mathbf{w}_\Delta \in \mathbf{W}_\Delta$, for a given $\mathbf{w}_\Delta \in \mathbf{W}_\Delta$, requires solving the following linear system

$$\begin{pmatrix} A_{II} & A_{\Delta I}^T & B_{II}^T & A_{\Pi I}^T & 0 \\ A_{\Delta I} & A_{\Delta\Delta} & B_{\Delta I}^T & A_{\Pi\Delta}^T & 0 \\ B_{II} & B_{\Delta I} & 0 & B_{\Pi I} & 0 \\ A_{\Pi I} & A_{\Pi\Delta} & B_{\Pi I}^T & A_{\Pi\Pi} & B_{\Pi 0}^T \\ 0 & 0 & 0 & B_{\Pi 0} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Delta \\ p_I \\ \mathbf{u}_\Pi \\ p_0 \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{w}_\Delta \\ 0 \\ 0 \\ 0 \end{pmatrix} . \tag{9}$$

In our FETI-DP algorithm, we solve this linear system by a Schur complement procedure. We first solve a coarse level problem

$$\begin{pmatrix} S_\Pi & B_{\Pi 0}^T \\ B_{\Pi 0} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_\Pi \\ p_0 \end{pmatrix} = \begin{pmatrix} \mathbf{f}_\Pi^* \\ 0 \end{pmatrix} , \tag{10}$$

and then the independent subdomain problems

$$\begin{pmatrix} A_{II} & A_{\Delta I}^T & B_{II}^T \\ A_{\Delta I} & A_{\Delta\Delta} & B_{\Delta I}^T \\ B_{II} & B_{\Delta I} & 0 \end{pmatrix} \begin{pmatrix} \mathbf{u}_I \\ \mathbf{u}_\Delta \\ p_I \end{pmatrix} = \begin{pmatrix} 0 \\ \mathbf{w}_\Delta \\ 0 \end{pmatrix} - \begin{pmatrix} A_{\Pi I}^T \\ A_{\Pi\Delta}^T \\ B_{\Pi I} \end{pmatrix} \mathbf{u}_\Pi. \tag{11}$$

In (10), S_Π is defined by:

$$A_{\Pi\Pi} - \begin{pmatrix} A_{\Pi I} & A_{\Pi\Delta} & B_{\Pi I}^T \end{pmatrix} \begin{pmatrix} A_{II} & A_{\Delta I}^T & B_{II}^T \\ A_{\Delta I} & A_{\Delta\Delta} & B_{\Delta I}^T \\ B_{II} & B_{\Delta I} & 0 \end{pmatrix}^{-1} \begin{pmatrix} A_{\Pi I}^T \\ A_{\Pi\Delta}^T \\ B_{\Pi I} \end{pmatrix}, \tag{12}$$

which corresponds to a discrete Stokes harmonic extension operator $\mathcal{SH}_\Pi : \mathbf{W}_\Pi \to \prod_{i=1}^N \mathbf{W}^h(\Omega^i)$ defined as: for any given primal velocity $\mathbf{u}_\Pi \in \mathbf{W}_\Pi$, find $\mathcal{SH}_\Pi \mathbf{u}_\Pi \in \prod_{i=1}^N \mathbf{W}^h(\Omega^i)$ and $p_I \in \prod_{i=1}^N \Pi_I^i$ such that on each subdomain $\Omega^i, i = 1, ..., N$,

$$\begin{cases} a(\mathcal{SH}_\Pi \mathbf{u}_\Pi, \mathbf{v}^i) + b(\mathbf{v}^i, p_I^i) &= 0, \quad \forall \mathbf{v}^i \in \mathbf{W}^h(\Omega^i) \\ b(\mathcal{SH}_\Pi \mathbf{u}_\Pi, q_I^i) &= 0, \quad \forall q_I^i \in \Pi^i \\ \mathcal{SH}_\Pi \mathbf{u}_\Pi &= \mathbf{u}_\Pi, \quad \text{in the primal space } \mathbf{W}_\Pi. \end{cases} \tag{13}$$

If we define an inner product $s_\Pi(.,.)$, corresponding to the Schur operator S_Π, on the space $\mathbf{W}_\Pi$ as

$$s_\Pi(\mathbf{u}_\Pi, \mathbf{u}_\Pi) = \mathbf{u}_\Pi^T S_\Pi \mathbf{u}_\Pi = a(\mathcal{SH}_\Pi \mathbf{u}_\Pi, \mathcal{SH}_\Pi \mathbf{u}_\Pi) \ , \ \forall \mathbf{u}_\Pi \in \mathbf{W}_\Pi, \tag{14}$$

then the matrix form of the coarse problem (10) can be written in the following variation form: find $\mathbf{u}_\Pi \in \mathbf{W}_\Pi$ and $p_0 \in \Pi_0$ such that,

$$\begin{cases} s_\Pi(\mathbf{u}_\Pi, \mathbf{v}_\Pi) + b(\mathbf{v}_\Pi, p_0) &= < \mathbf{f}_\Pi, \mathbf{v}_\Pi >, \forall \mathbf{v}_\Pi \in \mathbf{W}_\Pi \\ b(\mathbf{u}_\Pi, q_0) &= 0, \forall q_0 \in \Pi_0. \end{cases} \tag{15}$$

We can prove the following inf-sup stability estimate for this coarse saddle point problem.

Theorem 3.1

$$\sup_{\mathbf{w}_\Pi \in \mathbf{W}_\Pi} \frac{b(\mathbf{w}_\Pi, q_0)^2}{s_\Pi(\mathbf{w}_\Pi, \mathbf{w}_\Pi)} \geq \beta_C^2 ||q_0||_{L^2}^2, \forall q_0 \in \Pi_0, \tag{16}$$

where $\beta_C = C(1 + log(H/h))^{-1/2}$. C is a constant independent of h and H, but depends on the inf-sup stability constant of subdomain Stokes problem solver.

We have given a condition number bound for the preconditioned linear system (8) for two-dimensional case in [5]. Here we use some techniques from Klawonn et al [4] to obtain the following condition number bound for the three-dimensional case:

Theorem 3.2 *The condition number of the preconditioned linear system (8) is bounded from above by $C(1 + \log(H/h))^2$, where C is independent of h and H, but depends on the inf-sup stability constant of subdomain Stokes problem solver.*

4. Extension to nonlinear Navier-Stokes equations. The nonlinear problem is:

$$\begin{cases} -\mu\Delta\mathbf{u} + (\mathbf{u} \cdot \nabla)\mathbf{u} + \nabla p &= \mathbf{f}, \quad \text{in } \Omega \\ -\nabla \cdot \mathbf{u} &= 0, \quad \text{in } \Omega \\ \mathbf{u} &= \mathbf{g}, \quad \text{on } \partial\Omega \ , \end{cases} \tag{17}$$

where μ is the viscosity and $\int_{\partial\Omega} \mathbf{g} \cdot \mathbf{n} = 0$.
We solve this nonlinear problem by using a Picard iteration, where in each iteration step we solve a linearized Navier-Stokes problem:

$$\begin{cases} -\mu\Delta\mathbf{u}^{n+1} + (\mathbf{u}^n \cdot \nabla)\mathbf{u}^{n+1} + \nabla p^{n+1} &= \mathbf{f}, \\ -\nabla \cdot \mathbf{u}^{n+1} &= 0, \\ \mathbf{u}^{n+1}|_{\partial\Omega} &= \mathbf{g}. \end{cases}$$

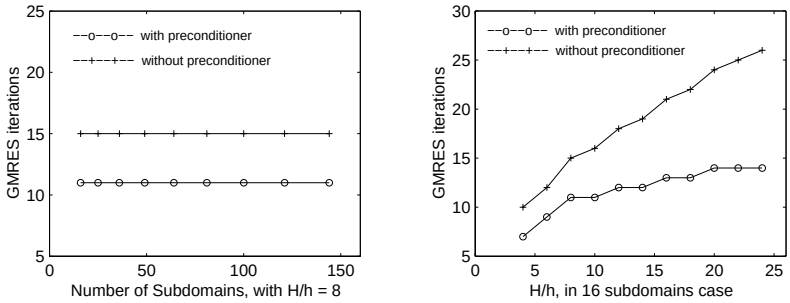


Figure 5.1: GMRES iterations counts for the Stokes solver vs. number of subdomains (left) and vs. H/h (right)

To solve this non-symmetric equation, the non-symmetric bilinear form $\int_{\Omega^i}(\mathbf{u}^n\cdot\nabla)\mathbf{u}^{n+1}\mathbf{v}$, on each subdomain Ω^i, is written as the sum of a skew-symmetric term and an interface term:

$$\left(\frac{1}{2}\int_{\Omega^i}(\mathbf{u}^n\cdot\nabla)\mathbf{u}^{n+1}\mathbf{v}-\frac{1}{2}\int_{\Omega^i}(\mathbf{u}^n\cdot\nabla)\mathbf{v}\mathbf{u}^{n+1}\right)+\frac{1}{2}\int_{\partial\Omega^i}(\mathbf{u}^n\cdot\mathbf{n})\mathbf{u}^{n+1}\mathbf{v}. \tag{18}$$

By doing this, we are identifying the correct bilinear form describing the action of the above non-symmetric operator on any given subdomain Ω^i, and the subdomain incompressible Navier-Stokes problem appears as:

$$\left\{\begin{array}{rcll} -\Delta\mathbf{u}^{n+1}+(\mathbf{u}^n\cdot\nabla)\mathbf{u}^{n+1}+\nabla p^{n+1} &=& \mathbf{f}, & \text{in } \Omega^i \\ -\nabla\cdot\mathbf{u}^{n+1} &=& 0, & \text{in } \Omega^i \\ \mathbf{u}^{n+1} &=& \mathbf{g}, & \text{on } \partial\Omega\cap\partial\Omega^i \\ \frac{\partial\mathbf{u}^{n+1}}{\partial\mathbf{n}}-p^{n+1}\mathbf{n}-\frac{\mathbf{u}^n\cdot\mathbf{n}}{2}\mathbf{u}^n &=& \lambda, & \text{on } \Gamma^{ij}\ . \end{array}\right. \tag{19}$$

The idea to write the non-symmetric bilinear form $\int_{\Omega^i}(\mathbf{u}^n\cdot\nabla)\mathbf{u}^{n+1}\mathbf{v}$ as in (18) was used by Achdou et al [1] to solve advection-diffusion problems. After discretizing the subdomain problems (19), we can use the same procedure as in section 2 to design the FETI-DP algorithm. We should also note that the conjugate gradient method cannot be used here to solve the preconditioned linear system, because this problem is no longer symmetric, positive definite.

5. Numerical Experiments. We have tested our algorithm by solving a lid driven cavity problem on the domain $\Omega=[0,1]\times[0,1]$, with $\mathbf{f}=\mathbf{0}$, $g_x=1, g_y=0$ for $x\in[0,1], y=1$, and $\mathbf{g}=\mathbf{0}$ elsewhere on the boundary (cf. Elman et al [2]). We have used GMRES to solve the preconditioned linear system (8), as well as the nonpreconditioned linear system (6). The initial guess is $\lambda=0$ and the stopping criterion is $||r_k||_2/||r_0||_2\leq 10^{-6}$, where r_k is the residual of the Lagrange multiplier equation at the kth iteration. Figure 5.1 gives the number of GMRES iterations for different number of subdomains with a fixed subdomain problem size $H/h=8$, and for different subdomain problem size H/h with 4×4 subdomains. We see, from the left figure, that the convergence of the augmented FETI-DP method, with or without a preconditioner, is independent of the number of subdomains, while the preconditioned version needs fewer iterations. The right figure shows that the GMRES iteration count increases, in both the preconditioned and the nonpreconditioned cases, with the increase of the size of subdomain problem, but that it is growing much slower with the Dirichlet preconditioner than without. Figure 5.2 shows that the coarse saddle point problem is inf-sup stable; cf. Theorem 3.1. We can see, from the left figure, that β_C is bounded away from zero while

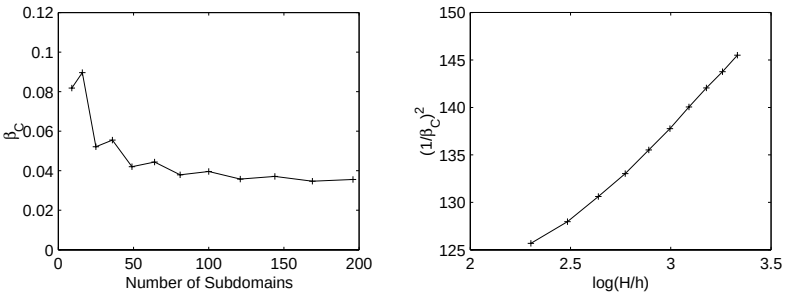


Figure 5.2: Inf-sup stability condition of the coarse level saddle point problem

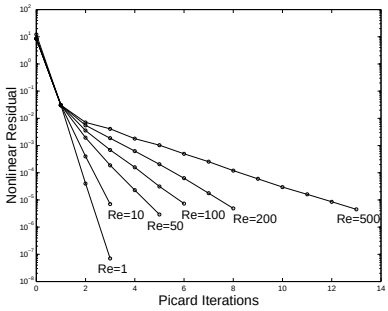


Figure 5.3: Convergence of Picard iteration for different Reynolds number

we increase the number of subdomains; from the right figure, $(1/\beta_C)^2$ appears to be a linear function of $log(H/h)$. In Figure 5.3 and Figure 5.4, we show the convergence behavior of the Picard iteration used to solve the nonlinear Navier-Stokes equation (17) for the 2D lid driven cavity problem. In our experiments, we start from a zero initial guess, and the Picard iteration is stopped when the nonlinear residual is reduced by 10^{-6}. For the GMRES solver, we reduce the linear residual by 10^{-3} in each iteration step. ¿From Figure 5.3, we see that the convergence of the Picard iteration depends on the Reynolds number: the larger is the Reynolds number, the slower is the convergence. Figure 5.4 shows that the convergence is independent of the mesh size. The left figure shows that the convergence is independent of the number of subdomains for fixed $H/h = 10$; the right figure shows that that the convergence is independent of H/h for the 64 subdomain case, except for a Reynolds number of 500. This can be explained by the fact that for high Reynolds number, the mesh has to be fine enough to achieve good convergence. **Acknowledgments.** The author is grateful to Olof Widlund for proposing this problem and giving many helpful suggestions.

REFERENCES

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

[2] H. C. Elman, D. J. Silvester, and A. J. Wathen. Iterative methods for problems in computational fluid dynamics. *Iterative Methods in Scientific Computing*, 1997.

[3] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.

[4] A. Klawonn, O. Widlund, and M. Dryja. Dual-primal FETI methods for three-dimensional

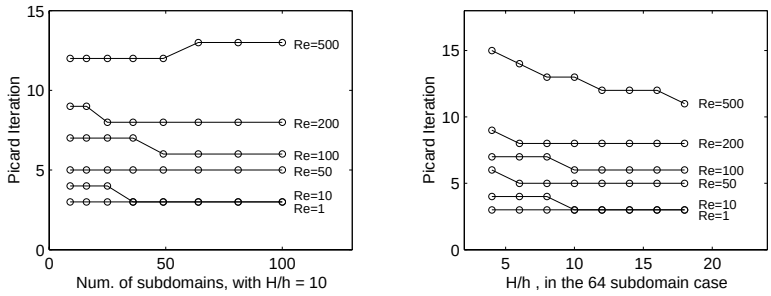


Figure 5.4: Convergence of Picard iteration for different meshes

elliptic problems with heterogeneous coefficients. Technical Report 815, Courant Institute of Mathematical Sciences, Department of Computer Science, April 2001. to appear in SIAM J. Numer. Anal.

[5] J. Li. A Dual-Primal FETI method for incompressible Stokes equations. Technical Report 816, Courant Institute of Mathematical Sciences, Department of Computer Sciences, 2001.

[6] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. *Numer. Math.*, 88:543–558, 2001.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

21. Experiences with FETI-DP in a Production Level Finite Element Application

K.H. Pierson[1], G.M. Reese[2], P. Raghavan[3]

Introduction The need for predictive, qualified models of very complex structures drives the requirements for large scale structural analysis. Reduced testing in the nuclear weapons program is a driving factor at the DOE labs. In addition, more detailed models reduce the need for engineering approximation, improve accuracy and often simplify model construction. Uncertainty in model parameters (for example, variations in joint preloads) can require multiple analyses for evaluation of a structure. Salinas was designed to meet the needs of a very large scale, general purpose, structural dynamics analysis ([1] and [14]).

Salinas was implemented with the goal of providing predictive modeling of complex structural systems. This necessitates a full suite of structural elements and solution methods which must perform reliably on serial and distributed memory machines. Robust solution methods and platform portability are critical. Sensitivity analysis and optimization capabilities are also required for application to the design and uncertainty quantification communities.

Salinas is implemented on a variety of Unix(tm) and Unix-like platforms. The core libraries are written in C++ using MPI communications. This facilitates extensibility to a full range of solvers, solution methods and element libraries. Scalability to thousands of processors is achieved through application of Finite Element Tearing and Interconnecting (FETI) methods ([2], [7], [6]). Recently, FETI-DP, the Dual-Primal Finite Element Tearing and Interconnecting method has been implemented as the replacement to the one-level FETI method previously used (see discussion below). High performance over a range of platforms is obtained through effective use of optimized BLAS routines. The BLAS routines are the building blocks for the sparse serial and parallel direct solvers used within Salinas/FETI-DP ([11], [13]).

Salinas has been used for production solutions of linear and nonlinear statics and implicit transient dynamics, and for eigen analysis and modal superposition solutions (such as frequency response, modal transient and random vibration). Extremely complex models have been analyzed utilizing combinations of beams, shells and solids. The models contain hundreds of different materials which may differ in modulus by ratios greater than 10^6. Models larger than 100M degrees of freedom (dof) have been solved with demonstrated scalability above 3000 processors. Salinas is limited to small deformation analysis, but some nonlinear elements have been added, and more are under development.

FETI-DP Overview We present an overview of the FETI-DP method to keep this paper self-contained. Let the global domain Ω be partitioned into a set of N_s, non-overlapping subdomains Ω^s. Select a set of corner points for each subdomain such that all zero energy modes are suppressed if Dirichlet boundary boundary conditions are applied to the set of corner points. The selected corner points remain primal unknowns which are used to define a sparse coarse grid matrix for FETI-DP. See [9] and more recent work by Lesoinne appearing in these proceedings about optimal corner point selection. Define u^s as the unknown solution vector associated with subdomain s. Split the global solution vector, u, into two sub-vectors

[1]Sandia National Laboratories, khpiers@sandia.gov
[2]Sandia National Laboratories, gmreese@sandia.gov
[3]The Pennsylvania State University, praghavan@psu.edu

such that:

$$u = \begin{bmatrix} u_r \\ u_c \end{bmatrix} = \begin{bmatrix} u_r^1 \\ \vdots \\ u_r^{N_s} \\ u_c \end{bmatrix} \tag{0.1}$$

where u_c is a primal unknown vector over all selected corner dof and u_r^s is the unknown vector for all remaining subdomain dof on subdomain s. The subdomain operator can be partitioned into the following 2x2 block matrix.

$$K^s = \begin{bmatrix} K_{rr}^s & K_{rc}^s \\ {K_{rc}^s}^T & K_{cc}^s \end{bmatrix} \tag{0.2}$$

Global equilibrium can be written by introducing unknown Lagrange multipliers exactly like the classical one-level FETI method.

$$\begin{bmatrix} K_{rr}^1 & \dots & 0 & K_{rc}^1 B_c^1 & {B_r^1}^T \\ \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & \dots & K_{rr}^{N_s} & K_{rc}^{N_s} B_c^{N_s} & {B_r^{N_s}}^T \\ {B_c^1}^T {K_{rc}^1}^T & \dots & {B_c^{N_s}}^T {K_{rc}^{N_s}}^T & \sum_{s=1}^{N_s} {B_c^s}^T K_{cc}^s B_c^s & 0 \\ B_r^1 & \dots & B_r^{N_s} & 0 & 0 \end{bmatrix} \begin{bmatrix} u_r^1 \\ \vdots \\ u_r^{N_s} \\ u_c \\ \lambda \end{bmatrix} = \begin{bmatrix} f_r^1 \\ \vdots \\ f_r^{N_s} \\ \sum_{s=1}^{N_s} {B_c^s}^T f_c^s \\ 0 \end{bmatrix} \tag{0.3}$$

where B_c^s maps the local corner equation numbering to global corner equation numbering, f_c^s is the external force applied on the corner dof, f_r^s is the external force applied on the remaining dof, ${B_r^s}^T$ is a boolean matrix that extracts the interface of a subdomain, and λ are the Lagrange multipliers. Let K_{rr} denote the block diagonal matrix of subdomain operators restricted to the remaining, r, points, K_{rc} the block column vector of subdomain coupling operator matrices and f_r the block column vector of subdomain force vectors. Using the same corner/remaining degrees of freedom matrix partitioning, we can rewrite the equilibrium equations compactly.

$$\begin{bmatrix} K_{rr} & K_{rc} & B_r^T \\ K_{rc}^T & K_{cc} & 0 \\ B_r & 0 & 0 \end{bmatrix} \begin{bmatrix} u_r \\ u_c \\ \lambda \end{bmatrix} = \begin{bmatrix} f_r \\ f_c \\ 0 \end{bmatrix} \tag{0.4}$$

The first equation can be solved for u_r since K_{rr} is a symmetric positive definite matrix if the selected corner points remove all of the local singularities. Then substitute the result into the compatablity equation (last equation in 0.4). The FETI-DP interface problem can be derived with some algebraic manipulation where the unknowns are λ, the Lagrange multipliers and u_c, the global corner degrees of freedom.

$$\begin{bmatrix} F_{rr} & F_{rc} \\ F_{rc}^T & -K_{cc}^* \end{bmatrix} \begin{bmatrix} \lambda \\ u_c \end{bmatrix} = \begin{bmatrix} d_r \\ -f_c^* \end{bmatrix} \tag{0.5}$$

where $F_{rr} = \sum_{s=1}^{N_s} B_r^s {K_{rr}^s}^{-1} {B_r^s}^T$, $F_{rc} = \sum_{s=1}^{N_s} B_r^s {K_{rr}^s}^{-1} K_{rc}^s B_c^s$, $d_r = \sum_{s=1}^{N_s} B_r^s {K_{rr}^s}^{-1} f_r^s$, and $f_c^* = \sum_{s=1}^{N_s} [{B_c^s}^T (f_c^s - {K_{rc}^s}^T {K_{rr}^s}^{-1} f_r^s)]$. The corner degrees of freedom, u_c, are condensed out to form the following symmetric positive definite Dual-Primal FETI interface problem which we solve

using a preconditioned conjugate gradient method. For a detailed derivation of this equation, please see [3]. Because of the preconditioning, the number of cg iterations (or FETI iterations) required for the solution is independent of model size. This scaling is demonstrated in the following sections.

$$\left[F_{rr} + F_{rc}{K^*_{cc}}^{-1} F^T_{rc}\right] \lambda = d_r - F_{rc}{K^*_{cc}}^{-1} f^*_c \tag{0.6}$$

The FETI operator defined above has an embedded coarse grid problem which can be written in the following form.

$$K^*_{cc} = \sum_{s=1}^{N_s} \left[{B^s_c}^T (K^s_{cc} - K^s_{cr}{K^s_{rr}}^{-1} K^s_{rc}) B^s_c)\right] \tag{0.7}$$

This new coarse problem has some highly beneficial properties over the previously defined two-level FETI coarse problem ([5]). First, this new coarse problem is symmetric positive definite sparse matrix. Secondly, only one forward/backward substitution has to be performed per FETI iteration. The original FETI algorithms required two forward/backward substitution operations per iteration. For a detailed derivation of FETI-DP see [9], [3], [4] and [12]. For a detailed mathematical analysis of the dual-primal FETI method one can review [8] and [10].

Scaled Problem Size Scalability We generate a series of model cube problems to assess scalability of Salinas and the underlying FETI-DP linear solver. The target platforms for assessing the scaled problem scalability of Salinas and FETI-DP are ASCI-Red, ASCI-Cplant and ASCI-White. The model cube problem is 13x13x13 hex elements per subdomain on ASCI-Red and ASCI-Cplant. On ASCI-White, we increased the model cube problem to 18x18x18 hex elements per subdomain to utilize the additional memory available. We scale the model cube problem with the number of processors keeping the size of the subdomains fixed. The number of subdomains is equal to the number of processors for all of our scalability experiments. The eight processor model cube problem is shown in figure 0.4.

For each of the platforms we evaluate the number of FETI iterations, the solver time, and the total time. The solver time (or FETI-DP time) represents the total time spent in the solver. This includes setup, factorization and solve time. The total time represents the time it takes to read the input geometry files, generate the subdomain matrices, solve a single $Ax = b$ problem and output the solution. The right hand side vector in all cases was a pressure load applied to the face opposite of the face where the Dirichlet boundary conditions were applied. The convergence tolerance was 0.001 for all platforms.

ASCI-Red The ASCI Option Red supercomputer, also known as the Intel Teraflops machine, is the first large-scale supercomputer built mostly of commodity, commercial, off-the-shelf (COTS) components. It has 4,536 compute and 72 service nodes each with 2 Pentium Pro processors. The system was delivered with 128 Mbytes of memory per node, but has been upgraded to 256 Mbytes of memory per node. The Pentium Pro processor runs at 333 MHz and has a peak floating-point rate of 333 Mflops. The system has over 1 Terabyte of real memory, and two independent 1-Terabyte disk systems. The system's 9216 Pentium Pro processors are connected by a 38x32x2 custom interconnect mesh with a bandwidth of 800 MB/s.

We show scalability results for up to 1000 nodes on ASCI-Red. Scaling the problem from sixty-four processors to one-thousand processors saw the number of iterations increase from 45 to 53. In figure 0.1 the total execution time is plotted for Salinas and FETI-DP running on ASCI-Red.

ASCI-Cplant CPlant is a large-scale massively parallel computer built from commodity computing and network computing components with a theoretical peak performance of

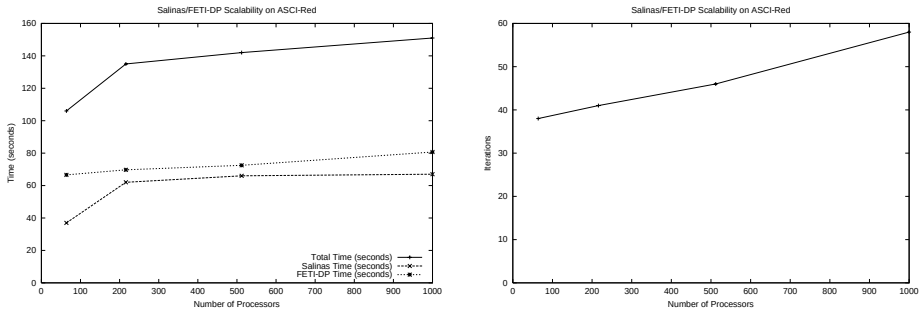


Figure 0.1: Performance on ASCI-Red for the following linear system sizes (processors): 446631 (64), 1479117 (216), 3472875 (512) and 6744273 (1000)

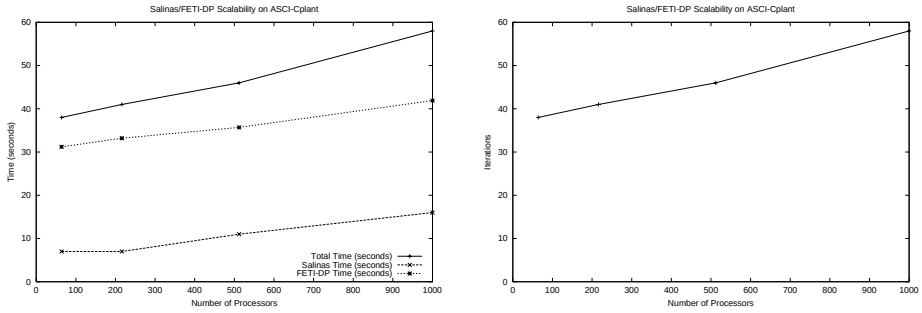


Figure 0.2: Performance on ASCI-Cplant for the following linear system sizes (processors): 446631 (64), 1479117 (216), 3472875 (512) and 6744273 (1000)

1.5 Tflops. The project goal of CPlant is to develop an architecture similar to the ASCI Red machine with entirely off-the-shelf commodity parts. Cplant uses the same partition model as the ASCI Red machine where pools of nodes can be divided into different categories, such as service, compute, and IO nodes.The compute processors are Compaq XP1000 workstations each containing a 500 MHz EV6 21264 microprocessor with 256 MB ECC SDRAM. The memory subsystem includes a 64KB instruction L1 cache and 4 MB L2 cache. The EV6 can issue four instructions per clock cycle and has two floating point units which amounts to a theoretical peak performance of 1 Gigaflops. The compute nodes are connected using Myrinet gigabit network.

We show scalability results for up to 1000 nodes on ASCI-Cplant. Scaling the problem from sixty-four processors to one-thousand processors saw the number of iterations increase from 45 to 53, identical to the ASCI-Red results. In figure 0.2 the total execution time is plotted for Salinas and FETI-DP running on ASCI-Cplant. The same model cube problem was tested on ASCI-Red and ASCI-Cplant. Therefore, one can directly compare the scalablity results for ASCI-Red in figure 0.1 versus the scalability results for ASCI-Cplant shown in figure 0.2. Based on these results, one can conclude that the total analysis time is approximately three times faster on ASCI-Cplant compared to ASCI-Red. This can be readily explained by the faster processors available on ASCI-Cplant. As expected, communication affects the overall scalability for large number of processors.

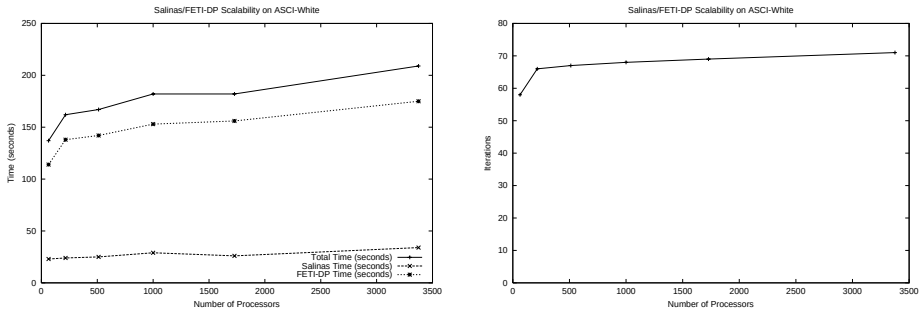


Figure 0.3: Performance on ASCI-White for the following linear system sizes (processors): 1167051 (64), 3885087 (216), 9145875 (512), 17789223 (1000), 30654939 (1728), 59707533 (3375)

ASCI-White ASCI White, the third step in a five step computational platform ladder, is currently the world's fastest computer with a peak speed slightly greater than 12 Tflops. The final ASCI platform has a goal to reach 100 Tflops peak performance by 2004. ASCI White is based upon IBM's latest SP technology. It comprises 512 symmetric multi-processor (SMP) machines, each possessing 16 processors, for a total of 8192 processing units. Each node consists of IBM's RS/6000 POWER3 symmetric multiprocessor (64 bit) architecture and this Nighthawk-2 SMP node is a stand-alone system with its own memory, operating system, local disk and 16 CPUs. POWER3-II processors are super-scalar pipelined 64 bit RISC chips with two floating point units and three integer units, capable of executing up to eight instructions per clock cycle and up to two floating point operations per cycle. At 375 Mhertz this processor is capable of producing a peak performance of 750 Mflops peak. The one cycle latency L1 cache is 128-way set associative and consists of 64KB data cache and a 32 KB instructions cache. The 4 MB L2 cache runs typically at half the processor speed and uses a direct mapped approach. Each processor has 1 GigaByte of available memory. All nodes are interconnected by the internal SP switch network, which has a bidirectional bandwidth of 300MB/second.

We show scalability results for up to 3375 nodes on ASCI-White. Scaling the problem from sixty-four processors to 3375 processors saw the number of iterations increase from 58 to 71. In figure 0.3 the total execution time is plotted for Salinas and FETI-DP running on ASCI-White.

Coarse Grid Solution Options We describe two FETI coarse grid solver technologies which are implemented in Salinas. The FETI-DP coarse grid as described above is a sparse matrix that couples all of the subdomains. This coarse sparse matrix has to be factored during the FETI-DP initialization step. During the FETI-DP solve step, one coarse matrix forward/backward solve is performed per iteration. The two coarse grid solver technologies are listed below.

- Redundant storage, factorization and forward/backward solves on each Salinas processor. In this option, a distributed inverse is computed and the relevant columns are stored on each processor. Solves are accomplished with local matrix-vector products.

- Distributed storage, parallel factorization and parallel forward/backward solves on a subset of the Salinas processors.

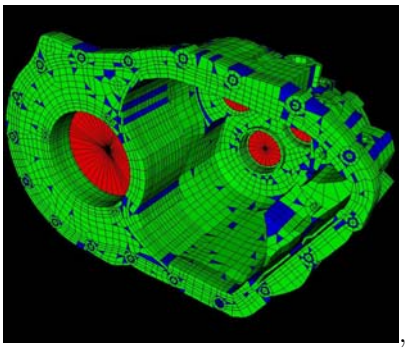
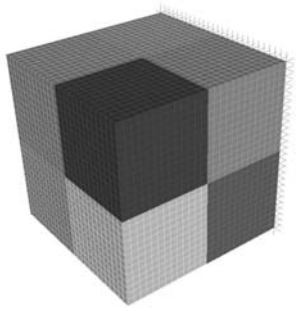



Figure 0.4: Engine block finite element model and Model cube

$N_p = N_s$	N_{eqn}^{coarse}	Serial Sparse	Parallel Sparse	Memory
25	831	8.3 sec.	7.9 sec. (8 processors)	1.7 MB
115	3999	9.7 sec.	5.9 sec. (16 processors)	14.6 MB
137	4815	14.0 sec.	8.4 sec. (32 processors)	19.9 MB
222	7425	25.8 sec.	11.4 sec. (32 processors)	34.0 MB
276	9339	36.4 sec.	12.8 sec. (32 processors)	45.6 MB

Table 0.1: Coarse grid setup and factorization time for engine block on ASCI-Red

For the redundant storage case, we choose a sparse matrix solver based on a multiple minimum degree ordering ([11]). The parallel factorization case is accomplished by the Domain Separator Cholesky package (DSCpack) ([13]).

Parallel Sparse Solver Experiments We experiment with solving large scale problems using large numbers of processors which results in increasingly larger FETI-DP coarse grid problems. A comparision is done between the redundant factorization and subsequent coarse grid matrix inverse technique versus using a parallel distributed memory sparse solver. At each iteration of FETI-DP, the parallel sparse solver does forward/backward solves in parallel. In the future, further studies will be conducted to determine if the parallel sparse solver in conjunction with the coarse grid matrix inverse technique will result in optimal CPU time.

Coarse Grid Scalability A finite element model of an engine block was chosen for coarse grid scalability studies. A picture of the engine block finite element model is shown in figure 0.4. This model is available in three increasing larger sizes. We choose the smallest model to illustrate the affect of increasing the number of subdomains for a fixed size problem on the FETI-DP coarse grid matrix. The small engine block model has 28498 nodes, 24363 elements and approximately 75000 degrees of freedom. This problem contains hex, wedge and triangular shell elements. We partition this model into 25, 115, 137, 222 and 276 subdomains respectively. We then solve an eigenvalue using Salinas and FETI-DP. In this Salinas solution method, FETI-DP is employed as the linear solver inside of a Lanzcos based parallel eigensolver. Table 0.1 shows the factorization times of the two coarse grid options on an increasing number of processors. Table 0.1 also shows the memory requirements of the serial sparse solver. The parallel sparse solver distributes the coarse grid LDL^T factor over the number of coarse solver processors. This effectively reduces the per processor memory requirements. For small coarse grid sizes, the redundant sparse direct method

outperforms the parallel sparse solver. Please note that this is mainly due to the calculation of the coarse grid inverse (a detailed description can be found in [6]) and subsequent replacement of sparse forward/backward substitution by matrix-vector multiplication during the FETI-DP iterations. For a sufficiently large problem, the factorization of the coarse grid becomes the dominant time and the parallel sparse solver begins to out-perform the serial sparse direct method. More importantly, the coarse grid eventually becomes too large to store on every processor and the only option is to use the parallel sparse distributed solver. Future studies will investigate tiling the parallel sparse solver on $N_p/N_{cs} = N_{tiles}$ while N_{cs} equals the number of processors solving the coarse grid matrix. This approach leaves an integer number of processors, $N_{rem} = mod(N_p/N_{cs})$ idle while the current parallel sparse solver implementation leaves $N_p - N_{cs}$ processors idle during coarse grid factorization and subsequent forward/backward solves required during FETI-DP iterations.

Conclusion We have shown that FETI-DP performs well in a production finite element application on a variety of massively parallel platforms. Scalablity was demonstrated on ASCI-Red, ASCI-Cplant and ASCI-White. We are actively pursuing parallel factorization of the FETI-DP coarse grid to enable further improvements in scalability.

Acknowledgements The authors would like to thank Charbel Farhat and Michel Lesoinne, from the University of Colorado at Boulder for all of their time, energy, and dedication in researching and implementing FETI methods into Salinas. The authors would also like to thank David Day, Manoj Bhardwaj, Tim Walsh, Ken Alvin, David Martinez, James Peery, Dan Segalman, and Sam Key for their support, contributions and insightful discussions. 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.

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 one thousand processors and discussion of highly heterogeneous problems. *Int. J. Numer. Meth. Engrg.*, 47:513–535, 2000.

[2] C. Farhat. A Lagrange multiplier based on divide and conquer finite element algorithm. *J. Comput. System Engrg*, 2:149–156, 1991.

[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.*, 50:1523–1544, 2001.

[4] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.

[5] C. Farhat and J. Mandel. The two-level feti method for static and dynamic plate problems - part i: an optimal iterative solver for biharmonic systems. *Computer Methods in Applied Mechanics and Engineering*, 155:129–152, 1998.

[6] C. Farhat, K. H. 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. *Computer Methods in Applied Mechanics and Engineering*, 184:333–374, 2000.

[7] C. Farhat and F.-X. Roux. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *Int. J. Numer. Meth. Engrg.*, 32:1205–1227, 1991.

[8] A. Klawonn and O. B. Widlund. FETI-DP Methods for Three-Dimensional Elliptic Problems with Heterogeneous Coefficients. Technical report, Courant Institute of Mathematical Sciences, 2000. In Preparation.

[9] M. Lesoinne and K. Pierson. Feti-dp: An efficient, scalable, and unified dual-primal feti method. In T. Chan, T. Kako, H. Kawarada, and O. Pironneau, editors, *Domain Decomposition Methods in Sciences and Engineering*, pages 421–428. DDM.org, 1999.

[10] J. Mandel and R. Tezaur. On the convergence of a dual-primal substructuring method. Technical report, University of Colorado at Denver, Department of Mathematics, January 2000. To appear in Numer. Math. `URL: http://www-math.cudenver.edu/ jmandel/papers/dp.ps.gz`.

[11] E. G. Ng and B. W. Peyton. Block sparse Cholesky algorithms on. *SIAM J. Sci. Stat. Comput.*, 14:1034–1056, 1993.

[12] K. H. Pierson. *A family of domain decomposition methods for the massively parallel solution of computational mechanics problems*. PhD thesis, University of Colorado at Boulder, Aerospace Engineering, 2000.

[13] P. Raghavan. Efficient parallel triangular solution with selective inversion. *Parallel Processing Letters*, 8:29–40, 1998.

[14] G. Reese, M. Bhardwaj, D. Segalman, K. Alvin, and B. Driessen. *Salinas User's Manual*. Sandia National Laboratories.

Part IV

Mini Symposium: Unified Approaches to Domain Decomposition Methods

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

22. Unified Theory of Domain Decomposition Methods

I. Herrera[1]

1. Introduction. Domain Decomposition Methods (DDM) have been derived by Herrera using a unifying concept, which consists in viewing DDM as procedures for gathering information at the internal boundary (Σ) of a partition, sufficient for defining well-posed problems at each one of its subdomains. Two broad categories of Domain Decomposition Methods are identified in this manner: 'direct' and 'indirect (or Trefftz-Herrera)' methods. Direct methods are usually understood as procedures for putting together local solutions, just as bricks, to build the global solution. However, for direct methods the point of view adopted by the unified theory, here presented, is different: the local solutions are used, as means for establishing compatibility relations that the global solution of the problem considered must fulfill. In Trefftz-Herrera methods, on the other hand, local solutions are used in an indirect manner; as specialized test functions with the property of supplying information on Σ, exclusively. Important features of Herrera's unified theory are the use, throughout it, of "fully discontinuous functions" and the treatment of a general boundary value problem with prescribed jumps. The generality of the resulting theory is remarkable, because it is applicable to any partial (or ordinary) differential equation or system of such equations, which is linear, independently of its type and with possibly discontinuous coefficients. The developments that have been carried out, thus far in this framework, have implications along two broad lines: as tools for incorporating parallel processing in the modeling of continuous systems and as an elegant and efficient way of formulating numerical methods from a *domain decomposition* perspective. In addition, the theory supplies a systematic framework for the application of fully discontinuous functions in the treatment of partial differential equations.

This paper is part of a sequence of papers, contained in these Proceedings, devoted to present, and further advance, this unified theory of Domain Decomposition Methods (DDM) and some developments associated with it. DDM have received much attention in recent years[2], mainly because they supply very effective means for incorporating parallel processing in computational models of continuous systems. Another aspect that must be stressed is that it is useful to analyze numerical methods for partial differential equations from a domain-decomposition perspective, since the ideas related to domain decomposition are quite basic for them. Indeed, developing numerical procedures as accurate as desired in small regions is an easy task that can be performed by many numerical schemes and, once this has been done, the remaining problem is essentially the same as that of Domain Decomposition Methods. In this respect, it is useful to recall the main objective of DDM:

Given a domain Ω and one of its partitions (Fig. 1.1), to obtain the solution of a boundary value problem defined on it (the 'global problem'), by solving problems formulated on the subdomains of the partition (the 'local problems'), exclusively. In what follows the subdomains of the partition will be denoted by $\Omega_i (i = 1, ..., E)$ and the internal boundary, which separates the subdomains from each other, will be Σ.

Herrera has proposed recently a unified theory of DDM [15],[14], in which most of the known methods may be subsumed, supplying more general formulations of them and hinting new procedures that should be investigated in the future. The sequence of papers mentioned above, intends to present briefly such theory in its different aspects. The present paper contains an exposition of the unified theory. Trefftz-Herrera Method is given in [20], while

[1]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM), Apartado Postal 22-582, 14000, México, D.F. e-mail: iherrera@servidor.unam.mx

[2]See: International Scientific Committee for Domain Decomposition "Proceedings of 13 conferences on Domain Decomposition Methods", www.ddm.org, 1988-2001

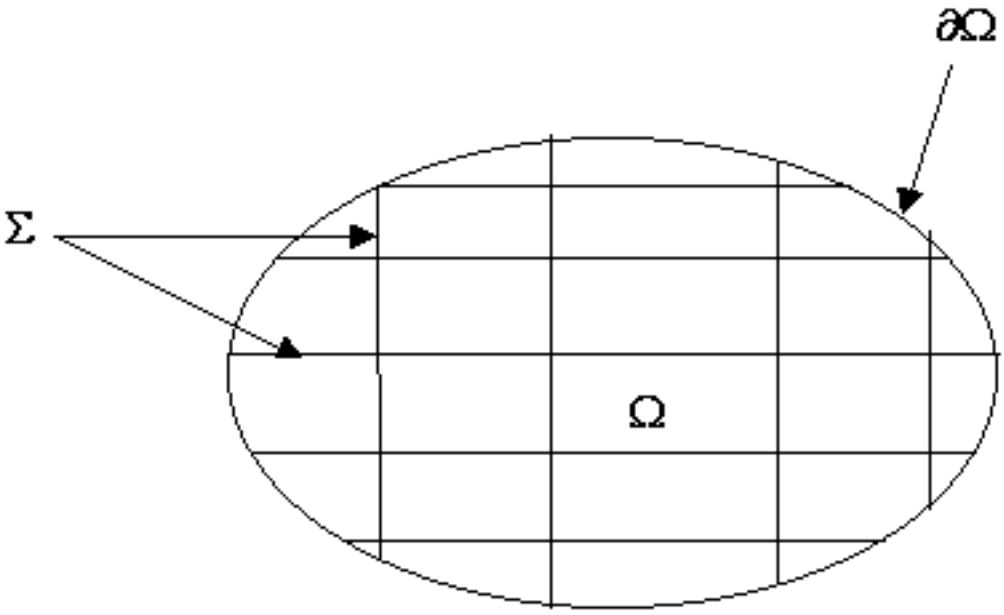


Figure 1.1: Partition of the domain Ω

direct methods are described in [9]. Applications to elliptic equations are presented in [6],[22] and [5] -second order equations are treated in [6] and the biharmonic equation in [5]-.

2. Some Unifying Concepts. Herrera's theory is formulated in function spaces whose elements are generally discontinuous, and the theory supplies systematic procedures for applying discontinuous functions in the numerical treatment of partial differential equations. Such function spaces have the following general form:

$$\hat{D}(\Omega) \equiv D(\Omega_1) \oplus ... \oplus D(\Omega_E); \tag{2.1}$$

If $u \in \hat{D}(\Omega)$, then $u \equiv (u_1, ..., u_E)$ where $u_i \in D(\Omega_i)$, $i = 1, ..., E$. Generally, when variational formulations are considered, as in the theory of indirect methods, two such spaces are introduced; namely, the space of *trial or base functions* $\hat{D}_1(\Omega)$ and the space of *test or weighting functions* $\hat{D}_2(\Omega)$. When $D(\Omega_i)$, $i = 1, ..., E$, are Sobolev spaces, a special kind of Sobolev space, $\hat{\mathbb{H}}^s(\Omega)$, is obtained: $\hat{\mathbb{H}}^s(\Omega) \equiv \mathbb{H}^s(\Omega_1) \oplus ... \oplus \mathbb{H}^s(\Omega_E)$. Of course, more complicated combinations are possible.

In addition, the theory deals with a very general boundary value problem, the Boundary Value Problem with prescribed Jumps (the BVPJ), in which, in addition to boundary conditions on the external boundary, $\partial\Omega$, jumps are prescribed across the internal boundary Σ. And it is also applicable when the coefficients of the differential operators are discontinuous across Σ. The general BVPJ considered by the theory is type-independent and has the form

$$\mathcal{L}u = f_\Omega; \quad in\ \Omega_i \quad i = 1, ..., E \tag{2.2}$$

$$B_j u = g_{\partial j}; \quad on \quad \partial\Omega \tag{2.3}$$

$$[J_k u] = j_{\Sigma k}; \quad on \quad \Sigma \tag{2.4}$$

Here $\mathcal{L}$ is a differential operator of any type; in particular it can be elliptic, hyperbolic or parabolic. Furthermore, it can be vector-valued and therefore the theory includes systems of equations and not just a single equation. The solution of the BVPJ will be denoted by $u \equiv (u_1, ..., u_E)$. In this setting, the objective of Domain Decomposition Methods is to find $u_i \in D(\Omega_i)$, for $i = 1, ..., E$. The unified theory is based on the following unifying principle:

Domain Decomposition Methods are procedures for gathering information, on the internal boundary Σ, sufficient for defining well-posed local problems in each one of the subdomains. Then it is possible to reconstruct the solution in the interior of the subdomains, $u_i \in D(\Omega_i)$, for $i = 1, ..., E$ by solving local problems exclusively.

3. The Sought Information. The information that one deals with, when formulating and treating partial differential equations (i.e., the BVPJ), is classified in two broad categories: *'data of the problem'* and *'complementary information'*. In turn, three classes of data can be distinguished: data in the interior of the subdomains of the partition (given by the differential equation, which in the BVPJ is fulfilled in the interior of the subdomains, exclusively), the data on the external boundary ($B_j u, \quad on \quad \partial\Omega$) and the data on the internal boundary (namely, $[J_k u], \quad on \quad \Sigma$). The complementary information can be classified in a similar fashion: the values of the sought solution in the interior of the subdomains ($u_i \in D(\Omega_i)$, for $i = 1, ..., E$); the complementary information on the outer boundary (for example, the normal derivative in the case of Dirichlet problems for Laplace's equation); and the complementary information on the internal boundary Σ (for example, the average of the function and the average of the normal derivative across the discontinuity for elliptic problems of second order [6]). In the unified theory of DDM, a target of information, which is contained in the complementary information on Σ, is defined; it is called *'the sought*

information'. It is required that the choice of *the sought information* fulfills the following assumption:

when 'the sought information' is complemented with the data of the problem, there is sufficient information available for defining well-posed problems in each one of the subdomains of the partition.

In general, however, the sought information may satisfy this property and yet be redundant, in the sense that if all of it is used simultaneously together with the data of the problem, ill-posed problems are obtained. Consider for example, a Dirichlet problem of an elliptic-type second order equation (see [6]), for which the jumps of the function and of its normal derivative have been prescribed. If for such problem the *sought information* is taken to be the average of the function -i.e., $(u_+ + u_-)/2$-, and the average of the normal derivative -i.e., $\frac{1}{2}\partial(u_+ + u_-)/\partial n$, on Σ-, then it may be seen that it contains redundant information. Indeed, $u_+ = \frac{1}{2}(u_+ + u_-) + \frac{1}{2}(u_+ - u_-)$, $u_- = \frac{1}{2}(u_+ + u_-) - \frac{1}{2}(u_+ - u_-)$, and a similar relation holds for the normal derivatives. Therefore, if the *'sought information'* and the *'data of the problem'* are used simultaneously, one may derive not only the value of the BVPJ solution on the boundary of each one of the subdomains, but also the normal derivative, at least in a non-void section of those boundaries. As it is well known, this is an ill-posed problem, because Dirichlet problem is already well-posed in each one of the subdomains. Thus, the *sought information* contains redundant information in this case.

Generally, in the numerical treatment of partial differential equations, efficiency requires eliminating redundant information. This fact motivates the following definition:

The sought information is 'optimal' when there is a family of well-posed problems -one for each subdomain of the partition- which uses all the sought information, together with the data of the BVPJ.

Analysis of existing methods reveals that there are some for which the sought information is optimal and others for which this is not the case. In general, except for the simple case of first order equations, methods for which the *sought information* is optimal are overlapping.

4. Direct and Indirect Methods. There are two main procedures for gathering the *sought information* on Σ: *'direct'* and *'indirect* (or *Trefftz-Herrera)'* methods. Both of them derive the *sought information*, on Σ, from compatibility conditions that the global solution of the BVPJ must satisfy locally and the local solutions are applied precisely for deriving such compatibility conditions. The global system of equations, for the *sought information*, is constructed in this manner. Trefftz-Herrera methods were introduced in numerical analysis by Herrera *et al.* [10], [16], [11], [4], [17], [12], [13] and [19], and its distinguishing feature is the use of specialized test functions which have the property of yielding any desired information on Σ. The guidelines for the construction of such weighting functions is supplied by a special kind of Green's formulas (Green-Herrera formulas), formulated in spaces of fully discontinuous functions [10],[16],[17], which permit analyzing the information on Σ, contained in approximate solutions. Using Green-Herrera formulas it has been possible to give a very general formulation of Indirect Methods in terms of a variational principle possessing great generality. This is Eqs. (7.4),(7.7) of reference [20](see also [19]), which corresponds to an Invited Plenary Talk of this Conference that was devoted to a full description of Trefftz-Herrera Methods and is contained in these Proceedings.

Conventional descriptions of Direct Methods present them as procedures for assembling, just as *bricks*, local solutions in order to build the global one. When these methods are formulated using the unified theory approach, direct methods derive the *sought information*, on Σ, from compatibility conditions that the global solution of the BVPJ must satisfy locally [9] and the local solutions are applied precisely for deriving such compatibility conditions. An important difference between direct and Trefftz-Herrera methods is that in the latter local solutions of equations formulated in terms of the adjoint differential operator are used, while

in the former such equations are formulated in terms of the original differential operator.

To finish this Section some general remarks are in order. In the methods of the unified theory the only information that is obtained when solving the global problem refers to information on the internal boundary Σ and no information at all is obtained in the interior of the subdomains. If such information is desired, it can be derived solving the well-posed local problems that are obtained in the manner explained before. When the unified theory is applied as a discretization procedure, the process described above for deriving the solution in the interior of the subdomains of the partition, which can be carried out by any numerical method, is referred as *'optimal interpolation'*. This is in agreement with, and supplements, the nomenclature that has been used in some past work, in which the specialized test functions that supply information at the internal boundary exclusively, are referred as *optimal test functions* [3].

5. General Conclusions. An elegant framework for Domain Decomposition Methods, which is quite general and simple, has been presented. The generality of the methodologies must be stressed, since they are applicable to any linear differential equation, or system of such equations and to problems with prescribed jumps and with discontinuous coefficients. In addition, the theory supplies systematic procedures for applying discontinuous functions in the numerical treatment of partial differential equations. Even more, its applicability is type-independent. Thus, it is not only applicable to elliptic equations, but also to hyperbolic and parabolic ones.

Thus far, DDM have been mainly applied as a tool for parallelizing numerical models of continuous systems [21]. However, Herrera's Unified Theory permits developing wide classes of numerical methods with many attractive features [19]. In addition, we claim that this theory subsumes most of the existing methods of domain decomposition. Using its framework Schwarz and Steklov-Poincaré methods were incorporated in this framework in [18] and [19], respectively, while Mixed Methods were derived in [17]. The theory also implies wide generalizations of the Projection Decomposition Method [1]. We suspect that the capacity of using fully discontinuous functions systematically -and the foundations of such capacity is one of the contributions of the theory- permits eliminating Lagrange multipliers in many instances and that it also has a bearing on partitions of unity and its applications. This, however, remains to be shown. Other subjects that should be investigated in the future are the implications of the unified theory on Mortar [2] and FETI methods [7],[8].

REFERENCES

[1] V. Agoshkov and E. Ovtchinnikov. Projection decomposition method. *Math. Models Methods Appl. Sci.*, 4:773–794, 1994.

[2] C. Bernardi, Y. Maday, and A. T. Patera. A new non conforming approach to domain decomposition: The mortar element method. In H. Brezis and J.-L. Lions, editors, *Collège de France Seminar*. Pitman, 1994. This paper appeared as a technical report about five years earlier.

[3] M. A. Celia. Eulerian-Lagrangian Localized Adjoint Methods for Contaminant Transport Simulations. In *Computational Methods in Water Resources X*, pages 207–216, 1994.

[4] M. A. Celia, I. Herrera, and T. Bouloutas. *Adjoint Petrov-Galerkin Methods for Multi-Dimensional Flow Problems*, pages 953–958. Finite Element Analysis in Fluids, T.J. Chung and R. Karr, Eds. University of Alabama Press, 1989.

[5] M. Diaz and I. Herrera. Indirect Method of Collocation for the Biharmonic Equation. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[6] M. Diaz, I. Herrera, and R. Yates. Indirect Method of Collocation: Second Order Equations. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[7] C. Farhat and F.-X. Roux. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *Int. J. Numer. Meth. Engrg.*, 32:1205–1227, 1991.

[8] C. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 2 (1), pages 1–124. North-Holland, 1994.

[9] F. Garcia-Nocetti, I. Herrera, R. Yates, E. Rubio, and L. Ochoa. The Direct Approach to Domain Decomposition Methods. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[10] I. Herrera. Unified Approach to Numerical Methods. Part 1. Green's Formulas for Operators in Discontinuous Fields. *Numerical Methods for Partial Differential Equations*, 1(1):12–37, 1985.

[11] I. Herrera. The Algebraic Theory Approach for Ordinary Differential Equations: Highly Accurate Finite Differences. *Numerical Methods for Partial Differential Equations*, 3(3):199–218, 1987.

[12] I. Herrera. *Localized Adjoint Methods: A New Discretization Methodology*, chapter 6, pages 66–77. Computational Methods in Geosciences. SIAM, 1992.

[13] I. Herrera. Trefftz-Herrera Domain Decomposition. *Advances in Engineering Software*, 24:43–56, 1995.

[14] I. Herrera. Trefftz Method: A General Theory. *Numerical Methods for Partial Differential Equations*, 16(6):561–580, 2000.

[15] I. Herrera. A General Theory of Domain Decomposition and Trefftz Methods. In *2nd. European Conference on Computational Mechanics, Cracovia, Polonia*, 2001.

[16] I. Herrera, L. Chargoy, and G. Alducin. Unified Approach to Numerical Methods. Part 3. Finite Differences and Ordinary Differential Equations. *Numerical Methods for Partial Differrential Equations*, 1:241–258, 1985.

[17] I. Herrera, R. Ewing, M. Celia, and T. Russell. Eulerian-Lagrangian Localized Adjoint Method: The Theoretical Framework. *Numerical Methods for Partial Differential Equations*, 9(4):431–457, 1993.

[18] I. Herrera and R. Yates. General Theory of Domain Decomposition: Beyond Schwarz Methods. *Numerical Methods for Partial Diffential Equations*, 17(5):495–517, 2001.

[19] I. Herrera, R. Yates, and M. Diaz. General Theory of Domain Decomposition: Indirect Methods. *Numerical Methods for Partial Differential Equations*, 18(3):296–322, 2002.

[20] I. Herrera, R. Yates, and M. Diaz. The Indirect Approach to Domain Decomposition. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

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

[22] R. Yates and I. Herrera. Parallel Implementation of Indirect Collocation Method. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

23. Indirect Method of Collocation: 2^{nd} Order Elliptic Equations

M.A. Diaz[1], I. Herrera[2], R. Yates[3]

1. Introduction. These papers is part of a group of papers [7],[9],[4], [3],[10], included in these Proceedings, devoted to present and illustrate the applications of Herrera's Unified Theory of Domain Decomposition Methods (DDM). As an example of the applications of indirect -or Trefftz-Herrera- methods, in the present paper a new method of collocation - Trefftz-Herrera collocation- is developed applicable to any elliptic equation of second order, which is linear. The general problem considered is one with prescribed jumps for the function and its first order derivatives; actually, the *'fluxes'*, as its explained later in the sequel. Differential operators with discontinuous coefficients are included.

The collocation method based on the use of Hermite cubic polynomials has a good number of attractive features such as its high accuracy and the simplicity of its formulation [1],[2]. However, it suffers computationally from several drawbacks, such as a large number of degrees of freedom associated with each node of the discretized mesh. Also, the global matrix of the system of equations does not enjoy the property of being positive definite even when the differential operator itself has this property. Up to now, collocation has been applied by means of splines. However, a broader and more efficient formulation is obtained when collocation is applied using fully discontinuous functions by means of the indirect (or Trefftz-Herrera) domain decomposition methodology. In this paper Trefftz-Herrera indirect method, in combination with orthogonal collocation, is applied to a general boundary value problem with prescribed jumps to produce a family of "indirect collocation methods (Trefftz-Herrera collocation)". In particular, when the differential equation (or system of such equations) is positive definite the global matrix is also positive definite. Also, a dramatic reduction in the number of degrees of freedom associated with each node is obtained. Indeed, in the standard method of collocation that number is two in one dimension, four in two dimensions and eight in three dimensions, while for some of the new algorithms they are only one in all space dimensions. A final comment worth doing refers to the fact that the treatment of problems with prescribed jumps is just as easy as that without them; as a matter of fact, the global matrix is exactly the same for both problems.

2. Trefftz-Herrera Approach to Elliptic Equations (2^{nd} Order) . The general theory of Trefftz-Herrera DDM, presented in [9], is applied in this Section to elliptic equations of second order. The boundary value problem with prescribed jumps (BVPJ) for this case was given as an illustration in [9]; it is:

$$\mathcal{L}u \equiv -\nabla\cdot(\underline{\underline{a}}\cdot\nabla u) + \nabla\cdot(\underline{b}u) + cu = f_\Omega \equiv \mathcal{L}u_\Omega, \quad in \quad \Omega_i, \quad i = 1, ..., E \tag{2.1}$$

subjected to the boundary conditions

$$u = u_\partial; \quad on \quad \partial\Omega, \tag{2.2}$$

and the jump conditions

$$[u] = [u_\Sigma] \equiv j^0_\Sigma \quad and \quad [\underline{\underline{a}}\cdot\nabla u]\cdot\underline{n} = [\underline{\underline{a}}\cdot\nabla u_\Sigma]\cdot\underline{n} \equiv j^1_\Sigma; \quad on \quad \Sigma, \tag{2.3}$$

[1]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM), mdiaz@tonatiuh.igeofcu.unam.mx
[2]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM) , iherrera@servidor.unam.mx
[3]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM) , yates@altcomp.com.mx

The notation is the same as that introduced in [9] and [4]. In particular, $u_\Omega \in \hat{D}_1$, $u_\partial \in \hat{D}_1$ and $u_\Sigma \in \hat{D}_1$ are any functions which satisfy the differential equation, the external boundary conditions and the jump conditions, respectively. and A partition of a domain Ω is being considered and the internal boundary is denoted by Σ (see [9] for further details).

The general theory introduces the following bilinear functionals:

$$\langle Pu, w\rangle \equiv \int_\Omega w\mathcal{L}u dx; \quad \langle Qw, u\rangle \equiv \int_\Omega u\mathcal{L}^* w dx \tag{2.4}$$

$$\langle Bu, w\rangle \equiv \int_{\partial\Omega} \mathcal{B}(u, w)dx; \quad \langle Cw, u\rangle \equiv \int_{\partial\Omega} \mathcal{C}(w, u)dx \tag{2.5}$$

$$\langle Ju, w\rangle \equiv \int_\Sigma \mathcal{J}(u, w)dx; \quad \langle Kw, u\rangle \equiv \int_\Sigma \mathcal{K}(w, u)dx \tag{2.6}$$

$$\langle S_J u, w\rangle \equiv \int_\Sigma \mathcal{S}_J(u, w)dx; \quad \langle R_J u, w\rangle \equiv \int_\Sigma \mathcal{R}_J(u, w)dx \tag{2.7}$$

$$\langle Sw, u\rangle \equiv \int_\Sigma \mathcal{S}(w, u)dx; \quad \langle Rw, u\rangle \equiv \int_\Sigma \mathcal{R}(w, u)dx \tag{2.8}$$

Where $\mathcal{J}(u, w)$ and $\mathcal{K}(w, u)$, are given by Eq. (5.4) of Ref. [9]:

$$\mathcal{J}(u, w) \equiv -\underline{\mathcal{D}}([u], \dot{w}) \cdot \underline{n} \quad \text{and} \quad \mathcal{K}(w, u) \equiv \underline{\mathcal{D}}(\dot{u}, [w]) \cdot \underline{n} \tag{2.9}$$

where

$$\underline{\mathcal{D}}\,(u, w) \equiv u\,(\underline{a}_n \cdot \nabla w + b_n w) - w\underline{a}_n \cdot \nabla u \tag{2.10}$$

has the property that

$$w\mathcal{L}u - u\mathcal{L}^* w \equiv \nabla \cdot \underline{\mathcal{D}}\,(u, w) \tag{2.11}$$

Here

$$\mathcal{L}^* w \equiv -\nabla \cdot (\underline{\underline{a}} \cdot \nabla w) - \underline{b} \cdot \nabla w + cw; \tag{2.12}$$

Then, for the case considered in this Section, the bilinear functions occurring in the integrals of Eqs. (2.4) to (2.8) are defined by [9]:

$$\mathcal{B}\,(u, w) \equiv u\,(\underline{a}_n \cdot \nabla w + b_n w) \cdot \underline{n}, \quad \mathcal{C}\,(w, u) \equiv w\underline{a}_n \cdot \nabla u \tag{2.13}$$

$$\mathcal{J}\,(u, w) \equiv \dot{w}\,[\underline{a}_n \cdot \nabla u] - [u]\,\dot{\overline{(\underline{a}_n \cdot \nabla w + b_n w)}} \tag{2.14}$$

$$\mathcal{K}\,(w, u) \equiv \dot{u}\,[\underline{a}_n \cdot \nabla w + b_n w] - [w]\,\dot{\overline{(\underline{a}_n \cdot \nabla u)}} \tag{2.15}$$

$$\mathcal{S}_J\,(u, w) \equiv \dot{w}\,[\underline{a}_n \cdot \nabla u]\,, \quad \mathcal{R}_J\,(u, w) \equiv -\,[u]\,\dot{\overline{(\underline{a}_n \cdot \nabla w + b_n w)}} \tag{2.16}$$

$$\mathcal{S}\,(w, u) \equiv \dot{u}\,[\underline{a}_n \cdot \nabla w + b_n w] \quad \textit{and} \quad \mathcal{R}\,(w, u) \equiv -\,[w]\,\dot{\overline{(\underline{a}_n \cdot \nabla u)}} \tag{2.17}$$

Define $\tilde{N}_1 \equiv N_P \cap N_B \cap N_{R_J}$ and $\tilde{N}_2 \equiv N_Q \cap N_C \cap N_R$, then a function $v \in \tilde{N}_1$, if and only if

$$Pv = 0, \quad Bv = 0 \quad \textit{and} \quad R_J v = 0 \tag{2.18}$$

and $w \in \tilde{N}_2$, if and only if

$$Qw = 0, \quad Cw = 0 \quad \textit{and} \quad Rw = 0 \tag{2.19}$$

The result that is basic for deriving the kind of domain decomposition to be applied in the present article, is given by the Theorem of Section 10 of Ref. [9]:

Theorem 2.1 *Assume $\mathcal{E} \subset \tilde{N}_2$ is a system of weighting functions TH-complete for S^* [9]. Let $u_P \in \hat{D}_1$ be such that*

$$Pu_P = Pu_\Omega, \quad Bu_P = Bu_\partial \quad and \quad R_J u_P = R_J u_\Sigma \tag{2.20}$$

Then there exists $v \in \tilde{N}_1$ such that

$$-\langle S^* v, w\rangle = \langle S_J\left(u_P - u_\Sigma\right), w\rangle, \quad \forall w \in \mathcal{E} \subset \tilde{N}_2 \tag{2.21}$$

In addition, define $\hat{u} \in \hat{D}_1$ by $\hat{u} \equiv u_P + v$. Then $\hat{u} \in \hat{D}_1$ contains the sought information. Even more, $\hat{u} \equiv u$, where u is the solution of the BVPJ.

Observe that Eq.(2.21) can also be written as

$$-\langle S^* v, w\rangle = \langle S_J u_P, w\rangle - \left\langle j^1, w\right\rangle, \quad \forall w \in \mathcal{E} \subset \tilde{N}_2 \tag{2.22}$$

where $\left\langle j^1, w\right\rangle \equiv \int_\Sigma w j_\Sigma^1 dx$.

3. Interpretation of the Algebraic Theory. According to the definitions given in Section 2, a function $v \in \tilde{N}_1$ if and only if

$$\mathcal{L}v \equiv -\nabla\cdot(\underline{\underline{a}}\cdot\nabla uv) + \nabla\cdot(\underline{b}v) + cv = 0, \quad v = 0, \quad on \quad \partial\Omega \quad and \quad [v] = 0, \quad on \quad \Sigma \tag{3.1}$$

In addition, a function $w \in \tilde{N}_2$, if and only if

$$\mathcal{L}^* w \equiv -\nabla\cdot(\underline{\underline{a}}\cdot\nabla w) - \underline{b}\cdot\nabla w + cw = 0, \quad w = 0, \quad on \quad \partial\Omega \quad and \quad [v] = 0, \quad on \quad \Sigma \tag{3.2}$$

i.e., such functions satisfy the homogenous adjoint equation, are continuous and vanish on the external boundary.

When $\mathcal{S}(w, u)$ is given by Eq.(2.17), the *sought information* is the average of the solution of the BVPJ on Σ. Even more, the choice of the pair of decompositions $\{S_J, R_J\}$ and $\{S, R\}$, is optimal [9], because the problem

$$(P - B - J)\,\hat{u} = Pu_\Omega - Bu_\partial - Ju_\Sigma \tag{3.3}$$

subjected to

$$S^*\hat{u} = S^* u_I \tag{3.4}$$

is well posed and local. Indeed, Eq.(2.3) corresponds to the following system of equations

$$\mathcal{L}\hat{u} \equiv -\nabla\cdot(\underline{\underline{a}}\cdot\nabla\hat{u}) + \nabla\cdot(\underline{b}\hat{u}) + c\hat{u} = f_\Omega \equiv \mathcal{L}u_\Omega, \quad in \quad \Omega_i, \quad i = 1, ..., E \tag{3.5}$$

subjected to the boundary conditions

$$\hat{u} = u_\partial; \quad on \quad \partial\Omega, \tag{3.6}$$

and the jump conditions

$$[\hat{u}] = [u_\Sigma] \equiv j_\Sigma^0; \quad on \quad \Sigma, \tag{3.7}$$

In addition, Eq.(2.4) corresponds to the condition

$$\dot{\hat{u}} = \dot{u}_I; \quad on \quad \Sigma, \tag{3.8}$$

i.e., the average across Σ, of $\hat{u}$, is prescribed. Therefore

$$\hat{u}_+ \equiv \dot{\hat{u}} + \frac{1}{2}[u] = \dot{\hat{u}}_I + \frac{1}{2} j_\Sigma^0 \quad and \quad \hat{u}_- \equiv \dot{\hat{u}} - \frac{1}{2}[u] = \dot{\hat{u}}_I - \frac{1}{2} j_\Sigma^0 \tag{3.9}$$

and it is seen that the system of equations (2.3) and (2.4), is equivalent to a family of well-posed local problems defined in each on of the subdomains of the partition.

The Eqs.(2.20), fulfilled by $u_P \in \hat{D}_1$, are

$$\mathcal{L}u_P \equiv -\nabla \cdot (\underline{\underline{a}} \cdot \nabla u_P) + \nabla \cdot (\underline{b} u_P) + c u_P = f_\Omega \equiv \mathcal{L}u_\Omega, \quad in \quad \Omega_i, \quad i = 1, ..., E \tag{3.10}$$

subjected to the boundary conditions

$$u_P = u_\partial; \quad on \quad \partial\Omega, \tag{3.11}$$

and the jump conditions

$$[u_P] = [u_\Sigma] \equiv j^0_\Sigma; \quad on \quad \Sigma, \tag{3.12}$$

This is the same as Eq. (2.3); i.e., the system of Eqs.(2.5) to (2.7). However, Eq.(2.3) is not imposed on u_P and, therefore, it is not uniquely determined. However, u_P is uniquely determined if it's average across Σ is specified. This can be chosen arbitrarily, except that it must be compatible with the external boundary conditions of Eq. (2.11). It must be observed that in a similar manner, elements of each one of the sets $\tilde{N}_1$ and $\tilde{N}_2$ are determined uniquely by the traces on Σ. A convenient manner of constructing such functions is, therefore, to specify their traces on Σ, and then solve each one of the well posed problems which in this manner are defined in the subdomains of the partition, as will be done numerically in the following Sections.

4. TH-Complete Systems of Test Functions. Discussions of TH-complete systems, in the context of the general theory, may be found in [6],[5]. Additional details in connection with applications to second order elliptic problems may be found in [8]. In what follows the traces on Σ, of the weighting functions, will be taken to be families of piecewise polynomials defined on Σ_{ij} (Fig. 4.1) . This kind of TH-complete families were first described in [5]. According to that figure, Σ_{ij} is the union of four intervals and using the numbering of internal boundaries of Fig. 4.1, associated with each node (x_i, y_j), five classes of weighting functions can be constructed [8]:

Class 0.- This is made of only one function, which is linear in each one of the four interior boundaries between the rectangles of Fig. 4.2, and such that $(x_i, y_j) = 1$.

Class 1.- The restriction to interval "1", of Fig. 4.2 is a polynomial which vanishes at the end points of interval "1". There is one such polynomial for each degree (G) greater than one.

Classes 2 to 4, are defined replacing interval "1" by the interval of the corresponding number in the definition of Class 1 [5].

5. The Numerical Implementation. In the theory that was presented in previous Sections, it is assumed that the exact local solutions are available. In numerical applications, they have to be produced by means of numerical methods and are, therefore only approximate solutions. Actually, the approximate nature of numerical solutions derived using TH-Domain Decomposition (TH-DD), stems from two sources: one of them is due to the approximate nature of the local solutions, which has just been mentioned, and the other one comes from the fact that TH-complete systems for problems in several dimensions constitute infinite families and in numerical implementations one can apply only finite sets of test functions. In particular, with reference to the families of functions introduced in the previous Section, one may construct algorithms in which only polynomials of degree less or equal to G, where G is a given number, are kept in each one of the Classes "1" to "4". In general, each choice of G will give rise to a different kind of algorithm.

In this Section the following notations are used, $H^0_i(x)$ is the one dimensional Hermite cubic polynomial with support in the interval (x_{i-1}, x_{i+1}), which takes the value 1 at node

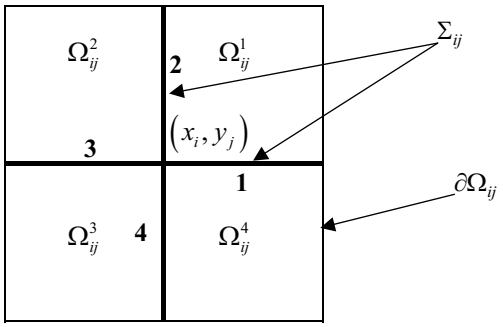


Figure 4.1: Subregion Ω_{ij} associated with the node (x_i, y_j).

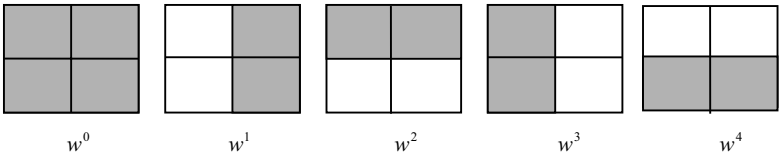


Figure 4.2: Supports of five classes of weighting functions.

x_i and zero at nodes x_{i-1} and x_{i+1}; and its first derivative is zero at all nodes x_{i-1}, x_i and x_{i+1}. Similarly, $H_i^1(x)$ - is the one dimensional Hermite cubic polynomial with support in the interval (x_{i-1}, x_{i+1}), which takes the value zero at nodes x_{i-1}, x_i and x_{i+1}; and its first derivative takes the value 1 at node x_i and zero at the other nodes x_{i-1} and x_{i+1}.

5.1. The Weighting Functions. In the numerical implementations reported in [8], two families of test functions were constructed:

$$\mathcal{F} \equiv \left\{ w_{ij}^0, w_{ij}^1, w_{ij}^2 \right\} \text{ and } \widehat{\mathcal{F}} \equiv \left\{ \overset{0}{\widehat{w}}_{ij}, \overset{1}{\widehat{w}}_{ij}, \overset{2}{\widehat{w}}_{ij}, \overset{3}{\widehat{w}}_{ij}, \overset{4}{\widehat{w}}_{ij} \right\} \tag{5.1}$$

Here, $w_{ij}^0 \equiv \overset{0}{\widehat{w}}_{ij}$ is the unique function belonging to Class "0"-i.e., piecewise linear on Σ-, of Section 6, and $\overset{\alpha}{\widehat{w}}_{ij}$ is a function of Class "α", for each $\alpha = 1, ..., 4$, which fulfills, at interval "α", the boundary condition $\overset{\alpha}{\widehat{w}}_{ij}(x, y_j) = H_i^1(x)$, for $\alpha = 1, 3$, and $\overset{\alpha}{\widehat{w}}_{ij}(x_i, y) = H_j^1(y)$, for $\alpha = 2, 4$. In addition, one defines

$$w_{ij}^1(x, y) \equiv \overset{1}{\widehat{w}}_{ij}(x, y) + \overset{3}{\widehat{w}}_{ij}(x, y) \text{ and } w_{ij}^2(x, y) \equiv \overset{2}{\widehat{w}}_{ij}(x, y) + \overset{4}{\widehat{w}}_{ij}(x, y) \tag{5.2}$$

Observe that the supports of w_{ij}^1 and w_{ij}^2 are the whole rectangle Ω_{ij} . In addition, they fulfill the local boundary conditions $w_{ij}^1(x, y_j) = H_i^1(x)$ at the interval $x_{i-1} \leqslant x \leqslant x_{i+1}$ together with $w_{ij}^2(x_i, y) = H_j^1(y)$ at the interval $y_{j-1} \leqslant y \leqslant y_{j+1}$.

In Ref. [8], the family $\widehat{\mathcal{F}}$ was first constructed and the family $\mathcal{F}$ was then derived by application of Eq.(4.2). The family $\widehat{\mathcal{F}}$ was built by solving local boundary value problems in each one of the subregions $\left\{\Omega_{ij}^1, \Omega_{ij}^2, \Omega_{ij}^3, \Omega_{ij}^4\right\}$, separately. This was done introducing a set of functions $\left\{B_{ij}^0, B_{ij}^1, B_{ij}^2, B_{ij}^3, B_{ij}^4\right\}$, which satisfy the boundary conditions and adding to it a linear combination of a family of functions $\left\{N_{ij}^1, N_{ij}^2, N_{ij}^3, N_{ij}^4\right\}$ which vanish on the

boundary of each one of the subdomains $\{\Omega_{ij}^1, \Omega_{ij}^2, \Omega_{ij}^3, \Omega_{ij}^4\}$, in order to fulfill the differential equation.

This leads to

$$\widehat{w}_{ij}^{\alpha}(x,y) = B_{ij}^{\alpha}(x,y) + \sum_{\beta=1}^{4} C_{ij}^{\alpha\beta} N_{ij}^{\beta}(x,y); \quad \alpha = 0, ..., 4 \tag{5.3}$$

The coefficients $C_{ij}^{\alpha\beta}$ are constant at each one of the subdomains $\{\Omega_{ij}^1, \Omega_{ij}^2, \Omega_{ij}^3, \Omega_{ij}^4\}$, but only piecewise constant in Ω_{ij} (Fig. 4.1). Therefore, each one of the functions $\widehat{w}_{ij}^{\alpha}(x,y)$ has different expressions at each one of the rectangles $\{\Omega_{ij}^1, \Omega_{ij}^2, \Omega_{ij}^3, \Omega_{ij}^4\}$. The same applies to the functions $\{B_{ij}^0, B_{ij}^1, B_{ij}^2, B_{ij}^3, B_{ij}^4\}$. The coefficients were obtained solving the system of collocation equations at four Gaussian points

$$\sum_{\beta=1}^{4} C_{ij}^{\alpha\beta} \mathcal{L}^* N_{ij}^{\beta}(x^p, y^p) = \mathcal{L}^* B_{ij}^{\alpha}(x^p, y^p); \quad p = 1, ..., 4 \tag{5.4}$$

5.2. Optimal Interpolation. According to the Theorem of Section 2, the approximate solution $\hat{u} \in \hat{D}_1$ is given by

$$\hat{u} = \hat{u}_P + v \tag{5.5}$$

The function fulfills Eqs.(2.9) to (2.11). As mentioned in Section 2, for its construction one can choose the average of this function arbitrarily, but compatible with the external boundary conditions of Eq.(2.11), and then solve the boundary value problems which are defined, when this specification is joined to the System of Eqs. (2.10) to (2.12), and Eq.(2.9) is also applied. These problems may be solved by any numerical method but in [8], orthogonal collocation was used and similar manner to that explained in the last Sub-Section.

The system of base functions used for building $v \in \hat{D}_1$ can be constructed in a similar manner. It is based on the fact that $v \subset \tilde{N}_1$. So those functions must fulfill the system of equations (2.1); i.e., the homogenous differential equation, and they be continuous and vanish on the external boundary. It is advantageous in many instances, to choose the traces on Σ of such base functions to be the same as those of the weighting functions, as was explained in the Sub-Section 5.1. In that case, Eq.(4.3) can also be applied for the construction the base functions, but to determine the coefficients $C_{ij}^{\alpha\beta}$ one has to replace in Eq.(4.4), the adjoint differential operator $\mathcal{L}^*$ by the differential operator $\mathcal{L}$, itself.

5.3. The Algorithms. To obtain the system of equations satisfied by the values of v on Σ (the values of v on both sides of Σ are the same since it is continuous), one has to apply Eq.(2.22). This is

$$-\int_{\Sigma} v\,[\underline{a}_n \cdot \nabla w] dx = \int_{\Sigma} w\,[\underline{a}_n \cdot \nabla u_P] dx - \int_{\Sigma} w j_{\Sigma}^1 dx \tag{5.6}$$

This form is simpler than that presented in [8], where additional details can be found.

In [8], two algorithms were developed. In **Algorithm 1**, both base and test functions are piecewise linear on Σ, while both of them are piecewise cubic on Σ in **Algorithm 2**.

6. Conclusions. This article illustrates the applications of Trefftz-Herrera methods to the derivation of new discretization procedures. In particular, in Trefftz-Herrera method, the order of approximation that is used in the internal boundary is independent of that used in the interior of the elements of the partition. Using this fact a non-standard method of collocation on Hermite cubics is presented which possesses many advantages over standard methods. Two algorithms are discussed, one in which the interpolation on Σ is piecewise linear and another in which it is piecewise cubic. Quadratic interpolation is also possible but was not discussed here. In this manner, a dramatic reduction in the number of degrees

of freedom associated with each node is obtained: in the standard method of collocation that number is two in one dimension, four in two dimensions and eight in three dimensions, while for some of the new algorithms they are only one in all space dimensions -this is due to the relaxation in the continuity conditions required by indirect methods-. Also, the global matrix is symmetric and positive definite when so is the differential operator, while in the standard method of collocation, using Hermite cubics, this does not happen. In addition, it must be mentioned that the boundary value problem with prescribed jumps at the internal boundaries can be treated as easily as the smooth problem -i.e., that with zero jumps-, because the solution matrix and the order of precision is the same for both problems. It must be observed also that, when the indirect method is applied, the error of the approximate solution stems from two sources: the approximate nature of the test functions, and the fact that TH-complete systems of test functions -which are infinite for problems in several dimensions- are approximated by finite families of such functions. In particular, when Hermite cubics are used to approximate the local solutions, in the problems treated in this paper, the error is $O\left(h^4\right)$, if the test functions are piece-wise cubic on Σ, and it is $O\left(h^2\right)$ when the test functions are only piece-wise linear, on that interior boundary. Finally, the construction of the test functions is quite suitable to be computed in parallel.

REFERENCES

[1] B. Bialecki and M. Dryja. Multilevel Additive and Multiplicative Methods for Orthogonal Spline Collocation Problems. *Numer. Math.*, 77:35–58, 1997.

[2] B. Bialecki and G. Fairweather. Orthogonal Spline Collocation Methods for Partial Differential Equations. *Journal of Computational and Applied Mathematics*, 128:55–82, 2001.

[3] M. Diaz and I. Herrera. Indirect Method of Collocation for the Biharmonic Equation. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[4] F. Garcia-Nocetti, I. Herrera, R. Yates, E. Rubio, and L. Ochoa. The Direct Approach to Domain Decomposition Methods. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[5] I. Herrera. Trefftz-Herrera Domain Decomposition. *Advances in Engineering Software*, 24:43–56, 1995.

[6] I. Herrera. Trefftz Method: A General Theory. *Numerical Methods for Partial Differential Equations*, 16(6):561–580, 2000.

[7] I. Herrera. Unified theory of domain decomposition methods. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[8] I. Herrera, R. Yates, and M. Diaz. General Theory of Domain Decomposition: Indirect Methods. *Numerical Methods for Partial Differential Equations*, 18(3):296–322, 2002.

[9] I. Herrera, R. Yates, and M. Diaz. The Indirect Approach to Domain Decomposition. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[10] R. Yates and I. Herrera. Parallel Implementation of Indirect Collocation Method. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

24. Dual preconditioners for mortar discretization of elliptic problems

M. Dryja[1], W. Proskurowski[2]

1. Introduction. In this paper, we discuss a second order elliptic problem with discontinuous coefficients defined on a polygonal region Ω which is a union of two polygons, Ω_1 and Ω_2. The problem is discretized by the finite element method on non-matching triangulation across $\overline{\Gamma} = \overline{\Omega}_1 \cap \overline{\Omega}_2$. The discrete problem is described using the mortar technique in the space with constraints (the mortar condition) and in the space without constraints using Lagrange multipliers, see [2] and [1].

The goal of this paper is to compare two preconditioners, dual Neumann-Dirichlet and dual Neumann-Neumann (or FETI, see [5], [6], [7]) used for solving the discrete problem formulated in the space without constraints using Lagrange multipliers. An analysis of convergence of the discussed preconditioners is given. Such analysis to our knowledge has not yet been previously established. The theory is supported by numerical experiments.

The paper is organized as follows. In Section 2, the differential and discrete problems are formulated. In Section 3, a matrix form of discrete problems is given. The preconditioners are described and analyzed in Sections 4, while some aspects of their implementation are presented in Section 5. Finally, numerical results and comparisons of the considered preconditioners are given in Section 6.

2. Mortar discrete problem. We consider the following differential problem:
Find $u^* \in H_0^1(\Omega)$ such that

$$a(u^*, v) = f(v), \qquad v \in H_0^1(\Omega), \tag{2.1}$$

where

$$a(u, v) = (\rho(x)\nabla u, \nabla v)_{L^2(\Omega)}, \; f(v) = (f, v)_{L^2(\Omega)}.$$

We assume that Ω is a polygonal region. Let Ω be a union of two disjoint polygonal subregions $\Omega_i, i = 1, 2$, of a diameter one. We additionally assume that $\rho(x) \geq \rho_0 > 0$ is a continuous function in each Ω_i and, for simplicity of presentation, that $\rho(x) = \rho_i =$ constant on Ω_i.

In each Ω_i, a triangulation is introduced with triangular elements $e_i^{(k)}$ and a parameter $h_i = \max_k h_i^{(k)}$, where $h_i^{(k)}$ is a diameter of $e_i^{(k)}$. The resulting triangulation of Ω is non-matching across $\overline{\Gamma} = \overline{\Omega}_1 \cap \overline{\Omega}_2$. We assume that the h_i-triangulation in each Ω_i is quasi-uniform, see [3].

Let $X_i(\Omega_i)$ be the finite element space of piecewise linear continuous functions defined on the triangulation of Ω_i and vanishing on $\partial\Omega_i \cap \partial\Omega$, and let

$$X^h(\Omega) = X_1(\Omega_1) \times X_2(\Omega_2).$$

Note that $X^h \not\subset H_0^1(\Omega)$; therefore it cannot be used for discretization of (2.1). To discretize (2.1) some weak continuity on Γ for $v \in X^h$ is imposed and it is called a *mortar condition*, see [2]. To describe the mortar condition we assume that $\rho_1 \leq \rho_2$ and select a face of Ω_2, geometrically equal to Γ, as a *mortar (master)* and denote it by γ, while $\delta = \Gamma$ as a face of Ω_1 as *non-mortar (slave)*. This choice is arbitrary in the case $\rho_1 = \rho_2$, however in

[1]Department of Mathematics, Warsaw University, Banacha 2, 02-097 Warsaw, Poland, dryja@mimuw.edu.pl. This work was supported in part by the National Science Foundation under Grant NSF-CCR-9732208 and in part by the Polish Science Foundation under Grant 2P03A 02116

[2]Department of Mathematics, University of Southern California, Los Angeles, CA 90089-1113, USA, proskuro@math.usc.edu.

our case, $\rho_1 \leq \rho_2$ and it is important for the analysis of convergence to choose as the mortar side the one where the coefficient is larger. In the analysis of the FETI method we need that $\frac{h_\gamma}{h_\delta}$ be uniformly bounded, where h_δ and h_γ are the steps of triangulation on δ and γ, respectively.

Let $W_1(\delta)$ and $W_2(\gamma)$ be the restrictions of $X_1(\Omega_1)$ and $X_2(\Omega_2)$ to δ and γ, respectively. Note that they are different because they are defined on different 1-D triangulations of Γ. Let $M(\delta)$ be a space of piecewise linear continuous functions defined on the triangulation of δ with constant values on elements which intersect $\partial\delta$.

We say that $u = (u_1, u_2) \in X^h(\Omega)$ satisfies the mortar condition on $\delta(\delta = \gamma = \Gamma)$ if

$$\int_\delta (u_1 - u_2)\psi ds = 0, \quad \psi \in M(\delta). \tag{2.2}$$

Note that (2.2) for a given u_2 can be written as $u_1 = \pi(u_2, Tr\ u_1)$ where $\pi(u_2, Tr\ u_1) : L^2(\delta) \to W_1(\delta)$ is defined by

$$\begin{cases} \int_\delta \pi(u_2, Tr\ u_1)\psi ds &= \int_\delta u_2\psi ds,\ \psi \in M(\delta), \\ \\ Tr\pi(u_2, Tr\ u_1) &= Tr\ u_1. \end{cases} \tag{2.3}$$

Here $Tr\ v$ is a trace of v on $\partial\delta$. In our case $Tr\ u_1 = 0$.

Let $V^h(\Omega)$ be a subspace of $X^h(\Omega)$ of functions which satisfy the mortar condition (2.2) on δ. The discrete problem for (2.1) in V^h is of the form:

Find $u_h^* = (u_{1h}^*, u_{2h}^*) \in V^h$ such that

$$\sum_{i=1}^{2} a_i(u_{ih}^*, v_{ih}) = f(v_h), \quad v_h = (v_{1h}, v_{2h}) \in V^h, \tag{2.4}$$

where $a_i(u_i, v_i) = \rho_i(\nabla u_i, \nabla v_i)_{L^2(\Omega_i)}$. This problem has a unique solution and its error bound is known, see [2].

The discrete problem (2.4) can be rewritten as a saddle-point problem using Lagrange multipliers as follows:

Let for $u = (u_1, u_2) \in X^h(\Omega)$ and $\psi \in M(\delta)$

$$b(u, \psi) \equiv \int_\delta (u_1 - u_2)\psi dx.$$

Find $(u_h^*, \lambda_h^*) \in X^h(\Omega) \times M(\delta)$ such that

$$\begin{cases} a(u_h^*, v_h) + b(v_h, \lambda_h^*) = f(v_h),\ v_h \in X^h(\Omega), \\ \\ b(u_h^*, \psi) = 0, \qquad \psi \in M(\delta). \end{cases} \tag{2.5}$$

It is easy to see that (2.5) is equivalent to (2.4), i.e. the solution u_h^* of (2.5) is the solution of (2.4) and vice versa. Therefore the problem (2.5) has a unique solution. An analysis of (2.5) can be done straightforwardly using the inf-sup condition, including the error bound, see [1], [2].

3. Matrix form. In this section we derive the matrix form of the discrete problem (2.5).

To provide a matrix form of (2.5) we need a matrix formulation of the mortar condition, i.e. the matrix form of $b(\cdot, \cdot)$. Using the nodal basis functions, $\varphi_k^{(1)} \in W_1(\delta)$, $\varphi_k^{(2)} \in W_2(\gamma)$, and $\psi_l \in M(\delta)$, one can rewrite the equation (2.2) as

$$B_\delta u_{1\delta} - B_\gamma u_{2\gamma} = 0, \tag{3.1}$$

where $u_{1\delta}$ and $u_{2\gamma}$ are vectors that represent $u_{1|\delta} \in W_1(\delta)$ and $u_{2|\gamma} \in W_2(\gamma)$, respectively, and

$$B_\delta = \{(\psi_l, \varphi_k^{(1)})_{L^2(\delta)}\}, \qquad l,k = 1,\dots,n_\delta,$$

$$B_\gamma = \{(\psi_l, \varphi_k^{(2)})_{L^2(\gamma)}\}, \qquad l = 1,\dots,n_\delta;\ k = 1,\dots,n_\gamma.$$

Here $n_\delta = \dim(M(\delta)) = \dim(W_1(\delta)), n_\gamma = \dim(W_2(\gamma))$. Note that B_δ is a square tridiagonal matrix $n_\delta \times n_\delta$, symmetric and positive definite, and $\mathrm{cond}(B_\delta) \sim 1$, while B_γ is a rectangular matrix $n_\delta \times n_\gamma$. Hence for $(u,\lambda) \in X^h(\Omega) \times M(\delta)$

$$b(u,\lambda) = (B_\delta u_{1\delta}, \lambda)_{R^{n_\delta}} - (B_\gamma u_{2\gamma}, \lambda)_{R^{n_\delta}},$$

where here and below a vector representation of λ is also denoted by λ.

Thus (2.5) can be presented in the form

$$\begin{pmatrix} A_{II}^{(1)} & A_{I\delta}^{(1)} & 0 & 0 & 0 \\ A_{\delta I}^{(1)} & A_{\delta\delta}^{(1)} & 0 & 0 & B_\delta \\ 0 & 0 & A_{II}^{(2)} & A_{I\gamma}^{(2)} & 0 \\ 0 & 0 & A_{\gamma I}^{(2)} & A_{\gamma\gamma}^{(2)} & -B_\gamma^T \\ 0 & B_\delta & 0 & -B_\gamma & 0 \end{pmatrix} \begin{pmatrix} u_I^{(1)} \\ u_\delta^{(1)} \\ u_I^{(2)} \\ u_\gamma^{(2)} \\ \lambda_\delta \end{pmatrix} = \begin{pmatrix} F_I^{(1)} \\ F_\delta^{(1)} \\ F_I^{(2)} \\ F_\gamma^{(2)} \\ 0 \end{pmatrix} \tag{3.2}$$

Here $\left\{u_I^{(1)}, u_\delta^{(1)}\right\}^T$ and $\left\{u_I^{(2)}, u_\gamma^{(2)}\right\}^T$ correspond to the nodal values of u_1^* and u_2^* at the interior nodal points of Ω_i, δ and γ, denoted by Ω_{ih}, δ_h and γ_h, respectively, and λ_δ is a vector representation of λ^*;

$A_{II}^{(1)} = \left\{a_1(\varphi_k^{(1)}, \varphi_l^{(1)})\right\} \quad x_k, x_l \in \Omega_{1h},$

$A_{I\delta}^{(1)} = \left\{a_1(\varphi_k^{(1)}, \varphi_l^{(1)})\right\} \quad x_k \in \Omega_{1h}$ and $x_l \in \delta_h,$

$A_{\delta\delta}^{(1)} = \left\{a_1(\varphi_k^{(1)}, \varphi_l^{(1)})\right\} \quad x_k, x_l \in \delta_h;$

$A_{II}^{(2)}$, $A_{I\gamma}^{(2)}$ and $A_{\gamma\gamma}^{(2)}$ are defined in a similar way. Note that $(A_{I\delta}^{(1)}) = (A_{\delta I}^{(1)})^T$ and $(A_{I\gamma}^{(2)}) = (A_{\gamma I}^{(2)})^T$. The matrix of (3.2) is invertible.

4. Preconditioners for (2.5). In this section we define and analyze preconditioners for problem (2.5). They will be defined for the Schur complement system with respect to unknowns λ_δ, the Lagrange multipliers.

Let

$$A^{(1)} = \begin{pmatrix} A_{II}^{(1)} & A_{I\delta}^{(1)} \\ A_{\delta I}^{(1)} & A_{\delta\delta}^{(1)} \end{pmatrix}, \quad A^{(2)} = \begin{pmatrix} A_{II}^{(2)} & A_{I\gamma}^{(1)} \\ A_{\gamma I}^{(2)} & A_{\gamma\gamma}^{(2)} \end{pmatrix}.$$

Their Schur complement matrices with respect to $u_\delta^{(1)}$ and $u_\gamma^{(2)}$, respectively, are of the form

$$S_1 = A_{\delta\delta}^{(1)} - A_{\delta I}^{(1)} \left(A_{II}^{(1)}\right)^{-1} A_{I\delta}^{(1)}, \quad S_2 = A_{\gamma\gamma}^{(2)} - A_{\gamma I}^{(2)} \left(A_{II}^{(2)}\right)^{-1} A_{I\gamma}^{(2)}. \tag{4.1}$$

We consider system (3.2). We first eliminate the unknowns $u_I^{(1)}$ and $u_I^{(2)}$. Using rows 1 and 3 of (3.2) and substituting the result in rows 2 and 4 of (3.2) we obtain

$$\begin{pmatrix} S_1 & 0 & B_\delta \\ 0 & S_2 & -B_\gamma^T \\ B_\delta & -B_\gamma & 0 \end{pmatrix} \begin{pmatrix} u_\delta^{(1)} \\ u_\gamma^{(2)} \\ \lambda_\delta \end{pmatrix} = \begin{pmatrix} F_\delta^{(1)} - (A_{I\delta}^{(1)})^T (A_{II}^{(1)})^{-1} F_I^{(1)} \\ F_\gamma^{(2)} - (A_{I\gamma}^{(2)})^T (A_{II}^{(2)})^{-1} F_I^{(2)} \\ 0 \end{pmatrix}, \tag{4.2}$$

where S_1 and S_2 are given by (4.1).

Then, we eliminate the unknowns $u_\delta^{(1)}$ and $u_\gamma^{(2)}$ from this system. Using rows 1 and 2 of (4.2) and setting $\hat{\lambda}_\delta = B_\delta \lambda_\delta$, we obtain

$$S_L \hat{\lambda}_\delta = F_\lambda, \tag{4.3}$$

where

$$S_L = S_1^{-1} + B_\delta^{-1} B_\gamma S_2^{-1} B_\gamma^T B_\delta^{-1}, \tag{4.4}$$

and

$$F_\lambda = S_1^{-1}(F_\delta^{(1)} - (A_{I\delta}^{(1)})^T (A_{II}^{(1)})^{-1} F_I^{(1)}) - B_\delta^{-1} B_\gamma S_2^{-1}(F_\gamma^{(2)} - (A_{I\gamma}^{(2)})^T (A_{II}^{(2)})^{-1} F_I^{(2)}).$$

The dual Schur complement matrix S_L is symmetric and positive definite, n_δ by n_δ.

Our goal is to define preconditioners for (4.3) dual to the Neumann-Dirichlet one and dual to the Neumann-Neumann one. The latter, for the matching triangulation, is called FETI (the Finite Element Tearing and Interconnecting), see [5], [6], [7].

4.1. Neumann-Dirichlet (N-D) preconditioner. The Neumann-Dirichlet dual preconditioner for S_L is defined by S_1^{-1}.

Theorem 4.1 *For any $\lambda \in R^{n_\delta}$ and $\rho_1 \leq \rho_2$ the following holds*

$$\left(S_1^{-1}\lambda, \lambda\right)_{R^{n_\delta}} \leq (S_L \lambda, \lambda)_{R^{n_\delta}} \leq C \left(S_1^{-1}\lambda, \lambda\right)_{R^{n_\delta}} \tag{4.5}$$

where C is a positive constant independent of h_i and ρ_i, $i = 1, 2$.

For the proof see [4].

4.2. FETI (N-N) preconditioner. We now discuss FETI method for solving (4.3). This preconditioner is of the form

$$G = \left(\frac{\rho_2}{\rho_1 + \rho_2} S_1 + \frac{\rho_1}{\rho_1 + \rho_2} B_\delta^{-1} B_\gamma S_2 B_\gamma^T B_\delta^{-1} \right)^{-1}. \tag{4.6}$$

Theorem 4.2 *Let $\frac{h_\gamma}{h_\delta}$ be uniformly bounded. For any $\lambda \in R^{n_\delta}$ and $\rho_1 \leq \rho_2$ holds*

$$\frac{1}{2}(G\lambda, \lambda)_{R^{n_\delta}} \leq (S_L \lambda, \lambda)_{R^{n_\delta}} \leq C (G\lambda, \lambda)_{R^{n_\delta}} \tag{4.7}$$

where C is a positive constant independent of h_i and ρ_i, $i = 1, 2$.

For the proof see [4].

5. Implementation aspects. In this section we discuss some implementation aspects of solving the Schur complement systems.

To solve the dual Schur complement equation (4.3) we use the preconditioned conjugate gradient (PCG) iterations. Here, we only need to describe the implementation of **1.** the multiplication of a vector by the dual Schur complement matrix $S_L \in R^{n_\delta \times n_\delta}$ (defined by (4.4)), and **2.** solving a system with **a.** the Neumann-Dirichlet dual preconditioner S_1 (defined by (4.1)), and with **b.** the Neumann-Neumann dual preconditioner G (defined by (4.6)).

Let us recall that the iterations are carried out on the non-mortar side δ of the interface Γ with the number of grid equal to n_δ. The mortar condition (2.3) ensures the proper transfer of information across the interface.

1. Compute $r^k = S_L \lambda^k$ for any given $\lambda^k \in R^{n_\delta}$. The multiplication by S_L reduces to solving two independent problems:

i) Compute $r_1^k = S_1^{-1}\lambda^k$, i.e. solve

$$S_1 r_1^k = \lambda^k. \tag{5.1}$$

This reduces to solving the Neumann problem on Ω_1, and more precisely, the problem with non-homogeneous Neumann boundary conditions on δ and homogeneous Dirichlet ones on $\partial\Omega_1\backslash\delta$, see (4.1)

$$\begin{pmatrix} A_{II}^{(1)} & A_{I\delta}^{(1)} \\ A_{\delta I}^{(1)} & A_{\delta\delta}^{(1)} \end{pmatrix} \begin{pmatrix} r_I^{(1)} \\ r_1^k \end{pmatrix} = \begin{pmatrix} 0 \\ \lambda^k \end{pmatrix}, \tag{5.2}$$

ii) Compute $r_2^k = B_\delta^{-1} B_\gamma S_2^{-1} B_\gamma^T B_\delta^{-1} \lambda^k$. This step is similar to (5.2). The only difference is that before solving the Neumann problem on Ω_2 we need first to solve $B_\delta z_\delta^{(1)} = \lambda^k$, then to compute $B_\gamma^T z_\delta^{(1)}$; and after solving the Neumann problem on Ω_2 we need to perform these operations in reversed order.

Finally, $r^k = r_1^k + r_2^k$.

2a. Compute $r_1^k = S_1\lambda^k$ for any given $\lambda^k \in R^{n_\delta}$.

We first compute, see (4.1),

$$S_1\lambda^k = A_{\delta\delta}^{(1)}\lambda^k - A_{\delta I}^{(1)}\left(A_{II}^{(1)}\right)^{-1} A_{I\delta}^{(1)}\lambda^k.$$

This reduces to solving the Dirichlet problem in Ω_1 as follows

$$A_{II}^{(1)} v_I^{(1)} = A_{I\delta}^{(1)}\lambda^k \tag{5.3}$$

and to computing $r_1^k = A_{\delta\delta}^{(1)}\lambda^k - A_{\delta I}^{(1)} v_I^{(1)}$.

2b. Compute $r^k = G^{-1}\lambda^k$ for any given $\lambda^k \in R^{n_\delta}$.

This step consists of solving two Dirichlet problems, one in Ω_1, the other in Ω_2 (with the pre- and post- multiplications by $B_\delta^{-1}B_\gamma$ and its transpose, respectively, as in **1.ii)**), see (4.6) .

6. Numerical experiments. The test example for all our experiments is the weak formulation, see (2.1), of

$$-div(\rho(x)\nabla u) = f(x_1, x_2) \text{ in } \Omega, \tag{6.1}$$

with the Dirichlet boundary conditions on $\partial\Omega$, where Ω is a union of two disjoint rectangular subregions Ω_i, $i = 1, 2$, of a diameter one, and $\rho(x) = \rho_i$ is a positive constant in each Ω_i.

The problem (6.1) is discretized by the finite element method on non–matching triangulation across the interface Γ. The grids used in our experiments are: 1. *double grids*, where the grid on one side of the interface Γ is twice the one on the other side of Γ, with every other position of the nodes coinciding, 2. *staggered grids*, where the grid size, h on both sides of Γ is the same but the nodes are staggered, with the distance of $\frac{h}{2}$ between the nearest two nodes on the opposite sides of Γ, and 3. *mixed grids*, where the grid on one side of Γ is coarse with the grid size $2h$, while the grid on the other side of Γ is fine with the grid size h and staggered by $\frac{h}{2}$. The mixed grids may better represent general non-matching grids.

We select a face of Ω_2 which coincides with the interface Γ as the mortar side, while the face of Ω_1 is the non–mortar one. We choose the following combinations of the diffusion coefficients: 1. $\rho_1 = \rho_2$, 2. $1 = \rho_1 < \rho_2 = 1000$, and 3. $1 = \rho_2 < \rho_1 = 1000$ (the case not covered by the theory).

To create a discrete driving function $f(x_1, x_2)$ we generate a random discrete solution $u(x_1, x_2)$ and multiply it by the matrix (3.2).

We solve the problems using the preconditioned conjugate gradient (PCG) iterations (see Section 5 for the implementation aspects). The iterations are terminated when the

Table 6.1: Performance of the dual Neumann-Dirichlet (dual N-D, $Q = S_1 S_L$) and dual Neumann-Neumann (dual N-N, or FETI, $Q = G^{-1} S_L$) preconditioners for the finest meshes on different grids. The number of iterations and the estimate of the condition number are displayed.

grids	precon-ditioner	n_δ	n_γ	continuous		$\rho_2 < \rho_1$		$\rho_1 < \rho_2$	
				no. iter.	$\kappa(Q)$	no. iter.	$\kappa(Q)$	no. iter.	$\kappa(Q)$
double	dual N-D	255	127	5	2.00	10	$*$	2	1.001
		127	255	4	1.34	6	1.85	2	1.001
	dual N-N	255	127	11	9.97	23	$*$	7	5.00
		127	255	6	1.73	7	2.26	5	1.28
staggered	dual N-D	256	255	8	1.93	115	997.	3	1.30
		255	256	9	3.08	114	1176.	2	1.002
	dual N-N	256	255	13	4.27	144	1003.	9	2.85
		255	256	12	5.07	146	2957.	8	1.91
mixed	dual N-D	256	127	7	2.28	16	$*$	3	1.31
		127	256	10	10.98	13	91.0	3	1.01
	dual N-N	256	127	14	19.23	35	$*$	12	9.98
		127	256	15	22.21	18	181.7	8	2.96

norm of the residual has decreased 10^6 times in the norm generated by the inverse of the preconditioner matrix.

To estimate the condition number of the PCG iteration matrix we compute the tridiagonal matrix representing the restriction of the preconditioned Schur complement matrix to the space spanned by the conjugate gradient residuals.

The preconditioners considered behave as predicted by the theory: for $\rho_1 \leq \rho_2$ the convergence is independent of the grid size, see Table 6.2. Table 6.1 presents performance of the preconditioners for the finest meshes on different grids. The N-D and dual N-D preconditioners converge somewhat faster than the N-N and dual N-N (FETI) preconditioners. All four preconditioners are robust for cases with the discontinuity ratio of 1000 across the interface, see Table 6.1.

The differences in performance on different grids are qualitatively insignificant, thus in Table 6.2 we present only one set of experiments. Comparison of the convergence rate for the preconditioned and non-preconditioned iterations (on a chosen set of problems, see Table 6.2) shows that the first remain constant independently of the grid size, while the latter depend roughly proportional to the square root of the size of the iteration matrix[3]. From this one can infer that $cond(S) = O(\frac{1}{h})$ and $cond(S_L) = O(\frac{1}{h})$ even for problems with jump discontinuity at the interface.

Additionally, we performed experiments with the grid ratio across the interface varying in the range $\frac{h_\gamma}{h_\delta} = 2^k, k = -5(1)5$, i.e. from $\frac{1}{32}$ to $\frac{32}{1}$ (and different diffusion coefficients ρ, as before). Performance for the dual N-D preconditioner was virtually independent of the grid ratio, as was for the FETI preconditioner and $\frac{h_\gamma}{h_\delta} < 1$. For $\frac{h_\gamma}{h_\delta} > 1$ the condition number of the FETI iteration matrix grows almost quadratically with the grid ratio while the number of iterations increases only very slowly (and depends also on the grid size).

7. Conclusions. The preconditioners considered behave as predicted by the theory: for $\rho_1 \leq \rho_2$ the convergence is independent of the grid size and the jump of the discontinuity. All preconditioners considered are very robust for cases with the discontinuity ratio of 1000

[3]the number of iterations for the non-preconditioned problems in the range $n_\delta = 16$ to 256 is proportional to n_δ^p, where $p = 0.5 \pm 0.03$ as computed using *polyfit* in the loglog scale.

Table 6.2: Examples of the PCG iterations convergence for (2.5) with and without preconditioners (for $\rho_1 < \rho_2$) as a function of grid sizes on the mixed grids. The number of iterations and estimate of the condition number are displayed.

		no precond.		dual N-D precond.		dual N-N precond.	
n_δ	n_γ	no. iter.	$\kappa(Q)$	no. iter.	$\kappa(Q)$	no. iter.	$\kappa(Q)$
16	7	12	14.35	4	1.30	9	9.88
32	15	18	27.06	4	1.30	12	9.96
64	31	24	52.39	4	1.31	12	9.97
128	63	33	102.9	3	1.31	12	9.98
256	127	44	203.8	3	1.31	12	9.98
7	16	7	6.29	3	1.01	7	2.81
15	32	11	12.95	3	1.01	8	2.96
31	64	17	25.68	3	1.01	8	2.96
63	128	23	51.29	3	1.01	8	2.96
127	256	33	102.7	3	1.01	8	2.96

across the interface. One should be cautious not to generalize conclusions drawn on such limited two subdomain case. Nevertheless, the results are illuminating, and we intend to extend the experimental evidence to more complex subdivisions.

REFERENCES

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

[2] C. Bernardi, Y. Maday, and A. T. Patera. A new non conforming approach to domain decomposition: The mortar element method. In H. Brezis and J.-L. Lions, editors, *Collège de France Seminar*. Pitman, 1994. This paper appeared as a technical report about five years earlier.

[3] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*, volume 15 of *Texts in Applied Mathematics*. Springer-Verlag, New York, 1994.

[4] M. Dryja and W. Proskurowski. On preconditioners for mortar discretization of elliptic problems. *Numerical Linear Algebra with Applications*, 2002. to appear.

[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.*, 32:1205–1227, 1991.

[6] A. Klawonn and O. B. Widlund. FETI and Neumann–Neumann Iterative Substructuring Methods: Connections and New Results. *Comm. Pure Appl. Math.*, 54:57–90, January 2001.

[7] J. Mandel and R. Tezaur. Convergence of a Substructuring Method with Lagrange Multipliers. *Numer. Math.*, 73:473–487, 1996.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

25. The Direct Approach to Domain Decomposition Methods

F. García-Nocetti [1], I. Herrera[2], E. Rubio [3], R. Yates[4], L. Ochoa [5]

1. Introduction. Recently, Herrera presented a general and unifying theory of domain decomposition methods (DDM), and this paper is part of a group of articles [5], included in these Proceedings, devoted to present an overview of this theory and some of its applications. According to it, DDM are classified into direct and indirect methods. This paper is devoted to briefly present direct methods from the point of view of the unified theory. A related and more detailed discussion may be found in [6]. It must be mentioned that Direct Methods subsume Schwartz and Steklov-Poincaré Methods among others [6], [7].

2. Notations. The notations will be as in [6].In what follows, unless otherwise explicitly stated, Ω will be an open, bounded region. The closure of any set Ω will be denoted by $\overline{\Omega}$. The (outer) boundary of Ω will be denoted by $\partial\Omega$.

As usual, a collection $\Pi = \{\Omega_1, ..., \Omega_E\}$ of open subregiones $\Omega_i\,(i = 1, ..., E)$ of Ω, is said to be a *partition* of Ω, *iff*

i. $\Omega_i \cap \Omega_j = \phi, for every i \neq j$ and

ii. $\bar{\Omega} = \bigcup_{i=1}^{i=E} \bar{\Omega}_i$

In addition, the partitions considered throughout this paper are assumed to be such that the subregiones Ω_i are *manifolds with corners* [6]. The manifold $\bigcup_{i=1}^{E} \partial\Omega_i$ will be referred to as the *'generalized boundary'*, while the *'internal boundary'* of Ω -to be denoted by Σ- is defined as the closed complement of $\partial\Omega$, considered as a subset of the generalized boundary. Observe that the internal boundary -and the generalized boundary as well- are concepts whose definition is relative to both the region Ω and the partition Π. Thus, when deemed necessary, the notation $\Sigma(\Omega, \Pi)$, which is more precise, will be used.

A partition $\Pi' = \left\{\Omega_1^{'}, ..., \Omega_{E'}^{'}\right\}$ of Ω, is said to be a *sub-partition* of Π, when for each given any $i = 1, .., E'$, there is a subset of natural numbers $\mathrm{N}(i) \subset \{1, .., E\}$, such that

$$\bar{\Omega}_i^{'} = \bigcup_{j \in \mathcal{N}(i)} \bar{\Omega}_j \tag{2.1}$$

Given a sub-partition $\Pi' = \left\{\Omega_1^{'}, ..., \Omega_{E'}^{'}\right\}$, the function $\mu' : \{1, ..., E\} \to \{1, ..., E'\}$ is defined, for every $j = 1, ..., E$, by the equation $\mu'(j) = i$, whenever $j \in \mathrm{N}(i)$. Two partitions: $\Pi' = \left\{\Omega_1^{'}, ..., \Omega_{E'}^{'}\right\}$ and $\Pi'' = \left\{\Omega_1^{''}, ..., \Omega_{E''}^{''}\right\}$, respectively, are said to be *conjugate* with respect to a partition Π, when:

i. They are both sub-partitions of Π ;

ii. In the measure of the generalized boundary, the sets

$$\Sigma^{'} - \left\{\Sigma^{'} \cap \left(\bigcup_{i=1}^{i=E^{'}} \Omega_i^{''}\right)\right\} \quad and \quad \Sigma^{''} - \left\{\Sigma^{''} \cap \left(\bigcup_{i=1}^{i=E^{''}} \Omega_i^{'}\right)\right\} \tag{2.2}$$

[1]Instituto de Investigaciones en Matemáticas Aplicadas y Sistemas (IIMAS), Universidad Nacional Autónoma de México (UNAM), fabian@uxdea4.iimas.unam.mx

[2]Instituto de Geofísica (IGF), Universidad Nacional Autónoma de México (UNAM) , iherrera@servidor.unam.mx

[3]IIMAS-UNAM, ernesto@uxdea4.iimas.unam.mx

[4]IGF-UNAM, yates@altcomp.com.mx

[5]IGF-UNAM, ochoa75@yahoo.com

have measure zero;
iii. And

$$\Sigma' \cup \Sigma'' = \Sigma \tag{2.3}$$

Here, $\Sigma' = \Sigma(\Omega, \Pi')$ and $\Sigma'' = \Sigma(\Omega, \Pi'')$.

When $\Pi' = \left\{\Omega'_1, ..., \Omega'_{E'}\right\}$ and $\Pi'' = \left\{\Omega''_1, ..., \Omega''_{E''}\right\}$ are conjugate partitions, in addition to the mapping μ' introduced above, it will be necessary to consider a second mapping μ'', associated with Π'', which is defined correspondingly.

The formulation and treatment of boundary problems with prescribed jumps requires the introduction of a special class of Sobolev spaces in which some of their functions are fully discontinuous [6]. The *jump* of u across Σ_{ij}, is defined by

$$[v] \equiv v_+ - v_- \tag{2.4}$$

and the *average* by

$$\dot{v} \equiv \frac{1}{2}(v_+ + v_-) \tag{2.5}$$

3. The General Problem with Prescribed Jumps (BVPJ). The direct approach to Domain Decomposition Methods, here presented, as well as Herrera's unified theory, can be applied to a very general class of boundary value problems for which jumps are prescribed in the internal boundaries. Given Ω, the region of definition of the problem, and a partition of Ω (or domain-decomposition) $\Pi \equiv \{\Omega_1, ..., \Omega_E\}$, let $\Sigma \equiv \Sigma(\Omega, \Pi)$ be the internal boundary. Then, using a notation similar to that presented in [8], the general form of such boundary value problem with prescribed jumps (BVPJ) is

$$\mathcal{L}u = \mathcal{L}u_\Omega \equiv f_\Omega; \quad in \quad \Omega_i, i = 1, ..., E \tag{3.1}$$

$$B_j u = B_j u_\partial \equiv g_j; \quad in \quad \partial\Omega \tag{3.2}$$

$$[J_k u] = [J_k u_\Sigma] \equiv j_k; \quad in \quad \Sigma \tag{3.3}$$

where the B_j's and J_j's are certain differential operators (the j's and k's run over suitable finite ranges of natural numbers) and $u_\Omega \equiv \left(u^1_\Omega, ..., u^E_\Omega\right)$, together with u_∂ and u_Σ are given functions of the space of trial functions. In addition, f_Ω, g_j and j_j may be defined by Eq. (3.1).

It must be emphasized that the scope of the methodology presented in this and the other papers of this series is quite wide, since in principle it is applicable to any partial differential equation or system of such equations that is linear, independently of its type and including equations with discontinuous coefficients. But, of course, every kind of equation has its own peculiarities, which require special developments that have to be treated separately.

4. The Elliptic Equation of Second Order. In this Section we describe the overlapping direct method under investigation, for the second-order differential equation of elliptic type, when the problem is defined in a space of arbitrary dimension. For definiteness, only boundary conditions of Dirichlet type will be presented, but the procedure is applicable to any kind of boundary conditions for which the problem is well posed, as was done in [4]. With the notation introduced in Section 2, a region Ω and a partition $\Pi \equiv \{\Omega_1, ..., \Omega_E\}$ of Ω, will be considered. The solution to the boundary value problem with prescribed jumps in this case, will be sought in a Sobolev space of the kind introduced in that Section. More precisely, a function $u \in \hat{H}^2(\Omega) \equiv H^2(\Omega_1) \oplus ... \oplus H^2(\Omega_E)$ is sought, such that

$$\mathcal{L}u \equiv -\nabla \bullet (\underline{\underline{a}} \bullet \nabla u) + \nabla \bullet (\underline{b} u) + cu = f_\Omega; \text{in} \Omega_i, i = 1, ..., E \tag{4.1}$$

subjected to the boundary conditions

$$u = u_\partial; \quad in \quad \partial\Omega \tag{4.2}$$

and jump conditions

$$[u] = j^0 = [u_\Sigma]; \quad on \quad \Sigma \tag{4.3}$$

$$[\underline{\underline{a}} \bullet \nabla u] \bullet \underline{n} = j^1 = [\underline{\underline{a}} \bullet \nabla u_\Sigma] \bullet \underline{n}; \quad on \quad \Sigma \tag{4.4}$$

The above formulation and the methodology that follows applies even if the coefficients of the differential operator are discontinuous. In the particular case when the coefficients are continuous, the jump condition of Eq. (4.4), in the presence of Eq. (4.3), is equivalent to

$$\left[\frac{\partial u}{\partial n}\right] = \left[\frac{\partial u_\Sigma}{\partial n}\right]; \quad on \quad \Sigma \tag{4.5}$$

In what follows, it will be assumed that this problem possesses one and only one solution. Conditions under which this assumption is fulfilled, are well-known.

According to the unified theory one has to choose an information-target, that is referred as *'the sought information'*, as a suitable part of the complementary information defined on Σ. In the procedure that is explained next, the *sought information* is taken to be the average, across Σ, of the solution of the BVPJ. This choice is suitable, because the boundary value problem defined by the system of equations (4.1) to (4.4), when this latter equation is replaced by

$$\overset{\bullet}{\hat{u}} = \overset{\bullet}{u_I} \tag{4.6}$$

is local and well-posed. Here, $u_I \in \hat{D}_1$ is a given function, This can be verified using the relation

$$u_+ = \overset{\bullet}{u} + \frac{1}{2}[u] \,\text{and} u_- = \overset{\bullet}{u} - \frac{1}{2}[u] \tag{4.7}$$

It permits evaluating the values of the function, on both sides of the internal boundary Σ, when the average is known. When this information is complemented with the boundary data on the external boundary, a Dirichlet problem can be formulated in each one of the subdomains of the partition.

In the Theorem that follows, two conjugate partitions $\Pi' = \left\{\Omega_1^{'}, ..., \Omega_{E'}^{'}\right\}$ and $\Pi'' = \left\{\Omega_1^{''}, ..., \Omega_{E''}^{''}\right\}$, as well as the mappings $\mu^{'}$ and $\mu^{''}$ associated to them in the manner explained in Section 2, will be considered. Also, the notations $\Sigma' \equiv \Sigma(\Omega, \Pi')$ and $\Sigma'' \equiv \Sigma(\Omega, \Pi'')$ will be adopted.

Theorem 4.1 .- *Let* $\Pi' = \left\{\Omega_1^{'}, ..., \Omega_{E'}^{'}\right\}$ *and* $\Pi'' = \left\{\Omega_1^{''}, ..., \Omega_{E''}^{''}\right\}$ *be two partitions of* Ω *which are conjugate with respect to* Π*, and let* $\left\{\widehat{u}^1, ..., \widehat{u}^{E'}\right\}$ *and* $\left\{\widehat{u}^1, ..., \widehat{u}^{E'}\right\}$ *be two families of functions, such that*

1) For every $i = 1, ..., E'$*, the function* $\widehat{u}^i \in \hat{H}^2(\Omega_i^{'}, \Pi^{'})$ *fulfills Eqs.(4.1) to (4.3) and satisfies Eq.(4.4) in* $\Sigma^{'}$

2) For every $j = 1, ..., E''$*, the function* $\breve{u}^j \in \hat{H}^2(\Omega_j^{''}, \Pi^{''})$ *fulfills Eqs.(4.1) to (4.3) and satisfies Eq.(4.4) in* $\Sigma^{''}$.

Then, define $u' = \left(u'^1, ..., u'^E\right) \in \hat{H}^2(\Omega, \Pi)$ *and* $u'' = \left(u''^1, ..., u''^E\right) \in \hat{H}^2(\Omega, \Pi)$, *by*

$$u'^i = \widehat{u}^{\mu'(i)}\Big|_{\Omega_i}; i = 1, ..., E \tag{4.8}$$

$$u''^j = \widehat{u}^{\mu''(j)}\Big|_{\Omega j}; i = 1, ..., E \tag{4.9}$$

Under these assumptions the following statements are equivalent:
i. $u^{'}$ *and* $u^{''}$ *are solutions of the BVPJ in* Ω*;*
ii.

$$u' \equiv u'' \tag{4.10}$$

iii.

$$\dot{u}'(\underline{x}) = \dot{u}''(\underline{x}), \quad a.e. \quad on \quad \Sigma = \Sigma^{'} \cup \Sigma^{''} \tag{4.11}$$

Proof..- That *i)* implies *ii)* is immediate, because of the assumption of uniqueness of solution for the BVPJ. That *ii)* implies *iii)* follows from the jump condition of Eq.(4.3) and the definition of the average across Σ. Eq.(4.11) in the presence of Eq.(4.3), in turn imply

$$u'(\underline{x}+) = \dot{u}'(\underline{x}) + \frac{1}{2}\left[u'\right] = \dot{u}'(\underline{x}) + \frac{1}{2}j^0 = \dot{u}''(\underline{x}) + \frac{1}{2}j^0 = \dot{u}''(\underline{x}) + \frac{1}{2}\left[u''\right] = u''(\underline{x}+) \tag{4.12}$$

Recalling that $\Sigma = \Sigma' \cup \Sigma''$ and that $\Sigma \cup \partial\Omega = \bigcup_{i=1}^{E} \partial\Omega_i$, it is seen that the boundary values of $u^{'}$ and $u^{''}$ coincide on each side of Σ. This, together with the assumed uniqueness of solution of the boundary value problem at each one of the sub-regions of the partition, imply $u' \equiv u''$. ■

It is timely to point out the connections between the method discussed in this paper and the Schwarz alternating methods. Indeed, this latter approach can be derived from Eqs.(4.1) to (4.3) and (4.11), when an iterative procedure is adopted for fulfilling Eq.(4.11). To show this, let $u^{2n}(n = 0, 1, ...)$ and $u^{2n+1}(n = 0, 1, ...)$ satisfy Eqs.(4.1) to (4.3), together with

$$\overbrace{u^{2n+1}}^{\bullet} = \overbrace{u^{2n}}^{\bullet}, \quad \text{on} \quad \Sigma', (n = 0, 1, ...) \tag{4.13}$$

$$\overbrace{u^{2n+2}}^{\bullet} = \overbrace{u^{2n+1}}^{\bullet}, \quad \text{on} \quad \Sigma'', (n = 0, 1, ...) \tag{4.14}$$

Then, if the sequence $u^{2n}(n = 0, 1, ...)$ converges to $\widehat{u}$, while the sequence $u^{2n+1}(n = 0, 1, ...)$ converges to $\breve{u}$, one has $\widehat{u} = \breve{u} = u$, and this function fulfills Eqs. (4.1) to (4.3), together with Eq.(4.11). In the cases when a variational principle can be applied, the projection interpretation is possible and the Schwarz alternating procedure can be derived (see, for example, [2], [3], [1]).

5. The One-Dimensional Case. The one dimensional version of the problem described in Section 4 corresponds to the two-point boundary value problem of the general differential equation of second order. Let be $\Omega \equiv (0, l)$ and $\Pi \equiv \{(0, x_1), (x_1, x_2), ..., (x_{E-1}, x_E = l)\}$. Then

$$\mathcal{L}u \equiv -\frac{d}{dx}(a\frac{du}{dx}) + \frac{d}{dx}(bu) + cu = f_\Omega, \quad in \quad (x_{i-1}, x_i), i = 1, ..., E \tag{5.1}$$

Assume that the boundary and jump conditions are:

$$u(0) = g_{\partial 0}, u(l) = g_{\partial l} \tag{5.2}$$

$$[u] = j_i^0 \equiv [u_\Sigma] \quad and \quad \left[\frac{du}{dx}\right] = j_i^1 \equiv \left[\frac{du_\Sigma}{dx}\right]; i = 1, ..., E-1 \tag{5.3}$$

respectively. In addition, it will be assumed that the Dirichlet problem is well-posed in each one of the subintervals and that $u(x) \in H^2(\Omega)$ is the unique solution of this BVPJ, in Ω. As in Section 4, the sought information will be the average of the solution, across Σ.

In every subinterval $(x_{i-1}, x_{i+1}), i = 1, ..., E-1$ define the function $u^i(x)$ to be the restriction of $u(x)$ to Ω_i. Then, for every $i = 1, ..., E-1$, $u^i(x)$, is the unique solution of a boundary value problem with prescribed jumps defined in the subinterval (x_{i-1}, x_{i+1}), which is derived from the following conditions:

$$\mathcal{L}u^i = f_\Omega, \quad \text{in} \quad (x_{i-1}, x_{i+1}); i = 1, ..., E-1 \tag{5.4}$$

$$[u^i]_i = j_i^0; \left[\frac{du^i}{dx}\right]_i = j_i^1; i = 1, ..., E-1 \tag{5.5}$$

$$u^i(x_{i-1}+) = u(x_{i-1}+) = \dot{u}(x_{i-1}) + \frac{1}{2} j_{i-1}^0; i = 2, ...E-1 \tag{5.6}$$

$$u^i(x_{i+1}-) = u(x_{i+1}-) = \dot{u}(x_{i+1}) - \frac{1}{2} j_{i+1}^0; i = 1, ..., E-2 \tag{5.7}$$

$$u^1(0) = u(0) = g_{\partial 0} \tag{5.8}$$

$$u^{E-1}(l) = u(l) = g_{\partial l} \tag{5.9}$$

Let the functions $u_H^i(x)$ and $u_P^i(x)$ be defined in (x_{i-1}, x_{i+1}) by the following conditions:

$$\mathcal{L}u_H^i = 0, \quad in \quad (x_{i-1}, x_{i+1}); i = 1, ..., E \tag{5.10}$$

$$[u_H^i]_i = \left[\frac{du_H^i}{dx}\right]_i = 0; i = 1, ...E-1 \tag{5.11}$$

$$u_H^i(x_{i-1}+) = u(x_{i-1}+) = \dot{u}(x_{i-1}) + \frac{1}{2} j_{i-1}^0; i = 2, ...E-1 \tag{5.12}$$

$$u_H^i(x_{i+1}-) = u(x_{i+1}-) = \dot{u}(x_{i+1}) - \frac{1}{2} j_{i+1}^0; i = 1, ..., E-2 \tag{5.13}$$

$$u_H^1(x_0) = u(0) = g_{\partial 0}; \tag{5.14}$$

$$u_H^{E-1}(x_E) = u(l) = g_{\partial l}; \tag{5.15}$$

together with

$$\mathcal{L}u_P^i = f_\Omega, \quad in \quad (x_{i-1}, x_i) \quad and \quad (x_i, x_{i+1}), \quad separately, \quad for \quad i = 1, ..., E-1 \tag{5.16}$$

$$u_P^i(x_{i-1}+) = u_P^i(x_{i+1}-) = 0, \quad for \quad i = 1, ..., E-1 \tag{5.17}$$

$$[u_P^i]_i = j_i^0 \quad \text{and} \quad \left[\frac{du_P^i}{dx}\right]_i = j_i^1; i = 1, ..., E-1 \tag{5.18}$$

Then, it can be verified that

$$u^i(x) = u_H^i(x) + u_P^i(x); i = 1, ..., E-1 \tag{5.19}$$

Even more:

$$u_H^i(x) = u_H^i(x_{i-1}-)\phi_-^i(x) + u_H^i(x_{i+1}+)\phi_+^i(x) \tag{5.20}$$

when $\phi_-^i(x)$ and $\phi_+^i(x)$ are defined by the conditions:

$$\mathcal{L}\phi_+^i = 0; \phi_+^i(x_{i-1}) = 0, \phi_+^i(x_{i+1}) = 1 \tag{5.21}$$

$$\mathcal{L}\phi_-^i = 0; \phi_-^i(x_{i-1}) = 1, \phi_-^i(x_{i+1}) = 0 \tag{5.22}$$

together with

$$[\phi_+^i]_i = [\phi_-^i]_i = \left[\frac{d\phi_+^i}{dx}\right]_i = \left[\frac{d\phi_-^i}{dx}\right]_i = 0 \tag{5.23}$$

From Eqs. (5.6), (5.7), (5.19), and (5.20), it follows that

$$\dot{u}(x_i) - \dot{u}_P^i(x_i) = \dot{u}_H^i(x_i) = \{\dot{u}(x_{i-1}) + \frac{1}{2}j_{i-1}^0\}\phi_-^i(x_i) + \{\dot{u}(x_{i+1}) - \frac{1}{2}j_{i+1}^0\}\phi_+^i(x_i) \tag{5.24}$$

Hence

$$-\rho_-^i\dot{u}_{i-1} + \dot{u}_i - \rho_+^i\dot{u}_{i+1} = \mu_i; i = 2, ..., E-2 \tag{5.25}$$

$$\dot{u}_i - \rho_+^i\dot{u}_{i+1} = \mu_i; i = 1 \tag{5.26}$$

$$-\rho_-^i\dot{u}_{i-1} + \dot{u}_i = \mu_i; i = E-1 \tag{5.27}$$

where

$$\rho_-^i = \phi_-^i(x_i), \rho_+^i = \phi_+^i(x_i); i = 1, ..., E-1 \tag{5.28}$$

$$\mu_i = \frac{\rho_-^i}{2}j_{i-1}^0 + \dot{u}_P^i(x_i) - \frac{\rho_+^i}{2}j_{i+1}^0; i = 2, ..., E-2 \tag{5.29}$$

$$\mu_i = \rho_-^i g_{\partial 0} + \dot{u}_P^i(x_i) - \frac{\rho_+^i}{2}j_{i+1}^0; i = 1 \tag{5.30}$$

$$\mu_i = \frac{\rho_-^i}{2}j_{i-1}^0 + \dot{u}_P^i(x_i) + \rho_+^i g_{\partial l}; i = E-1 \tag{5.31}$$

Eqs. (5.25) to (5.27), constitute an $E-1$ tridiagonal system of equations, which can be solved for $\dot{u}_i (i = 1, ..., E-1)$.

Once the averages $\dot{u}_i (i = 1, ..., E-1)$ are known, it is possible to apply *'optimal interpolation'* to obtain the solution in the interior of each one of the subintervals of the partition. This kind of interpolation consists in deriving enough information for defining well-posed problems in each of those subintervals. To this end, apply the identities

$$u\left(x_i+\right) \equiv \dot{u}_i + \frac{1}{2}\left[u\right]_i = \dot{u}_i + \frac{1}{2} j_i^0 \quad \text{and} \quad u\left(x_i-\right) \equiv \dot{u}_i - \frac{1}{2}\left[u\right]_i = \dot{u}_i - \frac{1}{2} j_i^0 \tag{5.32}$$

When these values are complemented with the prescribed boundary values of Eq. (5.2), well-posed boundary value problems in each one of the subintervals of the partition can be defined. In this manner, all that is required to reconstruct the solution of the BVPJ is to solve such "local problems", in each one of the subintervals. Using the previous developments, one can apply Eqs. (5.19), and (5.20), to obtain $u(x)$ in the interior of the subintervals of the partition.

Up to now, all the developments have been exact. However, one can apply the system of equations (5.25) to (5.27), as well as Eqs. (5.19), and (5.20), only if the functions ϕ_-^i, ϕ_+^i and u_P^i, $(i = 1, ..., E-1)$, are available. In general applications it will be necessary to resort to numerical approximations for the construction of such functions and the system of equations so obtained will not be exact any longer. Instead, its precision will depend on the error introduced by the numerical procedure that is applied for solving the problems defined by Eqs. (5.10) to (5.18). A similar comment can be made with respect to the construction of the solution of the local boundary value problem whose solution is given by Eqs. (5.19) and (5.20). In reference [6] collocation was used, obtaining in this manner a non-standard method of collocation.

REFERENCES

[1] B. Bialecki and M. Dryja. Multilevel Additive and Multiplicative Methods for Orthogonal Spline Collocation Problems. *Numer. Math.*, 77:35–58, 1997.

[2] M. Dryja and O. B. Widlund. Towards a unified theory of domain decomposition algorithms for elliptic problems. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 3–21. SIAM, Philadelphia, PA, 1990.

[3] M. Dryja and O. B. Widlund. Additive Schwarz methods for elliptic finite element problems in three dimensions. In D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs, and R. G. Voigt, editors, *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 3–18, Philadelphia, PA, 1992. SIAM.

[4] I. Herrera. Unified Approach to Numerical Methods. Part 1. Green's Formulas for Operators in Discontinuous Fields. *Numerical Methods for Partial Differential Equations*, 1(1):12–37, 1985.

[5] I. Herrera. Unified theory of domain decomposition methods. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[6] I. Herrera and R. Yates. General Theory of Domain Decomposition: Beyond Schwarz Methods. *Numerical Methods for Partial Diffential Equations*, 17(5):495–517, 2001.

[7] I. Herrera, R. Yates, and M. Diaz. General Theory of Domain Decomposition: Indirect Methods. *Numerical Methods for Partial Differential Equations*, 18(3):296–322, 2002.

[8] J.-L. Lions and E. Magenes. *Nonhomogeneous Boundary Value Problems and Applications*, volume I. Springer, New York, Heidelberg, Berlin, 1972.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

26. Parallel Implementation of Collocation Methods

R. Yates[1], I. Herrera[2]

1. Introduction. Domain decomposition methods (DDM) have received much attention in recent years[3] in that they offer very effective means for parallelizing computational models of continuous systems. Combining collocation procedures with domain decomposition methods, however, presents complications which must be overcome in order to profit from the advantages of parallel computing. Such methods can in fact be derived using a variety of approaches. One possibility involves the use of Steklov-Poincaré operators [7] while another is to apply an indirect formulation [3],[4],[6],[1]. In this paper, a method is derived based upon the application of collocation together with an indirect formulation which is suitable for parallel computation. As a first approach, we shall consider the case of a symmetric, elliptic differential operator which will allow for the utilization of the conjugate gradient method in a novel manner - where successive iterations involve the (parallel-computed) solutions of local problems.

2. Formulation. We shall use the indirect or Trefftz-Herrera formulation [5] for a boundary-value problem with prescribed jumps (BVPJ) for the case of a symmetric, elliptic operator $\mathcal{L}$ as follows:

Let Ω be a domain in $\mathbb{R}^n$ with external boundary $\partial\Omega$ together with a partition $\Pi = \{\Omega_1, ..., \Omega_E\}$ and internal boundary Σ (Fig. 2.1). Let

$$\mathcal{L}u = -\nabla \cdot \left(\underline{\underline{a}} \cdot \nabla u \right) + cu \tag{2.1}$$

be a symmetric, elliptic operator with $c \geq 0$ and

$$\mathcal{L}u = f \quad on \quad \Omega_i \quad for \quad i = 1, ..., E \tag{2.2}$$

$$u(x) = g(x) \quad on \quad \partial\Omega \tag{2.3}$$

$$[u] = j^0 \quad and \quad [\underline{a}_n \cdot \nabla u] = j^1 \quad on \quad \Sigma \tag{2.4}$$

Then u is said to be a solution of the BVPJ. Here, as in the general theory, the notation $[u] = u_+ - u_-$ and $\dot{u} = \frac{1}{2}(u_+ + u_-)$ is used for the jump of a function and average value across a (possibly discontinuous) internal boundary Σ.

The Green-Herrera formula [2] for this problem is given as

$$P - B - J = Q^* - C^* - K^*; \tag{2.5}$$

where

$$\langle Pu, w \rangle = \int_\Omega w \mathcal{L} u dx, \quad \langle Q^* u, w \rangle = \int_\Omega u \mathcal{L}^* w dx, \tag{2.6}$$

$$\langle Bu, w \rangle = \int_{\partial\Omega} u \underline{a}_n \cdot \nabla w ds, \quad \langle C^* u, w \rangle = \int_{\partial\Omega} w \underline{a}_n \cdot \nabla u ds, \tag{2.7}$$

$$\langle Ju, w \rangle = \langle (J^0 + J^1)u, w \rangle = -\int_\Sigma j^0 \dot{\underline{a}_n \cdot \nabla w} ds + \int_\Sigma j^1 w ds, \tag{2.8}$$

[1]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM) , yates@altcomp.com.mx

[2]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM) , iherrera@servidor.unam.mx

[3]See: International Scientific Committee for Domain Decomposition "Proceedings of 13 conferences on Domain Decomposition Methods", www.ddm.org, 1988-2001

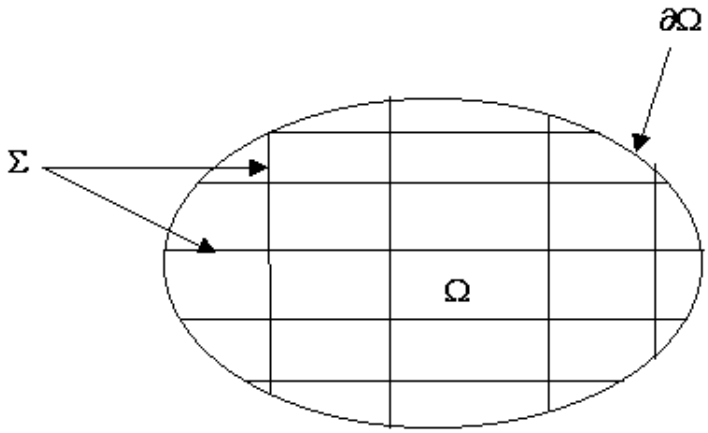


Figure 2.1: Partition of the domain Ω

$$\langle K^* u, w\rangle = \langle (R^* - S^*)u, w\rangle = -\int_\Sigma [w]\overline{\underline{a}_n \cdot \nabla u}\, ds + \int_\Sigma \dot{u}[\underline{a}_n \cdot \nabla w] ds \tag{2.9}$$

In the indirect formulation, the test functions w are chosen so that $w \in \tilde{N} \equiv N_Q \cap N_C \cap N_R$, or equivalently:

$$\mathcal{L}^* w = 0, \quad w = 0 \quad on \quad \partial\Omega \quad and \quad [w] = 0 \quad on \quad \Sigma \tag{2.10}$$

In this case, the resulting formula reduces to

$$\langle S^* u, w\rangle = \langle (P - B - J)u, w\rangle \tag{2.11}$$

which is a variational formulation of the problem. A straightforward calculation shows that

$$\langle S^* u, w\rangle = -\int_\Sigma \dot{u}\,[\underline{a}_n \cdot \nabla w] = \int_\Omega (\nabla u \cdot \underline{\underline{a}} \cdot \nabla w + cuw) \quad \forall u, w \in \tilde{N} \tag{2.12}$$

so that S^* is symmetric, positive-definite when $\mathcal{L}$ is.

In order to obtain the formulation suitable for parallelization, the notion of a *particular solution* must be introduced.

Definition 2.1.- *A function u_p is said to be a <u>particular solution</u> of the BVPJ provided*

$$(P - B - J^0)u_p = f - g - j^0 \tag{2.13}$$

or equivalently, if

$$Pu_p = \mathcal{L}u_p = f \quad in\ each \quad \Omega_i \tag{2.14}$$

$$u_p = g \quad on \quad \partial\Omega \tag{2.15}$$

and

$$[u_p] = j^0 \quad on \quad \Sigma \tag{2.16}$$

A particular solution is therefore a function which satisfies the differential operator locally, the external boundary conditions and the jump conditions of the function values on the internal boundary. Nothing is specified regarding the jump conditions of the normal derivative

(or flux) of a particular solution. As will be shown later, particular solutions can be obtained readily from solutions of the local problems only. Clearly, a particular solution u_p of the BVPJ is a solution of the BVJP if and only if $J^1 u_p = j^1$.

From this last remark, the following result is easily derived:

Theorem 2.1 *A function u is a solution of the BVPJ if and only if, for any particular solution u_p we have*

$$\langle S^* v, w\rangle = \left\langle J^1 u_p - j^1, w\right\rangle; \quad \forall w \in \tilde{N} \quad where \quad v = u - u_p \tag{2.17}$$

3. The Numerical Algorithm. To derive a numerical procedure for the BVPJ in a parallel processing environment, we first obtain a matrix equation from the above variational principle and then develop an iterative solution process using the conjugate gradient method in which each iteration involves the solution of local problems in the subregions Ω_i; these local problems can be effectively solved in parallel.

We first assume that we have a particular solution u_p of the BVPJ and, as above, let $v = u - u_p$ where u is the desired solution. Since the differential operator $\mathcal{L}$ is symmetric and $[v] = v|_{\partial\Omega} = 0$, then we have $v \in \tilde{N}$. From the above result Eq. (2.17), we have:

$$\langle S^* v, w\rangle = \left\langle J^1 u_p - j^1, w\right\rangle; \quad \forall w \in \tilde{N} \tag{3.1}$$

A more explicit form of the above equation is

$$-\int_\Sigma v[\underline{a}_n \cdot \nabla w] ds = \int_\Sigma w([\underline{a}_n \cdot \nabla u_p] - j^1) ds \tag{3.2}$$

To obtain the matrix equation, we will use a system of weighting functions $\{w_1, ..., w_N\}$ of $\tilde{N}$ whose restrictions to the internal boundary Σ form a suitable subspace of $L^2(\Sigma)$. These restrictions, $w_1|_\Sigma, ..., w_N|_\Sigma$ will be used as basis functions to represent v:

$$v = \sum_{j=1}^{N} c_j w_j \tag{3.3}$$

In this case we have:

$$-\sum_{j=1}^{N} c_j \int_\Sigma [\underline{a}_n \cdot \nabla w_i] w_j ds = \int_\Sigma w_i([\underline{a}_n \cdot u_p] - j^1) ds \tag{3.4}$$

which can be rewritten as:

$$\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{c}} = \underline{\mathbf{b}} \tag{3.5}$$

where $A_{ij} = -\int_\Sigma [\underline{a}_n \cdot \nabla w_i] w_j ds$ and $b_i = \int_\Sigma ([\underline{a}_n \cdot \nabla u_p] - j^1) w_i ds$. It should be noted that the matrix $\underline{\underline{\mathbf{A}}}$ is both symmetric and positive definite from Eq. (2.12) as $w_i, w_j \in \tilde{N}$.

The solution of this matrix equation will provide the solution $u = u_p + v$ to the BVPJ on the interior boundary Σ. The solution on Ω can then be obtained by obtaining local solutions in each subregion Ω_i with boundary values supplied by u.

However, a direct computation of the matrix $\underline{\underline{\mathbf{A}}}$ and the obtention of the solution vector $\underline{\mathbf{c}}$ is expensive since the subdomains can be quite large. An alternative approach involves the use of the conjugate gradient method, included in Appendix A, which requires the calculation of the matrix product $\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{c}}$ once for each iteration. In fact, this method does not require the calculation or storage of the components of the matrix $\underline{\underline{\mathbf{A}}}$. Rather, the product is derived from the local solutions of homogeneous BVPs in the subregions Ω_i. To this end, we make use of the following theorem:

Theorem 3.1 *Let* $\underline{\mathbf{p}} = (p_1, ..., p_n)$ *be a vector and let* φ *be the solution to the homogeneous boundary value problem in* Ω_i *defined by:*

$$\mathcal{L}^*\varphi = 0 \quad on \quad \Omega_{\mathrm{i}} \quad for\ each \quad i = 1, ..., E \tag{3.6}$$

$$\varphi = 0 \quad on \quad \partial\Omega \cap \partial\Omega_i \tag{3.7}$$

$$\varphi = \sum_{j=1}^{N} p_j w_j \quad on \quad \Sigma \tag{3.8}$$

then

$$(\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{p}})_i = \int_{\Sigma} w_i [\underline{a}_n \cdot \nabla\varphi] ds \tag{3.9}$$

Proof. Since φ is continuous and vanishes on the exterior boundary $\partial\Omega$, then $\varphi \in \tilde{N}$. Then we must have $\varphi \equiv \sum_{j=1}^{N} p_j w_j$ in each Ω_i so that $\sum_{j=1}^{N} p_j [\underline{a}_n \cdot \nabla w_j] = [\underline{a}_n \cdot \nabla\varphi]$ which is essentially the statement of the theorem. ■

4. The Numerical Procedure. Implementation of the above algorithm requires both the construction of a particular solution u_p and the test function basis $\{w_1, ..., w_N\}$. One way to obtain a particular solution is by setting $\dot{u}_p = 0$ on Σ ie. by solving local problems u_p^i on each Ω_i such that:

$$\mathcal{L}u_p^i = f \quad in\ each \quad \Omega_i \tag{4.1}$$

$$u_p^i = g \quad on \quad \partial\Omega \cap \partial\Omega_{\mathrm{i}} \tag{4.2}$$

$$u_p^i = \pm\tfrac{1}{2} j^0 \quad on \quad \Sigma \cap \partial\Omega_{\mathrm{i}} \tag{4.3}$$

where the sign is chosen according the outward normal.

The resulting u_p will satisfy the differential equation, the external boundary conditions and the internal jump operator J^0. An alternative way of deriving a u_p is to solve first the global BVPJ on the coarse grid and then solve the problem locally on each of the local domains Ω_i using boundary values supplied by the coarse solution. This latter method would tend to give an approximate solution "closer" to the desired solution.

To obtain the test functions, the discretization of the local subdomains gives rise to a discretization of the internal boundary Σ. For each such node point $n_i \in \Sigma$ a test function w_i can be constructed s.t.

$$\mathcal{L}^* w_i = 0 \tag{4.4}$$

$$w_i = 0 \quad on \quad \partial\Omega \cap \partial\Omega_j \quad for\ all \quad j \tag{4.5}$$

and

$$w_i(n_j) = \delta_{ij} \tag{4.6}$$

so that a function $\phi(x)$ on Σ can be approximated as $\phi(x) \simeq \sum_{j=1}^{N} \phi(n_j) w_j(x)$.

Techniques for such constructions using both linear and cubic polynomials on can be found in [5]. It should be stressed that in the computations of the required integrals of the form

$$\int_\Sigma w_i[\underline{a}_n \cdot \nabla u]ds = \int_\Sigma w_i \bar{a}\frac{\partial u}{\partial n}ds \quad where \quad \bar{a} = \underline{n} \cdot \underline{\underline{a}} \cdot \underline{n} \tag{4.7}$$

that care must be taken to evaluate $\frac{\partial u}{\partial n}$ with consistent precision.

5. Conclusions. The method presented above defines a procedure for parallelizing the numeric solution for second order symmetric elliptic equations. As the solution matrix obtained is symmetric and positive definite, direct application of the conjugate gradient metod is utilized to insure adequate convergence; no preconditioning techniques are required. Moveover, the method is applicable to problems with prescribed jumps (BVPJ) as well as to the case of discontinuous coefficients with no additional complications to the numerical procedure. Finally, it should be stressed that in the solution of the local problems, any numerical procedure can be successfully employed.

Appendix A: The Conjugate Gradient Algorithm

To solve $\underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{v}} = \underline{\mathbf{b}}$ where $\underline{\underline{\mathbf{A}}}$ is an $N \times N$ symmetric, positive definite matrix.

$\underline{\mathbf{v}}^0 = 0$

$\underline{\mathbf{r}}^0 = \underline{\mathbf{b}}$

$\underline{\mathbf{p}}^0 = \underline{\mathbf{r}}^0$

$k = 0$

while $\left\|\underline{\mathbf{r}}^k\right\|_\infty \geq \varepsilon$

$\underline{\mathbf{u}} = \underline{\underline{\mathbf{A}}} \cdot \underline{\mathbf{p}}^k$[Singlematrixmultiplication/iteration]

$\alpha^{k+1} = \underline{\mathbf{r}}^k \cdot \underline{\mathbf{r}}^k / (\underline{\mathbf{p}}^k \cdot \underline{\mathbf{u}})$

$\underline{\mathbf{v}}^{k+1} = \underline{\mathbf{v}}^k + \alpha^{k+1}\underline{\mathbf{p}}^k$

$\underline{\mathbf{r}}^{k+1} = \underline{\mathbf{r}}^k - \alpha^{k+1}\underline{\mathbf{u}}$

$\beta^{k+1} = \underline{\mathbf{r}}^{k+1} \cdot \underline{\mathbf{u}} / (\underline{\mathbf{p}}^k \cdot \underline{\mathbf{u}})$

$\underline{\mathbf{p}}^{k+1} = \underline{\mathbf{r}}^{k+1} - \beta^{k+1}\underline{\mathbf{p}}^k$

$k = k + 1$

REFERENCES

[1] M. Diaz, I. Herrera, and R. Yates. Indirect Method of Collocation: Second Order Equations. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[2] I. Herrera. Unified Approach to Numerical Methods. Part 1. Green's Formulas for Operators in Discontinuous Fields. *Numerical Methods for Partial Differential Equations*, 1(1):12–37, 1985.

[3] I. Herrera. Trefftz-Herrera Domain Decomposition. *Advances in Engineering Software*, 24:43–56, 1995.

[4] I. Herrera. Trefftz Method: A General Theory. *Numerical Methods for Partial Differential Equations*, 16(6):561–580, 2000.

[5] I. Herrera, R. Yates, and M. Diaz. General Theory of Domain Decomposition: Indirect Methods. *Numerical Methods for Partial Differential Equations*, 18(3):296–322, 2002.

[6] I. Herrera, R. Yates, and M. Diaz. The Indirect Approach to Domain Decomposition. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

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

Part V

Mini Symposium: Optimized Schwarz Methods

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

27. A Non-Overlapping Optimized Schwarz Method which Converges with Arbitrarily Weak Dependence on h

M.J. Gander[1], G.H. Golub[2]

1. Introduction. Optimized Schwarz methods have been introduced in [11] to correct the uneven convergence properties of the classical Schwarz method. In the classical Schwarz method high frequency components converge very fast, whereas low frequency components are only converging very slowly and hence slow down the performance of the overall method. This can be corrected by replacing the Dirichlet transmission conditions in the classical Schwarz method by Robin or higher order transmission conditions which approximate the classical absorbing boundary conditions used to truncate infinite domains for numerical computations on bounded domains. The new methods are called optimized Schwarz methods because the new transmission conditions are obtained by optimizing their coefficients for the performance of the method.

Using transmission conditions different from the Dirichlet ones is however not new. P.-L. Lions proposed in [14] to use Robin transmission conditions to obtain a converging non-overlapping variant of the Schwarz method, a result not possible with Dirichlet transmission conditions. But it was in the context of a particular problem, namely the Helmholtz equation, where the importance of radiation conditions was first realized in the PhD thesis of Deprés [5]. Several publications for the Helmholtz equation followed; in the context of control [1], for an overlapping variant in [2], and a first approach to optimize the transmission conditions without overlap in [4]. An interesting variant of a Schwarz method using perfectly matched layers can be found in [17]. Fully optimized transmission conditions were published in [9, 12] for the non-overlapping variant of the Schwarz method and a first approach for the overlapping case can be found in [11]. Very soon it was realized that approximations to absorbing boundary conditions were very effective for other types of equations as well. For the convection-diffusion equation, the first paper proposing optimized transmission conditions for a non-overlapping variant of the Schwarz method is [3]. Around the same time, a discrete version of such a Schwarz method was developed at the algebraic level in [16, 15], but it proved to be difficult to optimize the free parameters. Second order optimized transmission conditions for convection-diffusion were explored in [13] for the non-overlapping case and for symmetric positive definite problems in [7] with the first asymptotic results of the performance of those methods. Such transmission conditions are also crucial in the case of evolution problems, as shown in [10], and for systems of equations, for the Euler equations, see [6]. A complete survey for symmetric positive definite problems with all the asymptotic performances for overlapping and non-overlapping variants, is in preparation [8].

We show in this paper that the transmission conditions in the optimized Schwarz methods can be chosen such that the convergence rate of the method has an arbitrarily weak asymptotic dependence on the mesh parameter h, even if no overlap is used. This result is obtained by choosing a sequence of transmission conditions which is applied cyclicly in the optimized Schwarz iteration. Closed form expressions for the transmission conditions are derived which give an asymptotic convergence rate $\rho = 1 - O(h^{1/m})$ for m an arbitrary power of 2.

2. The Model Problem. We consider for this paper the self adjoint coercive model problem

$$\mathcal{L}(u) := (\eta - \Delta)u = f, \quad \text{in } \Omega = \mathbb{R}^2 \tag{2.1}$$

and we assume that the solution $u(x, y)$ stays bounded at infinity. We can pose an equivalent problem on $\mathbb{R}^2$ decomposed into two overlapping subdomains $\Omega_1 = (-\infty, L) \times \mathbb{R}$, $\Omega_2 =$

[1]McGill University, mgander@math.mcgill.ca, supported in part by NSERC grant 228061.
[2]Stanford University, golub@sccm.stanford.edu, supported in part by DOE DE-FC02-01ER41777.

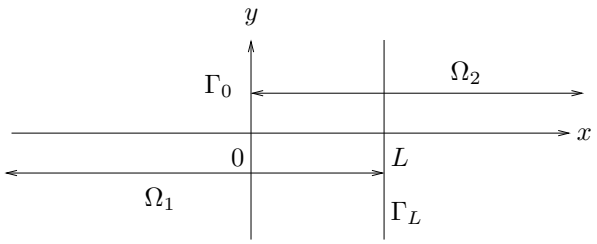


Figure 2.1: Decomposition for the model problem.

$(0,\infty)\times\mathbb{R}$, $L>0$, with boundaries Γ_L at $x=L$ and Γ_0 at $x=0$ as shown in Figure 2.1, namely

$$\begin{array}{rclll} (\eta-\Delta)v & = & f & \text{in } \Omega_1, & \qquad (\eta-\Delta)w = f \quad \text{in } \Omega_2, \\ v & = & w & \text{on } \Gamma_L, & \qquad w = v \quad \text{on } \Gamma_0. \end{array} \tag{2.2}$$

Then the restriction of the solution u of the original problem to Ω_1 coincides with the solution v of the partitioned problem and the restriction of the solution u to Ω_2 coincides with w of the partitioned problem. If the overlap however becomes zero, $L=0$, then the subdomain problems do not necessarily coincide with the solution u of the original problem any more, one has to introduce the additional condition that the derivatives need to match,

$$\partial_x w = \partial_x v \quad \text{on } \Gamma_0 = \Gamma_L.$$

To make the coupling more robust with respect to small overlap, we introduce the subdomain coupling

$$(\partial_x + S_v)v = (\partial_x + S_v)w \text{ on } \Gamma_L, \quad (\partial_x + S_w)w = (\partial_x + S_w)v \text{ on } \Gamma_0, \tag{2.3}$$

where S_v and S_w are for the moment undetermined linear operators acting in the y direction. Note that for example choosing $S_v = -S_w = p$ for some constant $p>0$ leads to subdomain solutions v and w which coincide with the solution u of the original problem even if the overlap is zero, $L=0$, since then the conditions $(\partial_x+p)v = (\partial_x+p)w$ and $(\partial_x-p)w = (\partial_x-p)v$ on Γ_0 imply both continuity of the subdomain solution and its derivative at $x=0$. The subdomain problems are then coupled by a Robin transmission condition, an idea introduced in [14]. The goal of optimized Schwarz methods is to determine good choices for the operators S_v and S_w to obtain fast domain decomposition methods at a computational cost comparable to the classical Schwarz method.

3. An Optimized Schwarz Method. We introduce a Schwarz relaxation to the system coupled with the new conditions,

$$\begin{array}{rcll} (\eta-\Delta)v^n & = & f, & \text{in } \Omega_1, \\ (\eta-\Delta)w^n & = & f, & \text{in } \Omega_2, \\ (\partial_x + S_v)v^n & = & (\partial_x + S_v)w^{n-1} & \text{on } \Gamma_L, \\ (\partial_x + S_w)w^n & = & (\partial_x + S_w)v^{n-1} & \text{on } \Gamma_0. \end{array} \tag{3.1}$$

This iteration can be analyzed using Fourier analysis, see for example [11]. The convergence rate of this algorithm is

$$\rho(k) = \frac{\sqrt{\eta+k^2}-\sigma_v(k)}{\sqrt{\eta+k^2}+\sigma_v(k)} \cdot \frac{\sqrt{\eta+k^2}+\sigma_w(k)}{\sqrt{\eta+k^2}-\sigma_w(k)} e^{-2\sqrt{\eta+k^2}L} \tag{3.2}$$

where k is the Fourier variable in the y direction and σ_v and σ_w denote the symbols of S_v and S_w. The optimal transmission operators S_v and S_w have thus the symbols $\sigma_v = \sqrt{\eta + k^2}$ and $\sigma_w = -\sqrt{\eta + k^2}$ because then the convergence rate vanishes and hence the algorithm converges in 2 steps, independent of the size of the overlap L. Unfortunately these operators are non-local, they require the evaluation of a convolution and hence polynomial approximations have been introduced for various types of partial differential equations, see [3, 13, 4, 7, 10, 11, 9, 12]. The simplest approximation is to use a constant, which leads to the Robin transmission conditions $S_v = -S_w = p$ for some constant $p > 0$. The sign of p is needed for well-posedness, but it also guarantees convergence of the algorithm, since then the convergence rate becomes

$$\rho(k,p) = \left(\frac{\sqrt{\eta + k^2} - p}{\sqrt{\eta + k^2} + p} e^{-\sqrt{\eta+k^2}L} \right)^2$$

which is less than one for all $k < \infty$, even if the overlap is zero, i.e. $L = 0$. To find the best Robin parameter, one minimizes the convergence rate over all the frequencies relevant to a given discretization, $k_{\min} < |k| < k_{\max}$, which leads to the min-max problem

$$\min_{p \geq 0} \left(\max_{k_{\min} < k < k_{\max}} \rho(k,p) \right).$$

The solution of this problem, with or without overlap, can be found in [8] and the convergence rate depends mildly on the mesh parameter h; for $L = 0$ one finds $\rho = 1 - O(\sqrt{h})$ and for $L = O(h)$ the result is $\rho = 1 - O(h^{1/3})$. In the following section we will make the convergence rate as weakly dependent on h as desired for the case $L = 0$.

4. Arbitrarily Weak Dependence on h. The idea is to use different parameters p_j for different steps of the iteration. Suppose we want to use m different values p_j, $j = 1, \ldots, m$ in the Robin transmission condition. We then cycle through these different parameters in the optimized Schwarz algorithm,

$$\begin{array}{rcll} (\eta - \Delta) v^n & = & f, & \text{in } \Omega_1, \\ (\eta - \Delta) w^n & = & f, & \text{in } \Omega_2, \\ (\partial_x + p_{n \bmod m+1}) v^n & = & (\partial_x + p_{n \bmod m+1}) w^{n-1} & \text{on } \Gamma_L, \\ (\partial_x - p_{n \bmod m+1}) w^n & = & (\partial_x - p_{n \bmod m+1}) v^{n-1} & \text{on } \Gamma_0. \end{array} \tag{4.1}$$

Performing again a Fourier analysis in y with the parameter k of this algorithm, we obtain the convergence rate depending on $\mathbf{p} = (p_1, p_2, \ldots, p_m)$

$$\rho(m, \mathbf{p}, \eta, k) = e^{-2\sqrt{\eta+k^2}L} \left(\prod_{j=1}^{m} \left(\frac{\sqrt{\eta + k^2} - p_j}{\sqrt{\eta + k^2} + p_j} \right)^2 \right)^{\frac{1}{m}}.$$

To optimize the performance of this new algorithm, the parameters p_j, $j = 1, \ldots, m$ in the vector $\mathbf{p}$ have to be the solution of the min-max problem

$$\min_{\mathbf{p} \geq 0} \left(\max_{k_{\min} < k < k_{\max}} \rho(m, \mathbf{p}, \eta, k) \right).$$

This optimization problem has to be solved numerically in general, but for $L = 0$ and $m = 2^l$ it has an elegant solution in closed form for the ADI method in [19].

Theorem 4.1 (Wachspress (1962)) *If $m = 2^l$ then the optimal choice for the parameters p_j, $j = 1, 2, \ldots, m$ is given by*

$$p_j = \alpha_{0,j}, \quad j = 1, 2, \ldots, m \tag{4.2}$$

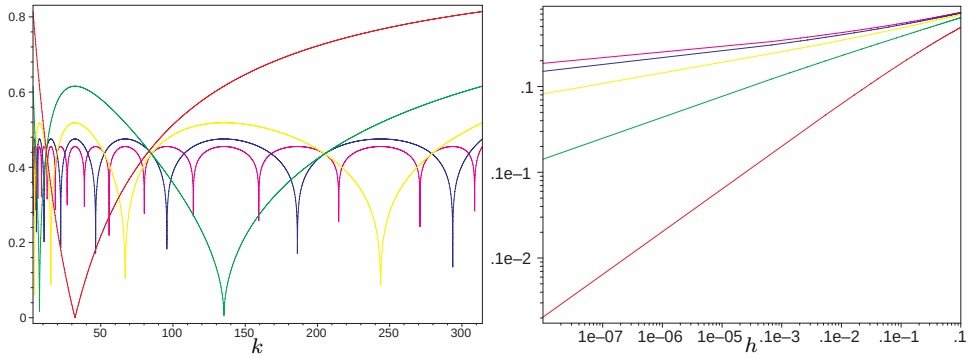


Figure 4.1: On the left dependence of $|\rho_{opt}|$ on the frequency k when 1, 2, 4, 8 and 16 optimization parameters are used on a fixed range of frequencies, $k_{\max} = 100\pi$, and on the right dependence of $1 - \rho_{\max}$ on h as h goes to zero for 1, 2, 4, 8 and 16 optimization parameters.

where the $\alpha_{0,j}$ are recursively defined using the forward recursion

$$\begin{array}{rclcrcll} x_0 & = & \sqrt{\eta + k_{\min}}\,, & & x_{i+1} & = & \sqrt{x_i y_i} & \\ y_0 & = & \sqrt{\eta + k_{\max}}\,, & & y_{i+1} & = & \frac{x_i + y_i}{2} & i = 0, 1, \ldots, l \end{array} \tag{4.3}$$

and the backward recursion

$$\begin{array}{rcl} \alpha_{l,1} = \sqrt{x_l y_l}, & \alpha_{i,2j-1} & = \alpha_{i+1,j} - \sqrt{\alpha_{i+1,j}^2 - x_i y_i} \\ & \alpha_{i,2j} & = \alpha_{i+1,j} + \sqrt{\alpha_{i+1,j}^2 - x_i y_i} \end{array} \tag{4.4}$$

where $i = l-1, l-2, \ldots, 0$ and $j = 1, 2, \ldots 2^{l-i-1}$ for each i. The convergence rate obtained with these parameters is given by

$$\max_{k_{\min} \le k \le k_{\max}} |\rho(k, m)| = \left(\frac{\sqrt{y_l} - \sqrt{x_l}}{\sqrt{y_l} + \sqrt{x_l}} \right)^{\frac{1}{m}}. \tag{4.5}$$

Proof. The proof uses the equioscillation property of the optimum similar to the case of the Chebyshev polynomials and is due to Wachspress in [19]. An elegant version of the proof can be found in Varga [18]. ∎

In Figure 4.1 we show how the optimal choice of an increasing number of parameters p_j affects the convergence rate of the optimized Schwarz method. From Figure 4.1 on the right we see that the more optimization parameters we use, the weaker the dependence on h of the convergence rate becomes. This indicates that we can define a sequence of non-overlapping optimized Schwarz methods with an arbitrarily weak dependence of the convergence rate on the mesh parameter h using m different constants in the Robin transmission conditions. To prove this result, we first need the following

Lemma 4.1 *For $k_{\max} = \pi/h$ the recursively defined x_i and y_i in equation (4.3) have for h small the asymptotic expansion*

$$\begin{array}{rcl} x_i & = & 2^{2-i-\frac{1}{2^{i-1}}} (\eta + k_{\min}^2)^{\frac{1}{2^{i+1}}} \left(\frac{\pi}{h}\right)^{1-\frac{1}{2^i}} + O((\frac{1}{h})^{1-\frac{5}{2^i}}) \\ y_i & = & \frac{1}{2^i}\frac{\pi}{h} + O((\frac{1}{h})^{1-\frac{1}{2^{i-1}}}) \end{array} \qquad i = 0, 1, \ldots \tag{4.6}$$

Proof. The proof is done by induction. For $i = 0$, we have

$$x_0 = \sqrt{\eta + k_{\min}^2}, \quad y_0 = \sqrt{\eta + \left(\frac{\pi}{h}\right)^2} = \frac{\pi}{h} + O(h).$$

Now we assume that (4.6) holds for i and compute for $i+1$, using the recursive definition (4.3) first for x_{i+1}

$$\begin{aligned}
x_{i+1} &= \sqrt{x_i y_i} \\
&= \sqrt{\left(2^{2-i-\frac{1}{2^{i-1}}}(\eta+k_{\min}^2)^{\frac{1}{2^{i+1}}}\left(\tfrac{\pi}{h}\right)^{1-\frac{1}{2^i}}+O((\tfrac{1}{h})^{1-\frac{5}{2^i}})\right)\left(\tfrac{1}{2^i}\tfrac{\pi}{h}+O((\tfrac{1}{h})^{1-\frac{1}{2^{i-1}}})\right)} \\
&= \sqrt{2^{2-2i-\frac{1}{2^{i-1}}}(\eta+k_{\min}^2)^{\frac{1}{2^{i+1}}}\left(\tfrac{\pi}{h}\right)^{2-\frac{1}{2^i}}+O((\tfrac{1}{h})^{2-\frac{5}{2^i}})} \\
&= 2^{2-(i+1)-\frac{1}{2^i}}(\eta+k_{\min}^2)^{\frac{1}{2^{i+2}}}\left(\tfrac{\pi}{h}\right)^{1-\frac{1}{2^{i+1}}}\sqrt{1+O(h^{\frac{1}{2^{i-2}}})} \\
&= 2^{2-(i+1)-\frac{1}{2^i}}(\eta+k_{\min}^2)^{\frac{1}{2^{i+2}}}\left(\tfrac{\pi}{h}\right)^{1-\frac{1}{2^{i+1}}}+O((\tfrac{1}{h})^{1-\frac{5}{2^{i+1}}})
\end{aligned}$$

and then for y_{i+1}

$$\begin{aligned}
y_{i+1} &= \frac{x_i+y_i}{2} \\
&= \frac{2^{2-i-\frac{1}{2^{i-1}}}(\eta+k_{\min}^2)^{\frac{1}{2^{i+1}}}\left(\frac{\pi}{h}\right)^{1-\frac{1}{2^i}}+O((\frac{1}{h})^{1-\frac{5}{2^i}})+\frac{1}{2^i}\frac{\pi}{h}+O((\frac{1}{h})^{1-\frac{1}{2^{i-1}}})}{2} \\
&= \tfrac{1}{2^{i+1}}\tfrac{\pi}{h}+O((\tfrac{1}{h})^{1-\frac{1}{2^i}})
\end{aligned}$$

which completes the induction. ∎

The asymptotic convergence rates for small mesh parameter h are given in the following

Theorem 4.2 *The non-overlapping optimized Schwarz method (4.1) with $m = 2^l$ optimally chosen parameters p_j, $j = 1, 2, \ldots, m$ in the Robin transmission conditions according to (4.2) has for small mesh parameter h the asymptotic convergence rate*

$$\rho_{opt} = 1 - 2\frac{2^{1-\frac{1}{m}}(\eta+k_{\min}^2)^{\frac{1}{4m}}}{m\pi^{\frac{1}{2m}}}h^{\frac{1}{2m}} + O(h^{\frac{1}{m}}). \tag{4.7}$$

Proof. We first need the asymptotic expansions of the square roots of x_l and y_l given in Lemma 4.1,

$$\begin{aligned}
\sqrt{x_l} &= 2^{1-\frac{l}{2}-\frac{1}{2^l}}(\eta+k_{\min}^2)^{\frac{1}{2^{l+2}}}\left(\tfrac{\pi}{h}\right)^{\frac{1}{2}-\frac{1}{2^{l+1}}}+O((\tfrac{1}{h})^{\frac{1}{2}-\frac{9}{2^{l+1}}}), \\
\sqrt{y_l} &= \left(\tfrac{1}{2}\right)^{\frac{l}{2}}\sqrt{\tfrac{\pi}{h}}+O((\tfrac{1}{h})^{\frac{1}{2}-\frac{1}{2^{l-1}}}).
\end{aligned}$$

Inserting these expansions into the expression for the optimized convergence rate (4.5) of Theorem 4.1 we obtain

$$\begin{aligned}
\rho_{opt} &= \left(\tfrac{\sqrt{y_l}-\sqrt{x_l}}{\sqrt{y_l}+\sqrt{x_l}}\right)^{\frac{1}{m}} = \left(\frac{1-2^{1-\frac{1}{2^l}}(\eta+k_{\min}^2)^{\frac{1}{2^{l+2}}}\left(\frac{h}{\pi}\right)^{\frac{1}{2^{l+1}}}+O(h^{\frac{1}{2^{l-1}}})}{1+2^{1-\frac{1}{2^l}}(\eta+k_{\min}^2)^{\frac{1}{2^{l+2}}}\left(\frac{h}{\pi}\right)^{\frac{1}{2^{l+1}}}+O(h^{\frac{1}{2^{l-1}}})}\right)^{\frac{1}{m}} \\
&= \left(1-2\frac{2^{1-\frac{1}{2^l}}(\eta+k_{\min}^2)^{\frac{1}{2^{l+2}}}}{\pi^{\frac{1}{2^{l+1}}}}h^{\frac{1}{2^{l+1}}}+O(h^{\frac{1}{2^l}})\right)^{\frac{1}{m}} = 1-2\frac{2^{1-\frac{1}{2^l}}(\eta+k_{\min}^2)^{\frac{1}{2^{l+2}}}}{m\pi^{\frac{1}{2^{l+1}}}}h^{\frac{1}{2^{l+1}}}+O(h^{\frac{1}{2^l}})
\end{aligned}$$

and the result follows by noting that $m = 2^l$. ∎

Hence increasing m we can achieve an as weak dependence of the convergence rate on the mesh parameter h as we like. The numerical experiments in the next section show that this result also holds for the discretized algorithm.

5. Numerical Experiments. We perform all our computations on a bounded domain for the model problem

$$\mathcal{L}(u) := (\eta - \Delta)u = f, \quad \text{in } \Omega = [0,1]^2$$

with homogeneous Dirichlet boundary conditions. We decompose the domain into two non-overlapping subdomains $\Omega_1 = [0, \frac{1}{2}] \times [0,1]$ and $\Omega_2 = [\frac{1}{2}, 1] \times [0,1]$ and apply the optimized non-overlapping Schwarz method for various values of the parameter m. We simulate directly the error equations, i.e. $f = 0$, and we show the results for $\eta = 1$. To solve the subdomain problems, we use the standard five point finite difference discretization with uniform mesh spacing h in both the x and y directions. We start the iteration with a random initial guess so that it contains all the frequencies on the given mesh and we iterate until the relative residual is smaller than $1e-6$. Table 5.1 shows the number of iterations required as one refines the mesh parameter h. There are two important things to notice: first one can

	Schwarz as a solver			Schwarz as a preconditioner		
h	$m=1$	$m=2$	$m=4$	$m=1$	$m=2$	$m=4$
1/50	24	6	3	10	5	3
1/100	34	8	3	12	6	3
1/200	48	10	4	14	6	3
1/400	68	12	4	17	7	4
1/800	95	14	4	20	8	4

Table 5.1: Dependence on h and m of the number of iterations when the optimized Schwarz method is used as a solver or as a preconditioner for a Krylov method.

see that the dependence of the number of iterations gets weaker as m becomes larger, as predicted by the analysis. Second for small m, using Krylov acceleration leads to significant improvement in the performance, whereas for bigger m, the improvement is almost negligible, Schwarz by itself is already such a good solver that Krylov acceleration is not needed. This is a property also observed for multi-grid methods applied to this problem. To see the dependence of the convergence rate on h more clearly, we plotted in Figure 5.1 the number of iterations together with the asymptotic rates expected from our analysis. One can see that the asymptotic analysis predicts very well the numerically observed results. One even gains almost the additional square-root from the Krylov method when Schwarz is used as a preconditioner.

Finally we emphasize that the number of iterations given in Table 5.1 is the number of times we cycled through all parameter values. In the current implementation therefore the cost of one iteration with $m = 4$ is four times the cost of one iteration with $m = 1$. But note that not each iteration of the Schwarz method needs the same resolution now, since it only needs to be effective in the frequency range around the corresponding p_j. The values of p_j for $m = 4$ with $h = 1/400$ are for example $p_1 = 4.78$, $p_2 = 25.85$, $p_3 = 160.26$ and $p_4 = 866.71$. Hence the solve with p_1 in the transmission condition can be done on a very coarse grid, the one with p_2 on quite a coarse grid, the one with p_3 on an intermediate grid and only the solve with p_4 needs to be on a fine grid. In addition we do not need to solve exactly; it is only required to reduce the error in the corresponding frequency range, using a relaxation iteration, which leads to an algorithm with natural inner and outer iterations. Doing this, the cost for arbitrary m will be only a constant times the cost for $m = 1$ and hence the number of iterations we gave become the relevant ones to compare. Furthermore in that case, the factor $1/m$ in the asymptotic convergence rate (4.7) disappears because now the relevant quantities to compare are ρ_{opt}^m and hence one can obtain a convergence rate

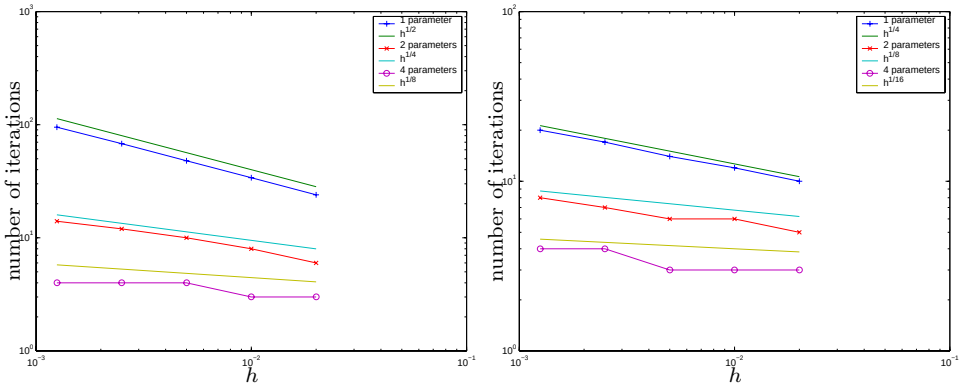


Figure 5.1: Asymptotic behavior of the optimized Schwarz method, on the left used as an iterative solver and on the right as a preconditioner.

independent of h by choosing the number m like the logarithm of $1/h$ as h is refined. Such an algorithm then has the key properties of multigrid, but is naturally parallel like the Schwarz algorithm.

REFERENCES

[1] J. D. Benamou. A domain decomposition method for the optimal control of system governed by the Helmholtz equation. In G. Cohen, editor, *Third international conference on mathematical and numerical wave propagation phenomena*. SIAM, 1995.

[2] 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)*, pages 391–399. Amer. Math. Soc., Providence, RI, 1998.

[3] P. Charton, F. Nataf, and F. Rogier. Méthode de décomposition de domaine pour l'equation d'advection-diffusion. *C. R. Acad. Sci.*, 313(9):623–626, 1991.

[4] P. Chevalier and F. Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.

[5] B. Deprés. *Méthodes de décomposition de domains pour les problèms de propagation d'ondes en régime harmonique*. PhD thesis, Université Paris IX Dauphine, 1991.

[6] V. Dolean and S. Lanteri. A domain decomposition approach to finite volume solutions of the Euler equations on triangular meshes. Technical Report 3751, INRIA, oct 1999.

[7] B. Engquist and H.-K. Zhao. Absorbing boundary conditions for domain decomposition. *Appl. Numer. Math.*, 27(4):341–365, 1998.

[8] M. J. Gander. Optimized Schwarz methods for symmetric positive definite problems. in preparation, 2000.

[9] M. J. Gander. Optimized Schwarz methods for Helmholtz problems. In *Thirteenth international conference on domain decomposition*, pages 245–252, 2001.

[10] M. J. Gander, L. Halpern, and F. Nataf. Optimal convergence for overlapping and non-overlapping Schwarz waveform relaxation. In C.-H. Lai, P. Bjørstad, M. Cross, and O. Widlund, editors, *Eleventh international Conference of Domain Decomposition Methods*. ddm.org, 1999.

[11] M. J. Gander, L. Halpern, and F. Nataf. Optimized Schwarz methods. In T. Chan, T. Kako, H. Kawarada, and O. Pironneau, editors, *Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan*, pages 15–28, Bergen, 2001. Domain Decomposition Press.

[12] M. J. Gander, F. Magoulès, and F. Nataf. Optimized Schwarz methods without overlap for the Helmholtz equation. *SIAM J. Sci. Comput.*, 2001. to appear.

[13] 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. Technical Report 373, CMAP (Ecole Polytechnique), 1997.

[14] P.-L. Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.

[15] H. San and W. P. Tang. An overdetermined Schwarz alternating method. *SIAM J. Sci. Comput.*, 7:884–905, 1996.

[16] W. P. Tang. Generalized Schwarz splittings. *SIAM J. Sci. Stat. Comput.*, 13:573–595, 1992.

[17] A. Toselli. Some results on overlapping Schwarz methods for the Helmholtz equation employing perfectly matched layers. Technical Report 765, Courant Institute, New York, June 1998.

[18] R. Varga. *Matrix Iterative Analysis*. Prentice Hall, first edition, 1962.

[19] E. L. Wachspress. Optimum alternating-direction-implicit iteration parameters for a model problem. *J. Soc. Indust. Appl. Math.*, 10:339–350, 1962.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

28. An optimized Schwarz method in the Jacobi-Davidson method for eigenvalue problems

M. Genseberger[1], G. L. G. Sleijpen[2], and H. A. van der Vorst[3]

1. Introduction. The Jacobi-Davidson method [3] is an iterative method suitable for the computation of solutions to large scale (generalized) eigenvalue problems. Most of the computational work of the Jacobi-Davidson method arises from performing (approximate) solves for the so-called correction equation. In order to relieve this amount of work and/or the local memory requirements we propose a strategy based on domain decomposition.

The domain decomposition method is based on previous work for ordinary systems of (definite) linear equations (§3). It requires specific knowledge of the underlying PDE's. For eigenvalue problems the situation is more complex as the correction equation is (highly) indefinite. In this paper we describe and analyze the situation for the correction equation (§4). Results of the analysis are of practical interest for more general cases like PDE's with variable coefficients, many subdomains in two directions and complex geometries (§5). The proposed domain decomposition approach enables a massively parallel treatment of large scale eigenvalue problems ([1, §4]).

2. The Jacobi-Davidson method. The Jacobi-Davidson method [3] projects the original eigenvalue problem on a suitable search subspace. From the projected eigenvalue problem approximate solutions to the original problem are computed. The search subspace is expanded iteratively with the most important direction in the residual not already present. Compared to other methods the Jacobi-Davidson method offers many advantages and flexibility such as the exploitation of a good preconditioner.

For a standard eigenvalue problem $\mathbf{A}\,\mathbf{x} = \lambda\,\mathbf{x}$ each iteration step Jacobi-Davidson

- **extracts** an approximate solution $(\theta, \mathbf{u}) \approx (\lambda, \mathbf{x})$ from a search subspace

  construct $H \equiv \mathbf{V}^*\mathbf{A}\,\mathbf{V}$,
  solve $H\,s = \theta\,s$, and compute $\mathbf{u} = \mathbf{V}\,s$
  where the columns of $\mathbf{V}$ form an orthonormal basis for the search subspace

- **corrects** the approximate eigenvector $\mathbf{u}$

  compute a correction $\mathbf{t}$ from the *correction equation*:
  $\mathbf{t} \perp \mathbf{u}, \quad \mathbf{P}\,\mathbf{B}\,\mathbf{P}\,\mathbf{t} = \mathbf{r}$
  where $\mathbf{P} \equiv \mathbf{I} - \frac{\mathbf{u}\,\mathbf{u}^*}{\mathbf{u}^*\mathbf{u}}$, $\mathbf{B} \equiv \mathbf{A} - \theta\,\mathbf{I}$, and $\mathbf{r} \equiv -\mathbf{B}\,\mathbf{u}$

- **expands** the search subspace with the correction $\mathbf{t}$

  $\mathbf{V}_{new} = \left[\mathbf{V} \mid \mathbf{t}^{\perp}\right]$
  where $\mathbf{t}^{\perp} = \alpha\,(\mathbf{I} - \mathbf{V}\,\mathbf{V}^*)\,\mathbf{t}$ such that $\|\mathbf{t}^{\perp}\|_2 = 1$

3. An optimized Schwarz method. For the domain decomposition technique we adapt a locally optimized additive Schwarz method based on work by Tan & Borsboom [5, 4] for linear systems which in its turn is a generalization of work by Tang [6]. We show the main ingredients and discuss some details for ordinary linear systems. The situation for the correction equation is described and analysed in §4.

[1]Utrecht University & CWI - Amsterdam, The Netherlands, menno.genseberger@wldelft.nl
[2]Utrecht University, The Netherlands, sleijpen@math.uu.nl
[3]Utrecht University, The Netherlands, vorst@math.uu.nl

We describe the domain decomposition technique for the two subdomain case. It can be generalized to more than two subdomains in a straightforward manner.

Let Ω be a domain over which some partial differential $\mathcal{L}\varphi = f$ is defined, together with appropriate boundary conditions on $\partial\Omega$. In order to compute numerical solutions, Ω is covered by a grid. The PDE is discretized accordingly, with unknowns defined on the grid points.

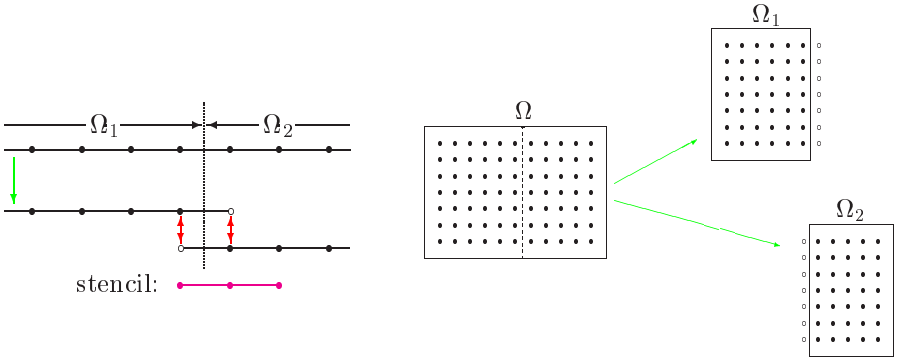


Figure 3.1: Decomposition in one (left picture) and two dimensions (right picture).

We decompose Ω in two nonoverlapping subdomains Ω_1 and Ω_2. The subdomains are covered by subgrids such that no splitting of the original discretized operator has to be made (see Figure 3.1). For that purpose additional grid points (the open bullets "$\circ$" in Figure 3.1) are introduced on the opposite side of the subgrids next to the internal interface between the subdomains. Since this introduces extra unknowns on the additional grid points, we must also provide extra equations that describe these extra unknowns. Furthermore, for the exact solution of the discretized PDE we want the function values on these additional points of one subgrid to be equal to the function values on the grid points of the other subgrid on the same location. Now, the *enhancement* consists of providing the original system with extra unknowns at the additional grid points and extra equations with precisely this property.

To do so, suppose we have ordered the discretized PDE in a linear system

$$\mathbf{B}\,\mathbf{y} = \mathbf{d}, \tag{3.1}$$

with unique solution and the following structure:

$$\begin{bmatrix} \mathbf{B}_{11} & \mathbf{B}_{1\ell} & \mathbf{B}_{1r} & \mathbf{0} \\ \mathbf{B}_{\ell 1} & B_{\ell\ell} & B_{\ell r} & \mathbf{0} \\ \mathbf{0} & B_{r\ell} & B_{rr} & \mathbf{B}_{r2} \\ \mathbf{0} & \mathbf{B}_{2\ell} & \mathbf{B}_{2r} & \mathbf{B}_{22} \end{bmatrix} \begin{bmatrix} \mathbf{y}_1 \\ y_\ell \\ y_r \\ \mathbf{y}_2 \end{bmatrix} = \begin{bmatrix} \mathbf{d}_1 \\ d_\ell \\ d_r \\ \mathbf{d}_2 \end{bmatrix}.$$

Here the labels $1, 2, \ell$, and r, respectively, refer to operations from/to and (un)knowns on subdomain Ω_1, Ω_2, and left, right from the interface, respectively. Subvector y_ℓ (y_r respectively) contains those unknowns on the left (right) from the interface that are coupled by the stencil both with unknowns in Ω_1 (Ω_2) and unknowns on the right (left) from the interface. This explains the zeros in the expression for matrix $\mathbf{B}$.

We enhance the linear system (3.1) to

$$\mathbf{B}_C\,\underset{\sim}{\mathbf{y}} = \underline{\mathbf{d}} \tag{3.2}$$

which has the following structure:

$$\left[\begin{array}{|ccc|ccc} \mathbf{B}_{11} & \mathbf{B}_{1\ell} & \mathbf{B}_{1r} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{B}_{\ell 1} & B_{\ell\ell} & B_{\ell r} & 0 & 0 & \mathbf{0} \\ \mathbf{0} & C_{\ell\ell} & C_{\ell r} & -C_{\ell\ell} & -C_{\ell r} & \mathbf{0} \\ \hline \mathbf{0} & -C_{r\ell} & -C_{rr} & C_{r\ell} & C_{rr} & \mathbf{0} \\ \mathbf{0} & 0 & 0 & B_{r\ell} & B_{rr} & \mathbf{B}_{r2} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{B}_{2\ell} & \mathbf{B}_{2r} & \mathbf{B}_{22} \end{array}\right] \left[\begin{array}{c} \mathbf{y}_1 \\ y_\ell \\ \tilde{y}_r \\ \tilde{y}_\ell \\ y_r \\ \mathbf{y}_2 \end{array}\right] = \left[\begin{array}{c} \mathbf{d}_1 \\ d_\ell \\ 0 \\ 0 \\ d_r \\ \mathbf{d}_2 \end{array}\right]. \tag{3.3}$$

Here $\tilde{y}_r$ ($\tilde{y}_\ell$ respectively) contains the unknowns at the additional grid points (the open bullets "$\circ$" in Figure 3.1) of the subgrid for Ω_1 (Ω_2) on the right (left) of the interface. So, for the exact solution of (3.2) we want that $\tilde{y}_\ell = y_\ell$ and $\tilde{y}_r = y_r$. The only requirement for the extra equations in (3.3) that the submatrix

$$C \equiv \begin{bmatrix} C_{\ell\ell} & C_{\ell r} \\ C_{r\ell} & C_{rr} \end{bmatrix},$$

the *interface coupling matrix*, is nonsingular. For nonsingular C it can be proven ([5, Theorem 1]) that the solution of the enhanced system (3.2) is unique, $\tilde{y}_\ell = y_\ell$ and $\tilde{y}_r = y_r$ as required, and the restriction of this solution $\underset{\sim}{\mathbf{y}}$ to $\mathbf{y}$ is the unique solution of the original system (3.1).

However, we want to perform solves on the subgrids only. For that purpose we split the matrix of the enhanced system (3.2) as $\mathbf{B}_C = \mathbf{M} - \mathbf{N}$. Here $\mathbf{M}$ is the boxed part in (3.3) that does not map elements from one subgrid to the other subgrid. Note that compared to $\mathbf{M}$ the remainder $\mathbf{N}$ has a relatively small number of nonzero elements. (The rank of $\mathbf{N}$ equals the dimension of C which corresponds to the amount of virtual overlap that we have created. For a five point stencil in the two subdomain case the dimension of C is for instance $2n_i$, where n_i is the number of grid points along the interface.)

A simple iterative solution method for the splitting $\mathbf{B}_C = \mathbf{M} - \mathbf{N}$ is the Richardson iteration:

$$\underset{\sim}{\mathbf{y}}^{(i+1)} = \underset{\sim}{\mathbf{y}}^{(i)} + \mathbf{M}^{-1} \left(\underline{\mathbf{d}} - \mathbf{B}_C \, \underset{\sim}{\mathbf{y}}^{(i)} \right). \tag{3.4}$$

Due to the splitting the iterates $\underset{\sim}{\mathbf{y}}^{(i)}$ of (3.3) are perturbed by errors. With $\mathbf{M}^{-1}\,\mathbf{B}_C = \mathbf{I} - \mathbf{M}^{-1}\,\mathbf{N}$ it can easily be verified that in each step these errors are amplified by the *error propagation matrix* $\mathbf{M}^{-1}\,\mathbf{N}$. Now, the idea is to use the degrees of freedom that we have created by the introduction of additional unknowns near the interface in order to damp the error components. Before we can perform this tuning of the interface coupling matrix C, we need to analyze the spectral properties of $\mathbf{M}^{-1}\,\mathbf{N}$ for the specific underlying PDE. For ordinary systems of linear equations originating from advection dominated problems this was done in [5, 4]. In §4 we describe and analyze the situation for the correction equation.

Besides the tuning of the interface coupling matrix we can further speed up the process for finding a solution of (3.3). The Richardson iteration uses only information from the last iterate for the computation of a new one. The process can be accelerated by interpreting the iterates as a Krylov subspace

$$\mathcal{K}_m \left(\mathbf{M}^{-1}\,\mathbf{B}_C, \mathbf{M}^{-1}\,\underline{\mathbf{d}} \right) = \text{span} \left(\mathbf{M}^{-1}\,\underline{\mathbf{d}}, \mathbf{M}^{-1}\,\mathbf{B}_C\,\mathbf{M}^{-1}\,\underline{\mathbf{d}}, \ldots, \left(\mathbf{M}^{-1}\,\mathbf{B}_C\right)^{m-1}\,\mathbf{M}^{-1}\,\underline{\mathbf{d}} \right)$$

and computing an approximate solution for (3.3) with respect to $\mathcal{K}_m$.

In fact, in this way the Krylov method computes a solution for the *left* preconditioned equation

$$\mathbf{M}^{-1}\,\mathbf{B}_C\,\underset{\sim}{\mathbf{x}} = \mathbf{M}^{-1} \left(\underline{\mathbf{d}} - \mathbf{B}_C \, \underset{\sim}{\mathbf{y}}^{(0)} \right), \tag{3.5}$$

where $\underset{\sim}{\mathbf{y}}^{(0)}$ is some initial guess ($\underset{\sim}{\mathbf{y}}^{(0)} = \mathbf{0}$ is convenient, but other good choices are possible as well) and a solution for (3.2) is computed from (3.5) via $\underset{\sim}{\mathbf{y}} = \underset{\sim}{\mathbf{y}}^{(0)} + \underset{\sim}{\mathbf{x}}$.

Note that *right* preconditioning is possible as well and has some nice additional properties. As it is slightly more complicated, we don't discuss it here but refer to [1, §3.2.4, §3.3.3].

4. The correction equation. In this section we describe and analyze the domain decomposition technique for the correction equation.
First it is shown how the correction equation is enhanced and how the preconditioner is incorporated. Then we pay attention to the spectrum of the error propagation matrix for a model eigenvalue problem. With this knowledge in mind, a strategy is developed for the tuning of the interface coupling matrix.

Similar to the enhancements (3.3) in 3, the following components of the correction equation have to be enhanced: the matrix $\mathbf{B} \equiv \mathbf{A} - \theta\,\mathbf{I}$ to $\mathbf{B}_C$, the correction vector $\mathbf{t}$ to $\underset{\sim}{\mathbf{t}}$, and the vectors $\mathbf{u}$ and $\mathbf{r}$ to $\underline{\mathbf{u}}$ and $\underline{\mathbf{r}}$, respectively. For the enhancement of the additional projection $\mathbf{P}$ see [1, §3.3.2].

The preconditioner $\mathbf{M}$ for $\mathbf{B}_C$ is constructed in the same way as in §3. In case of left preconditioning with $\mathbf{M}$ we compute approximate solutions to the correction equation from

$$\mathbf{P}'\,\mathbf{M}^{-1}\,\mathbf{B}_C\,\mathbf{P}'\,\underset{\sim}{\mathbf{t}} = \mathbf{P}'\,\mathbf{M}^{-1}\,\underline{\mathbf{r}} \quad \text{with} \quad \mathbf{P}' \equiv \mathbf{I} - \frac{\mathbf{M}^{-1}\,\underline{\mathbf{u}}\,\underline{\mathbf{u}}^*}{\underline{\mathbf{u}}^*\,\mathbf{M}^{-1}\,\underline{\mathbf{u}}}.$$

In [1, §3.4.3] the spectrum of the error propagation matrix is analyzed for the eigenvalue problem of an advection-diffusion operator with no cross terms and constant coefficients on two subdomains. We summarize the main results here.
The interface between the two subdomains Ω_1 and Ω is again in the y-direction. To facilitate the analysis, the discretized operator is written as a tensor product of one-dimensional discretized advection diffusion operators L_x and L_y: $L_x \otimes \mathbf{I} + \mathbf{I} \otimes L_y$. It turns out that the eigenvectors of the error propagation matrix show two typical types of behavior for the correction equation. This is illustrated in Figure 4.1.

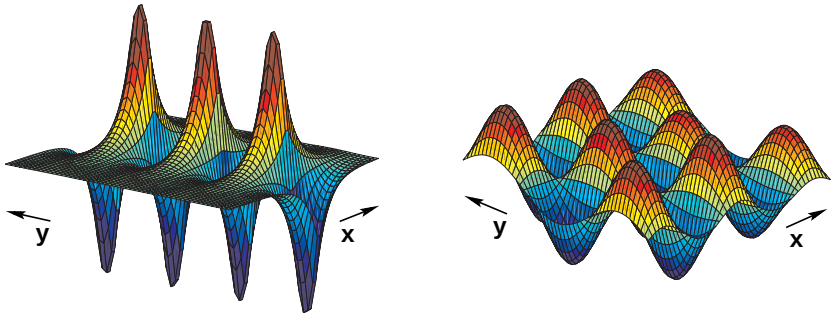


Figure 4.1: Typical eigenvectors of the error propagation matrix for the correction equation.

Parallel to the interface, all eigenvectors are coupled by eigenvectors of the one-dimensional operator L_y in the y-direction. Because of this, for the subblocks $C_{\ell\ell}$, $C_{\ell r}$, $C_{r\ell}$ and C_{rr} of the interface coupling matrix C we can take any linear combination of powers of L_y, for instance $C_{\ell\ell} = C_{rr} = \mathbf{I}$ and $C_{\ell r} = C_{r\ell} = \alpha\,\mathbf{I} + \beta\,L_y$. Here we are free to choose values for α and β, i.e. we can use these parameters for the minimization of the spectral radius of the error propagation matrix.
Perpendicular to the interface, however, there are differences. Most of the eigenvectors of the error propagation matrix show exponential behavior in the x-direction, the error grows exponentially fast when moving towards the interface (the left picture in Figure 4.1). A small number (this number depends on the location of the shift θ in the spectrum of matrix $\mathbf{A}$)

show harmonic behavior in the x-direction (the right picture in Figure 4.1), which has the disadvantage of being global.

For the eigenvectors of the error propagation matrix with exponential behavior in the x-direction we can estimate effective values for the interface coupling matrix C without specific knowledge of the subdomain size. In §5 we will see that this is of interest for more practical situations. For this reason we minimize the spectral radius of the error propagation matrix only with respect to these eigenvectors. With deflation the remaining eigenvectors, those with harmonic behavior in the x-direction, are controlled. We illustrate deflation by means of an example.

4.1. Deflation. Now we show, by example, how deflation improves the condition of the preconditioned correction equation. We consider θ equal to the 20th eigenvalue of the Laplace operator on a domain $[0,1] \times [0,1]$. The domain is covered by a 31×31 grid and decomposed in two equal subdomains.

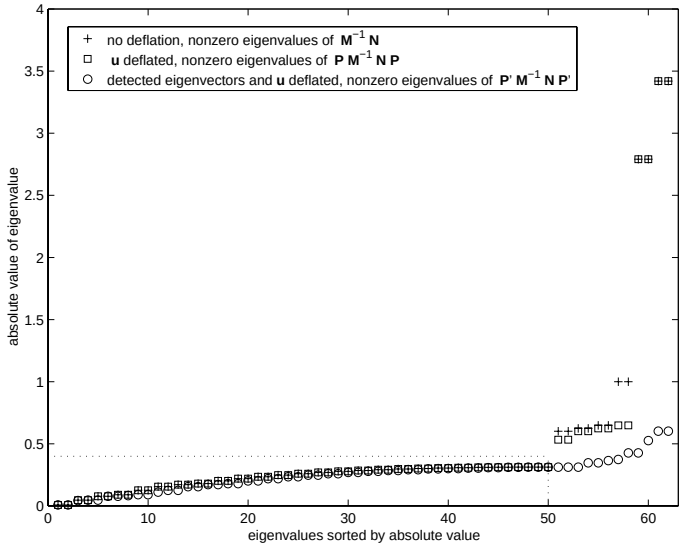


Figure 4.2: The effect of deflation.

In Figure 4.2 the nonzero eigenvalues of the error propagation matrix are shown for this situation.

Twelve eigenvectors of the error propagation matrix behave harmonic perpendicular to the interface. As we do not include them for the optimization, we do not necessarily damp these eigenvectors with the constructed interface coupling matrix as indicated by the twelve rightmost '+'-s (no deflation) in Figure 4.2.

Two of these eigenvectors are connected to the y-component of the eigenvector that corresponds to the 20th eigenvalue of the original eigenvalue problem: these eigenvectors can not be controlled at all with the interface coupling matrix because the operator $\mathbf{A}$ is shifted by this 20th eigenvalue and therefore singular in the direction of the corresponding eigenvector. In the correction equation the operator stays well-conditioned due to the projection $\mathbf{P}$ that deflates precisely this direction. Since the error propagator originates from the enhanced operator in the correction equation, this projection is actually incorporated in the error propagator and the '□'-s at positions 57 and 58 in Figure 4.2 show the positive effect. The other eigenvectors with harmonic behavior perpendicular to the interface can be controlled with information from the search subspace of Jacobi-Davidson itself: in practice one

starts the computation with the largest eigenvalues and when arrived at 20th one, the 19 largest eigenvalues with corresponding eigenvectors are already computed and will be deflated from the operator **B**. Deflation with these 19 already computed eigenvectors drastically reduces the absolute values, as the '$\circ$'-s at the horizontal positions $51, \ldots, 56$ and $59, \ldots, 62$ show in Figure 4.2.

From this example we learned that deflation may help to cluster the part of the spectrum that we can not control with the coupling parameters, and therefore improves the conditioning of the preconditioned correction equation. The remaining part of the spectrum, that is the eigenvalues that are in control (indicated by the dotted lines in Figure 4.2), can be damped even more with a stronger optimized coupling.

5. Applications. With the results from the analysis for the two subdomain case with constant coefficients in §4 we can accurately estimate optimal interface coupling matrices C for more than two subdomains, variable coefficients, and complicated geometries.
As an illustration we give two numerical examples, for specific details we refer to [1].

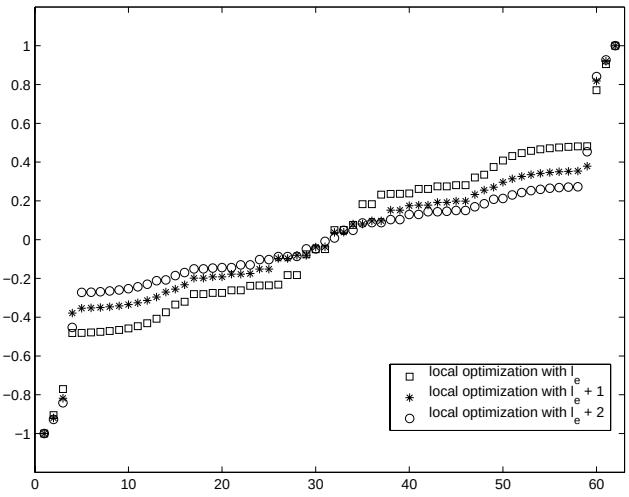


Figure 5.1: Eigenvalues of the error propagation matrix of the correction equation for an operator with a large jump.

5.1. Variable coefficients. In this example we illustrate the effectiveness of the determination of interface coupling matrices C for eigenvalue problems with variable coefficients.
Consider the following operator with a large jump:

$$\mathcal{L} \equiv \frac{\partial}{\partial x}[c(y)\frac{\partial}{\partial x}] + \frac{\partial}{\partial y}[c(y)\frac{\partial}{\partial y}] \text{ with } c(y) = \begin{cases} 1 \text{ for } 0 \le y < 0.25 \\ 1000 \text{ for } 0.25 \le y < 0.75 \\ 1 \text{ for } 0.75 \le y \le 1 \end{cases}$$

defined on $[0, 2] \times [0, 1]$. We focus on the largest eigenvalue of this operator, the corresponding eigenvector is the most smooth one among all eigenvectors. The domain is decomposed into two equal subdomains with physical sizes $[0, 1] \times [0, 1]$ and $[1, 2] \times [0, 1]$, and covered by a 31×31 grid.

Based on a local optimization strategy [1, §4.3], which uses results of the constant coefficients case, we determined appropriate values for the interface coupling matrix. Figure 5.1

shows the eigenvalues of the corresponding error propagation matrix. It shows the effectiveness of C: smaller eigenvalues result in faster damping of the errors.
For the optimization three values of l_e are considered. This l_e marks the subdivision in harmonic and exponential behavior perpendicular to the interface of the eigenvectors of the error propagation matrix. For constant coefficients we were able to determine the precise, fixed value of l_e. For variable coefficients the value of l_e varies.

If we concentrate on the eigenvalues at the horizontal positions $4, 5, \ldots, 59$ in Figure 5.1, then we see that, compared to the local optimization with l_e (the '$\square$'-s), these eigenvalues are closer to zero for the local optimization with $l_e + 1$ (the '$*$'-s). So, for variable coefficients, the outcome of this experiment indicates that the value of l_e should not be chosen too sharp.

From the figure it can be concluded that, except for a couple of outliers at the horizontal positions $1, 2, 3, 60, 61$, and 62 (which can be controlled by deflation and/or the Krylov acceleration), the local optimization strategy yields an effective interface coupling matrix C.

5.2. More than two subdomains. For this example, we start with an eigenvalue problem that is defined on two square subdomains of equal size. The subdomains are covered by a 63×63 subgrid. The number of subdomains is increased by pasting a new subdomain of the same size. So we model a channel that becomes larger each time. With Jacobi-Davidson we compute an approximate solution of the eigenpair that corresponds to the largest eigenvector of the two-dimensional Laplace operator. Each step of Jacobi-Davidson we use 4 steps of the Krylov method GMRES [2] preconditioned with the preconditioner based on domain decomposition. Given a number of subdomains (first row in Table 5.1) we compare the total number of Jacobi-Davidson steps that are needed such that the ℓ_2-norm of the residual $\mathbf{r}$ of the approximate eigenvalue is less than 10^{-9} for three kinds of coupling: Neumann-Dirichlet coupling ("ad hoc" choice for C: Neumann boundary condition on the left: $C_{\ell\ell} = \mathbf{I}$, $C_{\ell r} = -\mathbf{I}$ and Dirichlet boundary condition on the right: $C_{r\ell} = C_{rr} = \mathbf{I}$), simple optimized coupling ($C_{\ell\ell} = C_{rr} = \mathbf{I}$ and $C_{\ell r} = C_{r\ell} = \alpha\,\mathbf{I}$), and stronger optimized coupling ("finetuning" of C: $C_{\ell\ell} = C_{rr} = \mathbf{I} + \gamma L_y$ and $C_{\ell r} = C_{r\ell} = \alpha\,\mathbf{I} + \beta\,L_y$). For the simple and stronger optimized coupling we estimate optimal values for C by doing as if the decomposition is in two subdomains only. With the results from the analysis in §3 we determine optimal values for C for the two subdomain case. Because only the eigenvectors of the error propagation matrix that damp exponentionally when moving away from the interface are taken into account for the optimization, these values for C are also fairly good when the number of subdomains is larger than two.

number of subdomains	2	3	4	5	6
Neumann-Dirichlet coupling	5	9	19	21	22
simple optimized coupling	6	8	9	10	12
stronger optimized coupling	5	6	8	9	10

Table 5.1: Overall Jacobi-Davidson process on more subdomains for three different types of coupling.

From the table it can be concluded that a finer tuning of C pays off in the overall Jacobi-Davidson process. Note that for ease of presentation we used the Laplace operator here, experiments with more general advection-diffusion operators showed similar results.

REFERENCES

[1] M. Genseberger. *Domain decomposition in the Jacobi-Davidson method for eigenproblems*. PhD thesis, Utrecht University, September 2001.

[2] Y. Saad and M. H. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Stat. Comp.*, 7:856–869, 1986.

[3] 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(2):401–425, 1996. Reappeared in SIAM Review 42:267–293, 2000.

[4] K. H. Tan. *Local coupling in domain decomposition*. PhD thesis, Utrecht University, 1995.

[5] K. H. Tan and M. J. A. Borsboom. On generalized Schwarz coupling applied to advection-dominated problems. In D. E. Keyes and J. Xu, editors, *Seventh International Conference of Domain Decomposition Methods in Scientific and Engineering Computing*, pages 125–130. AMS, 1994. Held at Penn State University, October 27-30, 1993.

[6] W. P. Tang. Generalized Schwarz splittings. *SIAM J. Sci. Stat. Comp.*, 13(2):573–595, 1992.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

29. Optimization of Interface Operator Based on Algebraic Approach

François-Xavier Roux[1], Frédéric Magoulès[2], Stéphanie Salmon, Laurent Series

1. Introduction. This paper is dedicated to recent developments of an optimized two-Lagrange multiplier domain decomposition method [5], [8]). Most methods for optimizing the augmented interface operator are based on the discretization of approximations of the continuous transparent operator [4], [1], [2], [7]. At the discrete level, the optimal operator can be proved to be equal to the Schur complement of the outer domain. This Schur complement can be directly approximated using purely algebraic techniques like sparse approximate inverse methods or incomplete factorization. The main advantage of such algebraic approach is that it is much more easy to implement in existing code without any information on the geometry of the interface and the finite element formulation used. Convergence results and parallel efficiency of several algebraic optimization techniques of interface operator for acoustic analysis applications will be presented.

2. Algebraic Formulation of Domain Decomposition Methods.

2.1. General Presentation. Consider a splitting of the domain Ω as in Figure 2.1 and note by subscripts i and p the degrees of freedom located inside subdomain $\Omega^{(s)}$, $s = 1, 2$, and on the interface Γ_p. Then, the contribution of subdomain $\Omega^{(s)}$, $s = 1, 2$ to the matrix and the right-hand side of a finite element discretization of a linear PDE on Ω can be written as follows:

$$K^{(s)} = \begin{bmatrix} K_{ii}^{(s)} & K_{ip}^{(s)} \\ K_{pi}^{(s)} & K_{pp}^{(s)} \end{bmatrix} \quad , \quad b^{(s)} = \begin{bmatrix} b_i^{(s)} \\ b_p^{(s)} \end{bmatrix} \tag{2.1}$$

where $K_{pp}^{(1)}$ and $K_{pp}^{(2)}$ represent the interaction matrices between the nodes on the interface obtained by integration on $\Omega^{(1)}$ and on $\Omega^{(2)}$. The global problem is a block system obtained by assembling local contribution of each subdomain:

$$\begin{bmatrix} K_{ii}^{(1)} & 0 & K_{ip}^{(1)} \\ 0 & K_{ii}^{(2)} & K_{ip}^{(2)} \\ K_{pi}^{(1)} & K_{pi}^{(2)} & K_{pp} \end{bmatrix} \begin{bmatrix} x_i^{(1)} \\ x_i^{(2)} \\ x_p \end{bmatrix} = \begin{bmatrix} b_i^{(1)} \\ b_i^{(2)} \\ b_p \end{bmatrix}. \tag{2.2}$$

The block K_{pp} is the sum of the two blocks $K_{pp}^{(1)}$ and $K_{pp}^{(2)}$. In the same way, $b_p = b_p^{(1)} + b_p^{(2)}$ is obtained by local integration in each subdomain and sum on the interface.

[1]ONERA, francois-xavier.roux@onera.fr, stephanie.salmon@onera.fr, laurent.series@onera.fr
[2]Université Henri Poincaré, frederic.magoules@iecn.u-nancy.fr

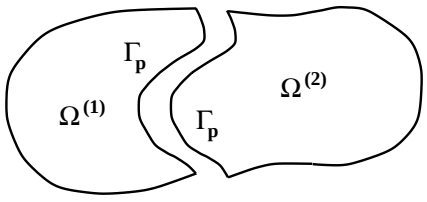


Figure 2.1: Non-overlapping domain splitting

Theorem 2.1 (existence and uniqueness) *Given any splitting of* $K_{pp} = K_{pp}^{(1)} + K_{pp}^{(2)}$ *and* $b_p = b_p^{(1)} + b_p^{(2)}$, *and any matrices* $A^{(1)}$, $A^{(2)}$ *there is only one pair of Lagrange multipliers* $\lambda^{(1)}$, $\lambda^{(2)}$ *such as the following coupled problem:*

$$\begin{bmatrix} K_{ii}^{(1)} & K_{ip}^{(1)} \\ K_{pi}^{(1)} & K_{pp}^{(1)} + A^{(1)} \end{bmatrix} \begin{bmatrix} x_i^{(1)} \\ x_p^{(1)} \end{bmatrix} = \begin{bmatrix} b_i^{(1)} \\ b_p^{(1)} + \lambda^{(1)} \end{bmatrix} \tag{2.3}$$

$$\begin{bmatrix} K_{ii}^{(2)} & K_{ip}^{(2)} \\ K_{pi}^{(2)} & K_{pp}^{(2)} + A^{(2)} \end{bmatrix} \begin{bmatrix} x_i^{(2)} \\ x_p^{(2)} \end{bmatrix} = \begin{bmatrix} b_i^{(2)} \\ b_p^{(2)} + \lambda^{(2)} \end{bmatrix} \tag{2.4}$$

$$x_p^{(1)} - x_p^{(2)} = 0 \tag{2.5}$$

$$\lambda^{(1)} + \lambda^{(2)} - A^{(1)} x_p^{(1)} - A^{(2)} x_p^{(2)} = 0 \tag{2.6}$$

is equivalent to the problem (2.2).

Proof. The admissibility condition (2.5) derives from the relation $x_p^{(1)} = x_p^{(2)} = x_p$.
If $x_p^{(1)} = x_p^{(2)} = x_p$, the first rows of local systems (2.3) and (2.4) are the same as the two first rows of global system (2.2), and adding the last rows of local systems (2.3) and (2.4) gives:

$$K_{pi}^{(1)} x_i^{(1)} + K_{pi}^{(2)} x_i^{(2)} + K_{pp}\ x_p - b_p = \lambda^{(1)} + \lambda^{(2)} - A^{(1)} x_p^{(1)} - A^{(2)} x_p^{(2)} \tag{2.7}$$

So, the last equation of global system (2.2) is satisfied only if:

$$\lambda^{(1)} + \lambda^{(2)} - A^{(1)} x_p^{(1)} - A^{(2)} x_p^{(2)} = 0 \tag{2.8}$$

Reversely, if $x_p^{(1)}$, $x_p^{(2)}$ and x_p are derived from global system (2.2), then local systems (2.3) and (2.4) define $\lambda^{(1)}$ and $\lambda^{(2)}$ in a unique way. ∎

2.2. Two-Lagrange Multiplier Domain Decomposition Method . If the local inner matrix $K_{ii}^{(s)}$ is non singular, a direct relation between $x_p^{(s)}$ and $\lambda^{(s)}$ can be obtained from (2.3) and (2.4):

$$x_p^{(s)} = [S^{(s)} + A^{(s)}]^{-1} (c_p^{(s)} + \lambda^{(s)}) \tag{2.9}$$

where $S^{(s)} = K_{pp}^{(s)} - K_{pi}^{(s)} [K_{ii}^{(s)}]^{-1} K_{ip}^{(s)}$ is the Schur complement and $c_p^{(s)} = b_p^{(s)} - K_{pi}^{(s)} [K_{ii}^{(s)}]^{-1} b_i^{(s)}$ is the condensed right hand side in subdomain $\Omega^{(s)}$.
After substitution of $x_p^{(1)}$ and $x_p^{(2)}$ in the interface continuity conditions (2.5) and (2.6) the following linear system is obtained:

$$\begin{bmatrix} [S^{(1)} + A^{(1)}]^{-1} & -[S^{(2)} + A^{(2)}]^{-1} \\ I - A^{(1)} [S^{(1)} + A^{(1)}]^{-1} & I - A^{(2)} [S^{(2)} + A^{(2)}]^{-1} \end{bmatrix} \begin{bmatrix} \lambda^{(1)} \\ \lambda^{(2)} \end{bmatrix} = \begin{bmatrix} -[S^{(1)} + A^{(1)}]^{-1} c_p^{(1)} + [S^{(2)} + A^{(2)}]^{-1} c_p^{(2)} \\ A^{(1)} [S^{(1)} + A^{(1)}]^{-1} c_p^{(1)} + A^{(2)} [S^{(2)} + A^{(2)}]^{-1} c_p^{(2)} \end{bmatrix} \tag{2.10}$$

The solution of this system by a Krylov method defines a non overlapping domain decomposition method.

2.3. Discrete Transmission Conditions as Local Preconditioner. Instead of relations (2.5) and (2.6) on the interface, it may be more interesting to consider another set of conditions:

$$C^{(1)}\ (2.5) + (2.6) \quad = \quad 0 \tag{2.11}$$
$$-C^{(2)}\ (2.5) + (2.6) \quad = \quad 0 \tag{2.12}$$

which are equivalent to the initial relations, as soon as the two matrices $C^{(1)}$ and $C^{(2)}$ are such that $C^{(1)} + C^{(2)}$ is invertible. Following the same steps than in section 2.2, the matrix and the right hand side of the linear system takes now the block form:

$$\begin{bmatrix} I - (A^{(1)} - C^{(1)})[S^{(1)} + A^{(1)}]^{-1} & I - (A^{(2)} + C^{(1)})[S^{(2)} + A^{(2)}]^{-1} \\ I - (A^{(1)} + C^{(2)})[S^{(1)} + A^{(1)}]^{-1} & I - (A^{(2)} - C^{(2)})[S^{(2)} + A^{(2)}]^{-1} \end{bmatrix} \tag{2.13}$$
$$\begin{bmatrix} (A^{(1)} - C^{(1)})[S^{(1)} + A^{(1)}]^{-1} c_p^{(1)} + (A^{(2)} + C^{(1)})[S^{(2)} + A^{(2)}]^{-1} c_p^{(2)} \\ (A^{(1)} + C^{(2)})[S^{(1)} + A^{(1)}]^{-1} c_p^{(1)} + (A^{(2)} - C^{(2)})[S^{(2)} + A^{(2)}]^{-1} c_p^{(2)} \end{bmatrix} \tag{2.14}$$

This manipulation on the interface relations simply correspond to a left multiplication of the linear system (2.10) by the following preconditioner:

$$\begin{bmatrix} C^{(1)} & I \\ -C^{(2)} & I \end{bmatrix} \tag{2.15}$$

Different choices can be considered for the matrices $C^{(1)}$ and $C^{(2)}$, but a natural choice consist in $C^{(1)} = A^{(1)}$ and $C^{(2)} = A^{(2)}$. Indeed, with this choice, the constraints on the interface becomes:

$$\lambda^{(1)} + \lambda^{(2)} - (A^{(1)} + A^{(2)})x_p^{(1)} \quad = \quad 0 \tag{2.16}$$
$$\lambda^{(1)} + \lambda^{(2)} - (A^{(1)} + A^{(2)})x_p^{(2)} \quad = \quad 0 \tag{2.17}$$

and the diagonal block of the matrix of the linear system reduces to the identity block:

$$\begin{bmatrix} I & I - (A^{(1)} + A^{(2)})[S^{(2)} + A^{(2)}]^{-1} \\ I - (A^{(1)} + A^{(2)})[S^{(1)} + A^{(1)}]^{-1} & I \end{bmatrix} \begin{bmatrix} \lambda^{(1)} \\ \lambda^{(2)} \end{bmatrix} = \begin{bmatrix} (A^{(1)} + A^{(2)})[S^{(2)} + A^{(2)}]^{-1} c_p^{(2)} \\ (A^{(1)} + A^{(2)})[S^{(1)} + A^{(1)}]^{-1} c_p^{(1)} \end{bmatrix} \tag{2.18}$$

3. Optimal Discrete Transmission Conditions. In the context of the additive Schwarz method with no overlap, it is shown in [9], [3] that the best choice for the continuous augmented operators $\mathcal{A}^{(s)}$, $s = 1, 2$ corresponds to the continuous transparent operators, which are not partial differential operators. Different techniques of approximation based on two dimensional Fourier analysis of Steklov-Poincaré operator in an half space have been analyzed in the recent years [4], [1] [2], [7].

In the following, a new analysis is performed directly on the discrete problem and shows that the optimal convergence of a two-Lagrange multiplier algorithm is obtained with a choice of the augmented term $A^{(s)}$, $s = 1, 2$ equal to the complete outer Schur complement. The extension to the case of a one-way splitting is analyzed.

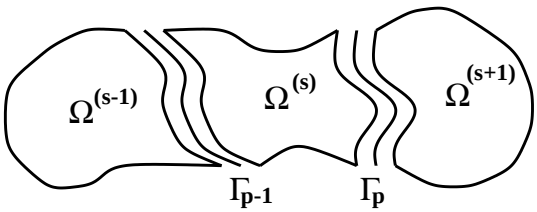


Figure 3.1: One-way decomposition without cross-points

3.1. Two-domain splitting. Eliminating of the inner unknowns of outer subdomain, $x_i^{(q)}$, $q = 1, 2$, $q \neq s$ in sysstem (2.2) leads to:

$$\begin{bmatrix} K_{ii}^{(s)} & K_{ip}^{(s)} \\ K_{pi}^{(s)} & K_{pp}^{(s)} + S^{(q)} \end{bmatrix} \begin{bmatrix} x_i^{(s)} \\ x_p \end{bmatrix} = \begin{bmatrix} b_i^{(s)} \\ b_p^{(s)} + c_p^{(q)} \end{bmatrix} \tag{3.1}$$

where $S^{(q)}$ and $c_p^{(q)}$ denote the Schur complement and condensed right hand side like in section 2.2. Equation (3.1) suggests that the optimal augmented term to add to local admittance matrix $K^{(s)}$ on interface is $S^{(q)}$, since then system (3.1) is similar to the augmented local problem:

$$\begin{bmatrix} K_{ii}^{(s)} & K_{ip}^{(s)} \\ K_{pi}^{(s)} & K_{pp}^{(s)} + A^{(s)} \end{bmatrix} \begin{bmatrix} x_i^{(s)} \\ x_p^{(s)} \end{bmatrix} = \begin{bmatrix} b_i^{(s)} \\ b_p^{(s)} + \lambda^{(s)} \end{bmatrix} \tag{3.2}$$

Theorem 3.1 *In a case of a two-domain splitting, the simple (Jacobi) iterative algorithm for 2-Lagrange multiplier with augmented term equal to the complete outer Schur complement defined as in equation (3.1) converges in one iteration at most.*

Proof. Choosing augmented local terms $A^{(s)} = S^{(q)}$, $s = 1, 2$, $q = 1, 2$, $s \neq q$ makes the matrix of condensed interface system (2.18) equal to identity. ■

3.2. One-way Splitting. Consider a one-way splitting of the domain as in Figure 3.1 and note by subscripts i, $p-1$ and p the degrees of freedom located inside subdomain $\Omega^{(s)}$, on left interface Γ_{p-1} and right interface Γ_p. Then, the contribution of subdomain $\Omega^{(s)}$ to admittance matrix and right-hand side can be written:

$$K^{(s)} = \begin{bmatrix} K_{ii}^{(s)} & K_{ip-1}^{(s)} & K_{ip}^{(s)} \\ K_{p-1i}^{(s)} & K_{p-1p-1}^{(s)} & 0 \\ K_{pi}^{(s)} & 0 & K_{pp}^{(s)} \end{bmatrix} \quad , \quad b^{(s)} = \begin{bmatrix} b_i^{(s)} \\ b_{p-1}^{(s)} \\ b_p^{(s)} \end{bmatrix} \tag{3.3}$$

The global system of equations can be reduced on the interfaces by elimination of inner degrees of freedom. The contribution of subdomain $\Omega^{(s)}$ to the condensed matrix and right-hand side is as follows:

$$\begin{bmatrix} S_{p-1p-1}^{(s)} & S_{p-1p}^{(s)} \\ S_{pp-1}^{(s)} & S_{pp}^{(s)} \end{bmatrix} = \\ = \begin{bmatrix} K_{p-1p-1}^{(s)} - K_{p-1i}^{(s)}[K_{ii}^{(s)}]^{-1}K_{ip-1}^{(s)} & -K_{p-1i}^{(s)}[K_{ii}^{(s)}]^{-1}K_{ip}^{(s)} \\ -K_{pi}^{(s)}[K_{ii}^{(s)}]^{-1}K_{ip-1}^{(s)} & K_{pp}^{(s)} - K_{pi}^{(s)}[K_{ii}^{(s)}]^{-1}K_{ip}^{(s)} \end{bmatrix} \\ \begin{bmatrix} c_{p-1}^{(s)} \\ c_p^{(s)} \end{bmatrix} = \begin{bmatrix} b_{p-1}^{(s)} & - & K_{p-1i}^{(s)} & [K_{ii}^{(s)}]^{-1} & b_i^{(s)} \\ b_p^{(s)} & - & K_{pi}^{(s)} & [K_{ii}^{(s)}]^{-1} & b_i^{(s)} \end{bmatrix} \tag{3.4}$$

The global condensed problem on interfaces is a block 3-diagonal system obtained by assembling local contribution of each subdomain:

$$\begin{bmatrix} ... & ... & 0 & 0 \\ S^{(s-1)}_{p-1p-2} & S^{(s-1)}_{p-1p-1} + S^{(s)}_{p-1p-1} & S^{(s)}_{p-1p} & 0 \\ 0 & S^{(s)}_{pp-1} & S^{(s)}_{pp} + S^{(s+1)}_{pp} & S^{(s+1)}_{pp+1} \\ 0 & 0 & ... & ... \end{bmatrix} , \begin{bmatrix} ... & & \\ c^{(s-1)}_{p-1} & + & c^{(s)}_{p-1} \\ c^{(s)}_{p} & + & c^{(s+1)}_{p} \\ ... & & \end{bmatrix} \quad (3.5)$$

If the system (3.5) is factorized by successive condensation of matrix and right-hand side starting from both ends up to block associated with subdomain $\Omega^{(s)}$, the following final condensed problem is obtained in one subdomain:

$$\begin{bmatrix} S^{-}_{p-1p-1} + S^{(s)}_{p-1p-1} & S^{(s)}_{p-1p} \\ S^{(s)}_{pp-1} & S^{(s)}_{pp} + S^{+}_{pp} \end{bmatrix} \begin{bmatrix} x_{p-1} \\ x_p \end{bmatrix} = \begin{bmatrix} c^{-}_{p-1} & + & c^{(s)}_{p-1} \\ c^{(s)}_{p} & + & c^{+}_{p} \end{bmatrix} \quad (3.6)$$

The condensed right and left blocks and right-hand sides of system (3.6) that are noted with plus and minus super-script are defined by the following recurrence relations:

$$\begin{array}{lcllll} S^{+}_{p-1p-1} & = & S^{(s)}_{p-1p-1} & - & S^{(s)}_{p-1p} & [S^{(s)}_{pp} + S^{+}_{pp}]^{-1} S^{(s)}_{pp-1} \\ S^{-}_{pp} & = & S^{(s)}_{pp} & - & S^{(s)}_{pp-1} & [S^{-}_{p-1p-1} + S^{(s)}_{p-1p-1}]^{-1} S^{(s)}_{p-1p} \\ c^{+}_{p-1} & = & c^{(s)}_{p-1} & - & S^{(s)}_{p-1p} & [S^{(s)}_{pp} + S^{+}_{pp}]^{-1} [c^{(s)}_{p} + c^{+}_{p}] \\ c^{-}_{p} & = & c^{(s)}_{p} & - & S^{(s)}_{pp-1} & [S^{-}_{p-1p-1} + S^{(s)}_{p-1p-1}]^{-1} [c^{-}_{p-1} + c^{(s)}_{p-1}] \end{array} \quad (3.7)$$

Equation (3.6) suggests that the optimal augmented term to add to local admittance matrix $K^{(s)}$ on left or right interface is respectively S^{-}_{p-1p-1} and S^{+}_{pp}, since then, if $\Omega^{(s)}$ is the only subdomain with non zero right-hand side, $c^{-}_{p-1} = 0$ and $c^{+}_{p} = 0$, and system (3.6) is exactly the condensation of the augmented local problem:

$$\begin{bmatrix} K^{(s)}_{ii} & K^{(s)}_{ip-1} & K^{(s)}_{ip} \\ K^{(s)}_{p-1i} & K^{(s)}_{p-1p-1} + S^{-}_{p-1p-1} & 0 \\ K^{(s)}_{pi} & 0 & K^{(s)}_{pp} + S^{+}_{pp} \end{bmatrix} \begin{bmatrix} x^{(s)}_{i} \\ x^{(s)}_{p-1} \\ x^{(s)}_{p} \end{bmatrix} = \begin{bmatrix} b^{(s)}_{i} \\ b^{(s)}_{p-1} \\ b^{(s)}_{p} \end{bmatrix} \quad (3.8)$$

Theorem 3.2 *In a case of a one-way splitting, the simple (Jacobi) iteration algorithm for 2-Lagrange multiplier with augmented term equal to complete outer Schur complement defined as in equation (3.8) converges in (number of subdomain - 1) iterations at most.*

Proof. If $\Omega^{(s)}$ is the only subdomain with non zero right-hand side, equations (3.6) and (3.8) mean that the first iteration with null initial Lagrange multipliers gives exact solution in Ω_s and zero in the other subdomains.
Since λ and x are zero everywhere except on Γ_{p-1} and Γ_p, the initial gradient is non zero on adjacent interfaces only:

$$\begin{array}{l} g^{(s-1)}_{p-1} = \lambda^{(s-1)}_{p-1} + \lambda^{(s)}_{p-1} - (S^{-}_{p-1p-1} + S^{+}_{p-1p-1}) x^{(s)}_{p-1} = -(S^{-}_{p-1p-1} + S^{+}_{p-1p-1}) x^{(s)}_{p-1} \\ g^{(s+1)}_{p} = \lambda^{(s+1)}_{p} + \lambda^{(s)}_{p} - (\ S^{-}_{pp} + S^{+}_{pp}\ )\ \ x^{(s)}_{p} = -(\ S^{-}_{pp} + S^{+}_{pp}\ )\ \ x^{(s)}_{p} \end{array} \quad (3.9)$$

By condensation of equation (3.6) it comes that initial solution on interface Γ_{p-1} satisfies:

$$(S^{-}_{p-1p-1} + S^{(s)}_{p-1p-1} - S_{p-1p}[S^{(s)}_{pp} + S^{+}_{pp}]^{-1} S_{pp-1}) x^{(s)}_{p-1} = \quad (3.10)$$

$$c^{(s)}_{p-1} - S_{p-1p}[S^{(s)}_{pp} + S^{+}_{pp}]^{-1} c^{(s)}_{p} \quad (3.11)$$

Under the assumption that right-hand-side is non zero in Ω_s only, $c_p^{(s)} = c_p^+$. So, from definition of condensed matrices and right-hand sides (3.7) it derives from equation (3.11) that:

$$(S_{p-1p-1}^- + S_{p-1p-1}^+) x_{p-1}^{(s)} = c_{p-1}^+ \tag{3.12}$$

and so:

$$g_{p-1}^{(s-1)} = - (S_{p-1p-1}^- + S_{p-1p-1}^+) x_{p-1}^{(s)} = -c_{p-1}^+ \tag{3.13}$$

A similar result is obtained for $g_p^{(s+1)}$.
The Jacobi algorithm on the condensed interface problem consists in updating λ by $\lambda - g$. So, at the second iteration, both subdomains $s-1$ and $s+1$ will have their complete condensed right-hand side, as well as subdomain s for which $\lambda_{p-1}^{(s)}$ and $\lambda_p^{(s)}$ will remain unchanged and equal to zero. After the second iteration, the solution in the three subdomain will be the exact restriction of the solution of the global problem.
It is easy to see now that, at iteration 2, the situation on interface Γ_{p-2} between subdomains Ω_{s-2} and Ω_{s-1} is exactly the same as, at iteration 1, on interface Γ_{p-1} between subdomains Ω_{s-1} and Ω_s. So, exact condensed right-hand side will be passed to subdomain Ω_{s-2} when updating λ at iteration 2.
On the other hand, if λ is such that in two neighboring subdomains Ω_{s-1} and Ω_s with interface Γ_{p-1}, the local condensed right-hand sides are complete, then $x_{p-1}^{(s)} = x_{p-1}^{(s)} = x_{p-1}$. Condensation on interface Γ_{p-1} of equation (3.6) gives:

$$(S_{p-1p-1}^- + S_{p-1p-1}^+)x_{p-1} = c_{p-1}^- + c_{p-1}^+ \tag{3.14}$$

So, if $\lambda_{p-1}^{(s-1)} + \lambda_{p-1}^{(s)} = c_{p-1}^- + c_{p-1}^+$ then:

$$g_{p-1}^{(s-1)} = g_{p-1}^{(s)} = \lambda_{p-1}^{(s-1)} + \lambda_{p-1}^{(s)} - (S_{p-1p-1}^- + S_{p-1p-1}^+)x_{p-1} = 0 \tag{3.15}$$

This is exactly the situation between Ω_{s-1} and Ω_s as well as between Ω_s and Ω_{s+1} at iteration 2. This means that the gradient will be zero on all the interfaces of these subdomains at iteration 2.
In the same way, it can be proved by recurrence that each Jacobi iteration will propagate the complete condensed right-hand side one subdomain further on the left and on the right while leaving the values of λ unmodified in all subdomains where the condensed right-hand side is already complete.
So, Jacobi method will converge in at most (number of subdomain - 1) iterations if the initial right-hand side is non zero in only one subdomain. As any general right-hand side can be decomposed in the sum of right-hand sides that are non zero in one subdomain only, and since the Jacobi procedure is additive, the same result holds for any case. ■

4. Approximation of Optimal Discrete Transmission Conditions. Unfortunately, the optimal choice derive in the previous section can not be done in practice since the computational cost of the complete Schur complement matrix is too expensive. A first natural step to reduce the cost consists in approximating the complete Schur complement with the Schur complement of the neighboring subdomains. Nevertheless, even with this approximation, the matrix $A^{(s)}$ is still dense and adding it to the local matrix $K^{(s)}$ increases its bandwidth a lot. So, rather than to consider the exact Schur complement, we can consider its approximation with a sparse matrix. One method of choice to approximate this dense matrix is based on the Sparse Approximate Inverse (SPAI) method. It consists in approximating the inverse of a $N \times N$ matrix A by a sparse matrix M which minimizes the Frobenius norm of $AM - I$ [10].
The ultimate step consists in approximating the Schur complement matrix by its first term

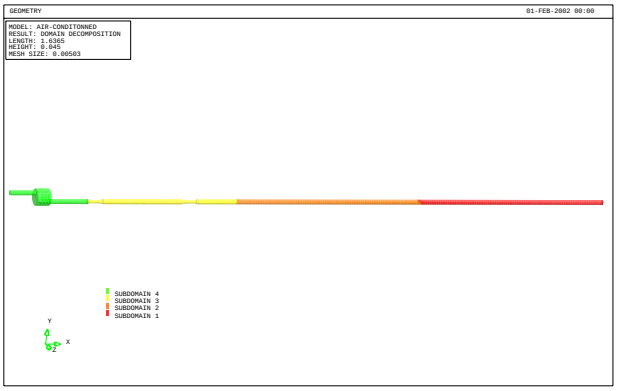


Figure 5.1: Decomposition of the air-cooling tube into four subdomains

Number of Subdomains	Exact Schur Complement		Approximation of Schur Complement		Absorbing Interface Conditions	
	Complete Schur Complement	Neighbor Schur Complement	Sparse Approx.	Lumped Approx.	Taylor	Optimized
2	1	1	12	10	92	86
4	3	4	27	30	155	137
6	5	8	41	46	212	174
8	7	12	56	77	311	247

Table 5.1: Number of iterations for different regularization matrix and different number of subdomains for the air-cooling tube problem

$K_{pp}^{(s)}$ like in the lumped preconditioner for the FETI method [6]. Such an approximation is extremely easy to implement and since $K_{pp}^{(s)}$ and $K_{pp}^{(q)}$ have the same sparse structure, the sparse structure of the local subdomain matrix is not modified.

5. Numerical Experiments. A three dimensional simulation of the noise level distribution in an air-conditionned tube is performed. Figure 5.1 shows the decomposition of the initial mesh into four subdomains. It is important to notice that the interface between the subdomains is irregular. The problem is characterized by a reduced frequency $\omega a = 75.60$ which corresponds, with the relation $\omega = 2\pi F/c$ with c the sound celerity in the fluid, a the length of the tube, to a frequency F of 2500 Hz. The length a is equal to 1.6365 and the diameter to 0.045. When using zeroth order Taylor conditions and a decomposition into 16 subdomains, the method needs 100 iterations to converge, whereas when using the "lumped" approximation of the Schur complement the method converges in 10 iterations. The SPAI approximation gives slightly faster convergence than the lumped for larger number of subdomains. The stopping criterion on the relative global error is set to 10^{-9}. More results are reported Table (5.1).

6. Conclusions. A general algebraic presentation of two-Lagrange multiplier domain decomposition method has been introduced. Optimal transmission conditions have been derived from this algebraic analysis. Since the optimal augmented operator in a subdomain is the Schur complement of the outer domain, it is not possible to compute it in practice. Promising results have been obtained using simple approximation techniques for this Schur

complement. The key issue to improve the method presented in this paper lies in the design of good sparse approximation method.

REFERENCES

[1] J. Benamou and B. Després. A domain decomposition method for the Helmholtz equation and related optimal control problems. *J. of Comp. Physics*, 136:68–82, 1997.

[2] P. Chevalier and F. Nataf. Symmetrized method with optimized second-order conditions for the Helmholtz equation. In *Domain decomposition methods, 10 (Boulder, CO, 1997)*, pages 400–407. Amer. Math. Soc., Providence, RI, 1998.

[3] F. Collino, S. Ghanemi, and P. Joly. Domain decomposition method for harmonic wave propagation: a general presentation. *Computer methods in applied mechanics and engineering*, 184:171–211, 2000.

[4] B. Després. Domain decomposition method and the Helmholtz problem.II. In *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993)*, pages 197–206, Philadelphia, PA, 1993. SIAM.

[5] C. Farhat, A. Macedo, M. Lesoinne, F.-X. Roux, F. Magoules, and A. de La Bourdonnaye. Two-level domain decomposition methods with lagrange multipliers for the fast iterative solution of acoustic scattering problems. *Computer Methods in Applied Mechanics and Engineering*, 184(2):213–240, 2000.

[6] C. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 2 (1), pages 1–124. North-Holland, 1994.

[7] M. J. Gander, L. Halpern, and F. Nataf. Optimized Schwarz methods. In T. Chan, T. Kako, H. Kawarada, and O. Pironneau, editors, *Twelfth International Conference on Domain Decomposition Methods, Chiba, Japan*, pages 15–28, Bergen, 2001. Domain Decomposition Press.

[8] M. J. Gander, F. Magoulès, and F. Nataf. Optimized schwarz method without overlap for the helmholtz equation. *SIAM Scientific Computing*, 2002.

[9] S. Ghanemi. A domain decomposition method for Helmholtz scattering problems. In P. E. Bjørstad, M. Espedal, and D. Keyes, editors, *Ninth International Conference on Domain Decomposition Methods*, pages 105–112. ddm.org, 1997.

[10] M. J. Grote and T. Huckle. Parallel preconditioning with sparse approximate inverses. *SIAM Scientific Computing*, 1996.

Part VI

Mini Symposium: The Method of Subspace Corrections for Linear and Nonlinear Problems

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

30. On multigrid methods for vector–valued Allen–Cahn equations with obstacle potential

R. Kornhuber[1], R. Krause [2]

1. Introduction. Phase field models provide a well–established framework for the mathematical description to free boundary problems for phase transitions. In contrast to sharp interface models, the phase field approach postulates a diffuse interface with a small but finite thickness. Approximations of the interface are recovered as level sets of a function u, called order parameter or phase field. The main advantage of this approach is that topological changes of the approximate interface cause no problems, because bulk phases and interface are treated in the same manner. In this paper, we consider multicomponent phase transitions as described by a vector–valued Allen–Cahn equation with obstacle potential [2, 3]. Semi–implicit discretization in time is unconditionally stable but, after finite element discretization in space, leads to large non–smooth algebraic systems. So far, fast solvers for such kind of problems were not available. As a consequence, explicit schemes are applied, in spite of severe stability restrictions on the time step [4]. We present a new class of multigrid methods based on successive minimization in the direction of well selected search directions and prove global convergence. Similar multigrid techniques have been applied in [6, 8] in a different context. Numerical experiments illustrate the reliability and efficiency of our method.

2. Vector–valued Allen–Cahn equations and discretization. We consider isothermal, multicomponent phase transitions in a polygonal (polyhedral) domain $\Omega \subset \mathbb{R}^d$, $d = 1, 2, 3$. Each phase at a particular point $(x, t) \in Q = \Omega \times [0, T_0]$, $T_0 > 0$, is represented by the value of a component $u_i(x, t)$ of the order parameter $u = (u_1, \ldots, u_N)^T$. In practical applications the components u_i may represent concentrations or volume fractions of the different phases in the system. Hence, we impose the condition that values of u_i are non–negative and add up to unity [3], i.e.

$$u(x,t) \in G = \{v \in \mathbb{R}^N \mid v_i \geq 0,\ \textstyle\sum_i v_i = 1\} \qquad \forall (x,t) \in Q.$$

We further assume that the Ginsburg–Landau total free energy of our system is given by

$$\mathcal{E}(u) = \int_\Omega \tfrac{1}{2}\varepsilon^2 \textstyle\sum_i |\nabla u_i|^2 + \Psi(u)\, dx, \qquad \varepsilon > 0.$$

The quadratic term describes interfacial energy and the non–convex free energy functional Ψ has N distinct local minima on G giving rise to phase separation. Phase kinetics should satisfy the second law of thermodynamics stating that total free energy is non–increasing along solution paths. The vector–valued Allen–Cahn equation

$$u_t = -\frac{d}{du}\mathcal{E}(u) = \varepsilon^2 \Delta u - T \nabla_u \Psi(u) \tag{2.1}$$

is the most simple model with this property. Denoting $\mathbf{1} = (1, 1, \ldots, 1) \in \mathbb{R}^N$, the projection $T : \mathbb{R}^N \to \Sigma_0 = \{v \in \mathbb{R}^N \mid \sum_i v_i = 0\}$, defined by

$$Tv = v - \tfrac{1}{N}(v \cdot \mathbf{1})\mathbf{1},$$

accounts for the fact that the values $u(x, t) \in G \subset \Sigma = \{v \in \mathbb{R}^N \mid \sum_i v_i = 1\}$ must only vary on the affine hyperplane Σ. See [3] for details.

[1]FU Berlin, kornhuber@math.fu-berlin.de
[2]FU Berlin, krause@math.fu-berlin.de

From now on, we concentrate on the obstacle potential $\Psi = \Psi_\infty$,

$$\Psi_\infty(u) = \begin{cases} \sum_{i<j} u_i u_j, & u \in G, \\ +\infty, & \text{else.} \end{cases}$$

Minimal values of Ψ_∞ on G are attained at the N unit vectors $e_1, \ldots, e_N \in \mathbb{R}^N$ which are associated with pure phases. Imposing Neumann boundary conditions, a weak formulation of (2.1) takes the form

$$\frac{d}{dt}(u,v) + \varepsilon^2(\nabla u, \nabla(v-u)) - (u, v-u) \geq -\tfrac{1}{N}(\mathbf{1}, v-u) \qquad \forall v \in \mathcal{G}, \tag{2.2}$$

where the time derivative is understood in an appropriate weak sense and

$$u(\cdot,t) \in \mathcal{G} := \{v \in H^1(\Omega)^N \mid v(x) \in G \text{ a.e. in } \Omega\}, \quad 0 < t \leq T_0.$$

In addition, we prescribe initial conditions $u(\cdot,0) = u_0 \in \mathcal{G}$.

Let $\mathcal{T}_J$ be a given partition of $\overline{\Omega}$ into triangles (tetrahedra) with minimal diameter $h_J = \mathcal{O}(2^{-J})$. The set of vertices is denoted by $\mathcal{N}_J$ and we set

$$\mathcal{S}_J = \{v \in C(\overline{\Omega}) \mid v_i|_t \text{ is linear } \forall t \in \mathcal{T}_J\}.$$

Now we discretize (2.2) in time by backward Euler with step size $\tau > 0$. The concave part $-(u,\cdot)$ of Ψ_∞ is taken explicitly (cf. e.g. [1]). Discretization in space by piecewise linear finite elements then leads to the discrete variational inequality

$$\begin{aligned} u_{J,k} \in \mathcal{G}_J: \quad \langle u_{J,k}, v - u_{J,k}\rangle + \tau\varepsilon^2(\nabla u_{J,k}, \nabla(v-u_{J,k})) \geq \\ \langle (1+\tau)u_{J,k-1} - \tfrac{\tau}{N}\mathbf{1}, v - u_{J,k}\rangle \quad \forall v \in \mathcal{G}_J \end{aligned} \tag{2.3}$$

to be solved in the k–th time step. Here, $\langle\cdot,\cdot\rangle$ stands for the lumped L^2–product and the continuous constraints $\mathcal{G}$ are approximated by

$$\mathcal{G}_J = \{v \in \mathcal{S}_J^N \mid v(p) \in G \; \forall p \in \mathcal{N}_J\}. \tag{2.4}$$

As $\mathcal{G}_J$ is a non–empty, closed, convex subset of $\mathcal{S}_J^N$ and the bilinear form appearing on the left hand side of (2.3) is symmetric, positive definite on $\mathcal{S}_J^N$ there is a unique solution $u_{J,k}$ for arbitrary step size $\tau > 0$, see [5].

3. Polygonal relaxation. We now derive a Gauß–Seidel type relaxation scheme for discrete variational inequalities of the form

$$u_J \in \mathcal{G}_J: \quad a(u_J, v - u_J) \geq \ell(u_J, v - u_J) \quad \forall v \in \mathcal{G}_J \tag{3.1}$$

with a symmetric, positive definite bilinear form $a(\cdot,\cdot)$ on $\mathcal{S}_J^N$, $\ell \in (\mathcal{S}_J^N)'$ and $\mathcal{G}_J$ defined in (2.4). Of course, (2.3) is a special case of (3.1).

Note that $\mathcal{G}_J$ is a subset of an affine subspace of $\mathcal{S}_J^N$ spanned by the hyperplane $\mathcal{H}_J = \{v \in \mathcal{S}_J^N \mid \sum_j v_j(p) = 0\}$. Hence, each splitting of $\mathcal{H}_J$ gives rise to a successive subspace correction method for (3.1). We consider the splitting

$$\mathcal{H}_J = \sum_{l=1}^{m_J} V_l, \quad V_l = \operatorname{span}\{\mu_l\}, \qquad \mu_{l(i,j)} = \lambda_{p_i}^{(J)} E_j, \;\; l = 1, \ldots, m_J, \tag{3.2}$$

where $\lambda_{p_i}^{(J)}$, $i = 1, \ldots, n_J$, denotes the nodal basis of $\mathcal{S}_J$, the vectors $E_j \in \mathbb{R}^N$, $i = 1, \ldots, M := \frac{1}{2}N(N-1)$ are given by the edges of G, $l = l(i,j)$ is some enumeration and $m_J := n_J M$.

The resulting successive subspace correction method reads as follows. Starting with the given ν–th iterate $u_J^\nu =: w_0^\nu \in \mathcal{G}_J$, we compute a sequence of intermediate iterates $w_l^\nu = w_{l-1}^\nu = v_l^*$, $l = 1, \ldots, m_J$. The corrections v_l^* are the unique solutions of the local subproblems

$$v_l^* \in \mathcal{D}_l^* \quad a(v_l^*, v - v_l^*) \geq \ell(v - v_l^*) - a(w_{l-1}^\nu, v - v_l^*) \quad \forall v \in \mathcal{D}_l^*, \tag{3.3}$$

where the closed convex subsets $\mathcal{D}_l^* = \mathcal{D}_l^*(w_{l-1}^\nu)$ are defined by

$$\mathcal{D}_l^*(w_{l-1}^\nu) = \{v \in V_l \mid w_{l-1}^\nu + v \in \mathcal{G}_J\}.$$

Finally, we obtain the next iterate $u_J^{\nu+1}$,

$$u_J^{\nu+1} = \mathcal{M}(u_J^\nu) := w_{m_J}^\nu = u_J^\nu + \sum_{l=1}^{m_J} v_l^*. \tag{3.4}$$

It is well–known from, e.g., [5] that (3.1) is equivalent to the constrained minimization problem

$$u_J \in \mathcal{G}_J : \quad \mathcal{J}(u_J) \leq \mathcal{J}(v) \quad \forall v \in \mathcal{G}_J \tag{3.5}$$

for the quadratic energy functional

$$\mathcal{J}(v) = \tfrac{1}{2} a(v, v) - \ell(v), \quad v \in \mathcal{S}_J^N.$$

Successive subspace correction (3.4) can be regarded as a successive minimization of $\mathcal{J}$ in the direction of μ_l, $l = 1, \ldots, m_J$. In particular, we have

$$\mathcal{J}(u_J^{\nu+1}) \leq \mathcal{J}(w_l^\nu) \leq \mathcal{J}(u_J^\nu), \qquad \forall l = 1, \ldots, m_J, \;\; \nu = 0, 1, \ldots \; . \tag{3.6}$$

The following lemma is crucial for the convergence of (3.4).

Lemma 3.1 *For any given $U, W \in G$ there is a decomposition*

$$W = U + \sum_{j=1}^{M} \omega_j E_j, \tag{3.7}$$

which is feasible in the sense that

$$U + \omega_j E_j \in G \qquad \forall j = 1, \ldots, M.$$

Proof. We only sketch the basic idea of the proof which is easy for $N = 2, 3, 4$, but becomes technical for arbitrary N. Let $U, W \in G$ be given. Recall that $e_1, \ldots, e_N \in \mathbb{R}^N$ denote the unit vectors in $\mathbb{R}^N$. By definition of G, there are coefficients $\alpha_1, \ldots, \alpha_N$ with the properties

$$w = \sum_{i=1}^{N} \alpha_i e_i, \qquad \alpha_i \geq 0, \quad \alpha_1 + \cdots + \alpha_N = 1. \tag{3.8}$$

Now it can be shown that the unit vectors e_i can be decomposed in such a way that insertion in (3.8) provides the desired feasible decomposition (3.7). ∎

We are ready to prove convergence.

Theorem 3.1 *For any initial iterate $u_J^0 \in \mathcal{G}_J$, the polygonal relaxation (3.4) converges to the solution u_J of (3.1).*

Proof. We only sketch the proof based on similar arguments as the proof of Theorem 2.1 in [7]. Utilizing (3.6), we have $\mathcal{J}(u_J^\nu) \le \mathcal{J}(u_J^0) < \infty$ for all $\nu \ge 0$. As a consequence, the sequence of iterates $(u_J^\nu)_{\nu\in\mathbb{N}}$ is bounded. As $\mathcal{S}_J^N$ has finite dimension, any subsequence of $(u_J^\nu)_{\nu\in\mathbb{N}}$ has a subsubsequence $(u_J^{\nu_k})_{k\in\mathbb{N}}$ that converges to some $u_J^* \in \mathcal{G}_J$. We now show that $u_J^* = u_J$. Observe that (3.6) leads to

$$\mathcal{J}(u_J^{\nu_{k+1}}) \le \mathcal{J}(\mathcal{M}(u_J^{\nu_k})) \le \mathcal{J}(u_J^{\nu_k}) \qquad \forall k = 1, \dots ,$$

where $\mathcal{M}$ is defined in (3.4). As $\mathcal{J}$, $\mathcal{M}$ are continuous on $\mathcal{G}_J$, we can pass to the limit in order to obtain

$$\mathcal{J}(\mathcal{M}(u_J^*)) = \mathcal{J}(u_J^*).$$

Hence, starting with $w_0 = u_J^*$, all corrections v_l^* computed from (3.3) are zero, giving

$$0 \ge \ell(v) - a(u_J^*, v) \qquad \forall v \in \mathcal{D}_l^*(u_J^*), \;\; l = 1, \dots, m_J. \tag{3.9}$$

Now, let $w \in \mathcal{G}_J$ be arbitrary chosen. As an immediate consequence of Lemma 3.1, there is a decomposition $w = u_J^* + \sum_{l=1}^{m_J} v_l$ such that $v_l \in V_l$ and $u_J^* + v_l \in \mathcal{G}_J$, i.e. $v_l \in \mathcal{D}_l^*(u_J^*)$. Inserting $v = v_l$ in (3.9) and summing up for $l = 1, \dots, m_J$, we obtain

$$a(u_J^*, w - u_J) \ge \ell(w - u_J).$$

Hence, u_J^* is a solution of (3.1). As u_J is the unique solution of (3.1), we get $u_J^* = u_J$. We have shown that any subsequence has a subsubsequence converging to u_J. Hence, the whole sequence $(u_J^\nu)_{\nu\in\mathbb{N}}$ must converge to u_J. ∎

Implementation of (3.4) is based on the representation

$$\mathcal{D}_{l(i,j)}^* = \{v \in V_l \mid v = z\lambda_{p_i}^{(J)} E_j,\ \underline{\psi}_{i,j} \le z \le \overline{\psi}_{i,j}\}$$

with local obstacles $\underline{\psi}_{i,j} \le 0 \le \overline{\psi}_{i,j}$ depending on the actual intermediate iterate w_l^ν. In contrast to box constraints, each correction $v_{l(i,j)}$ requires an update of *all* local obstacles $\underline{\psi}_{i,s}, \overline{\psi}_{i,s}$ $s = 1, \dots, M$. As a consequence, each iteration step of the polygonal relaxation requires $\mathcal{O}(M^2 n_J) = \mathcal{O}(N^4 n_J)$ point operations.

4. Extended polygonal relaxation. The convergence speed of Gauß–Seidel type relaxation (3.4) deteriorates rapidly with decreasing mesh size h_J. In order to accelerate convergence, we consider the extended splitting

$$\mathcal{H}_J = \sum_{l=1}^{m_J} V_l + \sum_{l=m_J+1}^{M_J^\nu} V_l^\nu, \qquad V_l^\nu = \operatorname{span}\{\mu_l^\nu\}, \quad \mu_l^\nu \in \mathcal{H}_J, \tag{4.1}$$

with V_l, $l = 1, \dots, m_J$, defined in (3.2). The additional search directions μ_l^ν are intended to improve the representation of the low–frequency contributions of the error and therefore should have large support. The μ_l^ν might be iteratively adjusted to the unknown solution u_J and, for this reason, are allowed to vary in each iteration step.

We consider the resulting *extended polygonal relaxation* defined as follows. Starting from a given iterate $u_J^\nu \in \mathcal{G}_J$, we first compute a *smoothed iterate* $\bar{u}_J^\nu = w_{m_J}^\nu = \mathcal{M}(u_J^\nu)$ by fine grid smoothing (3.4). Successive "coarse grid corrections" v_l are then obtained from

$$v_l \in \mathcal{D}_l : \quad a(v_l, v - v_l) \ge \ell(v - v_l) - a(w_{l-1}^\nu, v - v_l) \quad \forall v \in \mathcal{D}_l, \tag{4.2}$$

denoting $w_l^\nu = w_{l-1}^\nu + v_l$, $l = m_J + 1, \dots, M_J^\nu$. Due to large support of μ_l^ν, it might be too costly to check whether some $v \in V_l^\nu$ is contained in $\mathcal{D}_l^*$ or not. Hence, we may use approximate closed convex subsets $\mathcal{D}_l$, satisfying

$$0 \in \mathcal{D}_l \subset \mathcal{D}_l^* = \{v \in V_l^\nu \mid w_{l-1}^\nu + v \in \mathcal{G}_J\}.$$

The next iterate is given by

$$u_J^{\nu+1} = w_{M_J^\nu}^\nu = \bar{u}_J^\nu + \sum_{l=m_J+1}^{M_J^\nu} v_l. \tag{4.3}$$

The convergence proof is almost literally the same as for Theorem 2.1 in [6].

Theorem 4.1 *For any initial iterate $u_J^0 \in \mathcal{G}_J$, the extended polygonal relaxation (4.3) converges to the solution u_J of (3.1).*

The subset of all nodes with vanishing i–th phase is denoted by

$$\mathcal{N}_{J,i}^\bullet(u_J) = \{p \in \mathcal{N}_J \mid u_{J,i}(p) = 0\}, \quad i = 1, \dots, N.$$

It would be interesting to know whether

$$\mathcal{N}_{J,i}^\bullet(u_J^\nu) = \mathcal{N}_{J,i}^\bullet(u_J), \qquad \nu \geq \nu_0, \tag{4.4}$$

holds for some $\nu_0 \in \mathbb{N}$. In fact, assuming reasonable search directions μ_l^ν, a non–degeneracy condition of the form

$$a(u_J, \lambda_p^{(J)} E_j) < (-1)^{r_j} \ell(\lambda_p^{(J)} E_j) \qquad \forall j \text{ with } (e_i \cdot E_j) \neq 0 \quad \forall p \in \mathcal{N}_{J,i}^\bullet(u_J)$$

with suitable r_j depending on the orientation of E_j and finally that $u_J(p) \neq e_j$ holds for all $j \neq i$ and $p \in \mathcal{N}_{J,i}^\bullet(u_J)$, convergence of phases (4.4) can be shown in a similar way as Lemma 2.2 in [6]. Unfortunately, this result is of minor relevance for discretized vector–valued Allen–Cahn equation (2.3), because $u_J(p) = e_j$ stands for pure phase j. Recall that pure phases are local minima of Ψ_∞.

5. Monotone multigrid. Assume that $\mathcal{T}_J$ is resulting from J refinements of an intentionally coarse triangulation $\mathcal{T}_0$. In this way, we obtain a sequence of triangulations $\mathcal{T}_0 \subset \cdots \subset \mathcal{T}_J$ and corresponding nested finite element spaces $\mathcal{S}_0 \subset \cdots \subset \mathcal{S}_J$. Though the algorithms to be presented can be easily generalized to the non–uniform case, we assume for simplicity that the triangulations are uniformly refined. More precisely, each triangle $t \in \mathcal{T}_k$ is subdivided into four congruent subtriangles in order to produce the next triangulation $\mathcal{T}_{k+1}$.

Using the nodal basis functions $\lambda_p^{(k)}$, $p \in \mathcal{N}_k$ on all levels $k = J, \dots, 0$, we define the search directions μ_l^ν appearing in the splittings (3.2) and (4.1) by

$$\mu_{l(i,j,k)} = \lambda_{p_i}^{(k)} E_j, \qquad l = 1, \dots, M_J := M(n_J + \cdots + n_0).$$

The enumeration $l(i,j,k)$ is taken from fine to coarse, i.e. $l(i,j,k) > l'(i',j',k')$ implies $k \leq k'$. Approximate constraints in (4.2) have the form

$$\mathcal{D}_l = \{v = z\lambda_{p_i}^{(k)} E_j \in V_{l(i,j,k)} \mid \underline{\psi}_l \leq z \leq \overline{\psi}_l\}, \quad l = Mn_J + 1, \dots, M_J.$$

Local obstacles $\underline{\psi}_l$, $\overline{\psi}_l$ can be constructed by quasioptimal monotone restriction [6]. As a consequence of Theorem 4.1, the resulting *standard monotone multigrid method* converges for all initial iterates $u_J^0 \in \mathcal{G}_J$. It can be implemented as a multigrid V–cycle. Smoothing

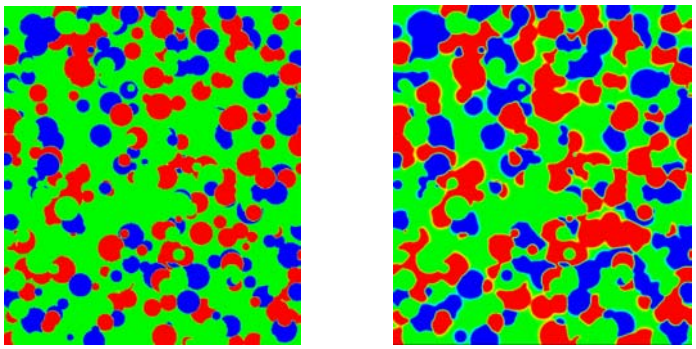


Figure 6.1: Initial condition u_0 and approximate solution at $t = 4$

is performed by polygonal relaxation (3.4) on each level. Restriction of stiffness matrix and residual and prolongation of corrections are canonical, if representation in terms of search directions $\lambda_p^{(k)} E_j$ is used. The numerical complexity of each iteration step is $\mathcal{O}(N^4 n_J)$, i.e. of the same order as fine–grid smoothing. Asymptotic multigrid convergence rates could be derived in the framework of linear successive subspace correction (cf. [6, 9]), provided that convergence of phases (4.4) holds for all $i = 1, \ldots, N$.

In related algorithms, convergence speed of standard monotone multigrid could be improved by so–called truncation of coarse grid nodal basis functions [6, 7, 8]. In the present case, truncation leads to the coarse grid search directions

$$\tilde{\mu}^\nu_{l(i,j,k)} = T^\nu_{J,k,j} \lambda^{(k)}_{p_i} E_j, \qquad l = Mn_J + 1, \ldots, M_J.$$

For each direction E_j, the truncation operators $T^\nu_{J,k,j} : \mathcal{S}_J \to \mathcal{S}_k$ are defined according to [6]. Truncation is implemented by modification of quasioptimal restriction and canonical restriction and prolongation: All entries from $\mathcal{N}^\bullet_{J,i}(\bar{u}^\nu_J)$ are set to zero. In this way, we obtain a *truncated monotone multigrid method*. Again, convergence follows from Theorem 4.1 and asymptotic multigrid convergence rates could be derived, if all phases $i = 1, \ldots, N$ converge according to (4.4). Mesh independent *global bounds* for convergence rates of monotone multigrid methods, e.g. from [6], are still an open problem.

6. Numerical experiments. We consider grain growth as described by the vector–valued Allen–Cahn equation (2.2) on the unit square $\Omega = (0, 1) \times (0, 1)$ with $N = 3$ and $\varepsilon = 0.002$. For example, each of the $N = 3$ different phases may reflect a different crystalline structure. The initial condition $u_0 \in \mathcal{G}$ is a randomly chosen superposition of 500 circular grains, each of which corresponds to a pure phase. The randomly chosen radii are ranging from 0.01 to 0.04. See the left picture in Figure 6.1 for illustration.

The continuous problem is approximated by the discretization (2.3) with step size $\tau = 1$ and triangulation $\mathcal{T}_J$ resulting from $J = 8$ uniform refinements. The initial triangulation $\mathcal{T}_0$ is obtained by subdivision of Ω into two congruent triangles and a subsequent refinement step. The right picture in Figure 6.1 and Figure 6.2 show the approximate discrete solution at $t = 4$, $t = 100$ and $T_0 = 600$, respectively. Observe that reduction of total free energy goes with a reduction of the (diffuse) interfaces by smoothing and coarsening. Interfaces at triple junctions tend to meet at an angle of 120°. This supports formal asymptotic analysis in [2]. In order to illustrate the convergence behavior of our iterative schemes, we consider the spatial problem to be solved in the first time step. The left picture in Figure 6.3 shows the iteration history of polygonal relaxation (cf. Section 3) as compared to the standard monotone multigrid method with V–cycle and three pre–smoothing and post–smoothing

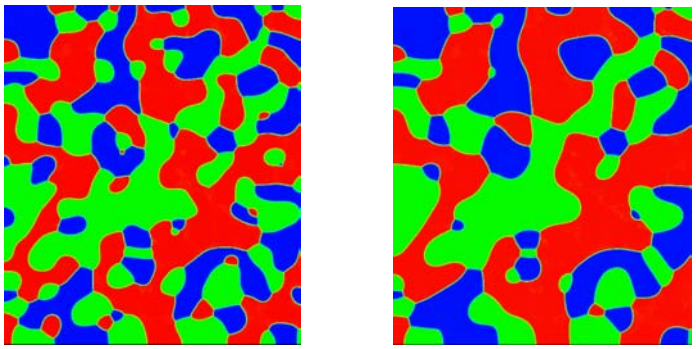


Figure 6.2: Approximate solutions at $t = 100$ and $T_0 = 600$

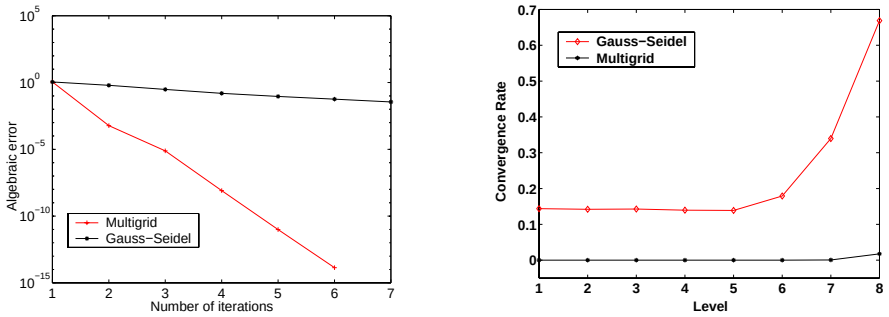


Figure 6.3: Iteration history and averaged convergence rates

steps, respectively (cf. Section 5). The algebraic error $\|u_J - u_J^\nu\|$ is measured by the energy norm $\|\cdot\| = a(\cdot,\cdot)^{1/2}$. The initial iterate $u_J^0 = e_3 \in \mathcal{G}_J$ has little to do with u_J. Nevertheless, we observe very fast convergence of our multigrid method throughout the iteration process. The averaged convergence rate is $\rho_J^{\mathrm{STD}} := \sqrt[\nu_0]{\|u_J - u_J^{\nu_0}\|/\|u_J - u_J^0\|} \approx 0.005$ where ν_0 is chosen such that $\|u_J - u_J^{\nu_0}\| < 10^{-12}$. Taking into account that each iteration step is much cheaper, polygonal relaxation performs reasonably well with averaged convergence rate $\rho_J^{\mathrm{GS}} = 0.56$. This seems to be a consequence of the redundancy of search directions in combination with a moderate number of grid points in the diffuse interface. The right picture in Figure 6.3 illustrates the mesh dependence of averaged convergence rates $\rho_j^{\mathrm{GS}}, \rho_j^{\mathrm{STD}}$, $j = 0, \ldots, 8$. Iteration always starts with the "arbitrary" initial iterate $u_j^0 = e_3 \in \mathcal{G}_j$. As expected, we observe only minor sensitivity of multigrid as compared to single grid. On the other hand, it seems that the mesh size $h_J = 2^{-9} \approx \frac{1}{2}\varepsilon$ is still too large to provide saturation.

7. Conclusion an perspective. We have introduced and analyzed new Gauß–Seidel type relaxation and monotone multigrid methods for systems of variational inequalities with local triangular constraints. Such problems arise in mathematical description of certain free boundary problems by phase field models. Future work will concentrate on more realistic Ginzburg–Landau functionals, involving anisotropic interfacial energy and logarithmic free energy [4]. Of course, adaptive mesh refinement will be indispensable for a better resolution of the diffuse interface. In this case, truncated multigrid might also be profitable.

REFERENCES

[1] J. Blowey and C. Elliott. The Cahn–Hilliard gradient theory for phase separation with non–smooth free energy. *Part II: Numerical analysis*, 3:147–179, 1992.

[2] L. Bronsard and F. Reitich. On three–phase boundary motion and the singular limit of a vector–valued Ginzburg–Landau equation. *Arch. Rational Mech. Anal.*, 124:355–379, 1993.

[3] H. Garcke, B. Nestler, and B. Stoth. On anisotropic order parameter models for multi–phase systems and their sharp interface limits. *Physica D*, 115:87–108, 1998.

[4] H. Garcke, B. Nestler, and B. Stoth. Anisotropy in multi phase systems: A phase field approach. *Interfaces Free Bound.*, 1:175–198, 1999.

[5] R. Glowinski. *Numerical Methods for Nonlinear Variational Problems*. Springer Verlag, New York, 1983.

[6] R. Kornhuber. Monotone multigrid methods for elliptic variational inequalities I. *Numer. Math*, 69:167–184, 1994.

[7] R. Kornhuber. *Adaptive Monotone Multigrid Methods for Nonlinear Variational Problems*. Teubner, Stuttgart, 1997.

[8] R. Kornhuber and R. Krause. Adaptive multilevel methods for Signorini's problem in linear elasticity. *Comp. Visual. Sci.*, 4:9–20, 2001.

[9] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, December 1992.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

31. Successive Subspace Correction method for Singular System of Equations

Young-Ju Lee[1], Jinchao Xu[2], Ludmil Zikatanov[3] [4]

1. Introduction. The method of successive subspace corrections, an abstraction of general iterative methods such as multigrid and Multiplicative Schwarz methods, is an algorithm for finding the solution of a linear system of equations. In this paper, we shall study in particular, Multiplicative Schwarz methods in a Hilbert space framework and present a sharp result on the convergence of the methods for singular system of equations.

For the symmetric positive definite (SPD) problems, a variety of literatures on the convergence analysis are available. Among others, we would like to refer to the upcoming paper by Xu and Zikatanov (Refer to [3]). In [3], the convergence rate of the method of subspace corrections has been beautifully established by introducing a new identity for the product of nonexpansive operators.

The main result in this paper is in that we obtained an appropriate identity for the non-SPD problems, which is suitably applied to devise or improve algorithms for singular and especially nearly singular system of equations. The related results and the corresponding estimate of the convergence rate of multigrid methods for singular system of equations shall be reported in the forthcoming paper.

The rest of this paper is organized as follows. In section 2, we set up a problem and review a successive subspace correction method in a Hilbert space setting. In section 3, we estabilsh the convergence factor of the algorithm and present an identity for the convergence rate of the method of successive subspace correction for singular system of equations. In section 4, we adapt our identity for Multiplicative Schwarz method and present various identities for the special algorithm such as Gauss-Seidel and Block Gauss-Seidel method. In the final section 5, we give some concluding remarks and future works.

2. MSC: The Method of Subspace Corrections. Let V be a Hilbert space with an inner product $(\cdot,\cdot)_V = (\cdot,\cdot)$ and an induced norm $\|\cdot\|_V = \|\cdot\|$. Let V^* denote the dual space of V. We consider the following variational problem: Find $u \in V$ for any given $f \in V^*$ such that

$$a(u,v) = \langle f, v\rangle \quad \forall v \in V \tag{2.1}$$

where $\langle\cdot,\cdot\rangle$ is a dual paring and $a(\cdot,\cdot)$ is a *symmetric* and *nonnegative* definite bilinear form satisfying $a(u,v) \le \|a\|\|u\|\|v\|$ where $\|a\| > 0$ is a constant. We shall define $\mathcal{N}$ and $\mathcal{N}^\circ$ by $\mathcal{N} = \{v \in V : a(v,w) = 0 \quad \forall w \in V\}$ and $\mathcal{N}^\circ = \{f \in V^* : \langle f, v\rangle = 0 \quad \forall v \in \mathcal{N}\}$ respectively. The latter is often called the polar set of $\mathcal{N}$. By usual convention, for any set $W \subset V$, $W^\perp$ shall denote the orthogonal complement of W with respect to the inner product, $(\cdot,\cdot)_V$. Throughout this paper, we shall assume that $f \in \mathcal{N}^\circ$ and the continuous bilinear form $a(\cdot,\cdot) : V \times V \mapsto \mathbb{R}$ satisfies the following coercivity conditions on $\mathcal{N}^\perp$, namely: There exists a constant $\alpha > 0$ such that $a(v,v) \ge \alpha\|v\|^2$. This assumption implies that the problem (2.1) is well-posed on $\mathcal{N}^\perp$. We would like to remark that the problem (2.1) is not well-posed on V in a sense that it has infinitely many solutions, namely if u is a solution to (2.1), then $u + c$ will be again a solution to the problem for any $c \in \mathcal{N}$.

Now we shall discuss the method of successive subspace correction for solving 2.1. The idea of the method of successive subspace correction is to solve the residual equation on some

[1]The Pennsylvania State University, lee_y@math.psu.edu
[2]The Pennsylvania State University, xu@math.psu.edu
[3]The Pennsylvania State University, ltz@math.psu.edu
[4]This work is supported by National Scientific Foundation Grant No. DMS-0074299

properly chosen subspaces. A decomposition of V consists of a number of closed subspaces $V_i \subset V, (1 \le i \le J)$ satisfying $V = \sum_{i=1}^{J} V_i$.

Associated with each subspaces V_i, we introduce a continuous bilinear form $a_i(\cdot,\cdot)$ which can be viewed as an approximation of $a(\cdot,\cdot)$ restricted on V_i. We shall assume that the following inf-sup conditions are satisfied for all $i = 1, 2, ..., J$,

$$\inf_{v_i \in V_i} \sup_{w \in V_i} \frac{a_i(v_i, w_i)}{\|v_i\|\|w_i\|} = \inf_{w_i \in V_i} \sup_{v \in V_i} \frac{a_i(v_i, w_i)}{\|v_i\|\|w_i\|} = \alpha_i > 0 \tag{2.2}$$

and for all $i = 1, 2, ..., J$, there exists $\beta_i > 0$ such that

$$a(v_i, v_i) \ge \beta_i \|v_i\|^2 \quad \forall v_i \in V_i. \tag{2.3}$$

These inf-sup conditions are often known as Babuska-Brezzi conditions or B-B conditions. (See e.g. [4]) This is equivalent to say that the approximate subspace problems and subspace problems are uniquely solvable. While we can not in general impose the inf-sup condition for $a(\cdot,\cdot)$ on V_i due to the fact that V_i may contain a non trivial subspace of $\mathcal{N}$. In this paper, we shall assume that $a(\cdot,\cdot)$ satisfies the B-B conditions since we are mainly concerned with Multiplicative Schwarzs methods.

2.1. SSC: Successive Subspace Corrections.. The method of successive subspace corrections (MSSC) is an iterative algorithm that corrects residual equation successively on each subspace.

ALGORITHM[MSSC] Let $u^0 \in V$ be given.

for $l = 1, 2, ...$
  $u_0^{l-1} = u^{l-1}$
  for $i = 1 : J$
    Let $e_i \in V_i$ solve
      $a_i(e_i, v_i) = f(v_i) - a(u_{i-1}^{l-1}, v_i) \quad \forall v_i \in V_i$
    $u_i^{l-1} = u_{i-1}^{l-1} + e_i$
  endfor
  $u^l = u_J^{l-1}$
endfor

We note that the above algorithm is well-defined, thanks to the inf-sup conditions for (2.2). For the analysis of this algorithm, let us introduce another class of linear operators $T_i : V \mapsto V_i$ defined by $a_i(T_i v, v_i) = a(v, v_i) \quad \forall v_i \in V_i$. Again, thanks to inf-sup condition (2.2), each T_i is well-defined and $\mathcal{R}(T_i) = V_i$. In the special case when the subspace equation is solved exactly, we shall use the notation P_i for T_i, namely $T_i = P_i$ if $a_i(\cdot,\cdot) = a(\cdot,\cdot)$.

It is easy to see that for given $u \in V$ a solution to (2.1),

$$u - u_i^{l-1} = (I - T_i)(u - u_{i-1}^{l-1}).$$

By a simple recursive application of the above identity, we obtain that

$$u - u^l = E_J(u - u^{l-1}) = ... = E_J^l(u - u^0) \tag{2.4}$$

where

$$E_J = (I - T_J)(I - T_{J-1}) \cdots (I - T_1). \tag{2.5}$$

which is often called an error transfer operator. Because of this special form of E_J, the successive subspace correction method is also known as the Multiplicative or Product (Schwarz) method. The general notion of subspace corrections by means of space decomposition was described in Xu[2].

3. An identity for the convergence factor of MSSC. In view of (2.4), the convergence of the method of subspace correction is equivalent to $\lim_{l\to\infty} E_J^l = 0$. As was discussed before in this paper, for the case when $a(\cdot,\cdot) : V \times V \mapsto \mathbb{R}$ is a symmetric positive definite bilinear form, the uniform convergence result under some natural conditions on the subspace solvers T_i was established as an identity for the convergence factor $\|E_J\|_a = \sup_{\|v\|_a=1} \|E_J v\|_a$, namely the norm of the product of nonexpansive operators. (Refer to [3].) In our case when $a(\cdot,\cdot)$ is *nonnegative* definite, two types of convergences can be considered, namely the classical convergence (or norm convergence in the space V):

$$\|u^l - u\|_V \to 0 \text{ as } l \to \infty$$

and quotient norm or energy norm convergence (Refer to [1]):

$$\|u^l - u\|_{V/\mathcal{N}} \to 0 \text{ as } l \to \infty,$$

where $V/\mathcal{N}$ is the quotient space. We shall present that the following quantity is both the norm and the quotient norm convergence factor for the MSSC under some suitable conditions.
DEFINITION[Convergence Factor]

$$\|E_J\|_{\mathcal{L}(\mathcal{N}^\perp,V)_a} = \sup_{v\in\mathcal{N}^\perp} \frac{|E_J v|_a}{\|v\|_a}$$

In the sequel of this paper, we shall establish an identity for the convergence factor $\|E_J\|_{\mathcal{L}(\mathcal{N}^\perp,V)_a}$ under certain assumptions.

3.1. Assumptions on subspace solvers. We shall now try to derive conditions on the subspaces and subspace solvers for the convergence of the MSSC.

First of all, we shall assume that

ASSUMPTION[$A0$] A decomposition of V consists of closed subspaces $V_i \subset V$, $i = 1, 2, ..., J$ satisfying

$$V = \sum_{i=1}^{J} V_i.$$

This assumption is necessary for any quantitative convergence even for SPD problems. (See [3] page 15.)

ASSUMPTION[$A1$] There exists $\alpha_i > 0$ such that

$$a(v_i, v_i) \geq \alpha_i \|v_i\|^2 \quad \forall v_i \in V_i$$

$$\inf_{v_i\in V_i} \sup_{w\in V_i} \frac{a_i(v_i, w_i)}{\|v_i\|\|w_i\|} = \inf_{w_i\in V_i} \sup_{v\in V_i} \frac{a_i(v_i, w_i)}{\|v_i\|\|w_i\|} = \beta_i > 0$$

This assumption implies that the subspace problems are well-posed and that $T_i : V_i \mapsto V_i$ is isomorphic for each $i = 1, 2, 3, ..., J$.

ASSUMPTION[$A2$] For each $1 \leq i \leq J$, there exists $\omega \in (0, 2)$ such that

$$a(T_i v, T_i v) \leq \omega a(T_i v, v) \quad \forall v \in V.$$

Let us discuss the assumption ($A1$) briefly for the finite dimensional case. For the notational simplicity and invoking Riesz Representation theorem (See e.g. [5] (e.g. $a(\cdot,\cdot) \Leftrightarrow A$ and $a_i(\cdot,\cdot) \Leftrightarrow R_i$), let us put the discretization of the system of equation (2.1) as following operator equation: Find $u \in \mathbb{R}^n$ such that

$$Au = b$$

$\mathcal{R}(A)$ and $\mathcal{N}(A)$ denote the range of A and kernel of A respectively. The iterative method is based on the classical matrix splitting as follows:

$$A = D - L - L^T$$

where D is the diagonal and L is the strictly lower triangular matrix. In this situation, one can easily show that the sufficient condition that $(A1)$ holds true is that A has a positive diagonal and the symmetric part of the approximate subspace operators, say R_i to A is positive definite.

Remark 3.1 *For the case multigrid method with nested subspaces, with the assumption that $a(\cdot,\cdot)$ satisfies the inf-sup condtion on $R(T_i)$, an appropriate identity can be deriven.*

3.2. On the Convergence factor of the MSSC. In this subsection, we shall see that the convergence factor of the MSSC is given by $\|E_J\|_{\mathcal{L}(\mathcal{N}^\perp,V)_a}$ as mentioned before. Let us begin with simple but important lemma.

Lemma 3.1 *Let $E_J = (I - T_J)\cdots(I - T_1)$. Then*

$$a(E_J v, E_J v) \le \|E_J\|_{\mathcal{L}(\mathcal{N}^\perp,V)_a} a(v,v) \quad \forall v \in V$$

The following lemma and the theorem shall reveal that $\|E_J\|_{\mathcal{L}(\mathcal{N}^\perp,V)_a}$ is indeed both the norm and quotient norm convergence rate of the MSSC.

Theorem 3.1 *Assume $(A1)$ and $(A2)$. Then for any initial guess $u^0 \in V$, the followings hold true:*

$$\|u - u^k\| \le C\|E_J\|^{k-1}_{\mathcal{L}(\mathcal{R}A,V)_a}\|u - u^{k-1}\|$$

and

$$\|u - u^k\|_{V/\mathcal{N}} \le C\|E_J\|^{k}_{\mathcal{L}(\mathcal{R}A,V)_a}\|u - u^{k-1}\|_{V/\mathcal{N}},$$

where u is a solution to (2.1).

3.3. An identity for the convergence factor for the MSSC. We are in a position to present the identity for the convergence factor for the MSSC. The theorem presented below is based on the aforementioned assumptions $(A0), (A1)$ and $(A2)$. Let us first introduce an operator $Q_A : V \mapsto \mathcal{N}^\perp$ defined by $\forall v \in V$ and $\forall w \in \mathcal{N}^\perp$, $(Q_A v, w) = (v, w)$ and define $Q_{i,A}$ by the restriction of Q_A on $\mathcal{R}(T_i) = V_i$. We also denote a space $Q_A W$ for any set $W \subset V$ by $Q_A W = \{Q_A w \in V : w \in W\}$. We shall also introduce linear operators $T_{i,A} : V \mapsto V$ defined by $T_{i,A} = Q_{i,A} T_i$.

Lemma 3.2 *Let us define $E_{J,A}$ by $E_{J,A} = (I - T_{J,A})\cdots(I - T_{1,A})$. Then,*

$$\|E_J\|_{\mathcal{L}(\mathcal{N}^\perp,V)_a} = \|E_{J,A}\|_a = \sup_{v\in\mathcal{N}^\perp} \frac{\|E_{J,A}v\|_a}{\|v\|_a}$$

Proof. The proof is completed by the simple observation that

$$|E_J v|_a^2 = \|Q_A E_J v\|_a^2 = \|E_{J,A} v\|_a^2.$$

■

We would like to remark that we use the notation $\|\cdot\|_a$ rather than $|\cdot|_a$. This is because $E_{J,A}$ is invariant operator on $\mathcal{N}^\perp$ and $a(\cdot,\cdot)$ is SPD on $\mathcal{N}^\perp$. We shall use this rule in the sequel of this paper if no confusion arises.

In view of the lemma (3.2), the convergence factor for the MSSC is transformed into the norm of a product of nonexpansive operators on $\mathcal{N}^\perp$. Now, by this observation, the acquisition of an identity for the convergence factor of the MSSC is in showing the three assumptions on $T_{i,A}$'s (Refers to [3]) under which we can apply the known theory in Xu and Zikatanov [3] and obtain the desired result.

Lemma 3.3 *Assume* $(A0), (A1)$ *and* $(A2)$*. Then the followings hold true.*

- *Each* $\mathcal{R}(T_{i,A}) = Q_A\mathcal{R}(T_i)$ *is closed and* $Q_{i,A} : \mathcal{R}(T_i) \mapsto \mathcal{R}(T_{i,A})$ *is an isomorphism.*
- *Each* $T_{i,A} : \mathcal{R}(T_{i,A}) \mapsto \mathcal{R}(T_{i,A})$ *is an isomorphism.*
- *The following holds true: for each* $1 \leq i \leq J$*, there exists* $\omega \in (0, 2)$ *such that*

$$a(T_{i,A}v, T_{i,A}) \leq \omega a(T_{i,A}v, v) \quad \forall v \in V.$$

- $\mathcal{N}^{\perp} = \sum_{i=1}^{J} \mathcal{R}(T_{i,A})$.

Theorem 3.2 *Under the assumptions* $(A0), (A1)$ *and* $(A2)$*, we obtain the following identity:*

$$\|E_J\|_{\mathcal{L}(\mathcal{N}^{\perp},V)_a} = \|E_{J,A}\|_a = \frac{c_0}{1+c_0}$$

where

$$c_0 = \sup_{v\in\mathcal{N}^{\perp}} \inf_{\sum_{i=1}^{J} T_{i,A}v_i = v} \frac{\sum_{i=1}^{J} (\bar{T}_{i,A}^{-1} T_{i,A}^{*} u_i, T_{i,A}^{*} u_i)_a}{(v,v)_a}$$

, $u_i = \sum_{j=i}^{J} T_{j,A}v_j - v_i$ *and* $\tilde{v}^T = (v_1, ..., v_J) \in \mathcal{R}(T_{1,A}) \times \cdots \times \mathcal{R}(T_{J,A})$.

Proof. From the lemma (3.3) and by applying the main result theorem 4.2 (page 10) in [3], the proof is completed. ∎

We would like to point out that c_0 is a bit different from that given in [3]. One can obtain this by the following simple change of variable: $T_{i,A}v_i \leftrightarrow v_i$.

4. Multiplicative Schwarz Method. We shall devote this section to write the expression c_0 in terms of the real subspace solvers T_i instead of $T_{i,A}$. We shall first discuss an adjoint operator of T_i.

4.1. On the adjoint operator T_i^*. It is easy to see that it is not possible to define a unique adjoint of T_i with respect to $a(\cdot,\cdot)$ in a classical sense due to the fact that $a(\cdot,\cdot)$ is semi definite. While this is the fact, we shall see that we need to define the adjoint of T_i in some sense so that we can write c_0 in terms of the real subspace solvers T_i. In doing so, let us introduce another class of operators as follows: For each $1 \leq i \leq J$, we define $R_i : V_i \mapsto V_i$ and $Q_i : V \mapsto V_i$ by $(R_i v_i, w_i) = a_i(v_i, w_i)$ and $(Q_i v, w_i) = (v, w_i)$ $\forall v \in V, w_i \in V_i$ respectively. We would like to remark that by inf-sup condition (2.3), R_i is an isomorphism. We can then introduce the adjoint T_i^* and symmetrization $\bar{T}_i$ of T_i as follows:

$$T_i^* = R_i^T Q_i A \text{ and } \bar{T}_i = T_i + T_i^* - T_i^* T_i.$$

where R_i^T is the adjoint of R_i with respect to $(\cdot,\cdot)_V$. We here point out that T_i^* satisfies

$$a(T_i v, w) = a(v, T_i^* w) \quad \forall v, w \in V.$$

Correspondingly, we also define $T_{i,A}^*$ and $\bar{T}_{i,A}$ by

$$T_{i,A}^* = Q_{i,A}^* T_i^* \text{ and } \bar{T}_{i,A} = T_{i,A} + T_{i,A}^* - T_{i,A}^* T_{i,A}.$$

where $Q_{i,A}^*$ is the restriction of Q_A on $\mathcal{R}(T_i^*)$. Note that $Q_{i,A} = Q_{i.A}^*$ if $\mathcal{R}(T_i) = \mathcal{R}(T_i^*)$.

Lemma 4.1 *Assume that* $(A1), (A2)$*. Then the followings hold true:*

- $\mathcal{R}(T_i) = \mathcal{R}(T_i^*) = \mathcal{R}(\bar{T}_i) = V_i$
- T_i, T_i^* *and* $\bar{T}_i$ *are all isomorphic from* V_i *to itself.*

- *$\bar{T}_i$ is nonnegative on V and symmetric positive definite on V_i.*

Here we provide the main theorem in the paper.

Theorem 4.1 *Assume that $(A0), (A1)$ and $(A2)$. Then the convergence rate of subspace correction method above is given by the following identity.*

$$\|E_J\|_{\mathcal{L}(\mathcal{N}^\perp, V)_a} = \frac{c_0}{1+c_0}$$

where

$$c_0 = \sup_{v\in\mathcal{N}^\perp} \inf_{\sum_i v_i = v \in \mathcal{N}^\perp} \inf_{\sum_i c_i = c \in \mathcal{N}} \frac{\sum_{i=1}^J (T_i \bar{T}_i^{-1} T_i^* w_i, w_i)_a}{\|v\|_a}$$

$w_i = \sum_{j=i}^J (v_j + c_j) - T_i^{-1}(v_i + c_i)$. *and $v_i, c_i \in V_i$.*

Proof. By theorem (3.2) and simple change of variable, it is easy to see that we can write an identity for the convergence rate as follows:

$$\|E_J\|_{\mathcal{L}(\mathcal{N}^\perp, V)_a} = \frac{c_0}{c_0+1}$$

where

$$c_0 = \sup_{v\in\mathcal{N}^\perp} \inf_{\sum_{i=1}^J T_i w_i = v + c} \frac{\sum_{i=1}^J (\bar{T}_{i,A}^{-1} T_{i,A}^* u_i, T_{i,A}^* u_i)_a}{(v,v)_a}$$

with $u_i = (\sum_{j=i}^J T_j w_j - w_i)$. Let us denote $\tilde{V}$ by $V_1 \times \cdots \times V_J$ and $\tilde{v}$ by $(v_1, \cdot, \cdot, \cdot, v_J) \in \tilde{V}$. We note that since $\tilde{T} : \tilde{V} \mapsto V$ is onto, c may vary arbitrarily in $\mathcal{N}$. Let us decompose $\tilde{w} \in \tilde{V}$ as followings: $\tilde{w} = \tilde{v} + \tilde{c}$ with $\tilde{v}, \tilde{c} \in \tilde{V}$ and $\tilde{T}\tilde{v} = v$ and $\tilde{T}\tilde{c} = c$. Thanks to this decomposition, we see that

$$c_0 = \sup_{v\in\mathcal{N}^\perp} \inf_{\tilde{T}\tilde{v} + \tilde{T}\tilde{c}} = \frac{\sum_{i=1}^J (\bar{T}_{i,A}^{-1} T_{i,A}^* u_i, T_{i,A}^* u_i)_a}{(v,v)_a}$$

with $u_i = \sum_{j=i}^J T_j^*(v_j + c_j) - (v_i + c_i)$.

Now let us set

$$\begin{aligned} c_1 &= \inf_{\tilde{T}(\tilde{v}+\tilde{c})} \frac{\sum_{i=1}^J (\bar{T}_{i,A}^{-1} T_{i,A}^* u_i, T_{i,A}^* u_i)_a}{(v,v)_a} \\ c_2 &= \inf_{\tilde{T}\tilde{v}=v} \inf_{\tilde{T}\tilde{c}=c} \frac{\sum_{i=1}^J (\bar{T}_{i,A}^{-1} T_{i,A}^* u_i, T_{i,A}^* u_i)_a}{(v,v)_a} \end{aligned}$$

and we shall show that $c_1 = c_2$. It is clear that $c_1 \geq c_2$, since if

$$\tilde{w} = \arg\Big(\inf_{\tilde{T}\tilde{w}=v+c} \frac{\sum_{i=1}^J (\bar{T}_{i,A}^{-1} T_{i,A}^* u_i, T_{i,A}^* u_i)_a}{(v,v)_a}\Big) \in \tilde{V}$$

with $u_i = \sum_{j=i} T_j w_j - w_i$. We can choose any decomposition of $\tilde{w} = \tilde{v} + \tilde{c}$ such that $\tilde{T}\tilde{v} = v$ and $\tilde{T}\tilde{c} = c$ with $\tilde{v}, \tilde{c} \in \tilde{V}$. Let us show the reverse inequality. Now for any given $\tilde{v} \in \tilde{V}$, let

$$\tilde{c}(\tilde{v}) = \arg\{\inf_{\tilde{T}\tilde{c}=c} \frac{\sum_{i=1}^J (\bar{T}_{i,A}^{-1} T_{i,A}^* u_i, T_{i,A}^* u_i)_a}{(v,v)_a})\}$$

with $u_i = \sum_{i=1}^J T_j(v_j + c_j) - (v_i + c_i)$ and now again set

$$\tilde{v} = \arg\Big(\inf_{\tilde{T}\tilde{v}=v} \frac{\sum_{i=1}^J (\bar{T}_{i,A}^{-1} T_{i,A}^* u_i, T_{i,A}^* u_i)_a}{(v,v)_a}\Big)\}$$

with $u_i = \sum_{i=1}^{J} T_j(v_j + c_j(\tilde{v})) - (v_i + c_i(\tilde{v}))$. Then it is easy to see that

$$\tilde{v} + \tilde{c} = \arg\{ \inf_{\tilde{T}\tilde{v}=v} (\inf_{\tilde{T}\tilde{c}=c} \frac{\sum_{i=1}^{J} (\bar{T}_{i,A}^{-1} T_{i,A}^{*} u_i, T_{i,A}^{*} u_i)_a}{(v,v)_a})\}$$

with $u_i = (\sum_{j=i}^{J} T_j(v_j + c_j) - (v_i + c_i))$ and $\tilde{T}(\tilde{v} + \tilde{c}) = v + c$, which implies that $c_1 \leq c_2$.

Hence it has been shown that

$$c_0 \quad = \quad \sup_{v \in \mathcal{N}^{\perp}} \inf_{\tilde{T}\tilde{v}=v \in \mathcal{N}^{\perp}} \inf_{\tilde{T}\tilde{c}=c \in \mathcal{N}} \frac{\sum_{i=1}^{J} (\bar{T}_{i,A}^{-1} T_{i,A}^{*} (v_i + c_i), (v_i + c_i))_a}{(v,v)_a}$$

Finally, we insert an explicit expression for $\bar{T}_{i,A}^{-1}$ as follows and obtain:

$$a(\bar{T}_i^{-1} Q_{i,A}^{-1} Q_{i,A} T_i^{*} v, w) = a(\bar{T}_i^{-1} T_i^{*} v, w) \quad \forall v \in \mathcal{N}^{\perp} \text{ and } \forall w_i \in V_i$$

This completes the proof. ■

Let us consider some special cases : in the case we use exact solvers $T_i \Leftrightarrow P_i$, c_0 in the theorem (4.1) is given by

$$c_0 = \sup_{v \in \mathcal{N}^{\perp}} \inf_{\sum_i v_i = v \in \mathcal{N}^{\perp}} \inf_{\sum_i c_i = c \in \mathcal{N}} \frac{\sum_{i=1}^{J} |P_i(\sum_{j=i+1}^{J} (v_j + c_j)|_a^2}{\|v\|_a} \tag{4.1}$$

where $v_i, c_i \in V_i$. and in particular, for Gauss-Seidel method, c_0 is given by

$$c_0 = \sup_{v \in \mathcal{N}^{\perp}} \inf_{c \in \mathcal{N}} \frac{(S(v-c), v-c)}{(v,v)_a} \tag{4.2}$$

where $A = D - L - L^T$ and $S = L^T D^{-1} L$.

5. Conclusion and extensions. We would like to remark that we can also consider the sharp result on the convergence rate of Multigrid methods with a nested subspace decomposition by modifying the assumption $(A1)$ slightly, in which case, the subspace problems are not well-posed. The theory presented in this paper can be applied to devise algorithms for Singular system of equations and especially Nearly singular system of equations. We shall report such related and further results in the forthcoming paper.

REFERENCES

[1] J. Xu. *Theory of Multilevel Methods.* PhD thesis, Cornell University, May 1989.

[2] J. Xu. Iterative methods by space decomposition and subspace correction. *SIAM Review*, 34(4):581–613, December 1992.

[3] J. Xu and L. Zikatanov. The method of alternating projections and the method of subspace corrections in hilbert space. To Appear in J. Amer. Math. Soc., 2001.

[4] J. Xu and L. Zikatanov. Some observation on babuska brezzi conditions. To Appear in Numer. Math., 2001.

[5] K. Yosida. *Functional Analysis.* Springer-Verlag, sixth edition, 1980.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

32. Some new domain decomposition and multigrid methods for variational inequalities

Xue–Cheng Tai[1]

1. Introduction. Domain decomposition (DD) and multigrid (MG) methods are powerful iterative methods for solving some partial differential equations. Some recent progress has shown that DD and MG methods are also efficient for some nonlinear elliptic problems and some convex minimization problems [15, 14, 18, 17]. Mesh independent convergence rate has been proved and it is shown that the convergence rate for some nonlinear problems is as good as the convergence rate of the methods when they are used for the Laplace equation. In many industrial applications, we need to solve nonlinear partial differential equations and at the same time the solutions of the equations need to satisfy some constraints. For such problems, the solutions always satisfy some variational inequalities. To apply DD and MG methods for variational inequalities is a difficult task, see [1, 3, 4, 5, 7, 8, 2, 9, 12] for some literature results. It is even more difficult to analyse the convergence rate. In this work, we shall propose some new algorithms using DD and MG methods for variational inequalities and at the same it is shown that the proposed algorithms have a convergence rate that is as good as DD and MG are used for some linear elliptic equations. Another feature of our approach is that we interpret DD and MG methods as space decomposition techniques [19, 18] and our algorithms are proposed for general space decomposition techniques. Thus, the algorithms and the analysis cover both DD and MG in the same frame work. The algorithms proposed here are different from the algorithms of [16, 6, 7].

Algorithms and convergence rate analysis for DD method with a coarse mesh seem still missing in the literature. When no coarse mesh is used, DD method is essentially a block relaxation method and some results are available in the literature, see [13] for some reference. For MG method, the only uniform convergence rate estimate we know is [6, 7] which is valid in the asymptotic sense and need very special conditions.

2. Some subspace correction algorithms. Consider the nonlinear convex minimization problem

$$\min_{v\in K} F(v), \quad K \subset V \ , \tag{2.1}$$

where F is a convex functional over a reflexive Banach space V and $K \subset V$ is a nonempty closed convex subset. The norm of V will be denoted by $\|\cdot\|$. In order to solve the minimization problem efficiently, we shall decompose V and K into a sum of subspaces and subsets of smaller sizes respectively as in [10] [17]. More precisely, we decompose

$$V = \sum_{i=1}^{m} V_i, \quad K = \sum_{i=1}^{m} K_i, \quad K_i \subset V_i \subset V \ , \tag{2.2}$$

where V_i are subspaces and K_i are convex subsets. We use two constants C_1 and C_2 to measure the quality of the decompositions. First, we assume that there exits a constant $C_1 > 0$ and some operators $R_i : K \mapsto K_i,\ i = 1, 2, \cdots m$, which are generally nonlinear operators, such that the following relations are correct for all $u, v \in K$

$$u = \sum_{i=1}^{m} R_i u \ , \quad v = \sum_{i=1}^{m} R_i v, \quad \text{and} \quad \left(\sum_{i=1}^{m} \|R_i u - R_i v\|^2\right)^{\frac{1}{2}} \leq C_1 \|u - v\|. \tag{2.3}$$

[1]Department of Mathematics, University of Bergen, Johannes Brunsgate 12, 5008, Bergen, Norway, email: tai@math.uib.no, http://www.mi.ubi.no/~ tai. This work is supported by the research council of Norway and also by grants NSF ACI-0072112, NSF INT-0072863 and ONR-No0014-96-1-10277 at UCLA.

We also need to assume that there is a $C_2 > 0$ such that for any $w_i \in V, \hat{v}_i \in V_i,\ \tilde{v}_j \in V_j$ it is true that

$$\sum_{i=1}^{m}\sum_{j=1}^{m} \left|\langle F'(w_{ij}+\hat{v}_i) - F'(w_{ij}), \tilde{v}_j\rangle\right| \le C_2 \Big(\sum_{i=1}^{m} \|\hat{v}_i\|^2\Big)^{\frac{1}{2}} \Big(\sum_{j=1}^{m} \|\tilde{v}_j\|^2\Big)^{\frac{1}{2}} . \tag{2.4}$$

In the above, F' is the Gâteaux differential of F and $\langle\cdot,\cdot\rangle$ is the duality pairing between V and its dual space V', i.e. the value of a linear function at an element of V. We also assume that there exists a constant $\kappa > 0$ such that

$$\langle F'(v_1) - F'(v_2), v_1 - v_2\rangle \ge \kappa \|v_1 - v_2\|^2, \quad \forall w, v \in V\ . \tag{2.5}$$

Under the assumption (2.5), problem (2.1) has a unique solution. For some nonlinear problems, the constant κ may depend on v_1 and v_2 and our algorithms and anaylsis are still valid for such cases [18]. For a given approximate solution $u \in K$, we shall find a better solution w using one of the following two algorithms.

Algorithm 1 *Choose a relaxation parameter $\alpha \in (0, 1/m]$. Find $\hat{w}_i \in K_i$ in parallel for $i = 1, 2, \cdots, m$ such that*

$$\hat{w}_i = \arg\min_{v_i \in K_i} G(v_i) \quad \textit{with} \quad G(v_i) = F\Big(\sum_{j=1, j\ne i}^{m} R_j u + v_i\Big). \tag{2.6}$$

Set $w_i = (1-\alpha)R_i u + \alpha \hat{w}_i$ and $w = (1-\alpha)u + \alpha \sum_{i=1}^{m} \hat{w}_i$.

Algorithm 2 *Choose a relaxation parameter $\alpha \in (0, 1]$. Find $\hat{w}_i \in K_i$ sequentially for $i = 1, 2, \cdots, m$ such that*

$$\hat{w}_i = \arg\min_{v_i \in K_i} G(v_i) \quad \textit{with} \quad G(v_i) = F\Big(\sum_{j<i} w_j + v_i + \sum_{j>i} R_j u\Big) \tag{2.7}$$

where $w_j = (1-\alpha)R_j u + \alpha \hat{w}_j,\ j = 1, 2, \cdots i-1$. Set $w = (1-\alpha)u + \alpha \sum_{i=1}^{m} \hat{w}_i$.

Denote u^* the unique solution of (2.1), the following convergence estimate is correct for Algorithms 1 and 2 (see Tai [13]):

Theorem 2.1 *Assuming that the space decomposition satisfies (2.3), (2.4) and that the functional F satisfies (2.5). Then for Algorithms 1 and 2, we have*

$$\frac{F(w) - F(u^*)}{F(u) - F(u^*)} \le 1 - \frac{\alpha}{(\sqrt{1 + C^*} + \sqrt{C^*})^2}, \quad C^* = \Big(C_2 + \frac{[C_1 C_2]^2}{2\kappa}\Big)\frac{2}{\kappa}. \tag{2.8}$$

Algorithms 1 and 2 are written for general space decompositions. In implementation for a specific space decomposition technique, auxiliary functions may be introduced to make the implementation simpler and easier. For example, by defining $e_i = \hat{w}_i - R_i u$, Algorithms 1 and 2 can be written in the following equivalent form:

Algorithm 3 *Choose a relaxation parameter $\alpha \in (0, 1/m]$. Find $e_i \in V_i$ in parallel for $i = 1, 2, \cdots, m$ such that*

$$e_i = \arg\min_{\substack{v_i + R_i u \in K_i \\ v_i \in V_i}} G(v_i) \quad \textit{with} \quad G(v_i) = F\Big(u + v_i\Big). \tag{2.9}$$

Set $w = u + \alpha \sum_{i=1}^{m} e_i$.

Algorithm 4 *Choose a relaxation parameter* $\alpha \in (0,1]$. *Find* $e_i \in V_i$ *sequentially for* $i = 1, 2, \cdots, m$ *such that*

$$e_i = \arg \min_{\substack{v_i + R_i u \in K_i \\ v_i \in V_i}} G(v_i) \quad \textit{with} \quad G(v_i) = F\left(u + \sum_{j<i} e_j + v_i\right) \tag{2.10}$$

Set $w = u + \alpha \sum_{i=1}^{m} e_i$.

3. Some Applications. We apply the algorithms for the following obstacle problem:

$$\text{Find } u \in K, \quad \text{such that} \quad a(u, v-u) \geq f(v-u), \quad \forall v \in K, \tag{3.1}$$

with

$$a(v,w) = \int_\Omega \nabla v \cdot \nabla w \, dx, \qquad K = \{v \in H_0^1(\Omega) |\ v(x) \geq \psi(x) \text{ a.e. in } \Omega\}.$$

It is well known that the above problem is equivalent to the minimization problem (2.1) assuming that $f(v)$ is a linear functional on $H_0^1(\Omega)$. For the obstacle problem (3.1), the minimization space $V = H_0^1(\Omega)$. Correspondingly, we have $\kappa = 1$ for assumption (2.5). Later, $|\cdot|_1$ and $\|\cdot\|_1$ are used to denote the semi-norm and norm of $H_0^1(\Omega)$. The finite element method shall be used to solve (3.1). It shall be shown that domain decomposition and multigrid methods satisfy the conditions (2.3) and (2.4). For simplicity of the presentation, it will be assumed that

$$\psi = 0.$$

3.1. Overlapping domain decomposition methods. For the domain Ω, let $\mathcal{T}_H$ be a shape regular quasi-uniform finite element division, or a coarse mesh, of Ω, with a mesh size H. Further more, assume that $\{\Omega_i\}_{i=1}^M$ is a non-overlapping decomposition of Ω where each Ω_i has a diameter of order H and is the union of several coarse mesh elements. We further refine $\mathcal{T}_H$ to get a fine mesh partition $\mathcal{T}_h$ with mesh size h. We assume that $\mathcal{T}_h$ forms a shape regular quasi-uniform finite element subdivision of Ω. We call this the fine mesh or the h-level subdivision of Ω. We denote by $S_H \subset W_0^{1,\infty}(\Omega)$ and $S_h \subset W_0^{1,\infty}(\Omega)$ the continuous, piecewise linear finite element spaces over the H-level and h-level subdivisions of Ω respectively. For each Ω_i, we consider an enlarged subdomain Ω_i^δ consisting of elements $\tau \in \mathcal{T}_h$ with $\textit{distance}(\tau, \Omega_i) \leq \delta$. The union of Ω_i^δ covers $\bar{\Omega}$ with overlaps of size δ. For the overlapping subdomains, assume that there exist m colors such that each subdomain Ω_i^δ can be marked with one color, and the subdomains with the same color will not intersect with each other. Let Ω_i^c be the union of the subdomains with the i^{th} color, and $V_i = \{v \in S_h |\ v(x) = 0,\ x \notin \Omega_i^c\},\ i = 1, 2, \cdots, m$. By denoting the subspaces $V_0 = S_H$, $V = S_h$, we find that

$$a). \quad V = \sum_{i=1}^{m} V_i \qquad \text{and} \qquad b). \quad V = V_0 + \sum_{i=1}^{m} V_i. \tag{3.2}$$

Note that the summation index is now from 0 to m instead of from 1 to m when the coarse mesh is added. For the constraint set K, we define

$$K_0 = \{v \in V_0 |\ v \geq 0\}, \quad \text{and} \quad K_i = \{v \in V_i |\ v \geq 0\},\ i = 1, 2, \cdots, m. \tag{3.3}$$

Under the condition that $\psi = 0$, it is easy to see that (2.2) is correct both with or without the coarse mesh. When the coarse mesh is added, the summation index is from 0 to m. Let $\{\theta_i\}_{i=1}^m$ be a partition of unity with respect to $\{\Omega_i^c\}_{i=1}^m$, i.e. $\theta_i \in V_i$, $\theta_i \geq 0$ and $\sum_{i=1}^m \theta_i = 1$. It can be chosen so that

$$|\nabla \theta_i| \leq C/\delta, \quad \theta_i(x) = \begin{cases} 1 & \text{if } x \in \tau, \text{ distance } (\tau, \partial\Omega_i^c) \geq \delta \text{ and } \tau \subset \Omega_i^c, \\ 0 & \text{on } \overline{\Omega \backslash \Omega_i^c}. \end{cases} \tag{3.4}$$

Later in this paper, we use I_h as the linear Lagrangian interpolation operator which uses the function values at the h-level nodes. In addition, we also need a nonlinear interpolation operator $I_H^{\ominus} : S_h \mapsto S_H$. Assume that $\{x_0^i\}_{i=1}^{n_0}$ are all the interior nodes for $\mathcal{T}_H$ and let ω_i be the support for the nodal basis function of the coarse mesh at x_0^i. The nodal values for $I_H^{\ominus} v$ for any $v \in S_h$ is defined as $(I_H^{\ominus} v)(x_0^i) = \min_{x \in \omega_i} v(x)$, c.f [13]. This operator satisfies

$$I_H^{\ominus} v \leq v, \ \forall v \in S_h, \quad \text{and} \quad I_H^{\ominus} v \geq 0, \ \forall v \geq 0, v \in S_h. \tag{3.5}$$

Moreover, it has the following monotonicity property

$$I_{h_1}^{\ominus} v \leq I_{h_2}^{\ominus} v, \quad \forall h_1 \geq h_2 \geq h, \quad \forall v \in S_h. \tag{3.6}$$

As $I_H^{\ominus} v$ equals v at least at one point in ω_i, it is thus true that for any $u, v \in S_h$

$$\|I_H^{\ominus} u - I_H^{\ominus} v - (u - v)\|_0 \leq c_d H |u - v|_1, \quad |I_H^{\ominus} v|_1 \leq c_d |v|_1, \tag{3.7}$$

where d indicates the dimension of the physical domain Ω, i.e. $\Omega \subset R^d$, and

$$c_d = \begin{cases} C & \text{if } d = 1; \\ C\left(1 + \left|\log \frac{H}{h}\right|^{\frac{1}{2}}\right) & \text{if } d = 2, \\ C\left(\frac{H}{h}\right)^{\frac{1}{2}} & \text{if } d = 3, \end{cases}$$

With C being a generic constant independent of the mesh parameters. See Tai [13] for a detailed proof.

3.2. Decompositions with or without the coarse mesh. We first give the definition for the operators $R_i : K \mapsto K_i$ for the decomposition (3.2.a), i.e. we consider the domain decomposition method without using the coarse mesh. For any given $v \in S_h$, we decompose v as

$$v = \sum_{i=1}^{m} v_i, \quad , \quad v_i = I_h(\theta_i v), \tag{3.8}$$

and we define the mapping from v to v_i as R_i, i.e. $R_i v = v_i, \forall v \in S_h$. In case that $v \geq 0$, it is true that $v_i \geq 0$, i.e. R_i is a mapping from K to K_i. In addition,

$$\sum_{i=1}^{m} \|R_i u - R_i v\|_1^2 \leq C\left(1 + \frac{1}{\delta^2}\right) \|u - v\|_1^2,$$

which shows that $C_1 \leq C(1 + \delta^{-1})$. It is known that $C_2 \leq \sqrt{m}$ with m being the number of colors. From Theorem 2.1, the following rate is obtained for the one level domain decomposition method (c.f. (3.2.a)):

$$\frac{F(w) - F(u^*)}{F(u) - F(u^*)} \leq 1 - \frac{\alpha}{1 + C(1 + \delta^{-2})}.$$

For Algorithm 2, we can take $\alpha = 1$.

Numerical experiments and the convergence analysis for the two-level domain decomposition method, i.e. an overlapping domain decomposition with a coarse mesh, seem still missing in the literature. To apply our algorithms for the two-level domain decomposition method , i.e. for the decomposition (3.2.b), the operators R_i are defined as

$$R_0 v = I_H^{\ominus} v, \ R_i v = I_h(\theta_i (v - I_H^{\ominus} v)), i = 1, 2, \cdots m \quad \forall v \in S_h. \tag{3.9}$$

For a given $v \geq 0$, it is true using (3.5) that $0 \leq R_0 v \leq v$ and so $R_i v \geq 0, \ i = 1, 2, \cdots, m$, which indicates that $R_0 v \in K_0$ and $R_i v \in K_i, \ i = 1, 2, \cdots, m$ for any $v \in K$. It follows from

(3.7) that for any $u, v \in K$ $\|R_0 u - R_0 v\|_1 \leq c_d \|u - v\|_1$. Note that $R_i u - R_i v = I_h\big(\theta_i(u - v - I_H^{\ominus} u + I_H^{\ominus} v)\big)$. Using estimate (3.7) and a proof similar to those for the unconstrained cases, c.f. [17], [18], it can be proven that $\|R_i u - R_i v\|_1^2 \leq c_d \left(1 + \frac{H}{\delta}\right) \|u - v\|_1^2$. Thus

$$\left(\|R_0 u - R_0 v\|_1^2 + \sum_{i=1}^{m} \|R_i u - R_i v\|_1^2\right)^{\frac{1}{2}} \leq C(m) c_d \left(1 + \left(\frac{H}{\delta}\right)^{\frac{1}{2}}\right) \|u - v\|_1.$$

The estimate for C_2 is known, c.f. [17], [18]. Thus, for the two-level domain decomposition method, we have $C_1 = C(m) c_d \left(1 + \frac{\sqrt{H}}{\sqrt{\delta}}\right)$, $C_2 = C(m)$, where $C(m)$ is a constant only depending on m, but not on the mesh parameters and the number of subdomains. An application of Theorem 2.1 will show that the following convergence rate estimate is correct for the two-level domain decomposition method (3.2.b):

$$\frac{F(w) - F(u^*)}{F(u) - F(u^*)} \leq 1 - \frac{\alpha}{1 + c_d^2 (1 + H\delta^{-1})}.$$

3.3. Multigrid decomposition. Multigrid methods can be regarded as a repeated use of the two-level method. We assume that the finite element partition $\mathcal{T}_h$ is constructed by a successive refinement process. More precisely, $\mathcal{T}_h = \mathcal{T}_J$ for some $J > 1$, and $\mathcal{T}j$ for $j \leq J$ is a nested sequence of quasi-uniform finite element partitions, see [13], [17], [18]. We further assume that there is a constant $\gamma < 1$, independent of j, such that h_j is proportional to γ^{2j}. Corresponding to each finite element partition $\mathcal{T}j$, a finite element space $\mathcal{M}_j$ can be defined by

$$\mathcal{M}_j = \{v \in W_0^{1,\infty}(\Omega) : v|_\tau \in \mathcal{P}_1(\tau), \forall \tau \in \mathcal{T}_j\}.$$

let $\{x_j^k\}_{k=1}^{n_j}$ be the set of all the interior nodes. Denoted by $\{\phi_j^i\}_{i=1}^{n_j}$ the nodal basis functions satisfying $\phi_j^i(x_j^k) = \delta_{ik}$. We then define a one dimensional subspace $V_j^i = \text{span}(\phi_j^i)$. Letting $V = \mathcal{M}_J$ and $K_j^i = \{v \in V_j^i |\ v \geq 0\}$, we have the following trivial space decomposition:

$$V = \sum_{j=1}^{J} \sum_{i=1}^{n_j} V_j^i, \qquad K = \sum_{j=1}^{J} \sum_{i=1}^{n_j} K_j^i, . \tag{3.10}$$

Each subspace V_j^i is a one dimensional subspace. Let $I_{h_j}^{\ominus}$ to be the nonlinear interpolation operator from $\mathcal{M}_j$ to $\mathcal{M}_J$. For any $v \geq 0$ and $j \leq J-1$, define $v_j = I_{h_j}^{\ominus} v - I_{h_{j-1}}^{\ominus} v \in \mathcal{M}_j$. Let $v_J = v - I_{h_{J-1}}^{\ominus} v \in \mathcal{M}_J$. A further decomposition of v_j is given by $v_j = \sum_{i=1}^{n_j} v_j^i$ with $v_j^i = v_j(x_j^i)\phi_j^i$. It is easy to see that

$$v = \sum_{j=1}^{J} v_j = \sum_{j=1}^{J} \sum_{i=1}^{n_j} v_j^i. \tag{3.11}$$

We define the mapping from v to v_j^i to be R_j^i, i.e. $R_j^i v = v_j^i$. It follows from (3.5) and (3.6) that $v_j^i \geq 0$ for all $v \geq 0$, i.e. R_j^i is a mapping from K to K_j^i under the condition that $\psi = 0$. Define

$$\tilde{c}_d = \begin{cases} C, & \text{if} \quad d = 1; \\ C(1 + |\log h|^{\frac{1}{2}}), & \text{if} \quad d = 2; \\ C h^{-\frac{1}{2}}, & \text{if} \quad d = 3. \end{cases}$$

For any given $u \in S_h$, we define u_j and u_j^i in a similar way as we did for v. The following estimate can be obtained using approximation properties (3.7) (see [13]):

$$\sum_{j=1}^{J} \sum_{i=1}^{n_j} \|R_j^i u - R_j^i v\|_1^2 \leq C \sum_{j=1}^{J} h_j^{-2} \|u_j - v_j\|_0^2 \leq \tilde{c}_d^2 \sum_{j=1}^{J} h_j^{-2} h_{j-1}^2 |u - v|_1^2 \leq \tilde{c}_d^2 \gamma^{-2} J |u - v|_1^2,$$

which proves that

$$C_1 \cong \tilde{c}_d \gamma^{-1} J^{\frac{1}{2}} \cong \tilde{c}_d \gamma^{-1} |\log h|^{\frac{1}{2}}.$$

The estimate for C_2 is known, i.e. $C_2 = C(1-\gamma^d)^{-1}$, see Tai and Xu [18]. Thus for the multigrid method, the error reduction factor for the algorithms is

$$\frac{F(w) - F(u^*)}{F(u) - F(u^*)} \leq 1 - \frac{\alpha}{1 + \tilde{c}_d^2 \gamma^{-2} J}.$$

3.4. Numerical experiments. For the implementation of the proposed algorithms, we need some subroutines to carry out the actions of the decomposition operators and some other subroutines to solve the sub-minimization problems in the algorithms. If we implement the algorithms in the form as given in Algorithms 3 and 4, it can be seen the decomposition operators R_i are only needed to determine the obstacles for the subproblems.

Let us first sketch the implementation for the DD methods with or without the coarse mesh. Without the coarse mesh, we need to have some subroutines to calculate the θ_i functions for the decomposition operators R_i given in (3.8). The construction of the θ_i functions are not unique and we just choose one of them. The sub-minimization problems over the subdomains are solved by the augmented Lagrangian method as stated in [11] (but without the dimensional splitting). Once the coarse mesh is added, we need a subroutine to calculate $I_H^{\ominus} v$ for any given $v \in S_h$ and this will give the decomposition operator R_0 as given in (3.9). Once this is done, the other operators R_i can be done using the functions θ_i. The sub-minimization problem over the coarse mesh is also solved by the augmented Lagrangian method, see [16, 11]. The subproblems need more computing time in the first iteration. From the second iteration, very good intial guess is available and the iterations are terminated after a few iterations.

For the multigrid decomposition (3.10), we need to calculate $I_{h_j}^{\ominus}$ to get the actions of operators R_j^i. The cost for calculating this is very cheap. For any $v \in S_h$ and $v \geq 0$, we use a vector z_j to store the values $\min_{\tau_j^i} v$ for all the elements $\tau_j^i \subset \mathcal{T} j$. As the meshes are nested, the vectors z_j can be computed recursively starting from the finest mesh and ending with the coarsest mesh. From the vectors z_j, it is easy to compute $I_{h_j}^{\ominus} v$ on each level. The value of $I_{h_j}^{\ominus} v$ at a given node is just the smallest value of z_j in the neighboring elements. The sub-minimization problems are just some minimization problems over a one-dimensional subspace and explicit formula can be given for these sub-minimization problems and they can be implemented similarly as for unconstrained problems, see [16]. The operation cost per iteration for the algorithms is $O(n_J)$.

We shall test our algorithms for the obstacle problem (3.1) with $\Omega = [-2,2] \times [-2,2]$, $f = 0$ and $\psi(x,y) = \sqrt{x^2+y^2}$ when $x^2 + y^2 \leq 1$ and $\psi(x,y) = -1$ elsewhere. This problem has an analytical solution [13]. Note that the continuous obstacle function ψ is not even in $H^1(\Omega)$. Even for such a difficult problem, uniform linear convergence has been observed in our experiments. In the discrete case, the non-zero obstacle can be shifted to the right hand side.

Figure 3.1 shows the convergence rate for Algorithm 2 with different overlapping sizes for decomposition (3.9). Figure 3.2 shows the convergence rate for Algorithm 2 with the multigrid method for decomposition (3.11) and J indicates the number of levels. In the figures en is the H^1-error between the computed solution and the true finite element solution and $e0$ is the initial error. $\log(en/e0)$ is used for one of the subfigures. The convergence rate is faster in the beginning and then approaches a constant after some iterations.

REFERENCES

[1] A. Brandt and C. W. Cryer. Multigrid algorithms for the solution of linear complementary problems arising from free boundary problems. *SIAM J. Sci. Comput.*, 4:655–684, 1983.

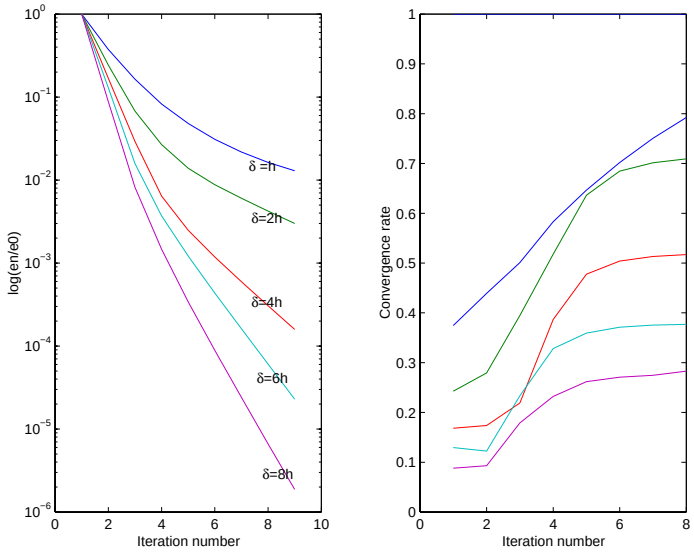


Figure 3.1: Convergence for the two-level method for decomposition (3.9) with different overlaps, $h = 4/128$, and $H = 4/8$.

[2] T. F. Chan and I. Sharapov. Subspace correction multi-level methods for elliptic eigenvalue problems. *Numer. Linear Algebra Appl.*, 9(1):1–20, 2002.

[3] W. Hackbusch and H. D. Mittelmann. On multilevel methods for variational inequalities. *Numer. Math.*, 42:65–76, 1983.

[4] R. H. W. Hoppe. Multigrid algorithms for variational inequalities. *SIAM J. Numer. Anal.*, 24(5):1046–1065, 1987.

[5] R. H. W. Hoppe and R. Kornhuber. Adaptive multilevel methods for obstacle problems. *SIAM J. Numer. Anal.*, 31:301–323, 1994.

[6] R. Kornhuber. Monotone multigrid methods for elliptic variational inequalities I. *Numer. Math*, 69:167–184, 1994.

[7] R. Kornhuber. Monotone iterations for elliptic variational inequalities, II. *Numer. Math.*, 72:481–499, 1996.

[8] J. Mandel. A multilevel iterative method for symmetric, positive definite linear complementary problems. *Appl. Math. Optim.*, 11:77–95, 1984.

[9] I. A. Sharapov. Multilevel subspace correction for large scale optimization problems. Technical Report CAM-97-31, UCLA, Computational and Applied Mathematics, 1997.

[10] X.-C. Tai. Parallel function and space decomposition methods. In P. Neittaanmäki, editor, *Finite element methods, fifty years of the Courant element, Lecture notes in pure and applied mathematics*, volume 164, pages 421–432. Marcel Dekker inc., 1994. Available online at http://www.mi.uib.no/~tai.

[11] X.-C. Tai. Parallel function decomposition methods and numerical applications. In J. Wang, M. B. Allen, B. M. Chen, and T. Mathew, editors, *Iterative methods in scientific computation*, volume 4 of *Series in computational and applied mathematics*, pages 231–238. IMACS, 1998.

[12] X.-C. Tai. Convergence rate analysis of domain decomposition methods for obstacle problems. *East-West J. Numer. Math.*, 9(3):233–252, 2001.

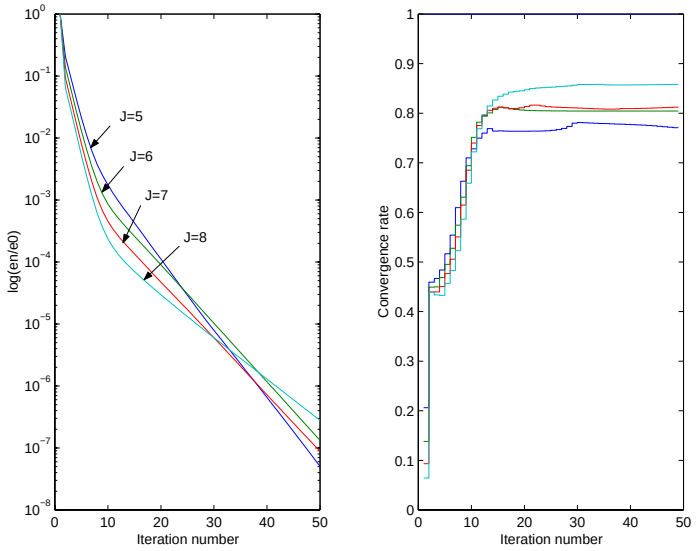


Figure 3.2: Convergence for the multigrid method

[13] X.-C. Tai. Rate of convergence for some constraint decomposition methods for nonlinear variational inequalities. *Numer. Math.*, accepted and to appear. Available online at http://www.mi.uib.no/~tai, report-150, Department of Mathematics, University of Bergen.

[14] X.-C. Tai and M. Espedal. Applications of a space decomposition method to linear and nonlinear elliptic problems. *Numer. Math. for Part. Diff. Equat.*, 14:717–737, 1998.

[15] X.-C. Tai and M. Espedal. Rate of convergence of some space decomposition method for linear and nonlinear elliptic problems. *SIAM J. Numer. Anal.*, 35:1558–1570, 1998.

[16] X.-C. Tai, B. ove Heimsund, and J. Xu. Rate of convergence for parallel subspace correction methods for nonlinear variational inequalities. In *Proceedings of the 13th International Conference on Domain Decomposition Methods in Lyon, France.*

[17] X.-C. Tai and P. Tseng. Convergence rate analysis of an asynchronous space decomposition method for convex minimization. *Math. Comput.*, 2001.

[18] X.-C. Tai and J.-C. Xu. Global and uniform convergence of subspace correction methods for some convex optimization problems. *Math. Comput.*, 71:105–124, 2001.

[19] J. Xu. Iteration methods by space decomposition and subspace correction. *SIAM Rev.*, 34:581–613, 1992.

Part VII

Contributed Papers

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

33. Flow in complex river networks simulation through a domain decomposition method

J. Aparicio[1], A. A. Aldama[2], H. Rubio[3]

1. Introduction. Lower river basins are characterized by rivers flowing on floodplains, usually forming interconnecting networks of streams that frequently interact with lagoons directly or indirectly connecting to the stream reaches. The flood plains both in the Pacific and in the Atlantic coasts of México have experienced, during the last few decades, an accelerated economic development and therefore an appreciable population growth, and some flood-related disasters have occurred in this zones recently. In order to avoid this kind of disasters and the consequent loss of human lives and property, the need to build flood defense infrastructure arises, constituted for instance by levees or dikes, and/or to develop real time flood-warning systems. In any case, computational models are needed to adequately simulate the passage of floods through the river networks. These computational models should take into account the fact that when river reaches flow into or from lagoons, their length is modified according to whether the free surface level in the lagoon is rising or lowering. A model of this kind is presented in this paper. Aldama and Aparicio (1994) [1] presented the fundamentals of this model elsewhere. Here, the complete development is addressed and an application to a real case in the lower Grijalva River is presented.

2. Fundamental equations. The equations on which the model is based are the one-dimensional, free surface Saint-Venant equations: [4]

Continuity

$$B\frac{\partial H}{\partial t} + \frac{\partial UA}{\partial x} = q \tag{2.1}$$

Momentum

$$\frac{\partial U}{\partial t} + U\frac{\partial U}{\partial x} + g\frac{\partial H}{\partial x} + gn^2\frac{U\,|U|}{R^{4/3}} = 0 \tag{2.2}$$

where B is the free surface width; H, free surface elevation or level; U, velocity; A, hydraulic area; q, lateral inflow per unit length; g, acceleration of gravity; n, Manning roughness coefficient; R, hydraulic radius and x and t represent distance and time respectively.

In a channel network such as that shown in fig. 1, two types of flooding areas (heretofore called "lagoons") may be formed: those directly connected to one or more channel reaches, which will be called *interconnecting lagoons* and those receiving or delivering water from or to the river, but not having any influence in the water level of any reach, which will be called *lateral lagoons*.

Interconnecting lagoons will be linked to the corresponding channel reaches by means of a mass conservation equation of the form

$$\frac{\partial V}{\partial t} + \oiint_{sc} \overline{U} \bullet \overline{dA} = 0 \tag{2.3}$$

where V is the lagoon volume, sc is the control surface defined by the lagoon boundaries and the scalar product $\overline{U} \bullet \overline{dA}$ represents the outflow discharge from the lagoon (see fig. 2; note that inflow to the lagoon is negative).

[1]Mexican Institute of Water Technology, aparicio@tlaloc.imta.mx
[2]Mexican Institute of Water Technology, aaldama@tlaloc.imta.mx
[3]National Water Commision, Mexico, hrubio@grfs.cna.gob.mx

On the other hand, lateral lagoons are connected to the channel reaches by means of the riverbanks. The unit discharge between river reaches and lateral lagoons will be assumed to be governed by a long-crested weir law:

$$q = C_q h\sqrt{|h|} \tag{2.4}$$

where C_q is a discharge coefficient and h is the net head. C_q is assumed to be a function of the parameter [3]

$$\phi = \frac{|\eta_2 - \eta_1|}{\eta - E} \tag{2.5}$$

where η_2 and η_1 are, respectively, the water surface elevation in the river reach and in the lagoon and E is the elevation of the river bank (see fig. 3). $\eta = \eta_1$ when flow is from lagoon to river and $\eta = \eta_2$ when the river flows into the lagoon. The discharge coefficient is then computed as [3]

$$C_q = \begin{cases} 0.871\sqrt{2g}\phi^{0.478} & for \quad 0 < \phi < 0.1 \\ 0.446\sqrt{2g} & for \quad \phi = 1.0 \\ 0.446\sqrt{2g}\phi^{0.155} & for \quad 0.1 < \phi < 1.0 \end{cases} \tag{2.6}$$

Due to the fact that flow in these conditions occurs in extremely flat terrain, only storage effects are taken into account and no dynamical effects will be considered neither in the interconnecting nor in the lateral lagoons.

3. Transformed equations. River reaches in flat floodplains are frequently confined between lagoons that change in size as floods progress, therefore changing the reach length, which requires solving eqs. (2.1) and (2.2) in variable domains. To avoid the sometimes severe inaccuracies arising from the use of fixed grids in these cases, and following Austria & Aldama [2] and Aldama & Aparicio [1], a coordinate transformation strategy of the following form is employed:

$$\xi = \frac{x - x_r(t)}{x_f(t) - x_r(t)} \tag{3.1}$$

$$\tau = t \tag{3.2}$$

where $x_r(t)$and $x_f(t)$are, respectively, the position of the rear and front of the size-changing river reach and ξ and τ are the transformed coordinates. Applying the coordinate transformation to eqs. (2.1) and (2.2), the following transformed equations are obtained: [1]

$$B(x_f - x_r)\frac{\partial H}{\partial \tau} - B\xi\frac{\partial H}{\partial \xi}\frac{dx_f}{dt} - B(\xi + 1)\frac{\partial H}{\partial \xi}\frac{dx_r}{dt} + \frac{\partial (UA)}{\partial \xi} = q(x_f - x_r) \tag{3.3}$$

$$(x_f - x_r)\frac{\partial U}{\partial \tau} - \xi\frac{\partial U}{\partial \xi}\frac{dx_f}{dt} - (\xi + 1)\frac{\partial U}{\partial \xi}\frac{dx_r}{dt} + U\frac{\partial U}{\partial \xi} + g\frac{\partial H}{\partial \xi} + g(x_f - x_r)n^2\frac{U\,|U|}{R^{4/3}} = 0 \tag{3.4}$$

4. Domain decomposition and numerical solution. Eqs. (3.3) and (3.4) are solved using an implicit, fractional step scheme [5], which leads to a system of algebraic linear equations of the form [1]

$$[A]^k \{H\}^{k+1} = \{C\}^k \tag{4.1}$$

where $[A]^k$ and $\{C\}^k$ are respectively a matrix and a vector which depend on known values at time level k. Matrix $[A]$ is tridiagonal for a single channel, which makes the solution of eq. (3.1) very efficient, while preserving a second order accuracy. However, in a complex channel

network such as that shown in fig. 1, due to interactions between the different reaches, nonzero elements appear outside the three main diagonals, thus making the solution far less efficient. Therefore, Aldama & Aparicio [1] proposed a solution algorithm based on the use of numerical Green's functions consisting of writing eq. (4.1) as

$$[A_R]^k \{H_R\}^{k+1} + B^k_{R,r} \{H_{R,r}\}^{k+1} + B^k_{R,f} \{H_{R,f}\}^{k+1} = \{C_R\}^k \tag{4.2}$$

where $[A]^k$ is a tridiagonal coefficients matrix, $\{H_R\}^{k+1}$ is the unknown water surface elevations vector within the reach, $B^k_{R,r}$ and $B^k_{R,f}$ are scalars and $\{H_{R,r}\}^{k+1} \equiv \left\{H^{k+1}_{R,r}, 0, ..., 0\right\}^T$ and $\{H_{R,f}\}^{k+1} \equiv \left\{0, ..., 0, H^{k+1}_{R,f}\right\}^T$, $H^{k+1}_{R,r}$ and $H^{k+1}_{R,f}$ representing the water surface elevations at the rear and front ends of the reach. The vector of unknowns is decomposed as the sum of a *homogeneous* and an *inhomogeneous* solutions:

$$\{H_R\}^{k+1} = \{H_{R,h}\}^{k+1} + \{H_{R,i}\}^{k+1} \tag{4.3}$$

where the homogeneous solution is defined by

$$[A_R]^k \{H_{R,h}\}^{k+1} = \{C_R\}^k \tag{4.4}$$

and the inhomogeneous solution is given by

$$\{H_{R,i}\}^{k+1} = H^{k+1}_{R,r} \{G_{R,r}\}^{k+1} + H^{k+1}_{R,f} \{G_{R,f}\}^{k+1} \tag{4.5}$$

where $\{G_{R,r}\}^{k+1}$ and $\{G_{R,f}\}^{k+1}$ are *rear* and *front numerical Green's functions*, representing the response of channel reach R to unit variations in the water surface elevations at its rear and front ends and defined respectively by

$$[A_R]^k \{G_{R,r}\}^{k+1} = -B^k_{R,r} \{1, 0, ..., 0\}^T \tag{4.6}$$

and

$$[A_R]^k \{G_{R,f}\}^{k+1} = -B^k_{R,f} \{0, ..., 0, 1\}^T \tag{4.7}$$

On the other hand, the mass conservation equation for interconnecting lagoons (eq. 3) is discretized as

$$A_L \frac{H^{k+1}_c - H^k_c}{\Delta\tau} - \sum_i A_i U_i = 0 \tag{4.8}$$

where A_L is the lagoon surface area, H_c is the free surface elevation at the lagoon and A_i are the hydraulic areas of the river reaches concurring to the lagoon. In the case of lateral lagoons, free surface elevations and therefore stored volumes are computed simply from

$$\frac{A_L}{\Delta\tau} \left(H^{k+1}_c - H^k_c\right) = \sum Q_i \tag{4.9}$$

where total discharge $Q_i = qL_i$, L_i being the *crest* length on the river bank. Note that several river reaches can be connected to the same lateral lagoon. For the sake of simplicity, discharges Q_i are computed explicitly from the previous time step.

In this way, eqs. (4.4), (4.6) and (4.7) are tridiagonal systems and eqs. (4.3) and (4.5), along with the mass conservation equation (4.8) for each interconnecting node, lead to a sparse but relatively small system of equations in terms of the water surface elevation at the node.

Therefore, with the above outlined procedure, three small tridiagonal systems for each channel and a small sparse system for the interconnection nodes are solved, which makes the overall solution considerably more efficient than the large, nonbanded system which would otherwise arise.

5. Boundary conditions. Every channel reach within the network is connected to a node with any of the several possible boundary conditions. The most common boundary conditions are known upstream or downstream discharge, specified upstream or downstream water level or interconnecting lagoon. In the first case, the known discharge is susbstituted into equation (3.3) an a two-term equation is obtained. In the second case, the known level is directly substituted into equation (4.1) and in the third case, equation (4.8) is used to couple the lagoon level with the corresponding channel reach level, either at the reach rear or front.

6. Application. The numerical model described above was integrated in a computational system called *Trans-R* and was applied to the uncontrolled part of the lower Grijalva river basin in the Southeastern region of México (see fig. 1). This river flows from a mountainous zone into a considerably flat region and finally into the Gulf of México. Four major tributaries can be identified as part of the river network: the La Sierra, Pichucalco, Teapa and Puyacatengo rivers. A main concern in this case is the city of Villahermosa, located downstream of the confluence of these major tributaries as shown in fig. 1. With a fast-growing population of about 400,000 inhabitants, the city and its surroundings are subject to flooding caused by the intense precipitations frequently produced by cyclones. A high population growth index produces a severe urban pressure on the Grijalva River and its naturally flooding lagoons, therefore requiring a real-time forecasting system and quantitative aids for the urban growth planning process. For the analysis, about 200 topographic maps of the zone were used, and some *ad hoc* topographic surveys were performed, from which channel sections for the whole network were obtained, and 21 lagoons were identified, including two interconnecting lagoons (lagoons 3 and 8) and 19 lateral lagoons (see fig. 1).

Data for boundary conditions were provided by four gauging stations in each of the major tributaries located at the boundary between the mountains and the floodplains and one at the downstream end of the considered region, called *Gaviotas*, where Villahermosa city is located.

Several flooding events were considered. Due to lack of space, only three of them will be shown here. Figures 4 to 6 show the May 1970 flood; in fig. 4 the measured hydrographs at each of the five gauging stations are shown. Only the stage-discharge relationship at the Gaviotas Station was used for the simulation as boundary condition and the measured hydrograph was reserved for comparison purposes. In figures 5 and 6 some of the results are shown. Figure 5 shows a comparison between measured and computed hydrographs at the Gaviotas gauging station. A reasonable agreement is observed. In fig. 6 a sequence of the flood progress in plan view is shown. It can be seen that two interconnecting lagoons are flooded in the first place (fig. 6 b); in fig. 6 c, one interconnecting lagoon is totally flooded and one lateral lagoon is affected. In fig. 6 d, the flood hydrograph has started to recede, one lagoon has disappeared and another has started to do so. In fig. 6 e, this interconnecting lagoon has completely disappeared. Figures 7 and 8 show recorded and simulated limnigraphs for the May-August, 1967 and September-October, 1999 flows at the Gaviotas Station. The latter event produced extensive flooding and damages in Villahermosa City and vicinity. Good agreement is observed.

7. Conclusions. A numerical model for transient flow simulation in complex river networks with interconnecting and lateral flooding lagoons has been developed. The model uses a coordinate transformation, which allows the channel-interconnecting lagoon interaction simulation and numerical Green's functions to decompose the domain and efficiently solve the problem in the whole river reaches-lagoons hydraulic system. Lateral lagoons connected to the river through the riverbanks are also taken into account in the model. Application to the lower Grijalva River network shows good agreement between computed and measured hydrographs and limnigraphs at the basin outlet.

Fig. 1. Lower Grijalva river network

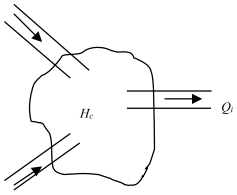


Fig. 2. Interconnecting lagoon

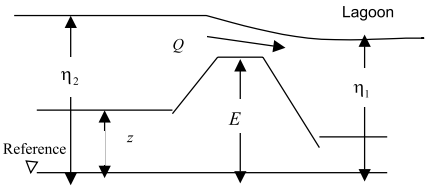


Fig. 3. Lateral lagoon

8. Acknowledgements. Marco A. Sosa programmed the Trans-R system and Ángeles Suárez did much of the computational test work.

REFERENCES

[1] A. A. Aldama and J. Aparicio. Numerical simulation of flow in river networks with complex topology. In A. P. et al, editor, *Computational Methods in Water Resources X*, pages 1131–1138, P.O. Box 17, 3300 AA Dordrecht, The Netherlands, 1994. Kluwer Academic Publishers.

[2] P. M. Austria and A. A. Aldama. Adaptive mesh scheme for free surface flows with moving boundaries. In G. G. et al, editor, *Computational Methods in Water Resources VIII*, pages 455–460. Springer-Verlag, 1990.

[3] C. Cruickshank. Modelos para el transito de avenidas en cauces con llanuras de inundacion. By Instituto de Ingenieria. UNAM, Ciudad Universitaria, Mexico, 1974.

[4] B. D. Saint-Venant. Theorie du movement non-permanent des eaux crues des rivieres et a lintroduction des marees dans leur lit. *Acad. Sci. (Paris) Comptes rendus*, 73:147–154, 237–240, 1871.

[5] N. N. Yanenko. *The method of fractional steps*. Springer-Verlag, 1971.

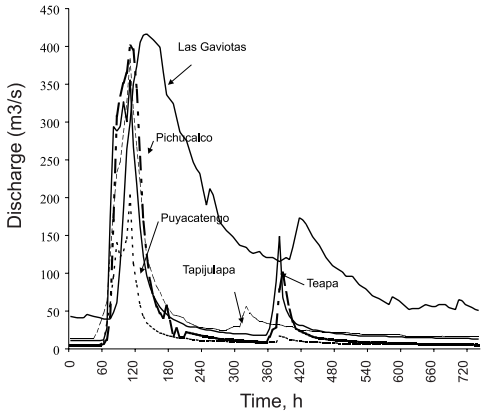


Fig. 4. Recorded Hydrographs, May, 1970

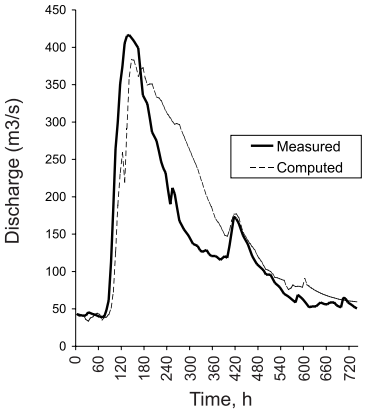


Fig. 5. Measured and computed hydrographs at the Gaviotas gauging station, May, 1970

Fig. 6a. Plan view, initial conditions

Fig. 6b. Plan view, time = 126 h

Fig. 6c. Plan view, time = 204 h

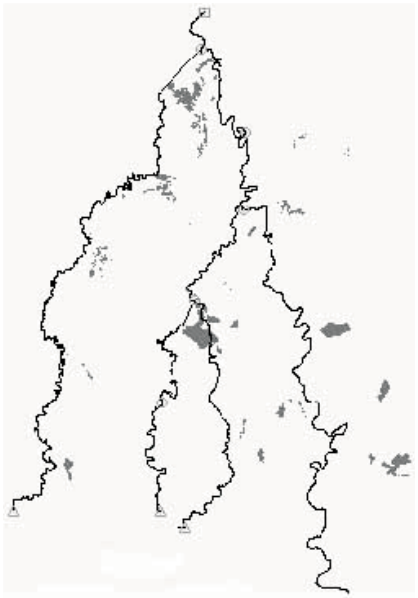


Fig. 6d. Plan view, time = 738 h

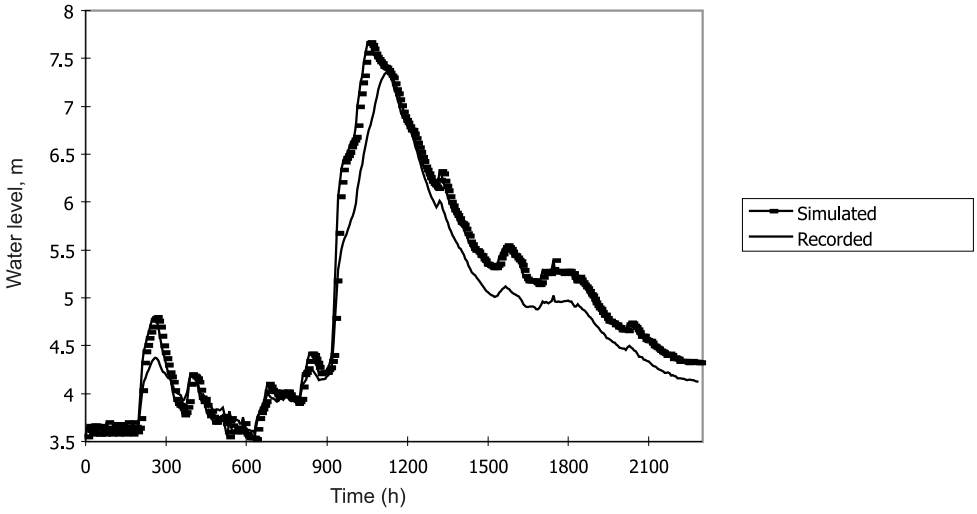


Fig. 7. Limnigraphs, May-August, 1967

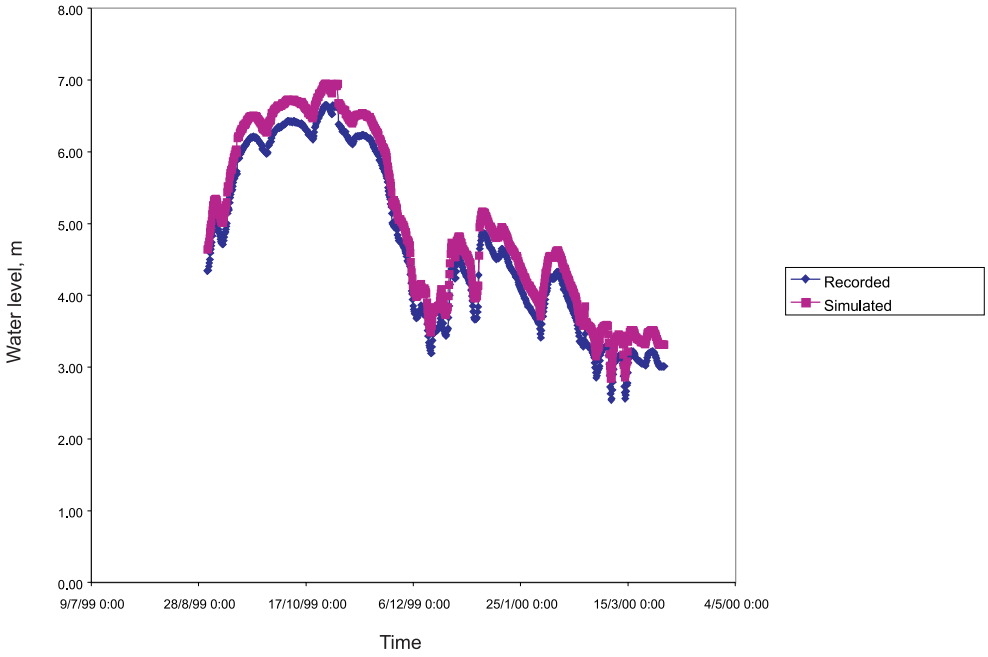


Fig. 8. Limnigraphs, September-October, 1999

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

34. On Aitken Like Acceleration of Schwarz Domain Decomposition Method Using Generalized Fourier

J.Baranger[1], M.Garbey[2] and F.Oudin-Dardun[3]

1. Introduction. The idea of using Aitken acceleration [5] [11], on the classical Schwarz additive domain decomposition (dd) method [9] [7] [8] [6] has been introduced in [3]. For an elliptic operator with constant coefficients on a regular grid, this method is called Aitken-Schwarz (AS) procedure, and is a direct solver. This method has shown very good numerical performance, and has been used in more complex situations [4]. We have also extended the Aitken-Schwarz procedure to the case of a 2-D cartesian grid, *not necessarily regular,* with two subdomains [1].

In the present paper, we extend this method to more complex situations and give a general framework for our method. We first consider overlapping strip domain decomposition with P domains on a *non-uniform* Cartesian grid. The key idea is the replacement of the 1-D Fourier transform used on the regular space step discretization of the artificial interface grid by a transform using the eigenvectors of a suitable 1-D operator. We give a direct solver version of the Aitken-Schwarz algorithm for arbitrary number of subdomains, as well as an iterative version when acceleration is applied to dominant modes only. In this last case, the number of iterates is not sensitive to the number of subdomains with overlap of few mesh steps. Second, we consider non-matching grids to apply our method to non-trivial geometries. We present some experimental results for the Poisson and Helmholtz operator and comment on an adaptive version of our acceleration technique for incompressible unsteady flow in a channel past a disc. This paper is restricted to problems in two space dimensions, but most of the concepts introduced here can be extended to 3 space dimensions.

2. A general framework. We briefly describe a general framework for AS method. For more details see [2]. The AS method is built on three ideas:

- Schwarz's method is an iterative method on a trace transfer operator acting on functions defined on the interfaces. Sparsity of the Jacobian of this operator is related to the domain decomposition (**dd**).

- discretization and choice of the interface representation may in some cases, and if well chosen, increase this sparsity.

- for an operator with a sparse matrix, simple acceleration processes can be constructed. The Aitken process, for example, provides an exact solver in the linear case if the trace transfer operator can be diagonallized .

• **Trace transfer operator for Schwarz iterative method:** we consider a bounded domain Ω in $\mathbb{R}^N$ with a strip dd in P domains Ω_p, i.e Ω_p only intersects Ω_{p-1} and Ω_{p+1}, with obvious modifications for p=1 and P.

The boundary Γ_p of Ω_p is decomposed into three subsets: Γ_p^l (resp.Γ_p^r) included in Ω_{p-1} (resp.Ω_{p+1}) and the remaining part $\tilde{\Gamma}_p$.

Let (Π) be a boundary value problem (**bvp**) well posed in Ω. One step of the additive Schwartz dd method with Dirichlet-Dirichlet boundary conditions (**bc**) is: for all p, given the Dirichlet bc l_p (resp.r_p) on Γ_p^l (resp.Γ_p^r) solve the problem (Π_p) the restriction of (Π) to Ω_p with these bc and the one of (Π) on $\tilde{\Gamma}_p$.

(Π_p) is assumed to be well posed. We denote $\bar{r}_{p-1}$ (resp. $\bar{l}_{p+1}$) the trace of the solution of (Π_p) on Γ_{p-1}^r (resp. Γ_{p+1}^l). So, one step of Schwarz method is described by one application

[1]MCS-ISTIL-Universit Lyon1, baranger@mcs.univ-lyon1.fr
[2]Dept. of Computer Science-University of Houston, garbey@cs.uh.edu
[3]MCS-ISTIL-Universit Lyon1, foudin@mcs.univ-lyon1.fr

of the trace transfer operator

$$(\bar{l}_2, \bar{r}_1, \ldots, \bar{l}_P, \bar{r}_{P-1}) = T(l_2, r_1, \ldots, l_p, r_{p-1}, \ldots, l_P, r_{P-1})$$

acting on trace spaces of functions or distributions adapted to the bvp.

T has the special structure:

$$\bar{l}_2 = T_1^r(r_1), \quad \ldots, \quad \left\{ \begin{array}{l} \bar{r}_{p-1} = T_p^l(l_p, r_p) \\ \bar{l}_{p+1} = T_p^r(l_p, r_p) \end{array} \right\} p = 2 \text{ to } P-1, \quad \ldots, \quad \bar{r}_{P-1} = T_P^l(l_P)$$

Here $(\bar{r}_{p-1}, \bar{l}_{p+1}) = T_p(l_p, r_p)$ is composed of a local solver of the bvp (Π_p) and the trace operators on Γ_{p-1}^r and Γ_{p+1}^l. These operators can be exact or approximated.

Formally, the 2(P-1) Jacobian matrix of T has the pentadiagonal structure, pointed out for a special case in [3]:

$$\begin{pmatrix} 0 & \delta_1^{rr} & 0 & 0 & & & \\ \delta_2^{ll} & 0 & 0 & \delta_2^{lr} & 0 & & \\ \delta_2^{rl} & 0 & 0 & \delta_2^{rr} & 0 & 0 & \\ 0 & 0 & \delta_3^{ll} & 0 & 0 & \delta_3^{lr} & \ddots \\ & 0 & \delta_3^{rl} & 0 & 0 & \delta_3^{rr} & \ddots \\ & & 0 & 0 & \delta_4^{ll} & 0 & \ddots \\ & & & \ddots & \ddots & \ddots & \ddots \end{pmatrix} \tag{2.1}$$

with $\delta_p^{lr} = \partial T_p^l / \partial r_p(l_p, r_p)$. The derivatives are assumed to exist in some sense in the traces functional spaces.

• **Discretization and interface representation:** we introduce a discrete approximation of the traces. Each trace l_p (resp. r_p) is approximated by J numbers l_{pj} (resp. r_{pj}), $j = 1$ to J. These numbers may be point values, coefficients in a basis, and so on. J may vary with p if, for example, one has non-matching grids between subdomains. Retaining the previous notations, l_p and r_p are now J-vectors and $\delta_p = \begin{pmatrix} \delta_p^{ll} & \delta_p^{lr} \\ \delta_p^{rl} & \delta_p^{rr} \end{pmatrix}$ is a 2J square matrix. T is an application from $\mathbb{R}^{2J(P-1)}$ into itself with a sparse Jacobian matrix. For some problems, dd and meshes, a well-chosen change of unknowns $l_{pj} \to \hat{l}_{pj}$ may greatly increase the sparsity of the Jacobian of the transformed trace transfer operator $\hat{T}$. This idea which is the core of AS method has been introduced on a uniform mesh -using Fourier transform- in [3]. An extension to non-uniform rectangular meshes is given in the next section.

• **Acceleration process:** Schwarz method can be considered as an iterative method for the hat transform (associated with the interface representation) of T which map vectors of size 2J(P-1). Any acceleration process can be used. The AS method uses Aitken method, taking advantage of the sparsity coming on the one hand from the special dd, and on the other hand from the generalized Fourier transform.

In the next section, we describe a special situation in which a good choice of the interface representation leads to a very sparse Jacobian. For proofs and extensions of the content of this section, we refer the reader to[2].

3. Generalized Fourier transform and interface representation on a non-uniform rectangular mesh. We restrict ourselves to two space dimensions and a rectangular domain Ω with a strip dd into rectangles. The left (resp. right) boundary of Ω_p is $x = x_p^l$ (resp. x_p^r). (Π) is a homogenous Dirichlet bvp whose equation $Lu = f$ has a separable second order operator $L = L_1 + L_2$ with

$$L_1 = a_1 \partial_{xx} + b_1 \partial_x + c_1, \quad L_2 = a_2 \partial_{yy} + b_2 \partial_y + c_2.$$

a_1, b_1, c_1 are functions of x, and a_2, b_2, c_2 functions of y.

The grid is a tensorial product of the following two irregular meshes: x-mesh $x_i, i \in \mathbf{I}$, and y-mesh $y_j, j = 0$ to $J+1$. L_1^h (resp.L_2^k) are discretization of L_1 (resp. L_2) on the x (resp. y)-mesh. The unknowns u_{ij} are approximations of $u(x_i, y_j)$ with f_{ij} some given approximation of $f(x_i, y_j)$. We use the notation $U_j = (u_{ij})_{i \in \mathbf{I}}$.

Let [e,w]x[n,s] be the generic rectangular subdomain. The discrete approximation (Π_p^{hk}) of problem (Π_p) can be written

$$L_1^h U_j + L_2^k U_j = F_j,\ U_j(w) \text{ and } U_j(e) \text{ given},\ u_{i0} = u_{iJ+1} = 0, i \in \mathbf{I}$$

The following result is proved in [2]:

Theorem 3.1 *Assume that the eigenvalue problem*

$$L_2^k \Phi_m = \lambda_m \Phi_m,\ \ \Phi_{m0} = \Phi_{mJ+1} = 0 \qquad (E)$$

has J linearly independent real eigenvectors associated with real eigenvalues. We define the generalized Fourier transform:

$$u_{ij} = \sum_{m=1}^{J} \hat{u}_{im} \Phi_{mj},\ \ j = 1 \text{ to } J.$$

Then $(\hat{\Pi}_p^{hk})$ -the hat transform of (Π_p^{hk})- is a set of J uncoupled discrete one-dimensional linear problems:

$$[L_1^h + \lambda_m]\hat{U}_m = \hat{F}_m,\ \ m = 1 \text{ to } J,\ \hat{u}_{0m} \text{ and } \hat{u}_{I+1m} \text{ given}\ .$$

The hat trace transfer operator is affine on $\mathbb{R}^{2J(P-1)}$ with a block-diagonal matrix of J blocks. The m-th diagonal block has the form (2.1) and corresponds to the mode Φ_m and the operator $L_1^h + \lambda_m$.

We are going to apply this result to construct the AS algorithm.

3.1. Algorithm. We apply an Aitken-like acceleration procedure to each mode of the generalized Fourier transform of the interfaces values given by Schwarz dd method. It follows from theorem 3.1 that the method is an exact solver in this context. The algorithm is:

Step 1: compute the eigenvectors $\lambda_m, \Phi_m, m = 1$ to J solution of problem (E).

Step 2: given traces on the interfaces, perform 3 steps of the Schwarz method.

Step 3: take the generalized Fourier transform of the last 4 traces.

Step 4: apply the one-dimension Aitken acceleration formula to each mode of these transformed traces.

Step 5: recompose the physical traces from the result of step 4.

Step 6: from these traces, make one step of the Schwarz method.

We observe that Step 1, 3, 5 and 6 can be processed in parallel. Step 2 is the additive Schwarz algorithm that has, in general, poor numerical efficiency but scales very well on a so called MIMD architecture. Step 4 requires global communication of the hat transform of the traces but makes the numerical algorithm efficient.

In order to minimize the amount of global communications in the parallel algorithm and decrease the number of arithmetic operations, it is interesting to accelerate only the eigenvector components of the traces that correspond to dominant eigenvalues $\lambda_m,\ m = 1..J'$, with $J' < J$. As a matter of fact, eigenvector components that corresponds to small eigenvalues λ_m converge fast with the Schwarz method itself. In that case, steps 3 and 5 are modified and the direct and inverse hat transforms use only the $J' < J$ first modes. Further, we may have to iterate step 2 to step 6 until convergence. We call this variant of our method as the Steffensen-Schwarz method.

We are going to apply this result to the Poisson problem discretized by FE as done in [1] on a rectangular irregular grid.

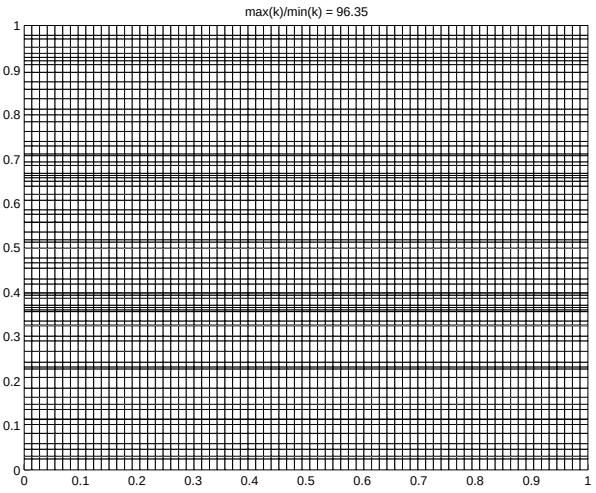


Figure 3.1: Mesh

3.2. Numerical experiments. We consider, on the domain $\Omega =]0,1[\times]0,1[$, the Poisson problem : $-(u_{xx}+u_{yy}) = f$, with $u = 0$ on $\partial\Omega$, such that the exact solution is : $u(x,y) = 150x(x-1)y(y-1)(y-1/2)$. We use a Cartesian grid of Ω with 73×73 elements, uniform in x, random in y (see Figure 3.1).

In Figure 3.2, we compare the error and the residual according to the number of subdomaims, and the number of modes that are accelerated. The error versus the exact discrete solution is then order 10^{-6}, after one Aitken acceleration, and becomes of order 10^{-5} with $J' = \frac{J}{2}$, *regardless of the number of subdomains used.*

Figure 3.3 shows error and residual at the first and second iteration, for different number of modes, and different sizes of overlap. We conclude that the larger the overlap, the better is the acceleration. These results suggest that one should adaptively select the minimum number of modes to accelerate as a function of the overlap and subdomain sizes. This is an essential feature of our method that may provide parallel scalability and should be the topic of further investigation.

We are going now to consider non-matching grids and application to CFD problems.

4. Experiments with Steffensen-Schwarz and Non-overlapping grids. We consider elliptic solvers with Dirichlet bc in a non-trivial geometric domain that are component of Navier Stokes incompressible flow simulations around obstacles. A good example is the two-dimensional test case proposed by Shäfer & Turek in [10] of incompressible flow in a straight channel around a disc. The domain Ω is $(0, L_x) \times (0, L_y)$ with a circular hole of radius R centered in (x_o, y_o). $\partial\Omega^R$ is the boundary of the rectangle and $\partial\Omega^C$ is the boundary of the disc. The linear elliptic solver corresponds either to the Poisson or the Helmholtz operator $-\epsilon\Delta + Id$. Figure 4.1 gives an illustration of the two non-matching grids that we do consider. This splitting of the domain is motivated by the physics for large Reynolds number. The boundary layer is approximated on the grid Ω^C in polar coordinates and the Cartesian grid Ω^R is used to approximate the main part of the flow. The overlap between subdomains is of the order of one mesh step of Ω^R.

We denote Γ^R (resp. Γ^C), the artificial boundary of the rectangular mesh Ω^R (resp. the mesh in polar coordinates Ω^C). If L^R (resp. L^C) represents the standard finite difference approximation of our linear operator on Ω^R in Cartesian coordinates, (resp. on Ω^C in polar

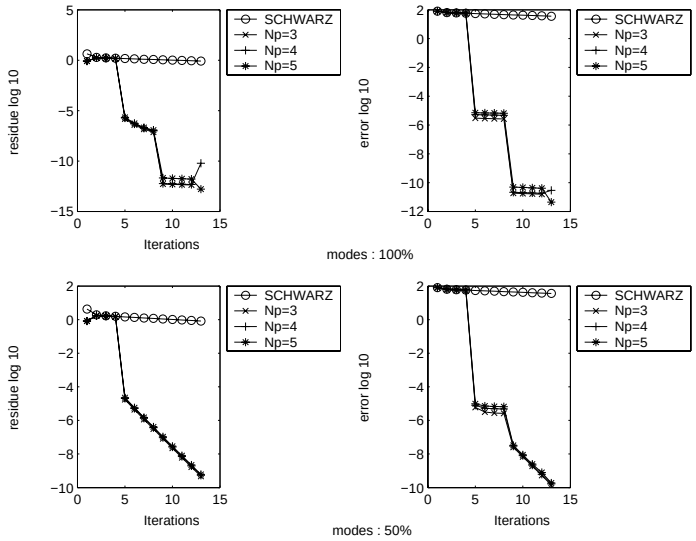


Figure 3.2: Error and residual - Np number of subdomains

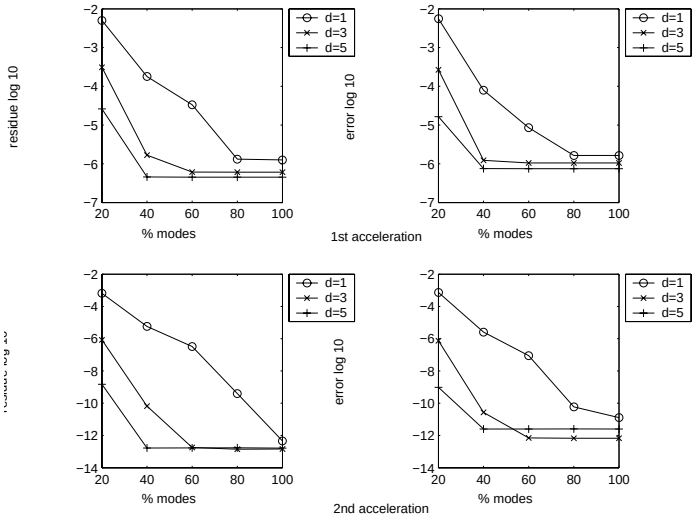


Figure 3.3: Error and residual - size d of overlap

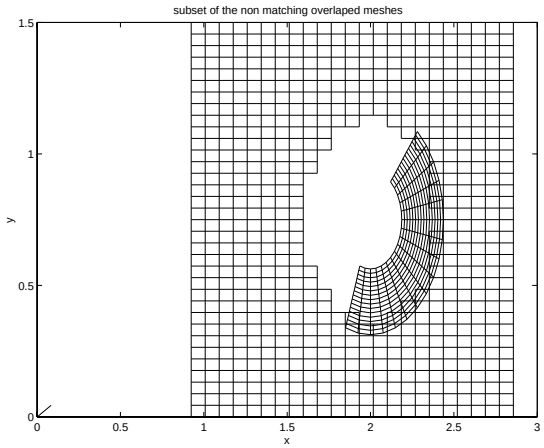


Figure 4.1: Representation of a subset of the overlapped non matching grids around the cylinder

coordinates), we close our discrete approximation problem by imposing:

$$I_R^C(U^R) = U^C \; on \; \Gamma^C, \; I_C^R(U^C) = U^R \; on \; \Gamma^R, \tag{4.1}$$

where I_C^R and I_R^C are linear second-order interpolation operators that satisfy a maximum principle. The discrete problem can be written as:

$$L^R[U^R] = f^R \; in \; \Omega^R, \; L^C[U^C] = f^C \; in \; \Omega^C, \tag{4.2}$$

with matching conditions (4.1), and Dirichlet bc on $\partial\Omega^R \bigcup \partial\Omega^C$.

The discrete solution process is the following alternate Schwarz iterative procedure:

$$L^R[U_n^R] = f^R, \; in \; \Omega^R, \; U_n^R = I_C^R(U_{n-1}^C) \; on \; \Gamma^R,$$

followed by $L^C[U_n^C] = f^C, \; in \; \Omega^C, \; U_n^C = I_R^C(U_n^R) \; on \; \Gamma^C$, using the corresponding Dirichlet bc on $\partial\Omega^R \bigcup \partial\Omega^C$, and an initial value for the artificial bc U_0^C.

From the maximum principle satisfied by the discrete operators L^R and L^C as well as the maximum principle satisfied by the interpolant operator I_C^R and I_R^C, one concludes the linear convergence of this iterative scheme to the unique solution of (4.2, 4.1), with Dirichlet bc on $\partial\Omega^R \bigcup \partial\Omega^C$. One applies then the Steffensen-Schwarz method described in [3] on the interface operator $U_{n\ |\Gamma^C}^C \rightarrow U_{n+1\ |\Gamma^C}^C$. To be more precise, let $\hat{U}^C = \sum_{k=-N/2,...,N/2} \hat{U}_k^C \; e^{ik\Theta}$ be the Fourier expansion of the discrete function U^C restricted to the circle Γ^C. The matrix P of the interface operator $(U_{n\ |\Gamma^C}^C \rightarrow U_{n+1\ |\Gamma^C}^C)$ in the set of basis function $e^{ik\Theta}, \; k = -N/2, ..., N/2$ satisfies

$$(\hat{U}_{n+1\ |\Gamma^C}^C - \hat{U}_{\infty\ |\Gamma^C}^C) = P \; (\hat{U}_{n\ |\Gamma^C}^C - \hat{U}_{\infty\ |\Gamma^C}^C).$$

One reconstructs a bandwidth approximation of P of size Z from the knowledge of the partial sequence $(\hat{U}_{0\ |\Gamma^C}^C, ..., \hat{U}_{n+Z+2\ |\Gamma^C}^C)$. The Aitken-like acceleration procedures can be written:

$$\hat{U}_{\infty\ |\Gamma^C}^C = (Id - P)^{-1} \; (\hat{U}_{n+1\ |\Gamma^C}^C - P \; \hat{U}_{n\ |\Gamma^C}^C). \tag{4.3}$$

We have observed that this Steffensen-Schwarz procedure is numerically most efficient with diagonal approximation of P. Each cycle of Steffensen-Schwarz algorithm requires two Schwarz

iterates to get from the sequence of interfaces, the diagonal approximation of P, and then one more Schwarz iterate to exploit the bc on the artificial interfaces (4.3). These Poisson or Helmholtz solvers have been used to solve an unsteady incompressible Navier Stokes (**NS**) equation written in Vorticity-Stream function ($\omega - \psi$) formulation, for the two-dimensional test case proposed by Shäfer & Turek in [10]. The main cost of the NS solution procedure corresponds to the Poisson problem for the stream function. The application of Steffensen-Schwarz procedure to ψ *at every time step* can take advantage of two interesting features. First, the initial guess for the trace of the stream function on the circle Γ^C in the iterative procedure is a second-order extrapolation in time of this trace value of ψ using the two previous time step's solution,i.e $\Psi_0^C = 2\Psi^C(t_n) - \Psi^C(t_{n-1})$. The diagonal approximation of the trace transfer operator T should be time-independent, but is in fact, with our approximation technique, solution-dependent. In practice, one can reuse the same diagonal approximation of P for O(10) time steps. The Steffensen-Schwarz cycle then reduces to two Schwarz iterates for those time steps that keep the same P approximation than the previous time step. For oscillatory flow with moderately large Reynolds number, time steps satisfying the CFL condition and grids of order 100×100, we can typically maintain the residual of order 10^{-6} with only one Steffensen-Schwarz cycle per time step.

5. Conclusion. We have presented a generalization of Aitken-Schwarz method [3] to grids that are tensorial products of one-dimensional grids with irregular mesh stepping and domain decomposition with non-matching grids. Our current work addresses the problem of the generalization of this method to unstructured meshes with Finite Volume approximation.

REFERENCES

[1] J. Baranger, M. Garbey, and F. Oudin-Dardun. Recent development on Aitken-Schwarz method. In N.Debit, M.Garbey, R.Hoppe, J.Periaux, D.Keyes, and Y.Kuznetsov, editors, *Domain Decomposition Methods in Science and Engineering*. CIMNE, UPS, Barcelone, 2001. to appear.

[2] J. Baranger, M. Garbey, and F. Oudin-Dardun. A generalisation of Aitken-Schwarz method on non uniform cartesian grids. in preparation, 2002.

[3] M. Garbey and D. Tromeur-Dervout. Operator splitting and domain decomposition for multi-clusters. In D. Keyes, A. Ecer, N. Satofuka, P. Fox, and J. Periaux, editors, *Proc. Int. Conf. Parallel CFD99*, pages 27–36. North-Holland, 2000.

[4] M. Garbey and D. Tromeur-Dervout. Two level domain decomposition for multi-clusters. In T. Chan and all editors, editors, *Domain Decomposition in Sciences and Engineering*, pages 325–339. DDM.org, 2001.

[5] P. Henrici. *Elements of Numerical Analysis*. John Wiley & Sons Inc, New York-London-Sydney, 1964.

[6] Y. A. Kuznetsov. Overlapping domain decomposition methods for fe-problems with elliptic singular perturbed operators. In R. Glowinski, Y. A. Kuznetsov, G. A. Meurant, J. Périaux, and O. Widlund, editors, *Fourth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1991. SIAM.

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

[8] P.-L. Lions. On the Schwarz alternating method. II. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods*, pages 47–70, Philadelphia, PA, 1989. SIAM.

[9] H. A. Schwarz. *Gesammelte Mathematische Abhandlungen*, volume 2, pages 133–143. Springer, Berlin, 1890. First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, volume 15, 1870, pp. 272–286.

[10] M. Shäfer and S. Turek. Benchmark computations of laminar flow around cylinder. In *Flow Simulation with High-Performance Computers II*. Vieweg, 1996.

[11] J. Stoer and R. Burlish. *Introduction to numerical analysis.* TAM 12 Springer, New York, 1980.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

35. An Aitken-Schwarz method for efficient metacomputing of elliptic equations

N. Barberou[1], M. Garbey[1,2], M. Hess[3], M. Resch[2,3], T. Rossi[4], J. Toivanen[4], D. Tromeur-Dervout[1]

1. Introduction. Metacomputing as defined by Larry Smarr [13] has been implemented in many projects among which GLOBUS is the most widely developed one [2]. Experiments with large configurations have however shown that the latency of wide area networks is prohibitively high and that substantial bandwidth can hardly be achieved [10]. From that some people have concluded that metacomputing does not make sense.

However, there are two strong arguments for metacomputing. First, with the introduction of clusters of fat nodes, varying latencies, and bandwidths are a characteristic feature of any modern hardware. Algorithms developed in metacomputing environments are therefore well suited also for such systems. Second, some large problems require a level of computing power not available on a single system. Especially in cases of industrial or natural disasters reliable predictions based on extremely large models may only be achievable on clustered supercomputers in a metacomputing environment. Such simulations of emergency scenarios will again need clever algorithms that can tolerate the bad performance of wide area communication networks.

The development of such algorithms is difficult. For Poisson or Helmholtz operators the speed of propagation of information in the spatial domain is infinite. However, two factors help to design latency-aware algorithms; firstly information propagating at infinite speed can be damped in space relatively fast, secondly, more than 90 percent of the information carried in a practical computation is noise.

In this paper, we address the significant challenge to build a fast solver for the Helmholtz operator. It combines the Aitken-Schwarz domain decomposition method [4, 5] associated with the Partial Solution variant of Cyclic Reduction (PSCR) method [11, 12] on large scale parallel computers.

2. Numerical methods.

2.1. The PDC3D inner solver.

The parallel solver PDC3D developed by T. Rossi and J. Toivanen [11] [12] following the ideas of Y. Kuznetsov [9] and P. Vassilevski [15] is a parallel fast direct solution method for linear systems with separable block tridiagonal matrices. Such systems appear, for example, when discretizing the Poisson equation in a rectangular domain using the seven-point finite difference scheme or piecewise linear finite elements on a triangulated, possibly nonuniform rectangular mesh. The method under consideration has the arithmetical complexity $\mathcal{O}(N \log^2 N)$ and is closely related to the cyclic reduction method. But instead of using the matrix polynomial factorization the so-called partial solution technique is employed. Based on the analysis of [12], the radix-4 variant is chosen for the parallel implementation using the MPI standard. However, the method works for blocks of arbitrary dimension. [11] [12] show that the sequential efficiency and numerical stability of the PSCR method compares favorably to the well-known BLKTRI implementation of the generalized cyclic reduction method. The current PDC3D code is using a two-dimensional domain decomposition. It requires a high performance communication network mainly because of global reduction operations to gather the partial solution. Its very good scalability has been shown on a CrayT3E (table 4.2).

[1]MCS/CDCSP/ISTIL - University Lyon 1, 69622 Villeurbanne, France
[2]Department of Computer Science, University of Houston, USA
[3]HLRS, University of Stuttgart, Germany
[4]Department of Mathematical Information Technology, University of Jyväskylä, Finland

2.2. The Aitken-Schwarz outer solver. Following a different approach than the PDC3D solver, M. Garbey and D. Tromeur-Dervout [4] [5] [6] have developed an Aitken-Schwarz algorithm (respectively Steffensen-Schwarz) for the Helmholtz operator, (respectively for general linear and non linear elliptic problems) that is highly tolerant to low bandwidth and high latency. In the specific case of separable elliptic operators, the Aitken-Schwarz algorithm might be less efficient in terms of arithmetic complexity than PDC3D as the number of processors increases [5], but it is very competitive when using $O(10)$ sub-domains.

Let us recall briefly the salient feature of the method in 1D to solve

$$L[u] = f, \quad on \ \ \Omega = [0,1] \tag{2.1}$$

$$B[u] = 0 \tag{2.2}$$

where operator B denotes the linear operator on boundary conditions, and L is some linear operator. Let $\Omega_i = (x_i^l, x_i^r),\ i = 1..q$ be a partition of Ω with $x_2^l < x_1^r < x_3^l < x_2^r, ..., x_q^l < x_{q-1}^r$. We consider the additive Schwarz algorithm:

$for\ i = 1..q, do \qquad L[u_i^{n+1}] = f\ in\ \Omega_i,\ u_i^{n+1}(x_i^l) = u_{i-1}^n(x_i^l),\ u_i^{n+1}(x_i^r) = u_{i+1}^n(x_i^r),$

enddo

Let $u_i^{l,n+1} = u_i^{n+1}(x_i^l),\ u_i^{r,n+1} = u_i^{n+1}(x_i^r)$ and $\tilde{u}^n$ (respectively $\tilde{u}$) be the n iterated (respectively exact) solution restricted at the interface, i.e

$$\tilde{u}^n = (u_2^{l,n}, u_1^{r,n}, u_3^{l,n}, u_2^{r,n}, ..., u_q^{l,n}, u_{q-1}^{r,n})$$

The operator $\tilde{u}^n - \tilde{u} \to \tilde{u}^{n+1} - \tilde{u}$ is linear. Let us denote its matrix by P. P has the following pentadiagonal structure:

$$\left|\begin{array}{ccccccccc}
0 & \delta_1^r & 0 & 0 & .... & & & & \\
\delta_2^{l,l} & 0 & 0 & \delta_2^{l,r} & ... & & & & \\
\delta_2^{r,l} & 0 & 0 & \delta_2^{r,r} & ... & & & & \\
 & & & & & & & & \\
 & & & ... & \delta_{q-1}^{l,l} & 0 & 0 & \delta_{q-1}^{l,r} & \\
 & & & ... & \delta_{q-1}^{r,l} & 0 & 0 & \delta_{q-1}^{r,r} & \\
 & & & ... & 0 & 0 & \delta_q^r & 0 &
\end{array}\right|$$

The sub-blocks $P_i = \left|\begin{array}{cc} \delta_i^{l,l} & \delta_i^{l,r} \\ \delta_i^{r,l} & \delta_i^{r,r} \end{array}\right|$ $i = 2..q-1$ can be computed with 3 Schwarz iterates as follows.

We have $(u_{i-1}^{r,n+1} - \tilde{u}_{i-1}^r, u_{i+1}^{l,n+1} - \tilde{u}_{i+1}^l)^t = P_i(u_i^{l,n} - \tilde{u}_i^l, u_i^{r,n} - \tilde{u}_i^r)^t$. Therefore

$$\begin{pmatrix} u_{i-1}^{r,n+3} - u_{i-1}^{r,n+2} & u_{i-1}^{r,n+2} - u_{i-1}^{r,n+1} \\ u_{i+1}^{l,n+3} - u_{i+1}^{l,n+2} & u_{i+1}^{l,n+2} - u_{i+1}^{l,n+1} \end{pmatrix} = P_i \begin{pmatrix} u_i^{l,n+2} - u_i^{l,n+1} & u_i^{l,n} - u_i^{l,n} \\ u_i^{r,n+2} - u_i^{r,n+1} & u_i^{r,n} - u_i^{r,n} \end{pmatrix}$$

In practice the last matrix on the right hand side of the previous equation is non singular and P_i can be computed, but it cannot be guaranteed. However, one can always compute beforehand the coefficients of P_i -see [4]. For the Helmholtz operator $L[u] = u'' - \lambda u$, or generally speaking elliptic problems with constant coefficients, the matrix P is known analytically.

From the equality $\tilde{u}^{n+1} - \tilde{u} = P(\tilde{u}^n - \tilde{u})$, one obtains the generalized Aitken acceleration as follows:

$$\tilde{u}^\infty = (Id - P)^{-1}(\tilde{u}^{n+1} - P\tilde{u}^n). \tag{2.3}$$

If the additive Schwarz method converges, then $||P|| < 1$ and $Id - P$ is non singular.

The aim of our paper is therefore to combine the two methods in order to have a highly efficient solver for the Helmholtz operator for metacomputing environments. This solver can be used to solve the elliptic equations satisfied by the velocity components of a incompressible Navier Stokes (NS) code written in velocity-vorticity formulation. The elliptic part of such an NS solver is usually the most time consuming part as these equations must be solved very accurately to satisfy the velocity divergence free constraint. Similarly this solver can be used for the pressure solve in NS written in velocity pressure formulation using the projection method.

Our parallel implementation is then as follows: first one decomposes the domain of computation into a one-dimensional domain decomposition (DD) of $O(10)$ macro sub-domains. This first level of DD uses the Aitken Schwarz algorithm. The macro sub-domains are distributed among clusters or distinct parallel computers. Secondly, each macro sub-domain is decomposed into a two-dimensional DD: this level of DD uses the PDC3D solver. Globally we have a three-dimensional DD and a two-level algorithm that matches the hierarchy of the network and access to memory.

3. Hardware and software components for metacomputing. Metacomputing in heterogeneous environments introduces problems that are partly similar to those well known from clusters and partly very new and specific. Among the most critical ones is the concurrent scheduling of resources. Another one is the mapping of processes to processors. For these problems we refer to projects like GLOBUS [2], Legion [7] or TME [14]. In this section we focus on communication.

The communication can be done using several MPI-implementations [1] [3][8]. All these implementations provide a simple way to start and run MPI-applications across a metacomputer. They differ, however, in completeness of implementing the MPI-standard and in the degree of optimization. For the experiments described in this paper we have chosen PACX-MPI [3] from the High Performance Computing Center Stuttgart (HLRS) which allows metacomputing for MPI-codes without any changes [10]. Based on the experience of several projects the library relies on four main concepts:

- For the programmer the metacomputer looks like any other parallel system.
- Usage of communication daemons to clearly split external from internal communication and ease support of different communication protocols (e.g. TCP/IP, ATM).
- Use of native MPI for internal communication and standard protocols for external communication. MPI-implementations based on native protocols typically are superior in performance to any other approach.
- Optimized global communication by minimizing traffic between systems.

4. Results. We are going to present some numerical experiences in metacomputing environments. For simplicity, we restrict ourselves to a network of two or three parallel computers. For large scale metacomputing experiments, we are using the hardware described in Table 4.1. Once and for all we denote **CrayS** the Cray of HLRS in Stuttgart University, **CrayN** of the von Neumann Institute in Jülich (NIC), **CrayP** of the Pittsburgh center of high performance computing in USA and **CrayH** the Cray T3E of the National Scientific Computing Center of Finland at CSC. The goal is to demonstrate on classical problems that make intense use of Poisson solves, that efficient numerical results and high performance are attainable in a metacomputing environment with standard network connections.

4.1. Fast Poisson solver experiment. We make three hypotheses:

- First, we restrict ourselves to the Poisson problem, i.e the Helmholtz operator with $\lambda = 0$. As a matter of fact, it is the worst situation for metacomputing because any perturbation at an artificial interface decreases linearly in space, instead of exponentially as for the Helmholtz operator.
- Second, we do a priori load balancing on the heterogeneous network of Cray supercom-

Machine	#proc	MHz	internal latency	internal bandwidth	Localization
CrayS	512	450	12 μs	320 MB/s	HLRS, Stuttgart
CrayP	512	450	12 μs	320 MB/s	PSC, Pittsburgh
CrayH	512	375	12 μs	320 MB/s	CSC, Helsinki
CrayN	512	375	12 μs	320 MB/s	NIC , Jülich

Table 4.1: System configuration at Stuttgart, Pittsburgh, Helsinki, Jülich.

puters. We verified that PDC3D solver is roughly 30% slower on CrayH than on CrayS for our test cases. The number of grid points in the Aitken domain decomposition is balanced in such way that PDC3D in each parallel computer uses approximatively the same CPU time.

• Third, and this is a key point, we are running our metacomputing experiment on two or three supercomputers with the existing ordinary area network. During all our experiments, the bandwidth fluctuated in the range of $(1.6Mb/s - 5.Mb/s)$ and the latency was about $30ms$.

Let us show that a fast elliptic solver that is quasi optimal on a single parallel computer gives poor performance in a metacomputing environment.

The PDC3D solver is an almost optimal solver with good parallel scalability in parallel computers having a good balance of network and processor. By analyzing the PDC3D algorithm, we can deduce that the number of communications per processor is of the order of $\log(p_x) + (N_x/p_x)\log(N_x)\log(p_y)$ and the total length of all messages for one processor is of the order of $(N_y/p_y)N_z\log(p_x) + (N_x/p_x)\log(N_x)N_z\log(p_y)$ floating point numbers.
Each processor stores $(N_x/n_x)(N_y/n_y)N_z$ floating point numbers which is considerably more than the amount of communication. Thus, the computational work per processor greatly exceeds the amount of data to be transferred. This leads to a rather efficient code as can be seen in Table 4.2, where the results of experiments made on CrayS are presented. Also, it can be seen from the communication estimates and the numerical results that it is favorable to choose p_y to be larger than p_x.

The PDC3D solver obviously cannot be used efficiently in a metacomputing environment. Based on the performances -see Table 4.2- of the PDC3D on CrayS, we select the most efficient data distribution and run the same problem on the metacomputing architecture, i.e on CrayS and CrayH that share equally the total number of processors used. Table 4.2 gives a representative set of the performance of PDC3D on the metacomputing architecture (CrayS-CrayH). We conclude that no matter what the number of processors, most of the elapsed time is spent in communication between the two computer sites. This conclusion holds for a problem of smaller size, that is 256^3: the elapsed time grows continuously from 0.76 s, up to 18.73 s with 512 processors. Obviously the PDC3D performance degrades drastically when using a slow network. In the following we show how Aitken-Schwarz can overcome this problem.

4.2. Aitken-Schwarz experiment. We proceed with a performance evaluation of our two level domain decomposition method combining Aitken-Schwarz and PDC3D (**AS**). We define the barrier between low and medium size frequencies in each space variable to be 1/4 of the number of waves; We do not accelerate the highest half of the frequencies. We checked that the impact on the numerical error against an exact polynomial solution is in the interval $[10^{-7}, 10^{-6}]$ for our test cases with minimum overlap between macro sub-domains. Let us give first the performance of our method on a single Cray.

Figure 4.1 gives the elapse time for the following growing size of Poisson problems $158 \times 192 \times 384$, $316 \times 192 \times 384, 633 \times 192 \times 384$. When increasing the number of do-

No metacomputing: localization of processors : 100% $\in$ **CrayS**		
128 procs ($p_x \times p_y$)	256 procs ($p_x \times p_y$)	512 procs ($p_x \times p_y$)
25.9 s (4 x 32)	17.6s (4 x 64)	
22.0 s (16 x 8)	11.5s (16 x 16)	7.2 (16 x 32)
21.8 s (64 x 2)	11.2s (64 x 4)	**5.77 (64 x 8)**
Metacomputing: localization of processors :50% $\in$ **CrayS** and 50% $\in$ **CrayH**		
72.0s (64 x 2)	77.2s (64 x 4)	**75.1 (64 x 8)**

Table 4.2: **Elapsed time in (s) for the PDC3D solver on CrayS and on metacomputing architecture (CrayS, CrayH) to solve a problem of global size** $511 \times 511 \times 512$

mains in the same proportion as the number of processors the elapsed time remains constant. Our solver has therefore good scalability properties on the Cray T3E. Further, our method requires no more than 6 seconds to solve the problem with 46 10^6 unknowns on a Cray T3E with 256 processors running at 450 MHz. Figure 5.2 shows also that the speedup of our solver is fairly good. Now let us proceed with the metacomputing experiment. We make the two following hypothesis:

• First we fix the size of our problem in such a way that it cannot be solved on one single computer at our disposal. As a matter of fact, we use almost all memory available on our network of supercomputers.

• Second, we focus our study in this context on the extensibility properties of our direct linear solver. To benefit of the Gustafson law for scalability, we believe that a direct measurement of the speedup is not appropriate. We have also not estimated the speedup from a model analysis, because our two level domain decomposition method is too complex to give any realistic estimate in a metacomputing environment. Table 4.3 summarizes our results. Let us notice that each case has been run several times and our measurements give elapsed time with a variation of few seconds, depending on the quality of the network during the experiment. We provide here an average value that corresponds to two or three consecutive runs excluding the cases where the network died during the runs.

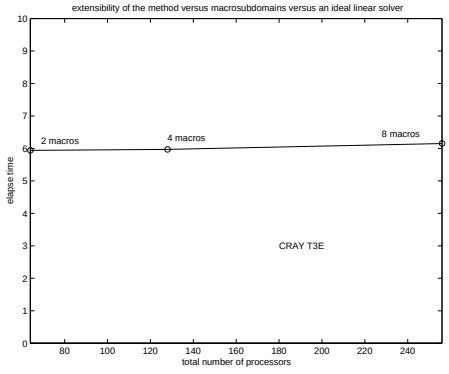


Figure 4.1: Extensibility of the Aitken Schwarz algorithm for the 3D Poisson problem

Our two main observations are as follows:

• We have in our experiments an irreducible overhead that varies from $17s$ to $24s$ and

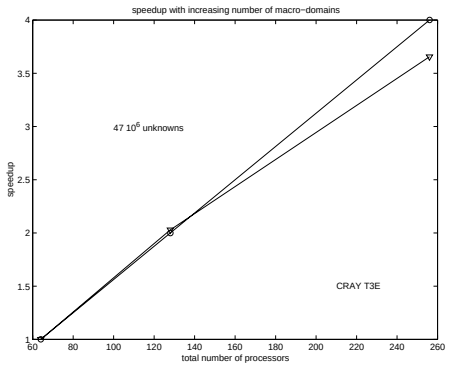


Figure 4.2: Speedup of the Aitken Schwarz algorithm for the 3D Poisson problem

that depends mostly on the speed of the network that interlinks our supercomputers. We recall that the bandwidth of the network was in the range of 2 to 5 Mb/s. This overhead is quasi independent of the size of the problem considered here.

• Beside the overhead due to the network between distant sites, we observe an excellent scalability of our Poisson solver. This result is a combination of two factors: first the arithmetic complexity of our solver grows almost linearly with the number of macrosubdomains. Second the ratio of computation time per macrodomains to communication time is large even with a slow network and fast supercomputers.

Finally, we would like to underline that the conditions of our experiments were by definition difficult. It is not realistic to stop simultaneously the production of several national computing centers for long. We had therefore few windows for experiments with few hours each time. We are extremely grateful to all centers participating in these sets of experiments for their cooperations. Further, to our knowledge, there are no other known results of efficient large scale metacomputing simulations of PDE problems that assume tidily coupled computation as it is the case in a Poisson solver. This work is currently extended to 3D Navier Stokes equation using our Poisson solver as a preconditioner.

5. Conclusions. In this paper, we demonstrate the feasibility of numerical efficient metacomputing between distantly located parallel computing resources for tidily coupled problems as Helmholtz solvers. In order to achieve this result, we use the best components we can get at each stage that is to say:

• PDC3D : one of the best efficient solvers for separable operators that scales well on homogeneous computers with fast communication network.

• PACX-MPI to achieve excellent performance of MPI communication on both the internal and external communication network.

• Aitken-Schwarz that is numerically efficient, tolerant to slow communication networks and high latencies and scales well up to $O(10)$ domains.

High latencies and slow communication networks with fluctuating bandwidth shared by thousands of users are typical difficulties encountered in grid computing. The Aitken-Schwarz DDM seems to be an example of a numerical tools that adresses to such difficulties.

Acknowledgment: The authors would like to thank the John von Neumann Institute and the Pittsburgh Supercomputing Center for providing access to their systems for experiments. All authors would like to thank their home organizations Cines, CSC and HLRS for support in this work. The work of T. Rossi and J. Toivanen was supported by the Academy of Finland, grants #43066, #53588 and #66407. The work of N. Barberou was supported

# of pts in x per MD			# of MD per machine			# of pts in y, z per proc.		# of proc. per MD		
N_{xJ}	N_{xS}	N_{xP}	MJ	MS	MP	n_y	n_z	p_y	p_z	Time (s)
225	321	0	1	1	0	49	49	16	16	60
225	321	0	1	2	0	49	49	16	16	58
0	321	0	2	0	0	49	49	8	16	30 .4
225	321	0	1	1	0	49	49	8	16	47
225	321	0	1	2	0	49	49	8	16	47
0	321	0	0	2	0	49	49	8	8	27.3
0	321	0	0	3	0	49	49	8	8	27.3
0	321	0	0	4	0	49	49	8	8	27.2
225	321	0	4	4	0	49	49	8	8	51
0	321	0	0	2	0	43	43	16	16	25,4
225	321	0	2	2	0	43	43	16	16	50
225	321	321	2	2	1	43	43	16	16	59
0	401	0	0	2	0	43	43	16	16	30.5
281	401	401	2	2	1	43	43	16	16	62

Table 4.3: Extensibility of the Aitken-Schwarz on the Poisson problem in a metacomputing framework

by the "abondement de l'ANVAR". The work of D. Tromeur-Dervout was supported by the ACI-Grid project of the French Ministry.

REFERENCES

[1] I. Foster and N. Karonis. A Grid-Enabled MPI: Message Passing in Heterogeneous Distributed Computing Systems.

[2] I. Foster and C. Kesselman. Globus: A metacomputing infrastructure toolkit. *Intl J. Supercomputer Applications*, 11(2):115–128, 1997.

[3] E. Gabriel, M. Resch, T. Beisel, and R. Keller. Distributed computing in a heterogenous computing environment. In V. Alexandrov and J. Dongarra, editors, *Recent Advances in Parallel Virtual Machine and Message Passing Interface, Lecture Notes in Computer Science*, pages 180–188. Springer, 1998.

[4] M. Garbey and D. Tromeur-Dervout. Two Level Domain Decomposition for Multi-cluster. In H. K. T. Chan, T. Kako and O. Pironneau, editors, *Proc. Int. Conf. on Domain Decomposition Methods DD12*, pages 325–340. DDM org, 2001.

[5] M. Garbey and D. Tromeur-Dervout. Aitken-Schwarz Method on Cartesian grids. In N. Debit, M. Garbey, R. Hoppe, J. Périaux, and D. Keyes, editors, *Proc. Int. Conf. on Domain Decomposition Methods DD13*, pages 53–65. CIMNE, 2002.

[6] M. Garbey and D. Tromeur-Dervout. On some Aitken like acceleration of the Schwarz method. *Int. J. of Numerical Methods in Fluids*, 2002. to appear.

[7] A. Grimshaw, A. Ferrari, F. Knabe, and M. Humphrey. Legion: An Operating System for Wide-Area Computing. Technical Report CS-99-12, University of Virginia, 1999.

[8] H. Koide, T. Imamura, and H. Takemiya. MPI based communication library for a heterogeneous parallel computer cluster, stampi. Technical report, Japan Atomic Energy Research Institute, 1997. http://ssp.koma.jaeri.go.jp/en/stampi.html.

[9] Y. Kuznetsov. Numerical methods in subspaces. *Vychislitel'nye Processy i Sistemy II*, 1985. G. I. Marchuk editor, Nauka, Moscow.

[10] S. Pickles, J. Brooke, F. Costen, E. Gabriel, M. Muller, M. Resch, and S. Ord. Metacomputing across intercontinental networks. *Future Generation Computer Systems*, 17(8):911–918, 2001.

[11] T. Rossi and J. Toivanen. A Nonstandard Cyclic Reduction Method, its Variants and Stability. *SIAM J. Matrix Anal. Appl.*, 3:628–645, 1999.

[12] T. Rossi and J. Toivanen. A Parallel Fast Direct Solver for Block Tridiagonal Systems with Separable Matrices of Arbitrary Dimension. *SIAM J. Sci. Comput.*, 5:1778–1796, 1999.

[13] L. Smarr and C. Catlett. Metacomputing. *Communications of the ACM*, 35(6):45–52, 1992.

[14] H. Takemiya, T.Imamura, and H. Koid. TME a visual programming and execution environment for a meta-application. Technical report, Jaeri Internal Report, 2000.

[15] P. S. Vassilevski. Fast Algorithm for Solving a Linear Algebraic Problem with Separable Variables. *C.R. Acad. Bulgare Sci.*, 37:305–308, 1984.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

36. The Mortar Method with Approximate Constraint

S. Bertoluzza[1], S. Falletta[2]

1. Introduction. The *Mortar* method is a non conforming approach for solving PDEs in domain decomposition. It consists in imposing weak continuity across the interfaces by requiring that the jump of the solution along two adjacent subdomains is orthogonal to a suitable *Multiplier* space. This method is particulary well suited for choosing different kinds of discretizations in each subdomain.
We will consider here the case of coupling finite elements with wavelets, which will allow us to overcome the limit of application of wavelet basis to tensor product domains, using FEM for more complicated shapes.
The constraint operator, which is used to impose weak continuity, leads to the problem of computing integrals of product of functions of different type and this can be extremely technical or even impossible. This is the case of wavelet/finite elements coupling, where such integral can not be computed exactly due to the particular nature of wavelets, which are not known in closed form. We will propose here to approximate it by a technique that is particulary well suited for the case we are treating. Moreover we will show that the use of such a technique allows to easily integrate new type of functions in existing codes, without the need of providing specific tools for computing the integrals of the product of a function of the new type with all functions of each of the types already present in the code.
The paper is organized as follows: in Section 2 we introduce the general context in the case of a simple splitting of the domain into two subdomains, introducing the approximate constraint and analizing the error estimate. In Section 3 we consider the particular case of coupling Wavelet and Finite Element disctretiations by studing the explicit form of the approximate constraint in both the cases of Wavelet type discretization in the Master subdomain and Finite Element discretization in the Slave one, and viceversa. Finally, Section 4 is devoted to a brief overwiev of the C++ code implemented for such an approach.

2. The mortar method with approximate constraint. Let $\Omega \subset \mathbb{R}^2$ be a bounded polygonal domain and consider the model problem: given $f \in L^2(\Omega)$ find $u : \Omega \to \mathbb{R}$ s.t.

$$-\nabla \cdot (a\nabla u) = f \ \text{ in } \Omega, \qquad u = 0 \ \text{ on } \Gamma = \partial\Omega, \tag{2.1}$$

where for simplicity we assume that the matrix a is constant symmetric positive definite. We consider here a very simple example of non conforming domain decomposition. More precisely consider a splitting of Ω in two subdomains as $\bar{\Omega} = \bar{\Omega}_+ \cup \bar{\Omega}_-, \qquad$ with $\qquad \gamma = \partial\Omega_+ \cap \partial\Omega_-$.

Denote by V_h^+ and V_h^-

$$V_h^+ \subset H^1_\Gamma(\Omega_+) = \{u \in H^1(\Omega_+) : \ u = 0 \text{ on } \Gamma \cap \underline{\Omega_+}\} \tag{2.2}$$

$$V_h^- \subset H^1_\Gamma(\Omega_-) = \{u \in H^1(\Omega_-) : \ u = 0 \text{ on } \Gamma \cap \underline{\Omega_-}\} \tag{2.3}$$

the two discrete spaces chosen for approximating u in Ω_+ and Ω_- respectively, and let $M_h \subset H^{-1/2}(\gamma)$, with $dim(M_h) = dim(V_h^-|_\gamma)$ be a suitable multiplier space — which in the mortar method is obtained from a subspace of $V_h^-|_\gamma$ with suitable modifications at the vertices of γ ([1]), or which coincides, in a more general formulation, with a suitable "dual space" of $V_h^-|_\gamma$ ([3], [5]). In the classical formulation of the mortar method, the approximation of the solution of (2.1) is sought in the constrained space $\mathcal{X}_h$ defined as

$$\mathcal{X}_h = \{u : \ u|_{\Omega_+} \in V_h^+, \ \ u|_{\Omega_-} \in V_h^-, \ \ \int_\gamma [u]\lambda = 0 \ \ \forall \lambda \in M_h\},$$

[1]I.A.N - C.N.R Pavia, Italy. aivlis@dragon.ian.pv.cnr.it
[2]Dip. MAtematica Università di Pavia, Italy. falletta@dragon.ian.pv.cnr.it

where introducing the notation $u^+ = u|_{\Omega_+}$ and $u^- = u|_{\Omega_-}$, $[u] = u^+|_\gamma - u^-|_\gamma$ denotes the jump of the function u across the interface γ. The solution u to problem (2.1) is approximated by looking for u_h in $\mathcal{X}_h$ such that for all $v_h \in \mathcal{X}_h$ it holds

$$\int_{\Omega_+} a\nabla u_h \nabla v_h + \int_{\Omega_-} a\nabla u_h \nabla v_h = \int_\Omega f v_h.$$

In the solution of the linear system resulting from such problem the need arises eventually of computing the integrals appearing in the constraint

$$\int_\gamma (u_h^+ - u_h^-)\lambda = 0, \qquad \forall \lambda \in M_h. \tag{2.4}$$

Since in the mortar method the multiplier space M_h is strongly related to the "slave" space V_h^-, it is not reasonable to assume that the integrals of the products $u_h^-\lambda$ are computable in practice. This is not necessarily the case of the products $u_h^+\lambda$ where functions originating from totally unrelated spaces are involved. We will concentrate here on approximating this term in the constraint.

In order to do that, let us introduce two auxiliary spaces $U_\delta^- \subset L^2(\gamma)$ and $U_\delta^+ \subset L^2(\gamma)$ depending on a parameter δ, which we assume to have the same finite dimension and to satisfy

$$\inf_{\zeta\in U_\delta^-} \sup_{\eta\in U_\delta^+} \frac{\int_\gamma \zeta\ \eta}{\|\zeta\|_{H_{00}^{1/2}(\gamma)}\|\eta\|_{H^{-1/2}(\gamma)}} \geq \alpha > 0. \tag{2.5}$$

Assume that the two auxiliary spaces are chosen in such a way that the integrals of the form $\int_\gamma \zeta\ \eta$ are computable provided either $\zeta \in V_h^+|_\gamma$ and $\eta \in U_\delta^+$ or $\zeta \in V_h^-|_\gamma$ and $\eta \in U_\delta^-$. For all $\zeta \in L^2(\gamma)$ let $P^-(\zeta) \in U_\delta^-$ be the unique element in U_δ^- such that

$$\int_\gamma P^-(\zeta)\ \eta = \int_\gamma \zeta\ \eta, \qquad \forall \eta \in U_\delta^+. \tag{2.6}$$

We propose here to approximate the integral of the product $u_h^+\lambda$ with the integral of $P^-(u_h^+)\lambda$ (where, by abuse of notation we will write u_h^+ instead of $u_h^+|_\gamma$). The constraint (2.4) is then replaced by the approximated constraint

$$\int_\gamma (P^-(u_h^+) - u_h^-)\ \lambda = 0, \qquad \forall \lambda \in M_h, \tag{2.7}$$

which corresponds to defining a new constrained space as

$$\mathcal{X}_h^* = \{u :\ \ u|_{\Omega_+} \in V_h^+,\ \ u|_{\Omega_-} \in V_h^-,\ \ \int_\gamma (P^-(u^+) - u^-)\lambda = 0\ \ \forall \lambda \in M_h\}, \tag{2.8}$$

and approximating the solution to (2.1) by the solution of the following discrete problem: find $u_h \in \mathcal{X}_h^*$ such that for all $v_h \in \mathcal{X}_h^*$ it holds

$$\int_{\Omega_+} a\nabla u_h \nabla v_h + \int_{\Omega_-} a\nabla u_h \nabla v_h = \int_\Omega f v_h. \tag{2.9}$$

Denoting by $\|\cdot\|_{1,*} = \|\cdot\|_{H^1(\Omega_+)} + \|\cdot\|_{H^1(\Omega_-)}$ the broken H^1 norms, we can prove the following bound [2]

Theorem 2.1 *Let the multiplier space M_h be chosen in such a way that the following assumptions are satisfied:*

(A1) *there exists a bounded projection* $\pi : L^2(\gamma) \to V_h^-|_\gamma$, *such that for all* $\eta \in H_{00}^{1/2}(\gamma)$ *and for all* $\lambda \in M_h$ *it holds that*

$$\int_\gamma (\eta - \pi(\eta))\ \lambda = 0, \qquad \text{and} \qquad \|\pi\eta\|_{H_{00}^{1/2}(\gamma)} \lesssim \|\eta\|_{H_{00}^{1/2}(\gamma)}. \tag{2.10}$$

(A2) *there exists a discrete lifting* $R_h : V_h^-|_\gamma \to V_h^-$ *such that for all* $\eta \in V_h^-|_\gamma$, $\|R_h\eta\|_{H^1(\Omega_-)} \lesssim \|\eta\|_{H_{00}^{1/2}(\gamma)}$.

Moreover let the two auxiliary spaces U_δ^+ *and* U_δ^- *be chosen in such a way that the following Jackson type inequality holds for some* $\tilde{R}, R \geq 1/2$*: for all* r, $1/2 \leq r \leq R$ *(resp. for all* $\tilde{r}$, $-1/2 \leq \tilde{r} \leq \tilde{R}$*)*

$$\forall \eta \in H_0^r(\gamma), \qquad \inf_{\eta_\delta \in U_\delta^-} \|\eta - \eta_\delta\|_{H^{1/2}(\gamma)} \lesssim \delta^{r-1/2} \|\eta\|_{H^r(\gamma)}, \tag{2.11}$$

$$\forall \eta \in H^{\tilde{r}}(\gamma), \qquad \inf_{\eta_\delta \in U_\delta^+} \|\eta - \eta_\delta\|_{H^{-1/2}(\gamma)} \lesssim \delta^{\tilde{r}+1/2} \|\eta\|_{H^{\tilde{r}}(\gamma)}, \tag{2.12}$$

Then, if u_h *is the solution of problem (2.9), and if the solution* u *of problem (2.1) verifies* $u \in H^s(\Omega)$ *for some* s, $2 \leq s \leq \min\{\tilde{R} + 3/2, R + 1/2\}$, *the following error estimate holds:*

$$\begin{aligned} \|u - u_h\|_{1,*} \quad \lesssim \quad & \delta^{s-1} \|u\|_{H^s(\Omega)} + \inf_{\lambda \in M_h} \|\partial_{\nu_a} u - \lambda\|_{H^{-1/2}(\gamma)} \\ & + \inf_{v_h \in V_h^+} \|u - v_h\|_{H^1(\Omega_+)} + \inf_{v_h \in V_h^-} \|u - v_h\|_{H^1(\Omega_-)} \end{aligned} \tag{2.13}$$

where ∂_{ν_a} *denotes the trace on* γ *of outer co-normal derivative to the subdomain* Ω_+.

Remark 2.1 *The extremely simple configuration considered (only two subdomains), hides some of the issues related to the analysis of the mortar method in more general configurations — namely the treatment of cross points. However, the approach used and the results obtained in this paper carry over to more complex cases (with the presence of cross-points), with, in the worse case, a loss of a logarithmic factor in the error estimate.*

3. Wavelet/FEM Coupling. Let us now consider the case of Wavelet/FEM coupling. In order to get two suitable auxiliary spaces, we will in such a case need a couple of biorthogonal multiresolution analyses $\{V_j\}_{j \geq j_0}$ and $\{\tilde{V}_j\}_{j \geq j_0}$ of $L^2(\gamma)$ with the following characteristics ([4]).

- $V_j \subset H^1(\gamma)$ is the subspace of P1 finite elements on the uniform grid $\mathcal{G}_j$ obtained by splitting γ into 2^j equal segments;
- $\tilde{V}_j \subset H^1(\gamma)$ is a subspace having the same dimension as V_j, which is *biorthogonal* to V_j in the following sense: denoting by $e_{j,k}$ $(k = 0, \ldots, 2^j)$ the nodal basis function in V_j corresponding to the k-th point in the grid $\mathcal{G}_j$, the space $\tilde{V}_j$ has a Riesz's basis $\{\tilde{e}_{j,k},\ k = 0, \ldots, 2^j\}$ which satisfies $\int_\gamma e_{j,k}\ \tilde{e}_{j,k'} = \delta_{kk'}$, $\forall k, k' = 0, \ldots, 2^j$;
- the functions $\tilde{e}_{j,k}$ can be obtained as linear combination of the restriction to γ (identified through a suitable mapping with the interval $(0,1)$), of the translates and contracted (with a contraction factor 2^j) of a compactly supported function $\tilde{e}$, which we assume to be *refinable*, to verify, for suitable values of the coefficients h_k, $\tilde{e}(s) = \sum_{k=0}^N h_k\ \tilde{e}(2s - k)$;

- $\tilde V_j$ satisfies a Strang-Fix condition of order M, that is it contains polynomials up to degree $M-1$, while, of course, V_j contains polynomials of order 1.

Let $V_j^0 = V_j \cap H_0^1(\gamma)$ and $\tilde V_j^0 = \tilde V_j \cap H_0^1(\gamma)$, it is possible (see [3]) to construct two subspaces $V_j^* \subset V_j$ and $\tilde V_j^* \subset \tilde V_j$ satisfying $dim(V_j^*) = dim\tilde V_j^0$, $dim(\tilde V_j^*) = dim V_j^0$, and

$$\inf_{\eta\in V_j^0}\sup_{\zeta\in\tilde V_j^*}\frac{\int_\gamma \eta\,\zeta}{\|\eta\|_{H_{00}^{1/2}(\gamma)}\|\zeta\|_{H^{-1/2}(\gamma)}}\geq\alpha_1,\qquad \inf_{\eta\in\tilde V_j^0}\sup_{\zeta\in V_j^*}\frac{\int_\gamma \eta\,\zeta}{\|\eta\|_{H_{00}^{1/2}(\gamma)}\|\zeta\|_{H^{-1/2}(\gamma)}}\geq\alpha_2,$$

in such a way that they satisfy a Strang-Fix condition with the same order as V_j and $\tilde V_j$ respectively. Moreover it is possible to construct Riesz's bases $e^*_{j,k}$ and $\tilde e^*_{j,k}$ for V_j^* and $\tilde V_j^*$ respectively in such a way that the two following biorthogonality relations hold:

$$\int_\gamma e_{j,k}\;\tilde e^*_{j,k'}=\delta_{k,k'},\qquad \int_\gamma \tilde e_{j,k}\;e^*_{j,k'}=\delta_{k,k'},\qquad \forall k,k'=1,\dots,2^j-1. \tag{3.1}$$

Thanks to the *refinable* property of the function $\tilde e$, it is well known that it is possible to compute integrals of the product of a wavelet type function times any function in $\tilde V_j$ (and therefore in $\tilde V_j^0$ and in $\tilde V_j^*$), while the product of a function in V_j, V_j^0 and V_j^* with a finite element type function can be computed by standard techniques, already implemented in the mortar method for finite elements with non-matching grids.

For using such spaces for coupling wavelets and finite elements in the mortar method we distinguish two cases.

Case 1. FEM master / Wavelet slave. In this case we set V_h^+ to be a finite element space on an unstructured, non uniform grid while V_h^- and M_h are two wavelet type spaces. The approximate integration is done by setting $U_\delta^+ = V_j^0$ and $U_\delta^- = \tilde V_j^*$.

Case 2. Wavelet master / FEM slave. In this case we set V_h^+ to be a wavelet type space, while V_h^- and M_h are two finite element spaces defined on unstructured, non uniform grid, the grid for M_h being the trace on γ of the grid for V_h^-. The approximate integration is this time performed by setting $U_\delta^+ = \tilde V_j^0$ and $U_\delta^- = V_j^*$.

Once $P^-(v_h^+)$ is known, the space U_δ^- is chosen in both cases in such a way that the integrals of the product $\lambda_h P^-(v_h^+)$ can be computed. We then only need to compute the $P^-(v_h^+)$. This can be done by taking advantage of the biorthogonality property (3.1). In fact it is not difficult to see that, depending on which of the two cases we are in, we have

$$\text{Case 1: } P^-u=\sum_{k=1}^{2^j-1}\left(\int_\gamma u\;e^*_{j,k}\right)\tilde e_{j,k},\qquad \text{Case 2: } P^-u=\sum_{k=1}^{2^j-1}\left(\int_\gamma u\;\tilde e^*_{j,k}\right)e_{j,k}.$$

Again, in both cases the two auxiliary spaces have been chosen in such a way that the two integrals defining the projectors are computable. Moreover, biorthogonality implies that *no linear system* has to be solved in order to compute the auxiliary projector.

By applying Theorem 2.1 we can finally estimate the effect of using the approximate integration technique proposed in the previous section. In both cases we get the following bound: if $u\in H^s(\Omega)$ with $2\leq s\leq T$ it holds

$$\begin{aligned}\|u-u_h\|_{1,*}\quad\lesssim\quad & 2^{-j(s-1)}\|u\|_{H^s(\Omega)}+\inf_{\lambda\in M_h}\|\partial_{\nu_a}u-\lambda\|_{H^{-1/2}(\gamma)}\\ &+\inf_{v_h\in V_h^+}\|u-v_h\|_{H^1(\Omega_+)}+\inf_{v_h\in V_h^-}\|u-v_h\|_{H^1(\Omega_-)}.\end{aligned}$$

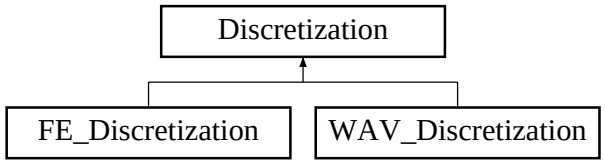


Figure 4.1: Inheritance diagram for Discretization

where, depending on the choice of the master and slave, the limit T in the bound is respectively $T = \min\{7/2, M + 1/2\}$ for the 'FEM master' case and $T = \min\{5/2, M + 3/2\}$ for the 'Wavelet master' case. The effect of approximating the constraint is contained in the first term on the r.h.s. which, by suitably choosing j can be tuned up in such a way it is comparable to the other three terms.

4. The implementation. The idea of replacing the classical Mortar method with the new approximate constraint, allows not only to overcome the problem of integrating functions of different kind, but gives also an advantage from the implementation point of view. In the first case, in fact, the introduction of a new discretization space in an existing code would require to provide specific tools for computing the integrals of the product of a function of the new type with all functions of each of the types already present in the code. On the other hand, the use of a projection on an auxiliary space to approximate the above integral reduces such a problem to the one of compute only the integral of an new type function with an auxiliary function.

We are now going to give a brief and schematic idea of the domain decomposition C++ code we implemented to couple Finite element and Wavelet discretitazions in the Mortar method. Without going into detailed descriptions, we will just give a brief overview to the two main classes defined in the code: the Class Discretization and the Class Mortar.

The Class Discretization It is a virtual class, from which the FE_Discretization (Finite Element Discretization) and the WAV_Discretization (Wavelet Discretization) classes are derived (Figure 4). It is associated to each subdomain of the global domain and provides the following main methods:

- **Trace_X_AuxBasis**: returns the integrals of a trace function with the auxiliary basis.
- **Get_Trace**: given a function, returns the trace of the function on an edge of the corresponding subdomain.
- **Set_Trace**: given a trace function f, sets the trace of the global function of the corresponding subdomain equals to f.
- **local_Stiff_x_u**: returns the Matrix-vector multiplication of the subdomain stiffness matrix with a vector u.

The Class Mortar The class Mortar is the class which allows to couple different kinds of discretization, in the sense that it is the way two Discretization classes comunicate with each other. It takes the traces of the functions of two adjacent subdomains and applies the approximate constraint operator, making use of the following methods:

- **Paux**: computes the projection of a trace function onto the auxiliary space.

- **Mortar_Projection**: returns the projection of a master edge function onto the multiplier space.
- **Local_Constraint**: applies the local Constraint operator.

In Figure (4.2), we show the numerical solution obtained by applying the approximate constraint to the Laplace equation

$$-\Delta u = 1 \ \text{ in } \Omega, \qquad u = 0 \ \text{ on } \Gamma = \partial\Omega,$$

REFERENCES

[1] C. Bernardi, Y. Maday, and A. T. Patera. A new non conforming approach to domain decomposition: The mortar element method. In H. Brezis and J.-L. Lions, editors, *Collège de France Seminar*. Pitman, 1994. This paper appeared as a technical report about five years earlier.

[2] S. Bertoluzza, S. Falletta, and V. Perrier. Wavelet/fem coupling by the mortar method. *Lecture Notes in Computational Science and Engineering*, 2001. (to appear).

[3] S. Bertoluzza and V. Perrier. The mortar wavelet method. *Mathem. Mod. and Numer. nal.*, 35:647–674, 2001.

[4] A. Cohen, I. Daubechies, and P. Vial. Wavelets on the interval and fast wavelet transform. *ACHA*, 1:54–81, 1993.

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

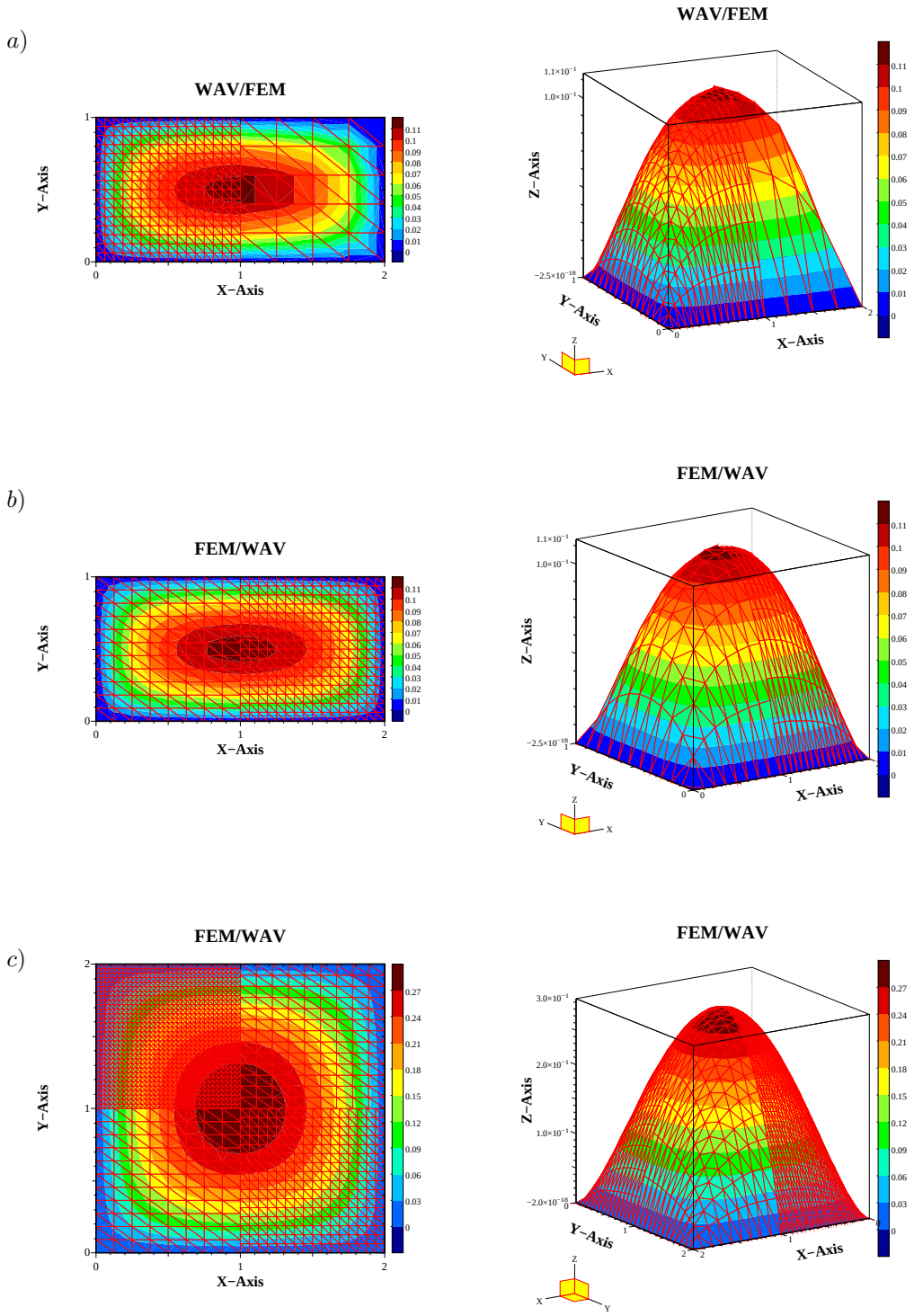


Figure 4.2: *a*):the case WAV master/FEM slave. *b*): the case FEM master/WAV slave *c*): the solution with mixed choice of discretizations and the presence of crosspoints.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

37. Generic parallel multithreaded programming of domain decomposition methods on PC clusters

A.S. Charão[1], I. Charpentier[2], B. Plateau[3], B. Stein[1]

1. Introduction. Since the early implementations of domain decomposition methods on parallel computers, programming techniques and computer architectures have significantly evolved. Due to the increasing availability of powerful microprocessors and high-speed networks, clusters of PCs become an attractive, low cost option for high-performance computing. While this trend makes parallel computing much more accessible, developing efficient programs for these architectures needs in general some expertise in parallel programming. In this paper we focus our attention on generic and object-oriented programming techniques for parallel implementation of domain decomposition methods. These techniques are central to Ahpik, our multithreaded programming tool targeted to such families of numerical methods.

Parallel efficiency is not the only goal when developing applications based on domain decomposition: flexibility and portability of parallel codes are also essential to preserve investments made in their development. Some existing parallel libraries, like PSPARSLIB[12] and PETSc[4], offer a compromise among all these goals. They provide a large set of linear equation system solvers which use domain decomposition methods as parallel preconditioners. Ahpik differs from these tools as it is rather an experimental library for doing research and experimentation involving parallel computing and numerical methods. It does not aim to provide a fairly complete set of algorithms and data structures for numerical computations, however it has some "plug-in" points allowing for easy integration of such components.

Ahpik offers a highly modular support for developing parallel domain decomposition solvers where numerical aspects are completely decoupled from parallel implementation details. This tool includes patterns of parallel coordination (task identification, communication and synchronization) that can be reused to implement different domain decomposition methods for the resolution of a PDE problem. These patterns can be viewed as drivers for parallel iterative computations, provided as C++ templates that must be "filled-in" with computations characterizing each numerical method. Performance results obtained with Ahpik were published in [2, 3, 1]. In this paper we concentrate on qualitative aspects of our approach. To do so, our evaluation criteria are based on the visualization of the parallel, multithreaded execution of some domain decomposition methods within different scenarios (good/bad workload distribution, synchronous/asynchronous iterations).

The outline of the paper is as follows. Our experience with a generic programming approach for parallel implementation of a large spectrum of domain decomposition methods is reported in section 2. Execution traces of a multithreaded driver are presented in section 3.

2. Genericity. A key idea within Ahpik is the representation of a parallel program as a graph of interacting tasks, namely *internal* and *interface* tasks. As far as domain decomposition methods are concerned, internal tasks correspond to local computations, *i.e.* computations that require only data local to a sub-domain (solving the linear system associated to a sub-domain for instance). Besides, interface tasks carry out operations requiring data from neighboring sub-domains.

[1]Departmento de Eletrônica e Computação (DELC-UFSM), Universidade Federal de Santa Maria, Brazil, andrea@inf.ufsm.br, benhur@inf.ufsm.br

[2]Projet Idopt (CNRS-INRIA-UJF-INPG), Laboratoire de Modélisation et Calcul (LMC-IMAG), Grenoble, France, Isabelle.Charpentier@imag.fr

[3]Projet Apache (CNRS-INRIA-UJF-INPG), Laboratoire Informatique et Distribution (ID-IMAG), Brigitte.Plateau@imag.fr

In previous works[2, 3, 1] Ahpik was presented as a three level library: the domain decomposition level containing specificities of some mainstream methods (Schwarz[13], FETI[8], Mortar[5]), the parallel drivers level (fixed-point, conjugate gradient and generalized minimum residual schemes), and the kernel of Ahpik which is based on a communicating threads library named Athapascan[6]. In this paper, Ahpik is described in an enhanced manner in order to point out the genericity of its parallel patterns and its abilities in the context of experimental studies.

2.1. Parallel patterns. While mathematical arguments differ significantly from one domain decomposition method to another, the parallel behavior of such methods is generally the same: it is dictated by the iterative, intrinsically synchronous, resolution procedure. In a traditional parallel programming method, the internal task (local computation) and three interface tasks (send, receive, interface computations) corresponding to a subdomain are gathered and executed sequentially on a unique UNIX process. Using multithreading, these four independent tasks may be assigned to different threads that are gathered on a UNIX process. There is no real order between them until one applies a parallel pattern for scheduling these tasks. This property eases the balancing of the computations and the implementation of asynchronous algorithms that allow for masking communication overhead. Such a programming model is particularly interesting when using clusters of shared-memory multiprocessors[10].

The assignment of the tasks onto threads depends on the iterative method one chooses to drive the parallel resolution of the domain decomposition problem. For the sake of simplicity, we present the *fixed point* parallel driver (coded in Ahpik) within a domain decomposition into two subdomains. Since we work with threads, we distinguish *read* and *write* tasks that are carried out through the memory shared by the threads associated to the same subdomain, from *send* and *receive* tasks that require communication through message passing. Threads are denoted using letters and numbers, the latter is equal to 1 (resp. 2) for threads of subdomain 1 (resp. 2), and equal to 0 for the thread devoted to the verification of the stopping criterion related to the convergence of the scheme. According to these definitions, threads perform the following tasks:

1. *Internal Thread* `IT1`: computes local PDE solutions, writes data for `ST1`, reads data from `ST1`, computes a local error and sends it to `CT0`, and finally reads data from `CT1`,
2. *Send Thread* `ST1`: reads data from `IT1`, sends them to `RT2`, reads data from `RT1`, computes and updates interface contributions and writes them for `IT1`,
3. *Receive Thread* `RT1`: receives data from `ST2` and writes them for `ST1`,
4. *Convergence Thread* `CT1`: receives data from `CT0` and writes them for `IT1`,
5. *Convergence Thread* `CT0`: receives data from `IT1` and `IT2`, computes a global error as specified by the user and sends it to `CT1` and `CT2`.

Send and receive tasks are assigned to different threads: this allows for overlapping communication with computations, because no sequential order is imposed between send and receive operations. For example, thread RT1 may receive data from ST2 before IT1 has finished its computations. There exist other manners of achieving that (for example using non-blocking send and receive primitives), but multithreading is a more elegant alternative.

When the *fixed point* driver is a synchronous one, these tasks are ordered uniquely. This may be observed on figure 2.1. In that picture, we draw the activity of threads, during one iteration, with respect to the execution time: a colored box signifies that the thread is active, a white box indicates the thread is blocked, waiting for data. Threads assigned to the same processor communicate through shared memory: a red arrow represents a synchronization point where one thread must wait for data made available by another. Message passing (blue arrows) is used to exchange data between threads running on different processing nodes. For the sake of clarity, the size of colored boxes corresponding to interface tasks has been enlarged

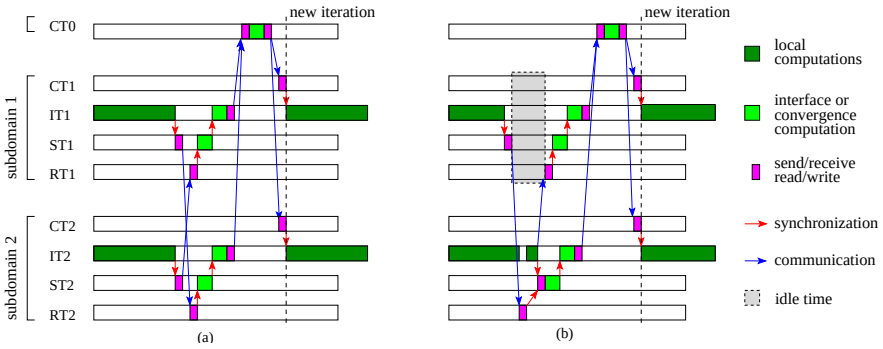


Figure 2.1: Theoretical traces for a synchronous "fixed point" parallel driver: (a) well-balanced workload (b) non-balanced workload

in order to avoid the superposition of blue and red arrows as it happens in actual execution traces presented in section 3.

Figure 2.1.a corresponds to a well-balanced domain decomposition in which the internal tasks of the two subdomains have the same computational cost. One observes that the two sets of threads have the same activity: the main steps described before are drawn on the scheme. In figure 2.1.b the two internal tasks have different computational costs. Since `IT1` performs less local computations than `IT2`, data are sent by `ST1` before the termination of local computations perfomed by `IT2`. In that case these data may be received by `RT2` quite immediately, this is why a break in the activity of `IT2` can occur[4]. Other steps are similar to those of figure 2.1.a. One observes that the four threads of subdomain 1 are inactive in the shadowed time interval. They wait for data sent by `ST2`, available only at the end of local computations performed by `IT2`. This induces idle times that may be reduced by placing multiple subdomains on each processing node.

An alternate solution relies on asynchronous iterations. As described in [9], an asynchronous scheme may be designed for the Schwarz alternating method. A theoretical trace is proposed in figure 2.2. There are no more synchronization points (no red arrows) between RT and ST threads (resp. CT and IT threads) because ST threads (resp. IT) do not wait for data made available by their RT (resp. CT) counterparts. In practice, ST and IT threads simply read data from shared memory without concern on the moment these data have been updated. As a consequence, there is always an active thread at any time.

On our trial trace, one observes that `IT1` is performing twice the same computation (first two iterations) because it uses the same interface data. The latter are updated when `IT2` has finished its first iteration. In more general situations (large number of subdomains), this problem is not so glaring because some interface data are usually updated before a new iteration begins.

2.2. An experimental library for domain decomposition methods. Ahpik is a generic parallel multithreaded environment that allows for the implementation of domain decomposition methods. We have been using generic programming facilities of the C++ programming language to allow users of Ahpik for a rather easy modification of the library with respect to the PDE problem of interest. This is why we decide to build Ahpik with regards to usual mathematical components. Moreover many "plug-in" points exist for

[4]Such breaks actually depend on the threads package and the operating system, they are not visible in the traces presented on section 3.

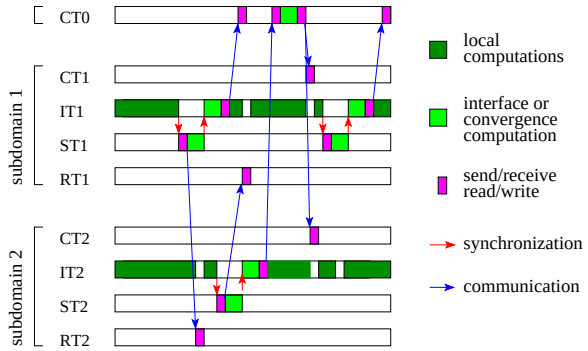


Figure 2.2: Theoretical trace for a non-balanced asynchronous "fixed point" parallel driver

coupling Ahpik with other existing libraries. The following classes and C++ templates are part of Ahpik[5]:

1. mesh capabilities: mesh data structures, mesh partitioning algorithms, subdomain and interface;
2. some discretization methods;
3. domain decomposition methods: additive Schwarz method as well as FETI and mortar methods;
4. local solvers: these are provided by SuperLU[7] for a LU direct solver and IML++[11] for iterative solvers. When appropriate, one may also define a matrix-vector product to perform local computation tasks;
5. generic parallel drivers: fixed point and conjugate gradient.

On one hand, Ahpik classes may be viewed as model classes for experimental solution of a PDE problem by a parallel domain decomposition method. Any user may plug his own C++ library at the level of interest (local solver, domain decomposition, discretization, etc.) as far as the mathematical aspects have been verified. For instance, changing the data structures representing matrices and vectors does not affect the code corresponding to subdomain or interface computations because these functions occur as C++ templates.

It is also possible to develop a new multithreaded parallel pattern that takes into account a preconditioner for example. Whatever the parallel pattern is, the management of the multithreading implementation remains hidden in the Ahpik kernel. Such an implementation allows to use Ahpik in a more general framework: grid-nesting and multigrid schemes are potential targets for our future works.

One the other hand, Ahpik is an experimental library devoted to parallel implementation of domain decomposition methods. It can be viewed as a set of trial problems (Laplace equation, various DDM, ...) for the evaluation of parallel programming alternatives. Developing new strategies to deal with parallelism only affects the kernel of Ahpik: the trial applications included in the object-oriented library are reusable.

3. Visualizing the execution of parallel drivers. Ahpik generates execution traces compatible with the post-processing tool Pajé[14]. This allows to make clear the role of each thread and the interactions between threads. Execution traces we present in this section are relative to a fixed point parallel driver again. The choice of this method against a

[5]More information on these components along with examples of their utilization will be included in the Ahpik distribution: `http://www.inf.ufsm.br/ahpik`

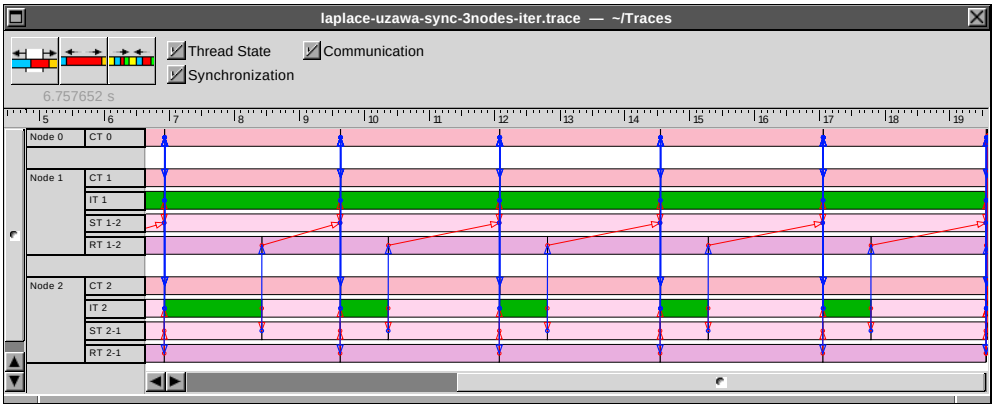


Figure 3.1: Synchronous fixed point driver, iterative local solver

conjugate gradient method is done for the sake of clarity of the drawings: the former induces a unique global synchronization point per iteration (global error computation), while the latter requires two global synchronizations (descent step and global error computation).

The main window of Pajé provides a space-time diagram which shows the activity of threads running on each processing node. As seen in section 2.1, Ahpik uses a set of threads for each subdomain. A single node can deal with multiple sets of threads *i.e* multiple subdomains. In the following, all executions are realized on 3 processing nodes, the first one checking the global convergence criterion of the domain decomposition method only.

Therefore node 0 runs a single thread (CT0) while nodes 1 and 2 run the sets of threads devoted to subdomain computations (CT, IT, ST, RT). The thread activity along the iterations is represented by a horizontal bar which is either green when the thread is working or pink when the thread is waiting for data. Two kinds of arrows are used to represent synchronization points. Red ones are synchronization between threads associated to the same subdomain (synchronized access to shared data) whereas blue ones show communication phases (message passing). The problem solved is the Laplace equation applied in rectangular domains, the geometric decompositions are described gradually.

3.1. General behavior of synchronous drivers. Figure 3.1 points out idle times that may appear when dealing with synchronous parallel drivers. For this experiment we used the domain decomposition of a rectangular domain into two well-balanced subdomains. Local computations are performed using a conjugate gradient solver which converges faster for subdomain 2 than for subdomain 1. The same behavior would have been observed using either a LU solver on a non-balanced domain decomposition or different discretization methods on each side of the interface. There are several ways of reducing idle times. Two of them, assignment of several process to a processor and implementation of asynchronous schemes, are discussed below. Solutions depending on dynamic load-balancing techniques will be discussed in a future work.

One of the key points of this work lies on the genericity of parallel drivers, which are completely independent of computations characterizing each domain decomposition method. As said before, the execution trace shows the behavior of a parallel driver. As a matter of fact, a Schwarz method or a FETI method applied to a domain decomposition in vertical stripes lead to the same kind of execution trace, the difference being in the computational cost and the number of iterations.

Figure 3.2: Synchronous fixed point driver, 2 subdomains assigned to node 1.

3.2. Reducing idle times. In MPI parallel programming, diminution of idle times can be achieved by placing multiple Unix processes (each one corresponding to a subdomain) to a processing node. A similar solution may be adopted for threaded programming. In that case a processor deals with a unique Unix process. The latter manages multiple sets of threads, each one being associated to a subdomain. Such a balancing method is presented in figure 3.2 for a decomposition of a rectangle into 3 subdomains (vertical bands). This execution trace, as well as others presented before, was generated on uniprocessor nodes. The first two (smaller) contiguous subdomains are assigned to node 1, while node 2 works on a single band. Even though subdomains have different computational costs (the first two subdomains are smaller than the third), one notices that the workload is well distributed over the processing nodes. Indeed, each node always has at least one active thread at any time interval. The interleaving of active threads on node 1 is due to the concurrent computation of two neighboring subdomains. When running the same experiment on 3 multiprocessor (SMP) nodes, the aspect of this execution trace changes for node 1 because the two subdomains can be treated not concurrently but in parallel.

Figure 3.3 presents an execution trace corresponding to an asynchronous parallel driver (fixed point) applied with the additive Schwarz method. The rectangular domain is decomposed in two overlapping (non-balanced) bands. One clearly observes that no idle times occur for this execution. Besides, no more red arrows representing synchronizations occur between RT and ST threads (resp. CT and IT threads). Indeed, in such asynchronous methods, threads devoted to local computations do not block, they do not wait for data. As a consequence, the overall iterative procedure is less structured and some processors may perform more iterations than others. In this execution trace, we can assume that each arrow arriving at a receive thread (the fourth thread on each node) indicates the beginning of a new iteration for each subdomain. Therefore node 1 performs 5 iterations while node 2 performs only 3. As predicted in theoretical traces, a same local computation can be performed twice. In practice, the use of asynchronous drivers could be interesting when the decomposition involves a larger number of subdomains distributed over non-homogeneous processing nodes.

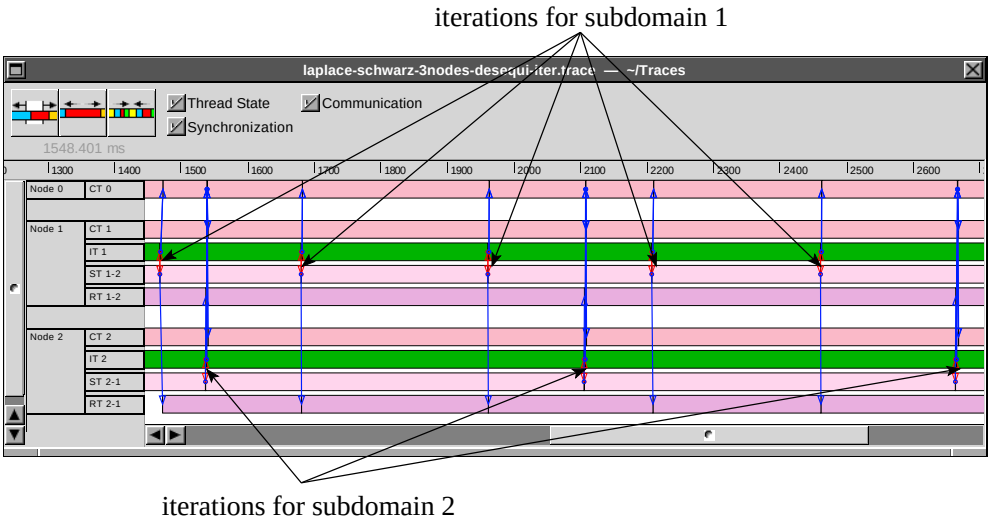


Figure 3.3: Asynchronous fixed point driver.

REFERENCES

[1] A. B. Abdallah, A. S. C. ao, I. Charpentier, and B. Plateau. Ahpik: A parallel multithreaded framework using adaptivity and domain decomposition methods for solving PDE problems. In R. H. J. P. D. K. Y. K. N. Debit, M. Garbey, editor, *13th International Conference on Domain Decomposition Methods*, pages 295–301, 2000.

[2] A. C. ao, I. Charpentier, and B. Plateau. A framework for parallel multithreaded implementation of domain decomposition methods. In E. H. D'Hollander, G. R. Joubert, F. J. Peters, and H. J. Sips, editors, *Parallel Computing: Fundamentals and Applications*, pages 95–102. Imperial College Press, 2000.

[3] A. S. C. ao, I. Charpentier, and B. Plateau. Programmation par objet et utilisation de processus légers pour les méthodes de décomposition de domaine. *Technique et Science Informatiques*, 5(19), 2000.

[4] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. *PETSc 2.0 User Manual*. Argonne National Laboratory, http://www.mcs.anl.gov/petsc/, 1997.

[5] C. Bernardi, Y. Maday, and A. T. Patera. A new non conforming approach to domain decomposition: The mortar element method. In H. Brezis and J.-L. Lions, editors, *Collège de France Seminar*. Pitman, 1994. This paper appeared as a technical report about five years earlier.

[6] J. Briat, I. Ginzburg, M. Pasin, and B. Plateau. Athapascan runtime : Efficiency for irregular problems. In *Proceedings of the Europar'97 Conference*, pages 590–599. Springer Verlag, august 1997.

[7] J. W. Demmel, J. R. Gilbert, and X. S. Li. An asynchronous parallel supernodal algorithm for sparse Gaussian elimination. *SIAM Journal on Matrix Analysis and Applications*, 20(4):915–952, 1999.

[8] C. Farhat and F.-X. Roux. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *Int. J. Numer. Meth. Engrg.*, 32:1205–1227, 1991.

[9] R. Guivarch and P. Spiteri. Implantation de méthodes de sous-domaines asynchrones avec PVM et MPI sur l'IBM-SP2. *Calculateurs Parallèles*, 10(4):431–438, 1998.

[10] E. L. Lusk and W. W. Gropp. A taxonomy of programming models for symmetric multiprocessors and SMP clusters. In *Proceedings of Programming Models for Massively Parallel Computers*, pages 2–7, 1995.

[11] R. Pozo et al. IML++ WWW home page, 1997.

[12] Y. Saad and A. V. Malevsky. PSPARSLIB: A portable library of distributed memory sparse iterative solvers. In *Proceedings of Parallel Computing Technologies (PaCT-95)*, 1995.

[13] H. A. Schwarz. *Gesammelte Mathematische Abhandlungen*, volume 2, pages 133–143. Springer, Berlin, 1890. First published in Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich, volume 15, 1870, pp. 272–286.

[14] B. O. Stein, J. C. de Kergommeaux, and P.-E. Bernard. Pajé, an interactive visualization tool for tuning multi-threaded parallel applications. *Parallel Computing*, 26:1253–1274, 2000.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

38. A preconditioner for the Schur complement domain decomposition method

J.-M. Cros [1]

1. Introduction. This paper presents a preconditioner for the Schur complement domain decomposition method inspired by the dual-primal FETI method [4]. Indeed the proposed method enforces the continuity of the preconditioned gradient at cross-points directly by a reformulation of the classical Neumann-Neumann preconditioner. In the case of elasticity problems discretized by finite elements, the degrees of freedom corresponding to the cross-points coming from domain decomposition, in the stiffness matrix, are separated from the rest. Elimination of the remaining degrees of freedom results in a Schur complement matrix for the cross-points. This assembled matrix represents the coarse problem. The method is not mathematically optimal as shown by numerical results but its use is rather economical. The paper is organized as follows: in sections 2 and 3, the Schur complement method and the formulation of the Neumann-Neumann preconditioner are briefly recalled to introduce the notations. Section 4 is devoted to the reformulation of the Neumann-Neumann preconditioner. In section 5, the proposed method is compared with other domain decomposition methods such as generalized Neumann-Neumann algorithm [7][9], one-level FETI method [5] and dual-primal FETI method. Performances on a parallel machine are also given for structural analysis problems.

2. The Schur complement domain decomposition method. Let Ω denote the computational domain of an elasticity problem. Consider a symmetric and positive definite linear system obtained by finite element discretization of the equations of equilibrium:

$$K\ u = f, \tag{2.1}$$

with the stiffness matrix K, the vector of degrees of freedom u, and the right-hand side f. The original domain Ω is partioned into n_s non-overlapping subdomains Ω^s. Let K^s be the local stiffness matrix and u^s the vector of degrees of freedom corresponding to subdomain Ω^s. Let N^s denote the Boolean matrix mapping the degrees of freedom u^s into global degrees of freedom u:

$$u^s = {N^s}^T\ u. \tag{2.2}$$

Then the stiffness matrix is obtained by the standard assembly process:

$$K = \sum_{s=1}^{n_s} N^s\ K^s\ {N^s}^T. \tag{2.3}$$

The union of all boundaries between subdomains is Γ such that $\Gamma = \cup_{s=1}^{n_s} \Gamma^s$ with $\Gamma^s = \partial\Omega^s \backslash \partial\Omega$. For each subdomain the total set of degrees of freedom is then split into two subsets, the interface degrees of freedom u_b^s associated with Γ^s and the other degrees of freedom u_i^s of the subdomain Ω^s. After this partition, the subdomain stiffness matrix, displacement vector, right-hand side and Boolean matrix take the following form:

$$K^s = \begin{bmatrix} K_{ii}^s & K_{ib}^s \\ {K_{ib}^s}^T & K_{bb}^s \end{bmatrix},\ \{u^s\} = \begin{Bmatrix} u_i^s \\ u_b^s \end{Bmatrix},\ \{f^s\} = \begin{Bmatrix} f_i^s \\ f_b^s \end{Bmatrix},\ \text{and } N^s = [N_i^s\ N_b^s]. \tag{2.4}$$

[1]Université d'ÉVRY-VAL d'ESSONNE, jean-michel.cros@iup.univ-evry.fr

With this notation, the linear system (2.1) takes the form:

$$\begin{bmatrix} K_{ii}^1 & \cdots & 0 & K_{ib}^1 {N_b^1}^T \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & K_{ii}^{n_s} & K_{ib}^{n_s} {N_b^{n_s}}^T \\ N_b^1 {K_{ib}^1}^T & \cdots & N_b^{n_s} {K_{ib}^{n_s}}^T & \sum_{s=1}^{n_s} N_b^s K_{bb}^s {N_b^s}^T \end{bmatrix} \begin{Bmatrix} u_i^1 \\ \vdots \\ u_i^{n_s} \\ u_b \end{Bmatrix} = \begin{Bmatrix} f_i^1 \\ \vdots \\ f_i^{n_s} \\ \sum_{s=1}^{n_s} N_b^s \ f_b^s \end{Bmatrix}. \tag{2.5}$$

After elimination of the interior degrees of freedom, the problem (2.5) reduces to a problem (2.6) posed on the interface Γ:

$$\Big(\sum_{s=1}^{n_s} N_b^s \big(K_{bb}^s - {K_{ib}^s}^T {K_{ii}^s}^{-1} K_{ib}^s\big) {N_b^s}^T \Big) u_b = \sum_{s=1}^{n_s} N_b^s \Big(f_b^s - {K_{ib}^s}^T {K_{ii}^s}^{-1} f_i^s \Big). \tag{2.6}$$

Defining the global Schur complement matrix S by:

$$S = \sum_{s=1}^{n_s} N_b^s \ S^s \ {N_b^s}^T \tag{2.7}$$

where the local Schur complement matrix is given by $S^s = K_{bb}^s - {K_{ib}^s}^T {K_{ii}^s}^{-1} K_{ib}^s$. The linear system (2.6) is solved iteratively without assembling S, using a preconditioned conjugate gradient algorithm.

3. Neumann-Neumann preconditioners. For mechanical problems, the most classical preconditioner used is the Neumann-Neumann method [1][7]. The preconditioner (3.1) is defined by approximating the inverse of the sum of local Schur complement matrices by the weighted sum of the inverses:

$$z = M\, r = \sum_{s=1}^{n_s} N_b^s \ D^s \ {S^s}^{-1} \ D^s \ {N_b^s}^T \ r, \tag{3.1}$$

where r is the conjugate gradient and z is the preconditioned conjugate gradient. For convergence reasons [7], the diagonal weight matrices D^s must verify:

$$\sum_{s=1}^{n_s} N_b^s \ D^s \ {N_b^s}^T = I_\Gamma. \tag{3.2}$$

However, the convergence rate decreases rapidly for a large number of subdomains. Then, the balancing domain decomposition method [8] includes a coarse problem in order to reduce significantly this dependence on the number of subdomains. The balancing method or the generalized Neumann-Neumann preconditioner [7] writes:

$$M = \big(I - G\,[G^T \ S \ G]^{-1} \ G^T S\big) \sum_{s=1}^{n_s} N_b^s \ D^s \ \tilde{S}^{s^{-1}} \ D^s \ {N_b^s}^T, \tag{3.3}$$

$$\text{with } G = [N_b^1 D^1 Z^1, ..., N_b^{n_f} D^{n_f} Z^{n_f}], \tag{3.4}$$

where n_f is the number of floating subdomains (subdomain without natural Dirichlet condition), the block matrices Z^s are boundary values of subdomain solutions with restriction of rigid body modes on Γ^s, and $\tilde{S}^{s^{-1}}$ is the pseudo inverse of the local Schur complement matrix. The method has been extended [9] for second or fourth order elasticity problems, by using corner modes. By definition, a corner or a cross-point is a node belonging to more than

two subdomains and also, for plate and shell problems, a node located at the beginning and the end of each edge of each subdomain. Then, the block matrices Z^s are boundary values of subdomain solutions with successively one degree of freedom fixed to one at one corner, all other corner degrees of freedom fixed to zero. The method is efficient, but the coarse matrix $[G^T\ S\ G]$ is costly to build because it involves a product with S.

4. A new coarse problem. We propose to build a Neumann-Neumann preconditioner by enforcing a continuous field at the cross-points. In the classical Neumann-Neumann preconditioner with or without coarse problem, the field is continuous only by averaging the contributions of each subdomain at the cross-points. Then, we introduce a new partitioning

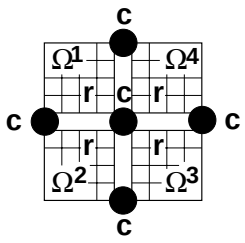


Figure 4.1: Mesh partition : corner (c) and remainder nodes (r)

(figure 4.1), by splitting u^s into two sub-vectors (4.1) where u_c is a global solution vector over all defined corner degrees of freedom and u_r^s is the remainder subdomain solution vector.

$$\{u\} = \left\{ \begin{array}{c} u_r \\ u_c \end{array} \right\} = \left\{ \begin{array}{c} u_r^1 \\ \vdots \\ u_r^{n_s} \\ u_c \end{array} \right\}. \tag{4.1}$$

The Boolean matrix N_c^s (4.2) maps the local corner equation to the global corner equation:

$$u_c^s \quad = \quad {N_c^s}^T\ u_c. \tag{4.2}$$

Then, we introduce new Boolean matrices (4.3) which extract from the interface Γ^s, the cross-points and the remainder unknowns:

$$\{u^s\} = \left\{ \begin{array}{c} u_r^s = [R_{ri}^s \ \ R_{rb}^s] \left\{ \begin{array}{c} u_i^s \\ u_b^s \end{array} \right\} \\ u_c^s = R_c^s\ u_b^s \end{array} \right\}. \tag{4.3}$$

According to this new partition of the degrees of freedom, the preconditioned gradient is the restriction on Γ of the solution of problem (4.4) with subdomains connected by the cross-points as shown in figure 4.1:

$$\left[\begin{array}{cccc} K_{rr}^1 & \cdots & 0 & K_{rc}^1 {N_c^1}^T \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & K_{rr}^{n_s} & K_{rc}^{n_s} {N_c^{n_s}}^T \\ N_c^1 {K_{rc}^1}^T & \cdots & N_c^{n_s} {K_{rc}^{n_s}}^T & \displaystyle\sum_{s=1}^{n_s} N_c^s K_{cc}^s {N_c^s}^T \end{array} \right] \left\{ \begin{array}{c} u_r^1 \\ \vdots \\ u_r^{n_s} \\ u_c \end{array} \right\} = \left\{ \begin{array}{c} f_r^1 \\ \vdots \\ f_r^{n_s} \\ \displaystyle\sum_{s=1}^{n_s} N_c^s\ f_c^s \end{array} \right\}, \tag{4.4}$$

with $f_r^s = R_{rb}^s\ D^s\ {N_b^s}^T r$ (with $f_i^s = 0$) and $f_c^s = R_c^s\ D^s\ {N_b^s}^T r$. It is noted that by definition of D^s, the quantity $\sum_{s=1}^{n_s} N_c^s\ f_c^s = \sum_{s=1}^{n_s} N_c^s\ R_c^s\ D^s\ {N_b^s}^T r$ is the restriction of gradient r at the cross-points (r_c). Thus the solution of (4.4) is written:

$$u_r^s\ =\ {K_{rr}^s}^{-1}\ \big(f_r^s - K_{rc}^s\ {N_c^s}^T\ u_c\big), \tag{4.5}$$

$$u_c\ =\ \Big(\sum_{s=1}^{n_s}\ N_c^s\big(K_{cc}^s - {K_{rc}^s}^T\ {K_{rr}^s}^{-1}\ K_{rc}^s\big)\ {N_c^s}^T\Big)^{-1}\ \sum_{s=1}^{n_s}\ N_c^s\ \Big(\ f_c^s - {K_{rc}^s}^T\ {K_{rr}^s}^{-1}\ f_r^s\ \Big). \tag{4.6}$$

Finally, the preconditioned gradient is given by:

$$z = Mr = \sum_{s=1}^{n_s} N_b^s D^s R_c^{s^T} N_c^{s^T} u_c + \sum_{s=1}^{n_s} N_b^s D^s R_{rb}^{s^T} u_r^s, \tag{4.7}$$

and the proposed preconditioner takes the form:

$$\begin{aligned} M = &\sum_{s=1}^{n_s} N_b^s D^s \Big(N_c^s (R_c^s - {K_{rc}^s}^T\ {K_{rr}^s}^{-1} R_{rb}^s)\Big)^T\ S_c^{-1}\ \sum_{s=1}^{n_s} \Big(N_c^s\ (R_c^s - {K_{rc}^s}^T\ {K_{rr}^s}^{-1} R_{rb}^s)\Big) D^s {N_b^s}^T \\ &+ \sum_{s=1}^{n_s} N_b^s D^s \Big[{R_{rb}^s}^T\ {K_{rr}^s}^{-1}\ R_{rb}^s\Big]\ D^s {N_b^s}^T. \end{aligned} \tag{4.8}$$

The first term is a coarse problem which couples all subdomains. We suppose that each subdomain owns enough cross-points to have local Neumann problems (4.5) well posed, otherwise artificial cross-points are added. The coarse matrix is built easily by forming the matrices S_c^s in each subdomain and by assembling S_c:

$$S_c = \sum_{s=1}^{n_s} N_c^s \Big[K_{cc}^s - {K_{rc}^s}^T\ {K_{rr}^s}^{-1}\ K_{rc}^s\Big]\ {N_c^s}^T = \sum_{s=1}^{n_s} N_c^s\ S_c^s\ {N_c^s}^T. \tag{4.9}$$

In comparison with coarse problem of the balancing method, the size of the coarse problem S_c (equals to the number of degrees of freedom per node multiplied by the total number of cross-points) is small, because of the definition of corner modes.

5. Numerical results. The parallel implementation of the different methods has been developed within message passing programming environment. Each subdomain is allocated to one processor. All coarse problems are assembled and solved by a skyline solver during the iterations of the preconditioned conjugate gradient algorithm.

In all the tables below, GNN denotes the Neumann-Neumann preconditioner with coarse grid solver based on rigid body modes (RBM) [8] or corner modes (CM) [9], NN+C is the proposed Neumann-Neumann preconditioner with coarse grid solver, FETI-DP is a dual-primal Finite Element Tearing and Interconnecting method [4] and FETI-1 is a classical FETI method [5]. In the absence of other specification, the FETI methods are equipped with the Dirichlet preconditioner. The stopping criterion to monitor the convergence is the same in all cases presented and it is related to the global residual:

$$\parallel K\,u - f \parallel\ /\ \parallel f \parallel\ \ \leq\ \ 10^{-6}. \tag{5.1}$$

We investigate the numerical scalability of the proposed method with respect to the mesh size h and to the number of subdomains n_s. For this purpose, we consider a cylindrical shell roof (figure 5.1) subjected to a loading of its own weight. The roof is supported by walls at each end and is free along the sides. For symmetry reasons only a quarter of the roof is

considered and is meshed with 3-node shell elements. We study three discretizations denoted respectively by h (3,750 degrees of freedom), $h/2$ (14,406 degrees of freedom) and $h/4$ (56,454 degrees of freedom). The meshes $h/2$ et $h/4$ are obtained from the first one by global regular refinement.

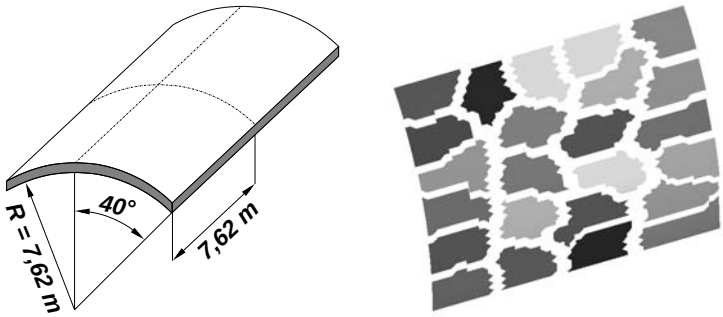


Figure 5.1: A cylindrical shell roof, mesh $h/2$ decomposed into 24 subdomains

First, the three meshes were decomposed into 24 subdomains, we note (figure 5.1) that automatic decomposition by METIS [6] induces rugged interfaces. The results were obtained on a Origin 2000 system (64 processors) of "Pôle Parallélisme Île de France Sud". We report

Table 5.1: A cylindrical shell roof, numerical scalability, $n_s = 24$

MESH	THICKNESS (m)	GNN (CM)	NN+C	FETI-DP
h	0.1	30 iter. (726)	42 iter. (270)	44 iter. (270)
	0.01	31 iter.	48 iter.	50 iter.
	0.001	54 iter.	99 iter.	106 iter.
$h/2$	0.1	36 iter. (714)	41 iter. (270)	44 iter. (270)
	0.01	37 iter.	48 iter.	50 iter.
	0.001	45 iter.	74 iter.	80 iter.
$h/4$	0.1	39 iter. (738)	40 iter. (276)	45 iter. (276)
	0.01	39 iter.	44 iter.	47 iter.
	0.001	42 iter.	67 iter.	73 iter.

(table 5.1) the number of iterations to converge for the different methods and in brackets the size of the coarse problem. The number of iterations remains roughly constant for the different methods (thickness = 0.1 m and 0.01 m). However all the methods are sensitive to the small thickness of the roof, and especially the FETI-DP method and the NN+C method.

The second test (table 5.2) consists in fixing the size of the problem ($h/4$, 56,454 degrees of freedom, thickness = 0.1 m) but we change the number of subdomains (12, 24, 48). The CPU times are reported (table 5.2) for both the preparation step (finite element operations, building of coarse problem,...) and for the solution. It appears clearly that the building of coarse problem takes a large part of the cpu time for GNN method. The two other methods have a good speed-up.

We consider now the modal analysis of a plate (1×1 m) embeded on one side. The problem is discretized in 10,086 degrees of freedom with 3-node shell elements. The mesh is partitioned into 20 subdomains. The two lowest eigenmodes are obtained in five iterations of the subspace iteration method. The conjugate gradient method with restart technique [2][5] is used to deal

Table 5.2: A cylindrical shell roof, parallel scalability, mesh $h/4$

n_s	INTERFACE	METHOD	ITERATION	CPU (sec.)
12	3,144	GNN (CM)	38 iter. (426)	46.22 + 9.81 = 56.03
		NN+C	40 iter. (216)	6.42 + 10.38 = 16.80
		FETI-DP	45 iter. (216)	6.38 + 12.79 = 19.17
24	4,818	GNN (CM)	39 iter. (738)	18.82 + 5.30 = 24.12
		NN+C	40 iter. (276)	1.59 + 5.21 = 6.80
		FETI-DP	45 iter. (276)	1.52 + 5.45 = 6.97
48	7,295	GNN (CM)	45 iter. (1602)	18.67 + 8.18 = 26.85
		NN+C	51 iter. (630)	0.81 + 3.21 = 4.02
		FETI-DP	56 iter. (630)	0.86 + 3.64 = 4.50

with successive and multiple right-hand sides. This technique is based on the exploitation of previously computed conjugate directions. Figure 5.2 shows the iteration history with respect to the number of right-hand sides using different methods, and we report also the total number of iterations, the size of the coarse problem (in brackets) and the CPU times. The GNN (CM) method converges quickly but the cost of one iteration is more important than the other methods, because of the large size of the coarse problem. Similar results are obtained for transient analysis. In addition, the solution of time-dependent problems by the implicit Newmark algorithm calls for successive solution of the linear system with the same matrix $[M + \beta\,\Delta t^2\ K]$. In this case, there are no longer floating subdomains due to the inertia term. Then, building a coarse grid based on the rigid body modes of stiffness matrices K^s becomes costly [3]. While the methods using the corner modes are not affected by this shifting of M.

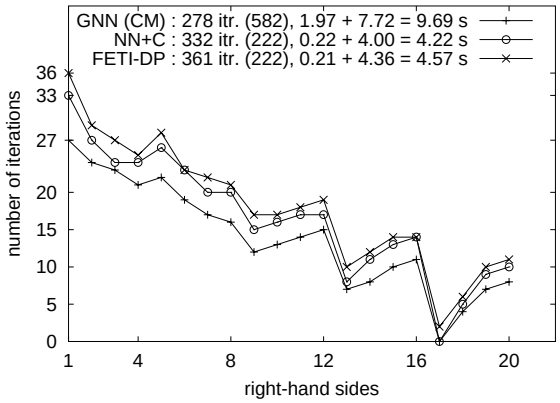


Figure 5.2: A shell problem, modal analysis

Another test example concerns a plane stress problem with a square (1×1 m) embeded on one side and subjected to a distributed load on the opposite side. The problem is discretized in 20,402 degrees of freedom with 3-node elements (101×101 nodes). The mesh is partitioned into 14 and 28 subdomains. The NN+C method is proved (table 5.3) to be efficient for this kind of problems.

Finally, we consider a three-dimensional cantilever beam (4×4×40 m) subjected to a

Table 5.3: 2D elasticity problem, 3-node elements, 101×101 nodes, 20,402 d.o.f.

n_s	GNN (RBM)	GNN (CM)	NN+C	FETI-DP	FETI-1
14	24 iter. (30)	14 iter. (132)	20 iter. (52)	22 iter. (52)	24 iter. (30)
28	26 iter. (72)	14 iter. (286)	23 iter. (108)	24 iter. (108)	26 iter. (72)

bending load. The finite element discretization is done with 8-node brick elements (12×12×76 nodes, 32,832 degrees of freedom). The beam (figure 5.3) is cut into 20 and 40 subdomains.

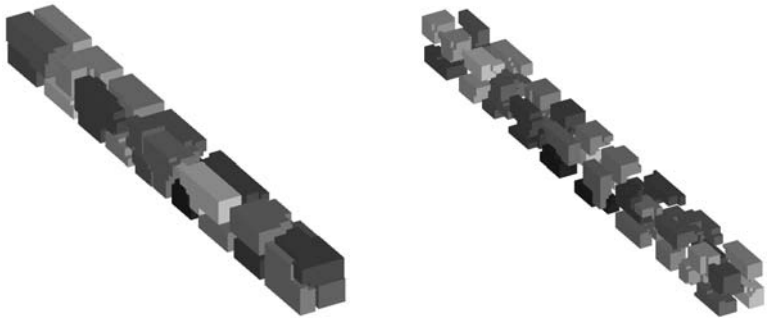


Figure 5.3: A cantilver beam, 20 subdomains (interface 7,614 d.o.f.) and 40 subdomains (interface 10,262 d.o.f.)

Table 5.4: Cantiler beam, 32,832 d.o.f.

			FETI-DP		FETI-1	
n_s	GNN (RBM)	NN+C	DIRICHLET	LUMPED	DIRICHLET	LUMPED
20	44 iter.	30 iter.	31 iter.	60 iter.	43 iter.	67 iter.
	(102)	(852)			(102)	
40	54 iter.	30 iter.	30 iter.	53 iter.	58 iter.	77 iter.
	(216)	(1968)			(216)	

For this analysis, the FETI methods use equally the lumped preconditioner. Table 5.4 summarizes the results with the size of the coarse problem (in brackets). Methods using corner modes have the best convergence rate, but the size of the coarse problem is very large (almost 20% of the interface for 40 subdomains). This size can be reduced easily. In fact, with brick elements, the number of cross-points can be chosen just enough to remove the singularities in subdomains.

6. Conclusion. In this paper, we have presented a modified Neumann-Neumann preconditioner validated by several examples. The results suggest that the proposed method (NN+C method) is numerically scalable with respect to the number of subdomains and to the mesh size. On the representative examples considered the NN+C method has the same performance as the FETI-DP method. Moreover, from the viewpoint of CPU time, the proposed method outperforms the optimal but expensive GNN preconditioner. However, the results depend largely on the implementation of the algorithm for solving the coarse problem.

REFERENCES

[1] J.-F. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods*, pages 3–16, Philadelphia, PA, 1989. SIAM.

[2] J.-M. Cros and F. Léné. Parallel iterative methods to solve large-scale eigenvalue problems in structural dynamics. In P. E. Bjørstad, M. Espedal, and D. Keyes, editors, *Domain Decomposition Methods in Sciences and Engineering*, pages 318–324. John Wiley & Sons, 1997. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.

[3] C. Farhat, P.-S. Chen, and J. Mandel. A scalable Lagrange multiplier based domain decomposition method for time-dependent problems. *Int. J. Numer. Meth. Eng.*, 38:3831–3853, 1995.

[4] C. Farhat, M. Lesoinne, and K. Pierson. A scalable dual-primal domain decomposition method. *Numer. Lin. Alg. Appl.*, 7:687–714, 2000.

[5] C. Farhat and F.-X. Roux. Implicit parallel processing in structural mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 2 (1), pages 1–124. North-Holland, 1994.

[6] G. Karypis and V. Kumar. Metis, unstructured graph partitioning and sparse matrix ordering system. version 2.0. Technical report, University of Minnesota, Department of Computer Science, Minneapolis, MN 55455, August 1995.

[7] P. Le Tallec. Domain decomposition methods in computational mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 1 (2), pages 121–220. North-Holland, 1994.

[8] J. Mandel. Balancing domain decomposition. *Comm. Numer. Meth. Engrg.*, 9:233–241, 1993.

[9] P. L. Tallec, J. Mandel, and M. Vidrascu. A Neumann-Neumann domain decomposition algorithm for solving plate and shell problems. *SIAM J. Numer. Math.*, 35:836–867, 1998.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

39. Interface Preconditioners for Splitting Interface Conditions in Air Gaps of Electrical Machine Models

H. De Gersem[1,2], S. Vandewalle[3], M. Clemens[1], T. Weiland[1]

1. Introduction. Electrical machine design is typically based on finite element (FE) simulations of steady-state working conditions. Motional eddy current effects are commonly resolved by transient simulation, which may be too expensive if only steady-state behaviour has to be simulated. This paper offers a time-harmonic FE approach for machines operating at steady-state, incorporating motional eddy current effects. The formulation incorporates interface conditions connecting the boundary of one stator model to the boundaries of several rotor models based on Fast Fourier Transforms and restriction operations. The matrix-free discretisation of the interface conditions excludes the use of algebraic iterative solution techniques. Instead, techniques related to iterative substructuring are proposed to solve the model.

2. Finite element machine models. Two common approaches for simulating electrical machines are the transient approach and the time-harmonic approach. The transient approach accounts for motional eddy currents by the Lagrange technique: between two successive time steps, the previous solution is azimuthally moved together with the rotor part. Accordingly, the interface conditions between stator and rotor are updated. The relative motion of both motor parts can be modelled by a moving band technique [5], a hybrid FE, boundary-element approach [8], discontinuous finite elements [1] or a sliding surface technique, possibly resolved by mortar finite elements [2]. Transient methods are however too expensive when only stationary operations have to be simulated.

For electrical machines excited by alternating current sources and rotating at constant velocities, formulations in frequency domain are preferred. The simplest case is when only one frequency f is present in the excitating voltages. Then, one can adopt the time-harmonic formulation

$$\nabla \times (\nu \nabla \times \underline{\mathbf{A}}) + \mathrm{j}\omega\sigma\underline{\mathbf{A}} = -\sigma\nabla\underline{V} \tag{2.1}$$

with the phasor $\underline{\mathbf{A}}$ related to the magnetic vector potential $\mathbf{A}$ by

$$\mathbf{A}(x,y,z,t) = \mathrm{Re}\left\{\underline{\mathbf{A}}(x,y,z)e^{\mathrm{j}\omega t}\right\} , \tag{2.2}$$

ν the reluctivity, σ the conductivity, $\underline{V}$ the phasor of the voltage and $\omega = 2\pi f$ the pulsation. Time-harmonic simulations are remarkably accurate and extremely efficient for the steady-state simulation of devices supplied with alternating currents. Unfortunately, accounting for motional effects in such simulations is not straightforward.

For many machines, a 2D FE model of the cross-section of the machine, extended with an equivalent circuit modelling the electric connections at the front and rear machine ends, achieves a sufficient accuracy [12]. Then, the vectorial PDE (2.1) simplifies to a scalar PDE in terms of the z-component $\underline{A}_z$ of $\underline{\mathbf{A}}$:

$$-\frac{\partial}{\partial x}\left(\nu\frac{\partial \underline{A}_z}{\partial x}\right) - \frac{\partial}{\partial y}\left(\nu\frac{\partial \underline{A}_z}{\partial y}\right) + \mathrm{j}\omega\sigma\underline{A}_z = \frac{\sigma}{\ell_z}\Delta\underline{V} \tag{2.3}$$

[1]Technische Universität Darmstadt, Computation Electromagnetics Laboratory, degersem/clemens/weiland@temf.tu-darmstadt.de
[2]H. De Gersem is working in the cooperation project "DA-WE1 (TEMF/GSI)" with the "Gesellschaft für Schwerionenforschung (GSI)", Darmstadt
[3]Katholieke Universiteit Leuven, Dep. Computer Science, stefan.vandewalle@cs.kuleuven.ac.be

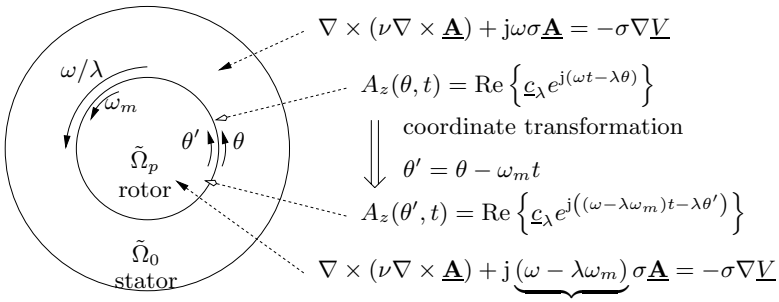


Figure 3.1: Slip transformation technique, illustrated for a simplified machine model.

with ℓ_z the device length and $\Delta\underline{V}$ the voltage drop between the machine's front and back side. The discretisation of (2.3) by linear triangular FE shape functions $N_i(x,y)$ yields the system of equations

$$K\underline{u} + \underline{g} = \underline{f} \tag{2.4}$$

with $\underline{u}$ containing the degrees of freedom for $\underline{A}_z$,

$$K_{ij} = \int_\Omega \left(\nu\frac{\partial N_i}{\partial x}\frac{\partial N_j}{\partial x} + \nu\frac{\partial N_i}{\partial y}\frac{\partial N_j}{\partial y} + \mathrm{j}\omega\sigma N_i N_j\right)\mathrm{d}\Omega\,, \tag{2.5}$$

$$\underline{f}_i = \int_\Omega \frac{\sigma}{\ell_z}\Delta\underline{V}\,N_i\,\mathrm{d}\Omega\,, \tag{2.6}$$

$$\underline{g}_i = -\int_{\partial\Omega} \nu\frac{\partial\underline{A}_z}{\partial n}\,N_i\,\mathrm{d}\Gamma \tag{2.7}$$

and $\partial/\partial n$ the normal derivative outward to Ω.

3. Slip transformation. Only in a very particular case, i.e., if the air gap field is a rotating wave, it is possible to account for motional eddy currents while keeping the classical time-harmonic formulation (2.3). Consider the simplified machine model of Fig. 3.1. Suppose the field at a circular interface between stator and rotor equals the rotating wave

$$A_z(\theta,t) = \mathrm{Re}\left\{\underline{c}_\lambda e^{\mathrm{j}(\omega t-\lambda\theta)}\right\} \tag{3.1}$$

with the phasor $\underline{c}_\lambda$, the pole pair number λ and the azimuthal coordinate θ along the interface. An observer attached to the stator experiences the wave as a cosine rotating at the velocity ω/λ along the interface. Consider a second observer attached to the rotor and hence inheriting its rotation at a constant mechanical velocity ω_m. The corresponding azimuthal coordinate θ' along the interface is related to θ by

$$\theta' = \theta - \omega_m t\,. \tag{3.2}$$

The rotating observer experiences the field at the interface as

$$A_z(\theta',t) = \mathrm{Re}\left\{\underline{c}_\lambda e^{\mathrm{j}((\omega-\lambda\omega_m)t-\lambda\theta')}\right\} \tag{3.3}$$

which is also a rotating wave with the same phasor and pole pair number, but with a different pulsation $\omega_{s,\lambda} = \omega - \lambda\omega_m$, called the *slip pulsation*. Hence, phenomena at the stator side induce phenomena at the rotor side at slip pulsation. Motional eddy currents are easily incorporated in (2.3) by replacing the pulsation ω by the slip pulsation $\omega_{s,\lambda}$ for the rotating

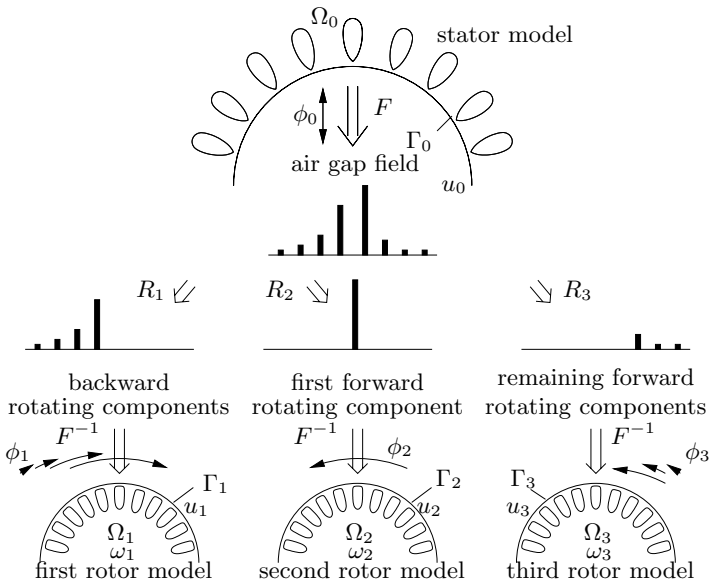


Figure 4.1: Scheme of the air gap flux decomposition approach illustrating the splitting of the total stator flux ϕ_0 into three parts ϕ_1, ϕ_2 and ϕ_3.

model parts. This procedure is called *slip transformation*. The assumption of a rotating wave form as field distribution in the air gap is approximately true for three-phase induction machines. Then, time-harmonic steady-state simulation with slip transformation commonly gives reliable results [4]. Other alternating current machines, e.g. single-phase induction machines, do not feature this property. Hence, time-harmonic simulation is at first glance not applicable.

4. Decomposition of the air gap field. The slip transformation technique is extendable to cases with more general air gap field distributions. The key point is to decompose the arbitrary air gap field into rotating field components and distribute these components towards distinct rotor models such that slip transformations can be defined for each rotating component independently [3]. Consider a model consisting of one stator model Ω_0 and n rotor domains Ω_p, $p = 1, \ldots, n$ (see Fig. 4.1 for an example with $n = 3$). The stator and rotor models share a circular interface Γ_b in the middle of the air gap. For each rotor domain, the slip pulsation $\omega_p = \omega - \lambda_p \omega_m$ is selected according to one of the field component present at Ω_p, i.e., the component with pole pair number λ_p. Because the stator windings do not experience eddy currents, ω_0 is set to zero. For each submodel independently, a FE subsystem is set up:

$$\begin{bmatrix} K_{p,aa} & K_{p,ab} \\ K_{p,ba} & K_{p,bb} \end{bmatrix} \begin{bmatrix} \underline{u}_{p,a} \\ \underline{u}_{p,b} \end{bmatrix} + \begin{bmatrix} 0 \\ \underline{g}_{p,b} \end{bmatrix} = \begin{bmatrix} \underline{f}_{p,a} \\ \underline{f}_{p,b} \end{bmatrix}, \quad p = 0, \ldots, n \tag{4.1}$$

where the subscripts a and b distinguish between degrees of freedom associated with inner nodes and degrees of freedom associated with nodes at Γ_b. Since in general $\omega_{p_1} \neq \omega_{p_2}$, the FE stiffness matrices for the rotor domains are different although they feature the same FE mesh and reluctivities. The subsystems are collected in the block diagonal matrices K_{aa}, K_{ab}, K_{ba} and K_{bb}, the vectors of unknowns $\underline{u}_a$ and $\underline{u}_b$, the boundary terms $\underline{g}_b$ and the load

vectors $\underline{f}_a$ and $\underline{f}_b$. For convenience, assume all submodels have equidistant and matching grids at Γ_b. An appropriate selection of rotating field components is performed by interface conditions applied at Γ_b:

$$F\underline{u}_{p,b} - R_p F \underline{u}_{0,b} = 0, \quad p = 1, \ldots, n \tag{4.2}$$

with F denoting the discrete Fourier transform and R_p a set of restriction operators such that $\sum_{p=1}^{n} R_p = I$. The interface conditions (4.2) take the distribution of $\underline{A}_z$ at the stator side of Γ_b ($\underline{u}_{0,b}$), transform this into harmonic components ($F\underline{u}_{0,b}$), next restrict these to a particular subset ($R_p F \underline{u}_{0,b}$) and finally equal this subset of harmonics to the harmonic components of the distribution of $\underline{A}_z$ at one of the rotor sides of Γ_b ($F\underline{u}_{p,b}$). The choices of the sets $\{R_p\}$ and $\{\omega_p\}$ are motivated by technical considerations [3]. For many electrical machines, only a few harmonics are responsible for the major machine behaviour whereas the remaining harmonics only have a marginal influence. Therefore, an important rotating field component λ_p is assigned to an individual rotor model Ω_p equipped with the corresponding slip pulsation $\omega_p = \omega - \lambda_p \omega_m$. The remaining harmonics are arbitrarily propagated to one of the already existing rotor models. Eddy current phenomena due to these harmonics are only approximately taken into account. The constraints (4.2) are added to the FE system (4.1). The boundary integral terms $\underline{g}_{p,b}$ are resolved in terms of a set of Lagrange multipliers $\underline{\xi}$, i.e., $\underline{g}_b = B^H \underline{\xi}$. The FE model including air gap flux decomposition corresponds to the saddle-point problem

$$\begin{bmatrix} K_{aa} & K_{ab} & 0 \\ K_{ba} & K_{bb} & B^H \\ 0 & B & 0 \end{bmatrix} \begin{bmatrix} \underline{u}_a \\ \underline{u}_b \\ \underline{\xi} \end{bmatrix} = \begin{bmatrix} \underline{f}_a \\ \underline{f}_b \\ 0 \end{bmatrix} \tag{4.3}$$

with

$$B = \begin{bmatrix} -R_1 F & F & & \\ \vdots & & \ddots & \\ -R_n F & & & F \end{bmatrix}. \tag{4.4}$$

Although the FE system part is complex symmetry, this property is not maintained in the system (4.3).

It is possible to eliminate the inner degrees of freedom $\underline{u}_{p,a}$ with respect to the degrees of freedom $\underline{u}_{p,b}$ at Γ_b for each submodel independently. The Schur complement subsystems $D_p \underline{u}_{p,b} = \underline{q}_p$ with stiffness matrices $D_p = K_{p,bb} - K_{p,ba} K_{p,aa}^{-1} K_{p,ab}$ and load vectors $\underline{q}_p = \underline{f}_{p,b} - K_{p,ba} K_{p,aa}^{-1} \underline{f}_{p,a}$, are collected in $D\underline{u}_b = \underline{q}$. The system with interface conditions reads

$$\begin{bmatrix} D & B^H \\ B & 0 \end{bmatrix} \begin{bmatrix} \underline{u}_b \\ \underline{\xi} \end{bmatrix} = \begin{bmatrix} \underline{q} \\ 0 \end{bmatrix}. \tag{4.5}$$

Two other reductions can be considered: one eliminating all $\underline{u}$ and hence left with the Lagrange multipliers only:

$$S\underline{\xi} = BD^{-1}\underline{q} \tag{4.6}$$

with $S = BD^{-1}B^H$ and one eliminating all $\underline{\xi}$ and $\underline{u}_{p,b}$, $p = 1, \ldots, n$ and hence left with an independent set of degrees of freedom for the magnetic vector potential:

$$\begin{bmatrix} I & \\ & Q^H \end{bmatrix} \begin{bmatrix} K_{aa} & K_{ab} \\ K_{ba} & K_{bb} \end{bmatrix} \begin{bmatrix} I & \\ & Q \end{bmatrix} \begin{bmatrix} \underline{u}_a \\ \underline{u}_{0,b} \end{bmatrix} = \begin{bmatrix} I & \\ & Q^H \end{bmatrix} \begin{bmatrix} \underline{f}_a \\ \underline{f}_b \end{bmatrix}; \tag{4.7}$$

$$Q^H = \begin{bmatrix} I & F^H R_1 F & \cdots & F^H R_n F \end{bmatrix} \tag{4.8}$$

with $F^H = F^{-1}$ the inverse discrete Fourier transform. The operator Q assigns a particular set of rotating field components generated by the stator winding at Γ_b to each of the rotor sides of Γ_b.

5. Solution of the coupled system. For computational efficiency, the operators F, F^H, R_p and B are not constructed as matrices. Instead, we apply Fast Fourier Transforms (FFTs) for F and F^H and explicit restrictions for R_p. This excludes the usage of direct solution and algebraic preconditioning techniques. The coupled systems (4.3), (4.5), (4.6) and (4.7) are neither Hermitian nor complex symmetric and are solved by a preconditioned Bi-Conjugate Gradient Stabilised (BiCGStab) method [10]. An appropriate algebraic multigrid preconditioner $\tilde{K}_{\mathrm{AMG}}$ is available for the FE matrix part with circuit equations [9]. The block corresponding to the Lagrange multipliers has to be preconditioned by an approximation to system S [11, 13].

The constraint equation $B\underline{u}_b = 0$ enforces flux continuity whereas the relation $\underline{g}_b = B^H\underline{\xi}$ ensures the correct distribution of the magnetic field strength. The Lagrange multipliers represent the Fourier coefficients of the boundary integral terms $\underline{g}_{p,b}$ of the individual rotor domains. This physical interpretation indicates a possible problem-based preconditioning technique. A preconditioner is constructed based on a classical analytical model for cylindrical induction machines which neglects the stator and rotor slotting and the saturation of the ferromagnetic materials. The approximate model consists of a set of concentric rings, each with equivalent homogeneous material properties [7]. Here, the stator and the rotor are represented by a single homogeneous domain: the stator domain $\tilde{\Omega}_0$ with equivalent reluctivity $\tilde{\nu}_{\mathrm{st}}$ and the rotor domains $\tilde{\Omega}_p$ with equivalent reluctivities $\tilde{\nu}_{\mathrm{rt}}$, equivalent conductivities $\tilde{\sigma}_{\mathrm{rt}}$ and the slip pulsations ω_p (Fig. 3.1). The analytical relations for the Fourier coefficients $\underline{\tilde{h}}_{p,b,\lambda}$ of the magnetic field strength at the stator and rotor sides of Γ_b with respect to the Fourier coefficients $\underline{\tilde{a}}_{p,b,\lambda}$ of $\underline{A}_z$ at Γ_b are

$$\underline{\tilde{h}}_{0,b,\lambda} = \underline{\tilde{\nu}}_{\mathrm{st}} \frac{\lambda}{r_b} \frac{\gamma^\lambda + \gamma^{-\lambda}}{\gamma^\lambda - \gamma^{-\lambda}} \underline{\tilde{a}}_{0,b,\lambda} \tag{5.1}$$

$$\underline{\tilde{h}}_{p,b,\lambda} = -\underline{\tilde{\nu}}_{\mathrm{rt}} \frac{\beta_p I'_\lambda(\beta_p r_b)}{I_\lambda(\beta_p r_b)} \underline{\tilde{a}}_{p,b,\lambda} \tag{5.2}$$

with the factor $\beta_p = \sqrt{\mathrm{j}\omega_p \sigma_{\mathrm{rt}}/\nu_{\mathrm{rt}}}$, I_λ the modified Bessel function of order λ, $\gamma = r_b/r_s$ the stator form factor, r_b the radius of Γ_b and r_s the outer radius of the stator. Weighting (5.1) and (5.2) by the FE shape functions and integration along Γ_b yields an approximate relation between $\underline{\tilde{a}}_{p,b,\lambda}$ and the weighted magnetic field strengths $\underline{\tilde{g}}_{p,b,\lambda}$ they exert at the stator and rotor sides of Γ_b:

$$\underline{\tilde{g}}_{0,b,\lambda} = \underline{\tilde{\nu}}_{\mathrm{st}} \Delta\theta \kappa_\lambda \lambda \frac{\gamma^\lambda + \gamma^{-\lambda}}{\gamma^\lambda - \gamma^{-\lambda}} \underline{\tilde{a}}_{0,b,\lambda} \tag{5.3}$$

$$\underline{\tilde{g}}_{p,b,\lambda} = \underline{\tilde{\nu}}_{\mathrm{rt}} \Delta\theta \kappa_\lambda \frac{\beta_p r_b I'_\lambda(\beta_p r_b)}{I_\lambda(\beta_p r_b)} \underline{\tilde{a}}_{p,b,\lambda} \tag{5.4}$$

with $\kappa_\lambda = \frac{\sin(\lambda\Delta\theta/2)}{\lambda\Delta\theta/2}$ and $\Delta\theta$ the angle between two successive FE nodes at Γ_b. Expressions (5.3) and (5.4) are gathered in the matrix systems $\underline{\tilde{g}}_{p,b} = \tilde{H}_p \underline{\tilde{a}}_{p,b}$, further collected in $\underline{\tilde{g}}_b = \tilde{H}\underline{\tilde{a}}_b$, combined into a preconditioner for D and inserted in an approximation to S:

$$\tilde{D} = \mathrm{diag}\left(F^H\tilde{H}_0F, F^H\tilde{H}_1F, \ldots, F^H\tilde{H}_pF\right), \tag{5.5}$$

$$\tilde{S}_{\mathrm{dyn}} = B\tilde{D}^{-1}B^H = \mathrm{diag}\left(\tilde{H}_1^{-1} + R_1\tilde{H}_0^{-1}, \ldots, \tilde{H}_n^{-1} + R_n\tilde{H}_0^{-1}\right) \tag{5.6}$$

where the factors R_p introduce appropriate weights in order to account for the flux splitting at Γ_b. A similar approximation $\tilde{S}_{\mathrm{stat}}$ is built based on a static analytical model with $\omega_p = 0$ for all rotor models. Since all matrices in (5.6) are diagonal, the cost of applying $\tilde{S}_{\mathrm{dyn}}^{-1}$ or $\tilde{S}_{\mathrm{stat}}^{-1}$ to a vector is negligible. The systems (4.3), (4.5) and (4.6) can be solved by BiCGStab

Table 6.1: Number of iteration of BiCGStab with a block Jacobi preconditioner applied to a small motor model.

		1 rotor domain			6 rotor domains			12 rotor domains		
conductivity σ in S/m		50	500	5000	50	500	5000	50	500	5000
K	S	3	3	3	3	3	3	3	3	3
$\tilde{K}_{\mathrm{AMG}}$	S	6	6	6	7	6	6	7	6	6
$\tilde{K}_{\mathrm{ILU(0)}}$	S	51	46	24	125	109	57	173	145	57
K	$\tilde{S}_{\mathrm{ILU(0)}}$	21	19	13	200	184	98	198	195	100
$\tilde{K}_{\mathrm{ILU(0)}}$	$\tilde{S}_{\mathrm{ILU(0)}}$	43	43	24	59	71	50	61	101	50
K	$\tilde{S}_{\mathrm{stat}}$	5	5	7	5	5	7	5	5	5
$\tilde{K}_{\mathrm{AMG}}$	$\tilde{S}_{\mathrm{stat}}$	6	14	20	12	24	53	12	24	55
$\tilde{K}_{\mathrm{ILU(0)}}$	$\tilde{S}_{\mathrm{stat}}$	51	52	44	110	167	200	150	192	198
K	$\tilde{S}_{\mathrm{dyn}}$	3	3	3	3	3	3	3	3	3
$\tilde{K}_{\mathrm{AMG}}$	$\tilde{S}_{\mathrm{dyn}}$	7	6	6	7	6	6	7	6	6
$\tilde{K}_{\mathrm{ILU(0)}}$	$\tilde{S}_{\mathrm{dyn}}$	78	49	34	125	109	57	173	145	57

using as a preconditioner $\mathrm{diag}(\tilde{K}, \tilde{S})$, $\mathrm{diag}(\tilde{D}, \tilde{S})$ and $\tilde{S}$ respectively. We prefer to solve (4.3), preconditioned by $\mathrm{diag}(\tilde{K}, \tilde{S})$.

6. Numerical experiments. The performance of the preconditioner is tested for a small technical induction machine model (Table 6.1). The system (4.3) is solved by BiCGStab and preconditioned by a block Jacobi preconditioner of which the diagonal blocks are indicated in the table. The dependence of the number of iterations is checked with respect to the number of rotor domains and the importance of the eddy current effects, characterised by the conductivity. The system (4.3) preconditioned by $\mathrm{diag}(K, S)$ has three eigenvalues and, hence, converges in 3 steps [6]. Replacing the exact solution of the FE system part by the AMG preconditioner $\tilde{K}_{\mathrm{AMG}}$ causes only a small increase of the number of iterations. The next numerical test with the Incomplete LU-preconditioner without fill-in (ILU(0)) $\tilde{K}_{\mathrm{ILU(0)}}$ indicates that, for this problem, the choice of a good preconditioner for the FE part is more critical than the choice of the preconditioner for the Lagrange multiplier space. Notice that the construction of a Schur complement preconditioner using the preconditioner for the FE part, e.g. $\tilde{S}_{\mathrm{ILU(0)}} = B\tilde{K}^{-1}_{\mathrm{ILU(0)}}B^H$, is in practice too expensive. The number of iterations with the ILU(0) preconditioner may decrease with respect to increasing conductivity since then, the FE system becomes more diagonally dominant which explains the better performance of ILU(0). The last two numerical experiments demonstrate the performance of the preconditioners $\tilde{S}_{\mathrm{stat}}$ and $\tilde{S}_{\mathrm{dyn}}$ based on a static and dynamic analytical model respectively. For the static Schur complement preconditioner $\tilde{S}_{\mathrm{stat}}$, the number of iterations increases significantly with the conductivity due to the fact that eddy current effects are neglected in the Schur complement preconditioner. The more sophisticated analytical model from which $\tilde{S}_{\mathrm{dyn}}$ is constructed, leads to an iteration number independent from the conductivity. Because of the factors R_p in the Schur complement preconditioners, the number of iterations is not affected by the number of rotor domains.

7. Capacitor motor. The air gap flux decomposition technique is applied to a capacitor motor (Fig. 7.1). The fundamental forward and backward rotating air gap flux components, i.e., those with pole pair numbers 1 and -1, produce the most important torque components. Two rotor models are considered: Ω_1 for all components with positive pole pair numbers and Ω_2 for all components with negative pole pair numbers (Fig. 7.2). At Ω_1, the slip pulsation $\omega_1 = \omega - \omega_m$ is applied whereas at Ω_2, the slip pulsation $\omega_2 = \omega + \omega_m$, is used. Hence, only motional eddy current effects with respect to the fundamental rotating air gap

(a)

(b)

Figure 7.1: (a) Stator and (b) rotor of the capacitor motor.

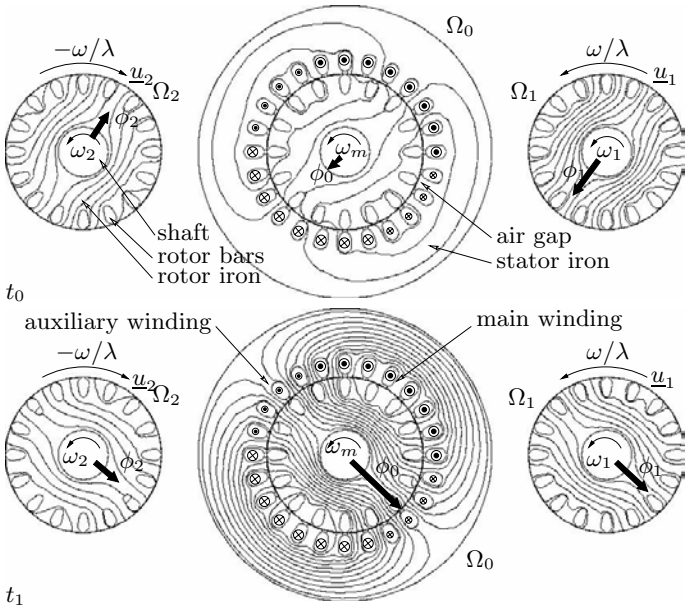


Figure 7.2: Plot of the magnetic flux lines at t_0 and t_1 of a capacitor motor operating at 1500 rotations per minute.

field components are correctly taken into account. The flux patterns plotted at two instants of time shifted over a quarter of a period, show the true alternating true rotor field ϕ_0 and the rotating forward and backward field components ϕ_1 and ϕ_2.

8. Conclusions. Motional eddy currents are considered within time-harmonic FE machine models by decomposing the air gap flux into rotating components and distributing these to independent rotor models. An appropriate solution scheme for the FE system incorporating interface conditions based on FFTs, consists of the BiCGStab algorithm and block preconditioning based on AMG for the FE part and an approximation for the interface Schur complement matrix based on approximate analytical machine models.

REFERENCES

[1] P. Alotto, A. Bertoni, I. Perugia, and D. Schötzau. Discontinuous finite element methods for the simulation of rotating electrical machines. *COMPEL*, 20(2):448–462, 2001.

[2] A. Buffa, Y. Maday, and F. Rapetti. A sliding mesh-mortar method for a two dimensional eddy-currents model of electric engines. Math. Meth. Anal. Num. (submitted), 1999.

[3] H. De Gersem and K. Hameyer. Air-gap flux splitting for the time-harmonic finite element simulation of single-phase induction machines. IEEE Trans. Magn., in print, mar 2002.

[4] R. De Weerdt, E. Tuinman, K. Hameyer, and R. Belmans. Finite element analysis of steady state behavior of squirrel cage induction motors compared with measurements. *IEEE Trans. Magn.*, 33(2):2093–2096, March 1997.

[5] A. Demenko. Movement simulation in finite element analysis of electric machine dynamics. *IEEE Trans. Magn.*, 32(3):1553–1556, May 1996.

[6] B. Fischer, A. Ramage, D. J. Silvester, and A. J. Wathen. Minimum residual methods for augmented systems. *BIT*, 38(3):527–543, 1998.

[7] E. M. Freeman. Equivalent circuits from electromagnetic theory: low-frequency induction devices. *IEE Proc.*, 121(10):1117–1121, October 1974.

[8] S. Kurz, J. Fetzer, G. Lehner, and W. M. Rucker. A novel formulation for 3D eddy current problems with moving bodies using a Lagragian description and FEM-BEM coupling. *IEEE Trans. Magn.*, 34(5):3068–3073, September 1998.

[9] D. Lahaye, K. Hameyer, and S. Vandewalle. An algebraic multilevel preconditioner for field-circuit coupled problems. *IEEE Trans. Magn.*, March 2002. in print.

[10] G. L. Sleijpen, H. A. Van der Vorst, and D. R. Fokkema. BICGSTAB(l) and other hybrid Bi-CG methods. *Num. Alg.*, 7:75–109, 1994.

[11] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations*. Cambridge University Press, 1996.

[12] I. A. Tsukerman, A. Konrad, G. Meunier, and J.-C. Sabonnadière. Coupled field-circuit problems: trends and accomplishments. *IEEE Trans. Magn.*, 29(2):1701–1704, March 1993.

[13] J. Xu and J. Zou. Some nonoverlapping domain decomposition methods. *SIAM Review*, 40:857–914, 1998.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

40. Indirect Method of Collocation for the Biharmonic Equation

M.A. Diaz[1], I. Herrera[2]

1. Introduction. Indirect methods of collocation (Trefftz-Herrera collocation), that were introduced in previous papers [5],[2], are formulated and applied to the biharmonic equation in two dimensions in combination with orthogonal collocation. The new approach allows relaxed continuity conditions. Two alternative procedures are considered and compared. The first one consists on the straight-forward application of the Trefftz-Herrera indirect collocation method to the biharmonic equation. From another hand, the second one uses the split formulation, also known as the mixed method of Ciarlet and Raviart, in which an auxiliary function is introduced and the biharmonic equation is rewritten as a coupled system of two Poisson equations. Then, to each one of these Poisson equations, Trefftz-Herrera indirect collocation method is applied. As illustration, some preliminary results of application of the last one approach to a numerical example are presented.

2. First Approach: Trefftz-Herrera Formulation for the Biharmonic Equation. In this Section, the general theory of Trefftz-Herrera DDM, presented in [6], will be applied to the biharmonic equation, when the problem is defined in a space of an arbitrary number of dimensions. The procedures are applicable to any kind of boundary conditions for which the problem is well-posed.

The notation is the same as that introduced in [6] and [3]. In particular, $u_\Omega \in \hat{D}_1$, $u_\partial \in \hat{D}_1$ and $u_\Sigma \in \hat{D}_1$ are any functions which satisfy the differential equation, the external boundary conditions and the jump conditions, respectively. and A partition of a domain Ω is being considered and the internal boundary is denoted by Σ (see [6] for further details).

Then, the boundary value problem with prescribed jumps (BVPJ) to be considered is

$$\Delta^2 u = f_\Omega, \quad in \quad \Omega \tag{2.1}$$

subjected to the boundary conditions

$$u = g^0_\partial \quad and \quad \Delta u = g^2_\partial, \quad on \quad \partial\Omega \tag{2.2}$$

and the jump conditions

$$[u] = j^0_\Sigma, \quad \left[\frac{\partial u}{\partial n}\right] = j^1_\Sigma, \quad [\Delta u] = j^2_\Sigma, \quad \left[\frac{\partial \Delta u}{\partial n}\right] = j^3_\Sigma, \quad on \quad \Sigma \tag{2.3}$$

Since the biharmonic operator $\mathcal{L}$ of Eq.(2.1) is self adjoint, i.e., $\mathcal{L} \equiv \mathcal{L}^*$, then its formal adjoint operator $\mathcal{L}^*$ is given by:

$$\mathcal{L}^* w \equiv \Delta\Delta w; \tag{2.4}$$

Introducing the bilinear vector valued function $\underline{\mathcal{D}}(u, w)$

$$\underline{\mathcal{D}}(u, w) \equiv w\nabla\Delta u + \Delta w\nabla u - \Delta u\nabla w - u\nabla\Delta w \tag{2.5}$$

which satisfies the property that

$$w\mathcal{L}u - u\mathcal{L}^* w = \nabla \cdot \underline{\mathcal{D}}\,(u, w) \tag{2.6}$$

[1]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM), mdiaz@tonatiuh.igeofcu.unam.mx

[2]Instituto de Geofísica Universidad Nacional Autónoma de México (UNAM) , iherrera@servidor.unam.mx

where

$$\begin{aligned} w\mathcal{L}u &\equiv -\nabla w\cdot(\nabla\Delta u)+\nabla\cdot(w\nabla\Delta u)\\ &= \Delta w\Delta u+\nabla\cdot(w\nabla\Delta u-\nabla w\Delta u)\end{aligned} \tag{2.7}$$

and

$$u\mathcal{L}^{*}w \equiv \Delta u\Delta w+\nabla\cdot(u\nabla\Delta w-\nabla u\Delta w) \tag{2.8}$$

Recalling that

$$\underline{\mathcal{D}}\,(u,w)\cdot\underline{n}=\mathcal{B}\,(u,w)-\mathcal{C}^{*}\,(u,w) \tag{2.9}$$

$$-\left[\underline{\mathcal{D}}\,(u,w)\right]\cdot\underline{n}=\mathcal{J}\,(u,w)-\mathcal{K}^{*}\,(u,w) \tag{2.10}$$

where

$$\mathcal{J}\,(u,w)=-\underline{\mathcal{D}}\,([u],\dot{w})\cdot\underline{n};\quad and\quad \mathcal{K}^{*}\,(u,w)=\underline{\mathcal{D}}\,(\dot{u},[w])\cdot\underline{n} \tag{2.11}$$

Then, the bilinear functions $\mathcal{B}(u,w)$ and $\mathcal{C}(w,u)$ in according to the boundary conditions given by Eqs. (2.2) may be defined as follow:

$$\mathcal{B}(u,w)\equiv(\Delta w)\frac{\partial u}{\partial n}-u\frac{\partial}{\partial n}(\Delta w),\quad \mathcal{C}(w,u)\equiv(\Delta u)\frac{\partial w}{\partial n}-w\frac{\partial}{\partial n}(\Delta u) \tag{2.12}$$

and correspondingly $\mathcal{J}(u,w)$ and $\mathcal{K}(w,u)$ are defined as:

$$\mathcal{J}(u,w)\equiv\ [u]\overline{\frac{\partial\dot{\Delta w}}{\partial n}}-\left[\frac{\partial u}{\partial n}\right]\overline{\dot{\Delta w}}+[\Delta u]\overline{\frac{\partial\dot{w}}{\partial n}}-\left[\frac{\partial\Delta u}{\partial n}\right]\dot{w}, \tag{2.13}$$

$$\mathcal{K}(w,u)\equiv\ -\dot{u}\left[\frac{\partial\Delta w}{\partial n}\right]+\overline{\frac{\partial\dot{u}}{\partial n}}[\Delta w]-\overline{\dot{\Delta u}}\left[\frac{\partial w}{\partial n}\right]+\overline{\frac{\partial\dot{\Delta u}}{\partial n}}[w] \tag{2.14}$$

Introducing the weak decompositions $\{S_J,R_J\}$ and $\{S,R\}$ of J and K, respectively, as was defined in [6]:

$$\mathcal{S}_J(u,w)\equiv-\dot{w}\left[\frac{\partial\Delta u}{\partial n}\right]-\overline{\dot{\Delta w}}\left[\frac{\partial u}{\partial n}\right],\quad \mathcal{R}_J(u,w)\equiv[u]\overline{\frac{\partial\dot{\Delta w}}{\partial n}}+[\Delta u]\overline{\frac{\partial\dot{w}}{\partial n}} \tag{2.15}$$

$$\mathcal{S}^{*}(u,w)\equiv\ -\dot{u}\left[\frac{\partial\Delta w}{\partial n}\right]-\overline{\dot{\Delta u}}\left[\frac{\partial w}{\partial n}\right],\quad \mathcal{R}^{*}(u,w)\equiv[w]\overline{\frac{\partial\dot{\Delta u}}{\partial n}}+[\Delta w]\overline{\frac{\partial\dot{u}}{\partial n}} \tag{2.16}$$

Thus, the bilinear functionals P, B, J, S_J, R_J, Q^{*}, C^{*}, K^{*}, S^{*} and R^{*} are defined in the same fashion of Eqs. (5.6)-(5.8) given in Ref. [6], by means of corresponding integrals.

Define $\tilde{N}_1\equiv N_P\cap N_B\cap N_{R_J}$ and $\tilde{N}_2\equiv N_Q\cap N_C\cap N_R$, but $\tilde{N}_1\equiv\tilde{N}_2\equiv\tilde{N}$ since the biharmonic operator is self adjoint. Then, a function $\phi\in\tilde{N}$, if and only if

$$\begin{aligned} &\Delta\Delta\phi=0,\quad in\quad \Omega_i\quad (i=1,...,E)\\ &\phi=\Delta\phi=0,\quad on\quad \partial\Omega\\ &[\phi]=[\Delta\phi]=0,\quad on\quad \Sigma\end{aligned} \tag{2.17}$$

Applying the Theorem of Section 10 of Ref. [6] a Trefftz-Herrera domain decomposition procedure can be obtained:

Assume $\mathcal{E}\subset\tilde{N}$ is a system of weighting functions TH-complete for S^{*} [6]. Let $u_P\in\hat{D}_1$ be such that

$$Pu_P=Pu_\Omega,\quad Bu_P=Bu_\partial\quad and\quad R_Ju_P=R_Ju_\Sigma \tag{2.18}$$

Then there exists $v \in \tilde{N}$ such that

$$-\langle S^* v, w\rangle = \langle S_J\left(u_P - u_\Sigma\right), w\rangle\,, \quad \forall w \in \mathcal{E} \subset \tilde{N} \tag{2.19}$$

In addition, define $\hat{u} \in \hat{D}_1$ by $\hat{u} \equiv u_P + v$. Then $\hat{u} \in \hat{D}_1$ contains the sought information. Even more, $\hat{u} \equiv u$, where u is the solution of the BVPJ.

The general outlines about the construction of a TH-complete system of weighting functions $\mathcal{E} \subset \tilde{N}$ for the biharmonic equation are given in [4]. As is known, TH-complete systems for such problems in several dimensions are constituted by infinite families, but, in numerical implementation, only one can use finite sets of test functions produced by means of numerical methods.

In particular, one may construct such systems of test functions solving local BVP problems of Eqs. (2.17) applying collocation method for families of piecewise polynomials of degree less or equal to G on the internal boundary Σ, where G is a given number, in the same manner as was developed in [5],[2] for the second order elliptic equation. In this sense, an indirect Trefftz-Herrera collocation method is obtained, which possesses the property that its global matrix is symmetric and positive definite.

3. Second Approach: Trefftz-Herrera Collocation for the Biharmonic Equation using the splitting formulation. A common approach for solving the biharmonic equation is to use the splitting principle in which an auxiliary function $v = \Delta u$ is introduced and the biharmonic equation is rewritten as a system of two Poisson equations in the form [7]:

$$\left\{\begin{array}{ll} -\Delta u = -v; & \\ -\Delta v = -f_\Omega; & in \quad \Omega \end{array}\right. \tag{3.1}$$

In the context of the finite element Galerkin method, this approach is known as the mixed method of Ciarlet and Raviart [1].

Using the splitting principle of Eq.(3.1), the boundary value problem with prescribed jumps (BVPJ) of the previous section Eqs. (2.1), (2.2) and (2.3), becomes one of solving sequentially two nonhomogeneous Dirichlet problems with prescribed jumps for Poisson's equation:

$$\left\{\begin{array}{l} -\Delta u = -v; \quad in \quad \Omega \\ u = g_1; \quad on \quad \partial\Omega \\ [u] = j^0, \quad \left[\frac{\partial u}{\partial n}\right] = j^1 \quad on \quad \Sigma \end{array}\right. \quad and \quad \left\{\begin{array}{l} -\Delta v = -f_\Omega; \quad in \quad \Omega \\ v = g_2; \quad on \quad \partial\Omega \\ [v] = j^2, \quad \left[\frac{\partial v}{\partial n}\right] = j^3, \quad on \quad \Sigma \end{array}\right. \tag{3.2}$$

The resulting coupled system of equations (3.2) can be solved applying the indirect Trefftz-Herrera collocation procedures, developed in [5],[2] for the second order elliptic equation, sequentially to each one of the BVPJs of Eq. 3.2).

In short, the algorithms reported in papers [5],[2] have the following features:

Algorithm I.- The family of test functions, using linear polynomials on the internal boundary Σ, contained only one member associated with each internal node. This leads to an algorithm in which only one degree of freedom is associated with each internal node. The resulting global matrix for each one of the BVPJs of Eqs. (3.2) is nine-diagonal, symmetric and positive definite.

Algorithm II.- Using cubic polynomials on the internal boundary Σ, a family of test functions is composed by three functions (or less, at those nodes in which some of the functions of this family do not satisfy the required zero boundary condition on the external boundary) associated with each node, including boundary nodes. This leads to an algorithm in which three, or less, degrees of freedom are associated with each node. The global matrix for each

one of the BVPJs of Eqs. (3.2) is block nine-diagonal, with blocks 3x3, symmetric and positive definite.

Here, it is worth to point out, that the resulting systems in both previous algorithm are symmetric and positive definite and consequently they can be solved using a Conjugate Gradient Method. In contrast, hermite collocation method does not enjoy this property.

4. The Numerical Experiments. In this section, some preliminary results of application of the second approach to a numerical example are presented.

The numerical experiments were carried out for the following BVPJ of the biharmonic equation in two dimensions:

$$\Delta^2 u = f_\Omega; \quad in \quad \Omega = [0,1] \times [0,1] \tag{4.1}$$

where the right hand side term is $f_\Omega = 24\left(e^x + e^y\right) + \left(y^2-1\right)^2 e^x + \left(x^2-1\right)^2 e^y + 8\left[\left(3y^2-1\right)e^x + \left(3x^2-1\right)e^y\right]$ and the corresponding analytical solution has the expression:

$$u(x,y) = \left(y^2-1\right)^2 e^x + \left(x^2-1\right)^2 e^y \tag{4.2}$$

Consequently, the imposed boundary conditions implied by the analytical solution were:

$$u = \left(y^2-1\right)^2 e^x + \left(x^2-1\right)^2 e^y; \quad on \quad \partial\Omega \tag{4.3}$$

$$\Delta u = \left(y^2-1\right)^2 e^x + 4\left(3x^2-1\right)e^y + 4\left(3y^2-1\right)e^x + \left(x^2-1\right)^2 e^y; \quad on \quad \partial\Omega \tag{4.4}$$

and it was considered the continuous case, i.e, the jump conditions imposed were taken equal to zero.

The numerical results are summarized in Figures 4.1 and 4.2. Each one of the examples was solved in a uniform rectangular partition ($E = E_x = E_y$) of the domain using Algorithm I and, subsequently, Algorithm II, for which the weighting functions are piecewise linear and piecewise cubic, respectively, on Σ. The convergence rate of the error -measured in terms of the norm $\|.\|_\infty$ - is $O\left(h^2\right)$ and $O\left(h^4\right)$ respectively, as shown in those figures.

5. Conclusions. In the present article, the indirect approach to domain decomposition methods has been applied to the BVPJ for the biharmonic equation using two different approaches. In the first one, the Trefftz-Herrera indirect method has been applied in straightforward manner to the biharmonic equation without further elaboration, while, in the second one, a BVPJ for the biharmonic equation has been reduced to a system of two BVPJs for Poisson's equation. In both cases, when the numerical procedure which is used for producing the local solutions is collocation, a non-standard method of collocation is obtained which possesses several attractive features. Indeed, a reduction with respect to other collocation methods, in the number of degrees of freedom associated with each node is obtained. This is due to the relaxation in the continuity conditions required by indirect methods-. Also, the global matrix is symmetric and positive definite when so is the differential operator, while in the standard method of collocation, using Hermite cubics, this does not happen. In addition, it must be mentioned that the boundary value problem with prescribed jumps at the internal boundaries can be treated as easily as the smooth problem -i.e., that with zero jumps-, because the solution matrix and the order of precision is the same for both problems. It must be observed also that, when the indirect method is applied, the error of the approximate solution stems from two sources: the approximate nature of the test functions, and the fact that TH-complete systems of test functions -which are infinite for problems in several dimensions- are approximated by finite families of such functions.

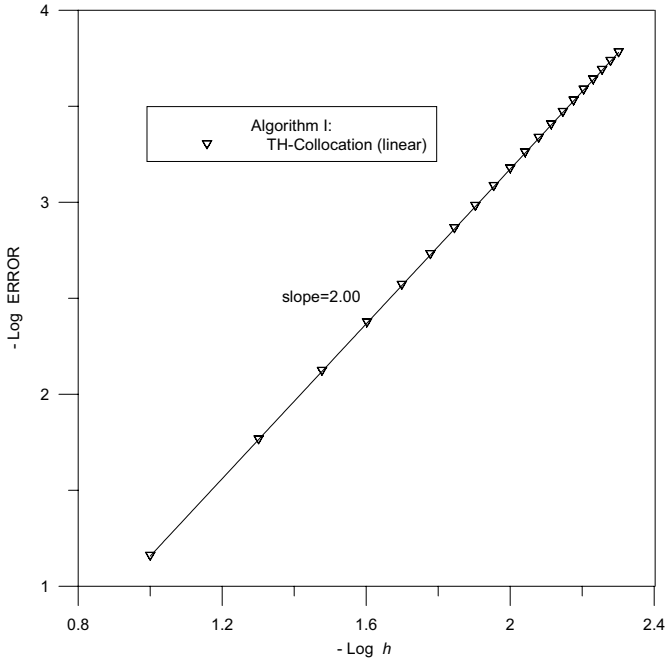


Figure 4.1: Convergence rate of Trefftz-Herrera collocation method for Algorithm I (using linear weighting functions).

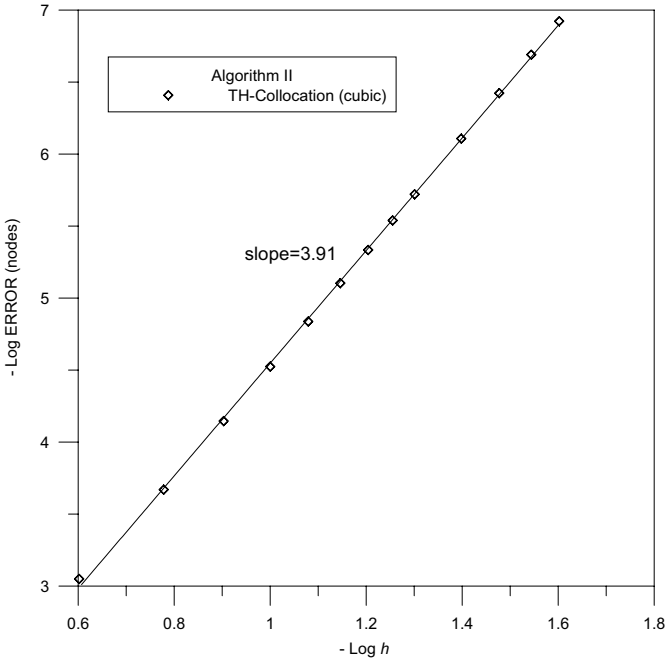


Figure 4.2: Convergence rate of Trefftz-Herrera collocation method for Algorithm II (using cubic weighting functions).

REFERENCES

[1] P. G. Ciarlet and P. A. Raviart. A Mixed Finite Element Method for the Biharmonic Equation. In C. de Boor, editor, *Proc. Symp. on Mathematical Aspects of Finite Elements in PDE, Academic Press, New York*, pages 125–145, 1974.

[2] M. Diaz, I. Herrera, and R. Yates. Indirect Method of Collocation: Second Order Equations. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[3] F. Garcia-Nocetti, I. Herrera, R. Yates, E. Rubio, and L. Ochoa. The Direct Approach to Domain Decomposition Methods. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[4] H. Gourgeon and I. Herrera. Boundary Methods. C-Complete Systems for Biharmonic Equations. In C. A. Brebbia, editor, *Boundary Element Methods, Springer-Verlag, Berlin*, pages 431–441, 1981.

[5] I. Herrera, R. Yates, and M. Diaz. General Theory of Domain Decomposition: Indirect Methods. *Numerical Methods for Partial Differential Equations*, 18(3):296–322, 2002.

[6] I. Herrera, R. Yates, and M. Diaz. The Indirect Approach to Domain Decomposition. In I. Herrera, D. E. Keyes, O. B. Widlund, and R. Yates, editors, *14th International Conference on Domain Decomposition Methods, Cocoyoc, Mexico*, 2002.

[7] Z.-M. Lou, B. Bialecki, and G. Fairweather. Orthogonal Spline Collocation Methods for Biharmonic Problems. *Numer. Math.*, 80:267–303, 1998.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

41. Toward scalable FETI algorithm for variational inequalities with applications to composites

Zdeněk Dostál, David Horák, Oldřich Vlach [1]

1. Introduction. In this paper we review our results related to development of scalable algorithms for solution of variational inequalities. After describing a model problem, we apply the FETI methodology to reduce it to the quadratic programming problem with equality and non-negativity constraints. Then we present the basic algorithm with a "natural coarse grid" proposed by Dostál, Friedlander, Santos and Gomes [9, 10, 12] and report recent theoretical results that may be used either to prove scalability of parts of the basic algorithm or to modify the basic algorithm so that it is scalable. Finally we give results of parallel solution of the model problem discretized by up to more than eight million of nodal variables and show application of the algorithm to analysis of fibrous composite material that was studied by Wriggers [20]. The results related to development of scalable algorithms for elliptic variational inequalities include experimental evidence of numerical scalability of the algorithm based on monotone multigrid [17] by Kornhuber. Another interesting algorithm was proposed by Schöberl [18]. Also the authors of the original FETI method proposed its adaptation to the solution of variational inequalities and gave experimental evidence of numerical scalability of their algorithm with a coarse grid initial approximation [14]. Let us recall that the FETI (Finite Element Tearing and Interconnecting) method proposed by Farhat and Roux [16] for solving of linear elliptic boundary value problems is based on the decomposition of the spatial domain into non-overlapping subdomains that are "glued" by Lagrange multipliers. Using the so called "natural coarse grid", Farhat, Mandel and Roux [15] modified the basic FETI algorithm so that they were able to prove its numerical scalability. These results are key ingredients in our research.

2. Model problem. Let $\Omega = \Omega^1 \cup \Omega^2$, $\Omega^1 = (0,1) \times (0,1)$ and $\Omega^2 = (1,2) \times (0,1)$ denote open domains with boundaries Γ^1, Γ^2 decomposed into $\Gamma^1_u = \{(x_1, x_2) \in \Gamma^1 : x_1 = 0\}$, $\Gamma^i_c = \{(x_1, x_2) \in \Gamma^i : x_1 = 1\}$, and Γ^i_f formed by the remaining sides of $\Omega^i, i = 1, 2$. Let $H^1(\Omega^i)$ denote the Sobolev space of first order on the space $L^2(\Omega^i)$ of the functions on Ω^i whose squares are integrable in the sense of Lebesgue. Let

$$V^1 = \left\{ v \in H^1(\Omega^1) : v^1 = 0 \quad \text{on} \quad \Gamma^1_u \right\}$$

denote the closed subspace of $H^1(\Omega^1)$, $V^2 = H^1(\Omega^2)$, and let

$$V = V^1 \times V^2 \qquad \text{and} \qquad \mathcal{K} = \left\{ (v^1, v^2) \in V : v^2 - v^1 \geq 0 \quad \text{on} \quad \Gamma_c \right\}$$

denote a closed subspace and a closed convex subset of $\mathcal{H} = H^1(\Omega^1) \times H^1(\Omega^2)$, respectively. The relations on the boundaries are in terms of traces. On $\mathcal{H}$ we shall define a symmetric bilinear form

$$a(u, v) = \sum_{i=1}^{2} \int_{\Omega_i} \left(\frac{\partial u^i}{\partial x} \frac{\partial v^i}{\partial x} + \frac{\partial u^i}{\partial y} \frac{\partial v^i}{\partial y} \right) d\Omega$$

and a linear form

$$\ell(v) = \sum_{i=1}^{2} \int_{\Omega_i} f^i v^i d\Omega,$$

[1]Department of Applied Mathematics, FEI VŠB-Technical University Ostrava, Tř 17. listopadu, CZ-70833 Ostrava, Czech Republic
zdenek.dostal@vsb.cz, david.horak@vsb.cz, oldrich.vlach2@vsb.cz,

where $f^i \in L^2(\Omega^i), i = 1, 2$ are the restrictions of

$$f(x,y) = \left\{ \begin{array}{c} -3 \quad \text{for} \quad (x,y) \in (0,1) \times [0.75,1) \\ 0 \quad \text{for} \quad (x,y) \in (0,1) \times [0,0.75) \quad \text{and} \quad (x,y) \in (1,2) \times [0.25,1) \\ -1 \quad \text{for} \quad (x,y) \in (1,2) \times [0,0.25) \end{array} \right\}.$$

Thus we can define a problem

$$\text{Minimize} \;\; q(u) = \frac{1}{2} a(u,u) - \ell(u) \;\; \text{subject to} \;\; u \in \mathcal{K}. \tag{2.1}$$

More details about this model problem including a discussion of the existence and uniqueness may be found in [9].

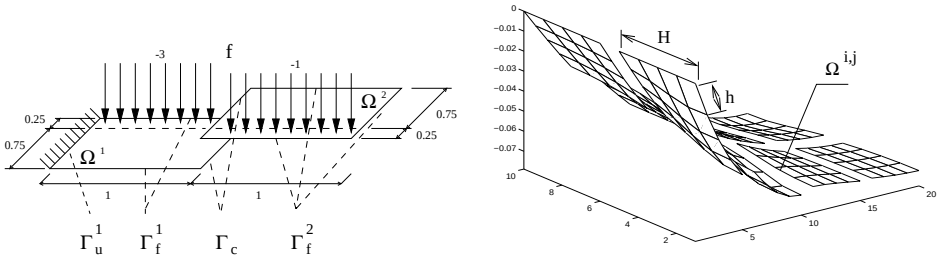


Figure 2.1: Model problem and its solution

3. Domain decomposition and discretized problem with a natural coarse grid. To enable efficient application of the domain decomposition methods, we can optionally decompose each Ω^i into square subdomains $\Omega^{i1}, \ldots, \Omega^{ip}, p = s^2 > 1$. The continuity in Ω^1 and Ω^2 of the global solution assembled from the local solutions u^{ij} will be enforced by the "gluing" conditions $u^{ij}(x) = u^{ik}(x)$ that should be satisfied for any x in the interface $\Gamma^{ij,ik}$ of Ω^{ij} and Ω^{ik}. After modifying appropriately the definition of problem (2.1), introducing regular grids in the subdomains Ω^{ij} that match across the interfaces $\Gamma^{ij,kl}$, indexing contiguously the nodes and entries of corresponding vectors in the subdomains, and using the finite element discretization, we get the discretized version of problem (2.1) with the auxiliary domain decomposition that reads

$$\min \frac{1}{2} x^T A x - f^T x \quad \text{s.t.} \quad B_I x \le 0 \quad \text{and} \quad B_E x = 0. \tag{3.1}$$

In (3.1), A denotes a positive semidefinite stiffness matrix, the full rank matrices B_I and B_E describe the discretized inequality and gluing conditions, respectively, and f represents the discrete analog of the linear term $\ell(u)$. Details may be found in [9]. Introducing the notation

$$\lambda = \left[\begin{array}{c} \lambda_I \\ \lambda_E \end{array} \right] \quad \text{and} \quad B = \left[\begin{array}{c} B_I \\ B_E \end{array} \right],$$

we can observe that B is a full rank matrix and write the Lagrangian associated with problem (3.1) briefly as

$$L(x,\lambda) = \frac{1}{2} x^T A x - f^T x + \lambda^T B x.$$

It is well known that (3.1) is equivalent to the saddle point problem

$$\text{Find} \quad (\overline{x}, \overline{\lambda}) \quad \text{s.t.} \quad L(\overline{x}, \overline{\lambda}) = \sup_{\lambda_I \geq 0} \inf_{x} L(x, \lambda). \tag{3.2}$$

After eliminating the primal variables x from (3.2), we shall get the minimization problem

$$\min \Theta(\lambda) \quad \text{s.t.} \quad \lambda_I \geq 0 \quad \text{and} \quad R^T(f - B^T\lambda) = 0, \tag{3.3}$$

where

$$\Theta(\lambda) = \frac{1}{2}\lambda^T B A^\dagger B^T \lambda - \lambda^T B A^\dagger f, \tag{3.4}$$

$A^\dagger$ denotes a generalized inverse that satisfies $AA^\dagger A = A$, and R denotes the full rank matrix whose columns span the kernel of A. Using the fact that $R^T B^T$ is a full rank matrix, it may be verified that the Hessian of Θ is positive definite. Even though problem (3.3) is much more suitable for computations than (3.1) and was used to efficient solving of the discretized variational inequalities [7], further improvement may be achieved by adapting some simple observations and the results of Farhat, Mandel and Roux [15]. Let us denote

$$\begin{aligned} F &= BA^\dagger B^T, & \widetilde{d} &= BA^\dagger f, \\ \widetilde{G} &= R^T B^T, & \widetilde{e} &= R^T f, \end{aligned}$$

and let $\widetilde{\lambda}$ solve $\widetilde{G}\widetilde{\lambda} = \widetilde{e}$. Let $d = \widetilde{d} - F\widetilde{\lambda}$ and let G denote a regular matrix with orthonormal rows and the same kernel as $\widetilde{G}$, so that

$$Q = G^T G \qquad \text{and} \qquad P = I - Q$$

are the orthogonal projectors on the image space of G^T and on the kernel of G, respectively. Problem (3.3) may then be reduced to

$$\min \quad \frac{1}{2}\lambda^T PFP\lambda - \lambda^T Pd \quad \text{s.t} \quad G\lambda = 0 \quad \text{and} \quad \lambda_I \geq -\widetilde{\lambda_I}. \tag{3.5}$$

The Hessian $H_\rho = PFP + \rho Q$ of the augmented Lagrangian

$$L(\lambda, \mu, \rho) = \frac{1}{2}\lambda^T(PFP + \rho Q)\lambda - \lambda^T Pd + \mu^T G\lambda \tag{3.6}$$

is decomposed by the projectors P and Q whose image spaces are invariant subspaces of H_ρ. The key point is that the analysis by Farhat, Mandel and Roux [15] implies that the *spectral condition number* $\kappa(H_\rho)$ of H_ρ is *bounded independently* of h for a regular decomposition provided H/h is uniformly bounded, where h and H are the mesh and subdomain diameters, respectively.

4. Solution of bound and equality constrained quadratic programming problems and optimal penalty. Dostál, Friedlander and Santos [8] proposed a variant of the augmented Lagrangian type algorithm by Conn, Gould and Toint [3] that fully exploits the specific structure of problem (3.3). To describe it, let us recall that the gradient of the augmented Lagrangian 3.6 is given by

$$g(\lambda, \mu, \rho) = PFP\lambda - Pd + G^T(\mu + \rho G\lambda),$$

so that the *projected gradient* $g^P = g^P(\lambda, \mu, \rho)$ of L at λ is given componentwise by

$$g_i^P = g_i \ \text{ for } \ \lambda_i > -\widetilde{\lambda}_i \ \text{ or } \ i \notin I \ \text{ and } \ g_i^P = g_i^- \ \text{ for } \ \lambda_i = -\widetilde{\lambda}_i \ \text{ and } \ i \in I$$

with $g_i^- = \min(g_i, 0)$, where I is the set of indices of constrained entries of λ.

Algorithm 4.1 (Quadratic programming with simple bound and equality constraints)

Step 0. Set $0 < \alpha < 1,\ 1 < \beta,\ \rho_0 > 0,\ \eta_0 > 0,\ M > 0,\ \mu^0$ and $k = 0$.

Step 1. Find λ^k so that $||g^P(\lambda^k, \mu^k, \rho_k)|| \leq M||G\lambda^k||$.

Step 2. If $||g^P(\lambda^k, \mu^k, \rho_k)||$ and $||G\lambda^k||$ are sufficiently small, then stop.

Step 3. $\mu^{k+1} = \mu^k + \rho_k G\lambda^k$

Step 4. If $||G\lambda^k|| \leq \eta_k$

Step 4a. then $\rho_{k+1} = \rho_k$, $\eta_{k+1} = \alpha\eta_k$

Step 4b. else $\rho_{k+1} = \beta\rho_k$, $\eta_{k+1} = \eta_k$

end if.

Step 5. Increase k by one and return to Step 1.

The algorithm has been proved [8] to converge for any set of parameters that satisfy the prescribed relations. Moreover, it has been proved that the asymptotic rate of convergence is the same as for the algorithm with an exact solution of the auxiliary QP problems (i.e. $M = 0$) and that the penalty parameter is uniformly bounded. These results give theoretical support to Algorithm 4.1. The performance of the algorithm depends essentially on the rate of convergence of the method that minimizes L in the inner loop as the number of the outer iterations was rather small ranging from two to six. We use the active set strategy in combination with the proportioning conjugate gradient algorithm [4] and the gradient projection [18]. We managed to get the rate of convergence for the inner loop in terms of $\kappa(H_\rho)$ [6]. Combining this result with that on the boundedness of $\kappa(H_\rho)$, *we find that the rate of convergence in the inner loop does not depend on the discretization parameter* h . The best results were achieved with relatively high penalty parameters which may be explained by the fact that it is possible to give the rate of convergence for the conjugate gradient method for minimization of the quadratic form with the Hessian H_ρ that depends neither on ρ nor on the rank of G [5]. This suggests that we could try to enforce the equality constraints by the *penalty method*. Closer inspection reveals nice optimality property of the penalty method applied to 3.5, namely if H/h is bounded, then there is a constant C independent of h such that if $\|g^P(\lambda, 0, \rho)\| \leq \epsilon\|Pd\|$, then

$$\|G\lambda\| \leq \frac{C(1+\epsilon)}{\rho}\|Pd\|.$$

It follows that using the penalty in combination with the algorithm with the rate of convergence, it is possible to *get an approximate solution with the prescribed precision in a number of iterations independent of the discretization parameter* h. We shall give the details elsewhere. Another way to achieve scalability, at least for coercive problems, is to apply FETI-DP method [2].

5. Numerical experiments. In this section we report some results of numerical solution of the model problem of Section 2 and of a problem with the fibrous composite material in order to illustrate the performance of the algorithm, in particular its numerical and parallel scalability. To this end, we have implemented Algorithm 4.1 in C exploiting PETSc [1] to solve the basic dual problem (3.3) so that we could plug in the projectors to the natural coarse space (3.5) and the dual penalty method. Each domain $\Omega^i,\ i = 1, 2$ was first decomposed into identical rectangles Ω^{ij} with the sides H that were discretized by the regular grids defined by the stepsize h as in Figure 2.1. The stopping criterium $\left\|g^P(\lambda, \mu, 0)\right\| \leq 10^{-4}\|d\|$ and $\|G\lambda\| \leq 10^{-4}\left\|(\widetilde{G}G^T)^{-1}\widetilde{e}\right\|$ was used in all calculations.

Solution of the model problem for $h = 1/8$ and $H = 1/2$ is in Figure 2.1. The experiments were run on the Lomond 18-processor Sun HPC 6500 Ultra SPARC-II based SMP system with 400 MHz, 18 GB of shared memory, 90 GB disc space, nominal peak performance 14.4 GFlops, 16 kB level 1 and 8 MB level 2 cache of the EPCC Edinburgh, and on the SGI Origin 3800 128-processor R12000 shared memory (MIMD) system with 400 Mhz, 48.128 GB of RAM, 500 GB disc space, FDDI 1 Gb/sec of the Johannes Kepler University Linz. All the computations were carried out with parameters $M = 1, \rho_0 = 10, \Gamma = 1, \lambda^0 = \frac{1}{2}Bf$.

Table 5.1: Parallel scalability for 128 subdomains

processors	1	2	4	8	16	32	64	128
Time[sec]	1814.0	566.4	185.9	54.5	32.0	32.7	62.5	147.0

Table 5.2: Performance for varying decomposition and discretization

H	1	1/2	1/4	1/8
$H/h \setminus$ procs	2	8	16	16
128	33282/129/41.95 28	133128/1287/89.50 59	532512/6687/74.9 36	2130048/29823/421.5 47
32	2178/33/0.20 17	8712/327/0.50 33	34848/1695/1.48 30	139392/7551/11.66 37
8	162/9/0.03 10	648/87/0.10 20	2592/447/0.39 23	10365/1983/2.06 27

Table 5.3: Highlights

h	H	prim. dim.	dual. dim.	num. of subdom.	procs	out. iter.	cg. iter.	time [sec]
1/1024	1/8	2130048	29823	128	32	2	47	167.8
1/2048	1/8	8454272	59519	128	64	2	65	1281.0

The selected results of the computations are summed up in Tables 5.1 - 5.4. Table 5.1 indicates that the algorithm presented enjoys high parallel scalability for problem with $h = 1/512, H = 1/8$, primal dimension 540800, and dual dimension 14975 that was solved on the computer SGI Origin. Table 5.2 indicates that Algorithm 4.1 may enjoy also high numerical scalability, even though the latter is so far supported by theory only for the inner loops of the algorithm. In particular, for varying decompositions and discretizations, the upper row of each field of the table gives the corresponding primal/dual dimensions and times in seconds on the Lomond, while the number in the lower row gives a number of the conjugate gradient iterations that were necessary for the solution of the problem to the given precision. We can see that the number of the conjugate gradient iterations for a given ratio H/h varies very moderately. The results for the largest problems using the SGI Origin are in Table 5.3. Optimality of dual penalty is illustrated in Table 5.4. We conclude that at least for our model problem, the experiments indicate that the cost of numerical solution of the variational inequalities may be comparable to the cost of solution of corresponding linear problem. To check the performance and robustness of our algorithm on a more challenging problem, we considered linear elastic response of a sample of fibrous composite material. For simplicity, we assumed that the fibers are inserted into the matrix

Table 5.4: Optimal enforcing of $\|G\lambda\| \,/\, \|d\|$

prim.dim./dual.dim.		1152/591	10368/1983	139392/7551	2130048/29823
$\|G\lambda\| \,/\, \|d\|$	$\rho = 10$	3.027e-03	3.108e-03	3.115e-03	3.117e-03
	$\rho = 1000$	3.144e-05	3.213e-05	3.222e-05	3.225e-05
	$\rho = 100000$	3.145e-07	3.212e-07	3.224e-07	

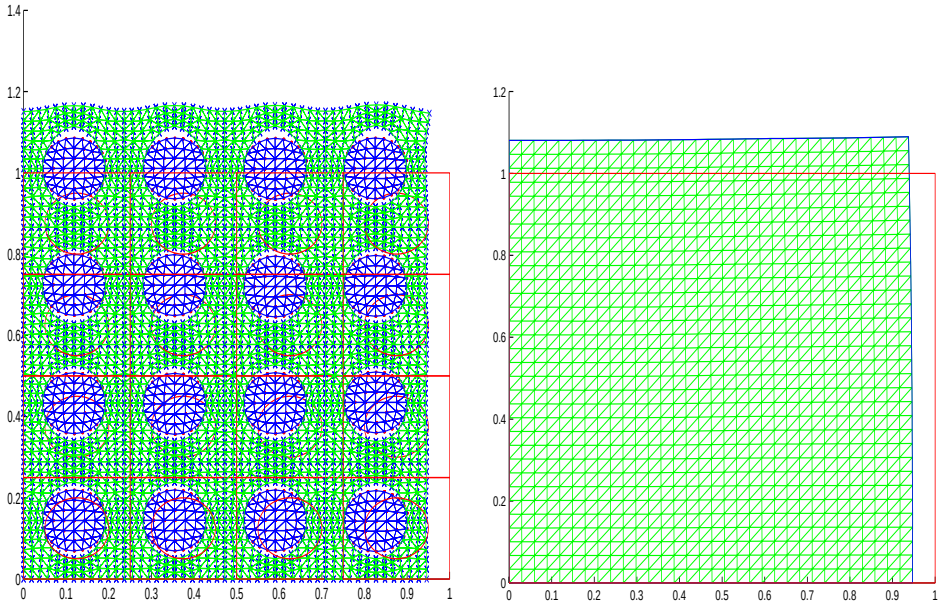


Figure 5.1: Fiber composite and homogeneous sample in strain

so that there is no adhesion. Such composites may be useful e.g. in design of materials with different response to strain and stress. The algorithm may be modified to model debonding of more realistic material as considered in Wriggers [20]. The problem is difficult due to the long, split, a priori unknown contact interface with many points in which the condition of strict complementarity is violated. We decomposed the space domain of the sample into squares, each square comprising two subdomains consisting of a circular fiber and corresponding part of the matrix, and discretized the problem as in Figure 5.1. The primal and dual dimensions of the problem were 6176 and 990, respectively. The number of outer iterations was only four, while the solution to the relative precision 1E-4 required 591 iterations with 157 faces examined. We conclude that the performance of the algorithm is acceptable even for problems where the results related to scalability mentioned above do not apply as in this case when the decomposition includes subdomains that are not simply connected.

6. Comments and conclusions. We have reviewed a domain decomposition algorithm for the solution of variational inequalities. The method combines a variant of the FETI method with projectors to the natural coarse grid and recently developed algorithms for the solution of special QP problems. We have also introduced the penalty approximation that is optimal in the sense that a fixed penalty parameter can enforce feasibility to the prescribed relative precision regardless of the discretization parameter. The theory gives relevant results concerning the scalability of main parts of the basic algorithm and yields full theoretical support of its variants presented in the paper. Numerical experiments with the model variational inequality discretized by up to more than eight million of nodal variables indicate that even the basic algorithm may enjoy full numerical and parallel scalability and confirm a kind of optimality for the dual penalty. Numerical solution of a problem with fibrous composite material confirm that the algorithm presented is effective for more challenging problems. In fact, it has already been exploited for solving 2D problems with Coulomb friction [11] and contact shape optimization [13, 19].

Acknowledgements: This research was supported by grants GAČR 103/01/0400, GAČR 101/01/0538, AV ČR S3086102, and CEZ J:17/98:272400019. The authors would like to acknowledge also the support of the European Commission through grant number HPRI-CT-1999-00026 (the TRACS Programme at EPCC). The experiments on the cluster in Linz were supported by the grant of *ÖAD* of Austrian government and hospitable environment of SFB013 at Johannes Kepler University Linz.

REFERENCES

[1] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. *PETSc 2.0 User Manual*. Argonne National Laboratory, http://www.mcs.anl.gov/petsc/, 1997.

[2] P. L. K. P. C. Farhat, M. Lesoinne and D. Rixen. Feti-dp: A dual-primal unified feti method - part i: A faster alternative to the two-level feti method. *International Journal for Numerical Methods in Engineering*, 2000. in press.

[3] A. R. Conn, N. I. M. Gould, and P. L. Toint. A globally convergent augmented Lagrangian algorithm for optimization with general constraints and simple bounds. *SIAM J. Num. Anal.*, 28:545–572, 1991.

[4] Z. Dostál. Box constrained quadratic programming with proportioning and projections. *SIAM J. Opt.*, 7:871–887, 1997.

[5] Z. Dostál. On preconditioning and penalized matrices. *Num. Lin. Alg. Appl. 6*, pages 109–114, 1999.

[6] Z. Dostál. A proportioning based algorithm for bound constrained quadratic programming with the rate of convergence. *submitted to Numerical Algorithms*, 2002.

[7] Z. Dostál, A. Friedlander, and S. A. Santos. Solution of coercive and semicoercive contact problems by FETI domain decomposition. *Contemporary Math.*, 218:82–93, 1998.

[8] 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.*, 2001. submitted.

[9] Z. Dostál, F. A. M. Gomes, and S. A. Santos. Duality based domain decomposition with natural coarse space for variational inequalities. *J. Comput. Appl. Math.*, 126:397–415, 2000.

[10] Z. Dostál, F. A. M. Gomes, and S. A. Santos. Solution of contact problems by FETI domain decomposition. *Computer Meth. Appl. Mech. Engrg.*, 190:1611–1627, 2000.

[11] Z. Dostál, J. Haslinger, and R. Kučera. Implementation of fixed point method for duality based solution of contact problems with friction. *Int. J.Comput. Appl. Math.*, 2000. submitted.

[12] Z. Dostál and D. Horák. Scalability and feti based algorithm for large discretized variational inequalities. *submitted to Math. and Comput. in Simulation*, 2002.

[13] Z. Dostál, V. Vondrák, and J. Rasmussen. Efficient algorithms for contact shape optimization. In V. Schulz, editor, *Workshop Fast Solution of Discretized Optimization Problems*, pages 98–106, WIAS Berlin, 2001.

[14] C. Farhat and D. Dureisseix. A numerically scalable domain decomposition method for solution of frictionless contact problems. *to appear in Int. J. Numer. Meth. Engng.*, 2002.

[15] C. Farhat, J. Mandel, and F.-X. Roux. Optimal convergence properties of the FETI domain decomposition method. *Comput. Methods Appl. Mech. Engrg.*, 115:367–388, 1994.

[16] 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.*, 13:379–396, 1992.

[17] R. Kornhuber. *Adaptive Monotone Multigrid Methods for Nonlinear Variational Problems*. Teubner, Stuttgart, 1997.

[18] J. Schöberl. Solving the signorini problem on the basis of domain decomposition techniques. *Computing*, 60:323–344, 1998.

[19] V. Vondrák, Z. Dostál, and J. Rasmussen. FETI domain decomposition algorithms for sensitivity analysis in contact shape optimization. In C.-H. Lai, P. E. Bjørstad, M. Cross, and O. B. Widlund, editors, *11th International Conference on Domain Decomposition in Science and Engineering*, pages 561–567, Bergen, 1999. Domain Decomposition Press.

[20] P. Wriggers, G. Zavarise, and T. I. Zohdi. A computational study of interfacial debonding damage in fibrous composite materials. *Comput. Materials Science*, 12:39–56, 1998.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

42. Error Estimation, Multilevel Method and Robust Extrapolation in the Numerical Solution of PDEs

M. Garbey [1], W.Shyy [2]

1. Introduction and Motivation. Richardson extrapolation (**RE**) is a simple, elegant and general mathematical idea that works for numerical quadrature with the *Romberg* method or ODE integrations that have smooth enough solution with the *Bulirsch-Stoer* method. Its use in Computational Fluid Dynamics (**CFD**) raises the following questions [3] [4] [5]:

- Does all mathematical hypotheses needed by RE are fulfilled by the numerical approximation ?

- Are the (3D) meshes fine enough to satisfy accurately the a priori convergence estimates that are only asymptotic relations in nature?

- What to do, if the order of convergence of a CFD code is space dependent and eventually solution dependent?

- Can we afford three grid levels with a coarse grid solution that has a satisfactory level of accuracy, to be used in RE?

Our objective is to use any PDE or CFD solvers, independent of their inner working algorithm and procedures, provided that they can offer the information including the residual of the numerical approximation, stability estimates, and varying grid resolutions and numerical solutions, to accomplish the following goals:

- *Automatic estimate* of the order of convergence in space,

- Using three different grid solutions (not necessarily with uniformly increasing mesh resolution), *obtain a solution with improved accuracy*

The extrapolation procedure is simple to implement and can be incorporated into any computer code without requiring detailed knowledge of the source code. Its arithmetic cost should be negligible compare to a direct computation of the fine grid solution. Finally the procedure should overall enhance the accuracy and trust of a CFD application in the context of code verification.

In this paper, we pursue the research presented in [2] as follows. We first summarize basic properties of Richardson extrapolation method and evaluate its application to CFD. Then we provide elementary approximation theory for least square extrapolation applied to grid functions. Further, we generalise this technique to PDEs, and provide some numerical results for a turning point problem. For a detailed version of this work with results on steady incompressible Navier Stokes flows, we refer to [7].

2. Basic Properties of Richardson Extrapolation and Computational Implications.

2.1. Asymptotic expansion for continuous function in a normed vector space . Let E be a normed linear space, $||\ ||$ its norm, $v \in E$, $p > 0$, and $h \in (0, h_0)$. $u^i \in E$, $i = 1..3$ have the following asymptotic expansion,

$$u^i \ = \ v + C(\frac{h}{2^{i-1}})^p + \delta,$$

with C positive constant independent of h, and $||\delta|| = o(h^p)$.

[1]Department of Computer Science, Houston TX 77204, University of Houston, USA
[2]Department of Aerospace Engineering, Mechanics and Engineering Science, Gainesville FL 32611, Univ. of Florida, USA

For known p, RE formula,

$$v_r^i = \frac{2^p\ u^{i+1} - u^i}{2^p - 1},\ i = 1, 2$$

provides improved convergence:

$$||v - v_r^i|| = o(h^p).$$

2.2. Numerical approximation for discrete functions defined on a mesh. Let E_i be a family of normed linear space, associated with a mesh $M_{h/2^{i-1}}$. We suppose a set of equations,

$$U^i\ =\ v + C_i(\frac{h}{2^{i-1}})^p + \delta_i,$$

with $C_i = (1+\epsilon_i)C$, and $\epsilon_i = o(1)$. δ_i is a model for the h independent numerical perturbation induced by consistency errors and/or arithmetic error. The Richardson extrapolate

$$V_r^2 = \frac{2^p\ U^3 - U^2}{2^p - 1},$$

has then for error in E_1,

$$v - V_r^2 = \frac{1}{2^p - 1}((\delta_2\ -\ 2^p\delta_3) + C\ (\epsilon_2 - \epsilon_3)(\frac{h}{2})^p).$$

The numerical perturbation is amplified by a factor $\frac{2^p+1}{2^p-1}$. For applications in (complex) CFD calculation, the asymptotic order of convergence is not well established and one uses:

$$p \sim \log_2 \frac{||u^1 - u^2||}{||u^2 - u^3||} \tag{2.1}$$

If one considers $\{u^i\}$ as a set of *real numbers* instead of a set of functions in $(E, ||\ ||)$, combining three ordered approximations gives the so-called Δ^2 Aitken formula,

$$v_r^2 \sim \frac{u^1u^3 - (u^2)^2}{u^1 - 2u^2 + u^3}.$$

But this formula has generally no rigorous basis in the corresponding space of approximation.

From the numerical point of view,

$$p = \log_2 ||(1 - \gamma(p))\ \frac{U^1 - U^2 - (\delta_1 - \delta_2)}{U^2 - U^3 - (\delta_2 - \delta_3)}||,$$

where $\gamma(p) \sim \kappa(2^p\epsilon_1 - (2^p + 1)\epsilon_2 + \epsilon_3)$, and $\kappa = (2^p - 1)^{-1}$.

In practice,

$$p \approx \log_2 ||\frac{U^1 - U^2}{U^2 - U^3}||,\ in\ (E_1, ||\ ||)$$

The second order error term ϵ_2 on u_2 (respt ϵ_1 on u_1) has therefore $2^p + 1$ (respt 2^p) more impact on p calculation error than the second order error term ϵ_3 on u_3. Further, the "pointwise" extrapolation

$$v_r^2 = \frac{U^1U^3 - (U^2)^2}{U^1 - 2U^2 + U^3},\ \forall x \in M_1$$

that is routinely used in CFD is very sensitive to numerical perturbation.

The convergence order approximation and RE presented so far is a common tool for solution quality assessement in CFD. In our experience [5] [7], we have observed, for example, that for two different codes for the steady state, 2-D laminar incompressible lid-driven square cavity flow with the Reynolds number (**Re**) in the range of 20 to 1000 and squared regular

meshes using either $\omega - \psi$ formulation and FD approximation or $v - p$ formulation and FV approximation with centered cells, RE can improve the order of accuracy, but not consistently. If the quality of the grid solution is poor then RE may provide worse approximations.

Further, the theoretical bases of classical RE formula presented here, are not satisfied when the convergence order of the solution process is space dependent and solution dependent, which is rather common in CFD. We propose in this paper a method that seems to be more robust than RE, and that further can be used as a framework for aposteri estimates.

3. Least Square Extrapolation for Numerical Functions. Let $E = L_2(0,1)$, $u \in E$. Let v_h^1 and v_h^2 be two approximations of u in E:

$$v_h^1,\ v_h^2 \to u \ in\ E\ as\ h \to 0.$$

A consistent linear extrapolation formula formally written

$$\alpha v_h^1 + (1-\alpha) v_h^2 = u.$$

In p order RE the α function is a constant. We adopt here a more general point of view than RE. We formulate the following problem as a Least Square Extrapolation (**LSE**):

P_α: Find $\alpha \in \Lambda(0,1) \subset L_\infty$ such that $\alpha v_h^1 + (1-\alpha) v_h^2 - u$ is minimum in $L_2(0,1)$.

If $1/(v_h^1 - v_h^2)$ is in $L_\infty(0,1)$, we get an explicit solution for this problem. If $v_h^1 - v_h^2$ vanishes, we can approximate then v_h^i by a w_h^i function in $L_2(0,1)$ such that

$$w_h^i - u_h^i = O(h^q), q >> p,\ and\ 1/(w_h^1 - w_h^2) \in L_\infty(0,1).$$

where p is the expected order of convergence of v_h as $h \to 0$. We get then

$$\alpha = \frac{u - w_h^2}{w_h^1 - w_h^2},\ and\ \alpha \in L_2(0,1).$$

We have easily

Lemma 1: If $\alpha_M - \alpha = 0(M^{-1})$ as $M \to \infty$ and $v_h^1 - v_h^2 = 0(h^p)$ then

$$u \ = \ \alpha v_h^1 + (1-\alpha) v_h^2 + O(h^p) \ \times \ O(M^{-1}).$$

In the present work, we set $\Lambda(0,1)$ to be the space of α functions

$$\alpha = \alpha^0 + \alpha^1 \cos(x\pi) + \Sigma_{j=1..M} \alpha^j \sin((j-1)x\pi).$$

with $\alpha^j,\ j = 0..M$ reals. We can then show using [1]

Lemma 2: Let α be in $L_2(0,1)$. Let $x_j = \frac{j}{N}$ be a regular discretization of $(0,1)$. Let M be an integer such that $M << N$. There is a unique trigonometric polynomial

$$\alpha_M = \alpha^0 + \alpha^1 \cos(x\pi) + \Sigma_{j=1..M} \alpha^j \sin((j-1)x\pi)$$

that minimizes the discrete L_2 norm

$$\Sigma_{j=0..N} (\alpha(x_j) - \alpha_M(x_j))^2.$$

α_M converges to α in $L_2(0,1)$ as $M \to \infty$ while the ratio $\frac{M}{N}$ stays constant and less than one. If $\alpha \in C^1(0,1)$, the convergence $\alpha_M \to v$ is pointwise and of order M^{-1} in (0,1) and M^{-2} away from the end points.

We have now a solution to the approximation problem P_α or its modified analog if we have possibly to modify locally the v_h^i function at neighborhood of points where $v_h^1 - v_h^2$ cancels.

From Lemma 1 and Lemma 2, we have

Theorem: if $u,\ v_h^i, \in\ C^1(0,1), i=1,\ 2$, if $\frac{1}{v_h^1-v_h^2}\ \in\ L_\infty(0,1)$ and $v_h^2-v_h^1=0(h^p)$ then $\alpha v_h^1+(1-\alpha)v_h^2$ is an $0(M^{-1})\times 0(h^p)$ approximation of u.

Special care must be done if $v_h^1-v_h^2 << u-v_h^2$, in some set of non zero measure Ω_S.

These outliers points should not affect globally the least square extrapolation as long as we impose that α be a bounded function independently of h. Further, a more robust approximation procedure consists to use three levels of grid solution as follows:

$P_{\alpha,\beta}$: *Find* $\alpha,\beta\in\Lambda(0,1)$ *such that* $\alpha v_h^1+\beta v_h^2+(1-\alpha-\beta)v_h^3-u$ *is minimum in* $L_2(0,1)$.

Existence of the solution (α,β) is established if one can partition $(0,1)$ into two overlapping subset $\Omega_1\bigcup\Omega_2=(0,1)$ of nonzero measure intersection, such that $1/(v_h^1-v_h^3)$ is in $L_\infty(\Omega_1)$ and $1/(v_h^2-v_h^3)$ is in $L_\infty(\Omega_2)$. But uniqueness is no longer guaranteed. We can use a Singular Value Decomposition method (**SVD**) then, to account for the fact that the linear system can be both over determined and under determined. But SVD requires many more arithmetic operations than a direct solve of the normal set of equations when $M << N$. In practice, if $v_h^1-v_h^3 << u-v_h^3$ and $v_h^2-v_h^3 << u-v_h^3$ in some set of non zero measure then there is no local convergence of our sequence of functions. We want to make sure that these outlier points do not affect the quality of the least square solution at points where convergence is achieved.

In practice, we work with *grid functions* solution of discretized PDE problem. In contrast to classical RE, where all grid solutions are projected onto a common coarse grid, our solution procedure consists of interpolating all data on a very fine grid denoted M^0 via a high order interpolant $\tilde{U}_i=I_i[U_i]$. We want then to get our best fitted extrapolation formula on the fine grid itself as follows.

P_α: *Find* $\alpha\in\Lambda(0,1)\ \subset\ L_\infty$ *such that* $\alpha\tilde{U}^1+(1-\alpha)\tilde{U}^2-U$ *is minimum in* $L_2(M^0)$.

The three-level extrapolation problem is analogous.

We have checked the numerical accuracy and sensitivity to perturbation of LSE on numerical function examples that possess different type of asymptotic behavior and different degree of smoothness. In all cases our least square extrapolation method seems to give improved accuracy and robustness. In particular our least square extrapolation is definitively an improvement on fixed order RE when the solution has a hybrid order of convergence that is first order in some subset of the domain and second order elsewhere.

The extension to multidimensional problem with rectangular grid that are tensorial products of one-dimensional regular grids is straightforward. The generalization to body fitted meshes generated by PDEs [6], is easy since the Fourier expansion technique is insensitive to change of variables as long as they are smooth transformations. However generalisation to FE approximation with unstructured grid will require obviously a different space of approximation for the weight functions α and β.

4. Least Square Extrapolation for PDEs and Computational Algorithm.

The idea is now to use the PDE in the RE process to find an improved solution on the fine grid.

Let us denote formally the linear PDE

$$L[u]=f,\ \textit{with}\ u\in(E_a,||\ ||_a)\ \textit{and}\ f\in(E_b,||\ ||_b),$$

and its numerical approximation,

$$L_h[U]=f_h,\ \ \textit{with}\ U\in(E_a^h,||\ ||_a)\ \textit{and}\ f_h\in(E_b^h,||\ ||_b),$$

parameterized by a mesh step h.

We suppose that we have a priori a stability estimate for these norms

$$||U||_a \ \leq \ C\ h^s\ (||f_h||_b), \tag{4.1}$$

with s real not necessarily positive. We will look for consistant extrapolation formula that minimize the residual.

Let us restrict for simplicity to a two-point boundary value problems in $(0,1)$. Our least square extrapolation is now defined as follows:

P_α: Find $\alpha \in \Lambda(0,1) \ \subset \ L_\infty$ such that $\alpha L_h[\tilde{U}^1] + (1-\alpha)L_h[\tilde{U}^2] - f_h$ is minimum in $L_2(M^0)$.

The three levels version is analogue. To focus on the practical use of this method, we should make the following observations. It is essential that the interpolation operator gives a smooth interpolant depending on the order of the differential operator. For conservation laws, one may require that the interpolation operator satisfies the same conservation properties. For chemical problems, one may require that the interpolant preserves the positivity of species. For elliptic problems, it is convenient to postprocess the interpolated functions $\tilde{U}^i$, by few steps of the relaxation scheme

$$\frac{V^{k+1} - Vk}{\delta t} \ = \ L_h[V^k] - f_h,\ V^0 = \tilde{U}^i,$$

with appropriate artificial time step δt. This will readily smooth out the interpolant.

Let $G_i,\ i = 1..3$, be three embedded grids that does not necessary match and their companion grid solutions U_i. Let M^0 be a regular grid that is finer than the grids G_i. The solution process of P_α and/or $P_{(\alpha,\beta)}$ can be decomposed into three consecutive steps.

- First, interpolation from $G_i, i = 1..3$ to M^0. We choose interpolation tools that have a number of arithmetic operations proportional to $Card(M^0)$, i.e. the number of grid points of M^0.
- Second, the evaluation of the residual on the fine grid M^0, that has the same asymptotic order of arithmetic operations.
- Third the solution of the linear least squares problem with M unknowns.

If we keep M of the same order as $Card(M^0)^{1/3}$, and use a standard direct solver for symmetric system to solve the normal set of equations, the arithmetic complexity of the overall procedure is still of order $Card(M^0)$, i.e., it is linear.

The application to nonlinear PDE problem is done via a Newton-like loop [7]. The algorithm is coded in an independent program from the main code application.

We choose a Fourier expansion for each weight function α and β, that has M terms with $M \approx Card(M^0)^{1/3}$, to keep a linear cost for the complete procedure when the direct solution of the normal set of equations is giving a good result. An SVD, if needed, will lead however to more intense computation.

Let us now illustrate the numerical efficiency and robustness of our method with a 2D Turning Point Problem:

$$\epsilon\ \Delta u + a(x,y)\frac{\partial u}{\partial x} = 0,\ x \in (0,\pi)^2,$$

with Dirichlet boundary conditions of opposite signs at $x = 0$ and $x = \pi$, and homogeneous Neumann at $y = 0/\pi$. We take

$$a(x,y) = x - (\frac{\pi}{2} + 0.3(y - \frac{\pi}{2})).$$

We have then a transition layer (**TL**) of ϵ order thickness centered on the curve $a(x, y) = 0$, which is not parallel to the x or y axis.

The application code uses second order central FD of the diffusion term and first order upwinding for the convection term with either direct sparse LU linear or GMRES solver. There are no spurious oscillations because of the discrete maximum principle.

Figures 4.1 and 5.2 report on the accuracy of the two-level and three-level least squares extrapolation versus RE assuming either first or second order convergence. The errors are given in L_∞ norm. The curve with hexagram signs gives an accurate estimation of the discrete solution error between the exact grid solution on the fine grid M^0 of size $N \times N$, versus the exact continuous solution of the turning point problem. Let G_I be square grids of size $N_i \times N_i$.

The number of Fourier modes in the approximation of the weight α, β is 4 in each space direction. We observe that for both cases $\epsilon = 0.1$ and $\epsilon = 0.01$ in Figure 4.1 and 5.2, R1 gives better results than R2. This is an indication of the fact that the transition layer is not under-resolved. We observe in Figure 4.1 with $\epsilon = 0.1$, and modest base grid sizes, namely, $N_1 = 17,\ N_2 = 23,\ N_3 = 29$, meaning that we have on average only one or two grid points in the transition layer for the G3 solution, our least squares is as accurate as the fine grid solution. This is still true when the Richardson extrapolation fails for $N \geq 70$. The least squares extrapolation also gives satisfactory results in Figure 5.2, where $\epsilon = 0.01,\ N_1 = 39,\ N_2 = 49,\ N_3 = 59$, but R1 predicts the grid solution on M^0 with an error less than or equal to the error with the exact continuous solution for $N \leq 110$. In all cases LS2 is more accurate than LS1, especially for large N values. In these experiments, LS1 and LS2 predict the fine grid solution with an error less than the fine grid approximation of the exact solution for N as large as 150: we gain therefore more than one order of convergence. Similar results on the lid-driven cavity flow confirm the capabilities of our method [7].

5. Conclusions and Discussions. We have studied a new extrapolation method for PDEs that is more robust and accurate than RE applied to numerical solutions with inexact or varying convergence order. Our method provides a better tool to establish a posteriori estimate than Richardson extrapolation when the convergence order of a CFD code is space dependent. However there are still many open questions. To cite some but a few, we still need to establish a criterion to relax the constraint on the accuracy of the coarse grid data for efficient least squares extrapolation. Further, from the application point of view, it might be interesting to test the robustness of our least squares extrapolation method to elliptic problems with general geometry domains via fictitious domain technique.

REFERENCES

[1] D. Gottlieb and C.W.Shu, *On the Gibbs Phenomenon and its Resolution*, SIAM Review, Vol 39, No4,pp 644-668, 1997.

[2] M. Garbey, *Some Remarks on Multilevel Method, Extrapolation and Code Verification* Proceedings of the 13th international conference on Domain Decomposition, Lyon Oct. 2000, pp379-386, CIMNE Barcelona, 2002.

[3] A.G.Hutton and M.V.Casey, *Quality and Trust in Industrial CFD - A European Initiative*, 39th AIAA Aerospace Sciences Meeting, 8-11 January, 2001/ Reno, NV, AIAA Paper 2001-0656.

[4] W.L. Oberkampf, F.G. Blottner and D.Aeshliman, Methodology for Computational Fluid Dynamics Code Verification and Validation, 26th AIAA Fluide Dynamic Conference, June 19-22, 1995/ San Diego, CA, AIAA Paper 95-2226.

[5] W. Shyy, M. Garbey, A. Appukuttan and J. Wu, *Evaluation of Richardson Extrapolation in Computational Fluid Dynamics* Numerical Heat Transfer, Part B: Fundamentals 41 (2): 139-164, 2002.

[6] J.F.Thompson, Z.V.A.Warsi and C.W. Mastin, *Numerical Grid Generation: foundation and applications,* North Holland, New York, 1985.

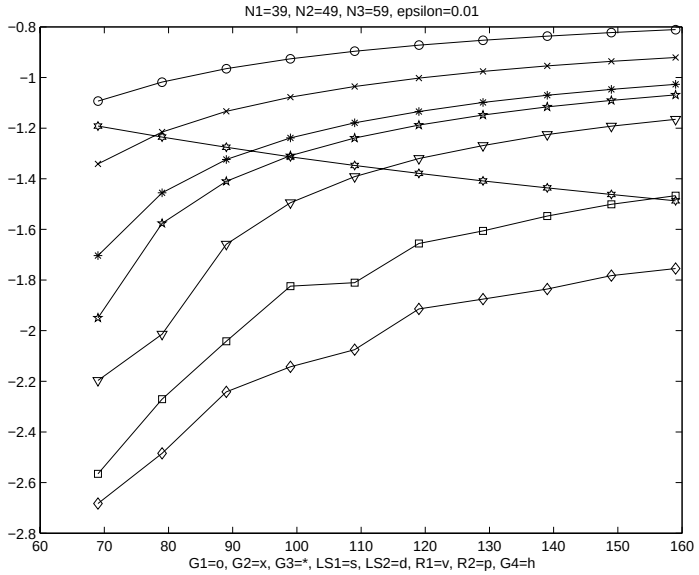


Figure 5.1: x axis is for the number of grid points N in each space direction for the fine grid M^0. y axis gives in log_{10} scale the errors in maximum norm. Labels of curves are as follows: 'o' for G_1 solution, 'x' for G_2 solution, '*' for G_3 solution, $\star$ for R2, 'v' for R1, *square* for LS1, $\diamond$ for LS2.

[7] M.Garbey and W.Shyy, *A Least Squares Extrapolation Method for PDEs,* to appear in JCP 2003.

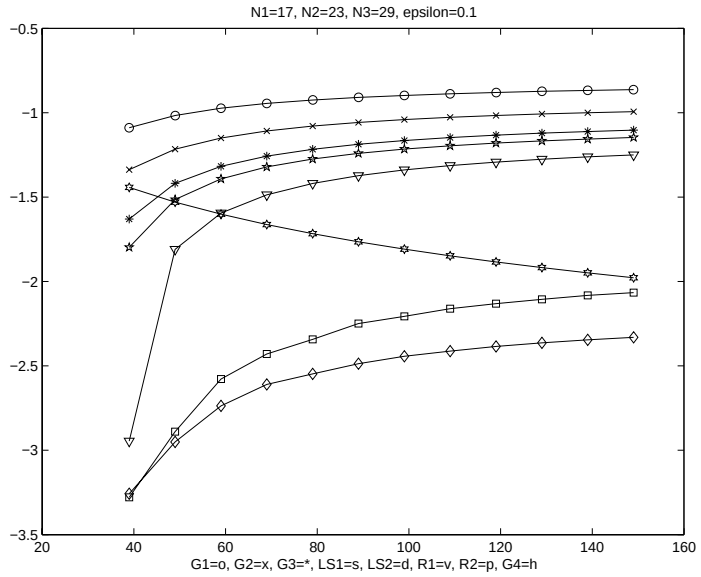


Figure 5.2: Same labels as in Figure 5.1.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

43. A Robin-Robin preconditioner for strongly heterogeneous advection-diffusion problems

L. Gerardo Giorda[1], P. Le Tallec[2], F. Nataf[3]

1. Introduction. We consider an advection-diffusion problem with discontinuous viscosity coefficients. We apply a substructuring technique and we extend to the resulting Schur complement the Robin-Robin preconditioner used for problems with constant viscosity. In Section 2 the algorithm is analyzed theorically by means of Fourier techniques, and we show that its convergence rate is independent of the coefficients: this allows to treat large discontinuities. Section 3 is dedicated to the variational generalization to an arbitrary number of subdomains, while in Section 4 we give some numerical result in 3D.

1.1. Statement of the problem. Let Ω be bounded domain in $\mathbf{R}^2$. We consider the following general advection-diffusion problem

$$\begin{array}{rcll} -\mathrm{div}\,(\nu(x)\nabla u) + \vec{b}\cdot\nabla(u) + au & = & f & \text{in } \Omega \\ u & = & 0 & \text{on } \partial\Omega_D \end{array} \tag{1.1}$$

where $\vec{b}$ is the convective field $\vec{b} = (b_x, b_y)$ while the constant a may arise from an Euler implicit time discretization for the time dependent problem, and represent the inverse of the time step, i.e. $a = 1/\Delta t$.
We assume the function $\nu(x)$ to be piecewise constant

$$\nu(x) = \begin{cases} \nu_1 & \text{if } x \in \Omega_1 \\ \nu_2 & \text{if } x \in \Omega_2 \end{cases}$$

with $\nu_1 < \nu_2$, where Ω_1 and Ω_2 are two non overlapping subsets which cover Ω $\Omega_1 \cup \Omega_2 = \Omega$. Γ denotes the interface between the two subdomains, i.e. $\Gamma = \overline{\Omega}_1 \cap \overline{\Omega}_2$, while $\mathcal{L}_j$ $(j = 1, 2)$ denotes the operator

$$\mathcal{L}_j(w) := -\nu_j \Delta w + \vec{b}\cdot\nabla w + aw$$

2. The Continuous Algorithm. We introduce, at the continuous level, the operator

$$\begin{array}{rcl} \Sigma : H^{1/2}_{00}(\Gamma) \times L^2(\Omega) & \longrightarrow & H^{-1/2}(\Gamma) \\ (u_\Gamma, f) & \longmapsto & \left(\nu_1 \frac{\partial u_1}{\partial n_1} + \nu_2 \frac{\partial u_2}{\partial n_2}\right)_\Gamma \end{array} \tag{2.1}$$

where u_j $(j = 1, 2)$ is the solution to problem

$$\begin{array}{rcll} \mathcal{L}_j(u_j) & = & f & \text{in } \Omega_j \\ u_j & = & 0 & \text{on } \partial\Omega_D \cap \partial\Omega_j \\ u_j & = & u_\Gamma & \text{on } \Gamma \end{array} \tag{2.2}$$

It is evident that u_Γ satisfies the Steklov-Poincaré equation

$$\mathcal{S}(u_\Gamma) = \chi \tag{2.3}$$

where $\mathcal{S}(.) := \Sigma(., 0)$ and $\chi := -\Sigma(0, f)$. We split the operator $\mathcal{S} = \mathcal{S}_1 + \mathcal{S}_2$, with

[1]Dipartimento di Matematica, Università di Trento, gerardo@science.unitn.it
[2]École Polytechnique, patrick.letallec@polytechnique.fr
[3]CMAP - CNRS, UMR 7641, École Polytechnique, nataf@cmapx.polytechnique.fr

$$\mathcal{S}_j : u_\Gamma \mapsto \left(\nu_j \frac{\partial u_j}{\partial n_j} - \frac{\vec{b}\cdot\vec{n}_j}{2} u_j \right)_\Gamma ,$$

for $j = 1, 2$ (since $\vec{n}_1 = -\vec{n}_2$ and $u_1 = u_2 = u_\Gamma$, the terms $\frac{1}{2}\vec{b}\cdot\vec{n}_j\, u_j$ cancel by summation). Following ([1]), ([2]) and ([8]), we propose as a preconditioner for the Steklov-Poincaré equation at the continuous level a weighted sum of the inverses of the operators $\mathcal{S}_1$ and $\mathcal{S}_2$,

$$\mathcal{T} = N_1 \mathcal{S}_1^{-1} N_1 + N_2 \mathcal{S}_2^{-1} N_2, \tag{2.4}$$

with $N_1 = \frac{\nu_1}{\nu_1+\nu_2}$, $N_2 = \frac{\nu_2}{\nu_1+\nu_2}$, which is defined by

$$\begin{array}{rcl} \mathcal{T} : H^{-1/2}(\Gamma) & \longrightarrow & H^{1/2}_{00}(\Gamma) \\ g & \longmapsto & (N_1\, v_1 + N_2\, v_2)_\Gamma \end{array} \tag{2.5}$$

where v_j $(j = 1, 2)$ is the solution to

$$\begin{array}{rcll} \mathcal{L}_j(v_j) & = & 0 & \text{in } \Omega_j \\ v_j & = & 0 & \text{on } \partial\Omega_D \cap \partial\Omega_j \\ \left(\nu_j \frac{\partial v_j}{\partial n_j} - \frac{\vec{b}\cdot\vec{n}_j}{2} v_j\right)_\Gamma & = & N_j\, g & \text{on } \Gamma. \end{array} \tag{2.6}$$

2.1. The vertical strip case - Uniform velocity. In this section we consider the case where $\Omega = \mathbf{R}^2$ is decomposed into the left ($\Omega_1 =]-\infty, 0[\times \mathbf{R}$) and right ($\Omega_2 =]0, +\infty[\times \mathbf{R}$) half-planes, we assume the convective field to be uniform $\vec{b} = (b_x, b_y)$, with the additional requirement on the solutions u_j to be bounded as $|x| \to +\infty$. We express the action of the operator $\mathcal{S}$ in terms of its Fourier transform in the y direction as

$$\mathcal{S} u_\Gamma = \mathcal{F}^{-1}\left(\hat{\mathcal{S}}(\xi) \hat{u}_\Gamma(\xi) \right), \quad u_\Gamma \in H^{1/2}_{00}(\Gamma)$$

where ξ is the Fourier variable and $\mathcal{F}^{-1}$ denotes the inverse Fourier transform. We consider, for $j = 1, 2$, the problem

$$\begin{array}{rcll} \mathcal{L}_j(u_j) & = & 0 & \text{in } \Omega_j \\ u_j & = & u_\Gamma & \text{on } \Gamma, \end{array} \tag{2.7}$$

and we have to compute $\hat{\mathcal{S}}\hat{u}_\Gamma$. Performing a Fourier transform in the y direction on the operators $\mathcal{L}_j$, we get

$$\left(a + b_x \partial_x - \nu_j \partial_{xx} + i b_y \xi + \nu_j \xi^2 \right) \hat{u}_j(x, \xi) = 0, \tag{2.8}$$

for $j = 1, 2$, where $i^2 = -1$. For a given ξ, equation (2.8) is an ordinary differential equation in x whose solutions have the form $\alpha_j(\xi) \exp\{\lambda_j^-(\xi) x\} + \beta_j(\xi) \exp\{\lambda_j^+(\xi) x\}$, where

$$\lambda_j^\pm(\xi) = \frac{b_x \pm \sqrt{b_x^2 + 4a\nu_j + 4\nu_j^2\xi^2 + 4ib_y\nu_j\xi}}{2\nu_j}, \tag{2.9}$$

with $\mathrm{Re}(\lambda_j^\pm) \gtrless 0$, as $\mathrm{Re}(z)$ indicates the real part of a complex number z. The solutions u_j $(j = 1, 2)$ must be bounded at infinity, so $\alpha_1(\xi) = \beta_2(\xi) = 0$, while the Dirichlet condition on the interface provides $\beta_1(\xi) = \alpha_2(\xi) = \hat{u}_\Gamma$. Hence,

$$\hat{\mathcal{S}}\hat{u}_\Gamma = \frac{1}{2}\left(\sqrt{b_x^2 + 4a\nu_1 + 4\nu_1^2\xi^2 + 4ib_y\nu_1\xi} + \sqrt{b_x^2 + 4a\nu_2 + 4\nu_2^2\xi^2 + 4ib_y\nu_2\xi} \right) \hat{u}_\Gamma \tag{2.10}$$

In a similar way we compute $\hat{\mathcal{T}}\hat{g}$ for $g \in H^{-1/2}(\mathbf{R})$, and we have $(\hat{\mathcal{T}} \circ \hat{\mathcal{S}})\hat{u}_\Gamma = \Phi(\xi)\hat{u}_\Gamma$, with

$$\Phi(\xi) = N_1^2 \cdot [1 + z(\xi)] + N_2^2 \cdot \left[1 + \frac{\bar{z}(\xi)}{|z(\xi)|^2} \right], \tag{2.11}$$

where we have set $z(\xi) := \sqrt{\frac{b_x^2+4a\nu_2+4\nu_2^2\xi^2+4ib_y\nu_2\xi}{b_x^2+4a\nu_1+4\nu_1^2\xi^2+4ib_y\nu_1\xi}}$. We have $1 < |z(\xi)| \leq \nu_2/\nu_1$, with $|z(\xi)|$ decreasing in $(-\infty, 0)$ and increasing in $(0, +\infty)$.

Theorem 2.1 (Main Result) *In the case where the plane* $\mathbf{R}^2$ *is decomposed into the left and right half planes, and the convective field is uniform, the reduction factor for the associated* GMRES *algorithm can be bounded from above by a constant independent of the time step* Δt*, the convective field* $\vec{b}$ *and the viscosity coefficients* ν_1 *and* ν_2*.*

Proof. Let $\Phi(\xi)$ be the function defined in (2.11). The GMRES reduction factor is given, for a positive real matrix A with symmetric part M, by

$$\rho_{\text{GMRES}} = 1 - \frac{(\lambda_{\min}(M))^2}{\lambda_{\max}(A^T A)}.$$

Therefore, it is enough to show that

$$\frac{\max_\xi |\Phi(\xi)|^2}{(\min_\xi \operatorname{Re}\Phi(\xi))^2} \in O(1) \tag{2.12}$$

independentely of a, b_x, b_y, ν_1 and ν_2.
If $b_y \neq 0$, since $\operatorname{Re} z(\xi) \geq 0$, we have from (2.11)

$$\operatorname{Re}\Phi(\xi) \geq N_1^2 + N_2^2 > \frac{\nu_2^2}{(\nu_1+\nu_2)^2}, \tag{2.13}$$

for all ξ, as well as, focusing on $|\Phi(\xi)|^2$,

$$|\Phi(\xi)|^2 \leq \left[N_1^2 + N_2^2 + N_1^2 \cdot |z(\xi)| + \frac{N_2^2}{|z(\xi)|}\right]^2 + \left[N_1^2 \cdot |z(\xi)| - \frac{N_2^2}{|z(\xi)|}\right]^2 = \Psi(\xi) \tag{2.14}$$

which is increasing in $(-\infty, 0)$ and decreasing in $(0, +\infty)$.
i) If $b_x \neq 0$, we define $\eta := 4a/b_x^2$ and we have

$$\Psi(0) = \left[N_1^2\left(1+\sqrt{\frac{1+\eta\nu_2}{1+\eta\nu_1}}\right) + N_2^2\left(1+\sqrt{\frac{1+\eta\nu_1}{1+\eta\nu_2}}\right)\right]^2 + \left[N_1^2\sqrt{\frac{1+\eta\nu_2}{1+\eta\nu_1}} - N_2^2\sqrt{\frac{1+\eta\nu_1}{1+\eta\nu_2}}\right]^2$$

The right hand term is decreasing as a function of η. This provides

$$\max_\xi |\Phi(\xi)|^2 \leq (2N_1^2 + 2N_2^2)^2 + (N_1^2 - N_2^2)^2 \tag{2.15}$$

From (2.13) and (2.15), we get

$$\frac{\max_\xi |\Phi(\xi)|^2}{(\min_\xi \operatorname{Re}\Phi(\xi))^2} \leq 5 + 6\cdot\left(\frac{\nu_1}{\nu_2}\right)^2 + 5\cdot\left(\frac{\nu_1}{\nu_2}\right)^4 < 16. \tag{2.16}$$

ii) If $b_x = 0$ (flux parallel to the interface), $|z(0)| = \sqrt{\nu_2/\nu_1}$, and we have

$$\max_\xi |\Phi(\xi)|^2 \leq \left[N_1^2\left(1+\sqrt{\frac{\nu_2}{\nu_1}}\right) + N_2^2\left(1+\sqrt{\frac{\nu_1}{\nu_2}}\right)\right]^2 + \left[N_1^2\sqrt{\frac{\nu_2}{\nu_1}} - N_2^2\sqrt{\frac{\nu_1}{\nu_2}}\right]^2 \tag{2.17}$$

From (2.13) and (2.17), we get

$$\frac{\max_\xi |\Phi(\xi)|^2}{(\min_\xi \operatorname{Re}\Phi(\xi))^2} \leq 1 + 2\sum_{n=1}^{7}\left(\frac{\nu_1}{\nu_2}\right)^{n/2} + \left(\frac{\nu_1}{\nu_2}\right)^4 < 16. \tag{2.18}$$

If $b_y = 0$, the complex function $\Phi(\xi)$ reduces to a real one which is symmetric in ξ, decreasing in $[0, +\infty)$ and satisfies $\Phi(\xi) \geq 1$ for all ξ. Therefore

$$\frac{\max_\xi |\Phi(\xi)|^2}{(\min_\xi \operatorname{Re}\Phi(\xi))^2} = \left[\frac{\max_\xi \Phi(\xi)}{\min_\xi \Phi(\xi)}\right]^2 \leq \left[\max_\xi \Phi(\xi)\right]^2 = [\Phi(0)]^2 .$$

i) if $b_x \neq 0$, we define $\eta := 4a/b_x^2$, and we have

$$[\Phi(0)]^2 = \left[N_1^2\left(1+\sqrt{\frac{1+\eta\nu_2}{1+\eta\nu_1}}\right) + N_2^2\left(1+\sqrt{\frac{1+\eta\nu_1}{1+\eta\nu_2}}\right)\right]^2 \tag{2.19}$$

where the right hand side attains its maximum for $\eta = 0$. Hence

$$\left[\frac{\max_\xi \Phi(\xi)}{\min_\xi \Phi(\xi)}\right]^2 < \left[2\cdot\frac{\nu_1^2}{(\nu_1+\nu_2)^2} + 2\cdot\frac{\nu_2^2}{(\nu_1+\nu_2)^2}\right]^2 < 4. \tag{2.20}$$

ii) if $b_x = 0$ (purely elliptic case) we simply have

$$\left[\frac{\max_\xi \Phi(\xi)}{\min_\xi \Phi(\xi)}\right]^2 < \left[N_1^2\left(1+\sqrt{\frac{\nu_2}{\nu_1}}\right) + N_2^2\left(1+\sqrt{\frac{\nu_1}{\nu_2}}\right)\right]^2 < 4. \tag{2.21}$$

□

Remark 2.1 The argument above is based only on the assumption $\nu_1 < \nu_2$, and it can be easily seen that a symmetric argument would give the same result as long as $\nu_2 < \nu_1$. In a forthcoming paper ([6]), a more detailed proof of the main result will be given. More, it appears that the condition number of the preconditioned system improves with the growth of the ratio ν_2/ν_1.

3. Variational Generalization. We consider in $\mathbf{R}^d$ (with $d = 2, 3$) the domain $\Omega = \bigcup_{k=1}^N \Omega_k$, with $\Omega_j \cap \Omega_k = \emptyset$ for $j \neq k$, in which we solve

$$\begin{array}{rcll} -\operatorname{div}(\nu(x)\nabla u) + \vec{b}(x)\cdot\nabla(u) + a(x)u &=& f & \text{in } \Omega \\ u &=& 0 & \text{on } \partial\Omega_D \end{array} \tag{3.1}$$

with piecewise constant viscosity $\nu(x) = \nu_k$ in $\Omega_k(x)$. We restrict ourselves to well-posed problems, and we assume $\vec{b} \in W^{1,\infty}(\Omega)$ and there exists $\mu > 0$ such that $a - 1/2\operatorname{div}(\vec{b}) \geq \mu > 0$. We introduce the space $\mathbb{H}(\Omega) = \left\{v \in H^1(\Omega) : v_{|\partial\Omega_D} = 0\right\}$, and the variational form of (3.1)

$$\text{Find } u \in \mathbb{H}(\Omega): \qquad a(u,v) = L(v) \quad \forall v \in \mathbb{H}(\Omega), \tag{3.2}$$

with

$$a(u,v) = \int_\Omega \nu\nabla u\nabla v + (\vec{b}\cdot\nabla u)v + auv, \qquad L(v) = \int_\Omega fv.$$

We define the local interfaces $\Gamma_k := \partial\Omega_k \setminus \partial\Omega$ and the global interface $\Gamma = \cup_k \Gamma_k$, and we introduce the local form

$$a_k(u,v) = \int_{\Omega_k} \left\{\nu_k\nabla u\nabla v + (\vec{b}\cdot\nabla u)v + auv\right\} - \int_{\Gamma_k} \frac{1}{2}\,\vec{b}\cdot\vec{n}_k uv$$

where the interface terms $-\int_{\Gamma_k} 1/2\,\vec{b}\cdot\vec{n}_k uv$ added locally cancel each other by summation, but their presence guarantees nevertheless that the local bilinear form is positive on the space of restrictions $\mathbb{H}(\Omega_k) = \left\{v_k = v_{|\Omega_k},\ v \in \mathbb{H}(\Omega)\right\}$. Summing up on k, and letting $L_k(v) := \int_{\Omega_k} fv$, the variational problem (3.2) is equivalent to

$$\text{Find } u \in \mathbb{H}(\Omega): \qquad \sum_{k=1}^n \{a_k(u,v) - L_k(v)\} = 0 \quad \forall v \in \mathbb{H}(\Omega). \tag{3.3}$$

3.1. Finite Element Approximation. In order to approximate problem (3.3) with finite elements, we assume that the domain Ω is polygonal, and that the triangulations respect the geometry of subdomain decomposition: the interfaces Γ_k will coincide with interelement boundaries, and each subdomain can be obtained as the union of a given subset of elements in the original triangulation.
In several cases of practical interest, problem (3.1) is advection-dominated and must be stabilized. We will use *Galerkin Least-Squares* techniques (*GALS*), which consists in adding to the original variational formulation the element residuals

$$\int_{T_i} \delta_i(h) \left(-\mathrm{div}\,(\nu\nabla u) + \vec{b}\cdot\nabla(u) + au - f\right)\left(-\mathrm{div}\,(\nu\nabla v) + \vec{b}\cdot\nabla(v) + av\right)$$

where T_i is an element of the triangulation, with a suitable choice of the local positive stabilization parameter $\delta_i(h)$. The stabilized finite elements formulation then reads

$$\text{Find } u_h \in \mathbb{H}_h(\Omega): \qquad \sum_{k=1}^{n} \{a_{kh}(u_h, v_h) - L_{kh}(v_h)\} = 0 \quad \forall v_h \in \mathbb{H}_h(\Omega), \tag{3.4}$$

3.2. Substructuring. The variational structure of problems (3.3) and (3.4) allows to reduce them to an interface problem by means of standard substructuring techniques. Following ([2]), we introduce the space $\mathbb{H}^0(\Omega_k) = \left\{v_k \in \mathbb{H}(\Omega), v_k = 0 \text{ in } \overline{\Omega \setminus \Omega_k}\right\}$, the global and local trace spaces $\mathbb{V}$ and $\mathbb{V}_k$, the restriction operators $R_k : \mathbb{H}(\Omega) \to \mathbb{H}(\Omega_k)$ and $\bar{R}_k : \mathbb{V} \to \mathbb{V}_k$, the a_k-harmonic extension $\mathrm{Tr}_k^{-1} : \mathbb{V}_k \to \mathbb{H}(\Omega_k)$, defined as

$$a_k(\mathrm{Tr}_k^{-1}\bar{u}_k, v_k) = 0 \quad \forall v_k \in \mathbb{H}^0(\Omega_k), \qquad \mathrm{Tr}(\mathrm{Tr}_k^{-1}\bar{u}_k)_{|\Gamma_k} = \bar{u}_k, \tag{3.5}$$

with its adjoint Tr_k^{-*}. The bilinear form a_k is elliptic on $\mathbb{H}^0(\Omega_k)$ so problem (3.5) is well-posed, and we can define the local Schur complement operator $S_k : \mathbb{V}_k \to \mathbb{V}'_k$ as

$$\langle S_k\bar{u}_k, \bar{v}_k\rangle = a_k(\mathrm{Tr}_k^{-1}\bar{u}_k, \mathrm{Tr}_k^{-*}\bar{v}_k) \qquad \forall \bar{u}_k, \bar{v}_k \in \mathbb{V}_k$$

If we decompose the local degrees of freedom U_k of $u_k = R_k u$ into internal (U_k^0) and interface ($\bar{U}_k$) degrees of freedom, the matrix A_k associated to the bilinear form a_k can be represented as

$$A_k = \begin{bmatrix} A_k^0 & B_k \\ \tilde{B}_k^T & \bar{A}_k \end{bmatrix},$$

and we eliminate the local internal component U_k^0 as solution of a well-posed local problem, to get

$$S_k\bar{U}_k = \left(\bar{A}_k - \tilde{B}_k^T(A_k^0)^{-1}B_k\right)\bar{U}_k.$$

The global Schur complement operator

$$S = \sum_{k=1}^{N} \bar{R}_k^T S_k \bar{R}_k \tag{3.6}$$

follows and we reduce problems (3.3) and (3.4) to the interface problem $S\bar{u} = F$ in $\mathbb{V}$, with a right-hand side defined as $\langle F, \bar{v}\rangle = \sum_k L_k(Tr_k^{-*}(\bar{R}_k\bar{v}))$, where v_k is any function in $\mathbb{H}(\Omega_k)$ such that $v_k = \bar{v}$ on Γ_k.

ν_1, ν_2	ν_1/ν_2	$\vec{b} = (\pm 1, 0, 0)$	$\vec{b} = (0, 1, 1)$	$\vec{b} = (\pm 1, 3, 5)$
$10^{-1}, 10^{-5}$	10^4	10 11	17	15 17
$10^{-2}, 10^{-6}$		12 16	13	7 8
$10^{-1}, 10^{-6}$	10^5	10 11	17	15 17
$10^{-6}, 10^{-11}$		5 5	2	7 7
$10^{-1}, 10^{-7}$	10^6	10 11	17	15 17
$10^{3}, 10^{-3}$		3 3	3	3 3
$1, 10^{-7}$	10^7	6 7	9	11 11

Table 4.1: Number of iterations for the two-domain problem

3.3. Construction of the preconditioner. We extend the preconditioner $\mathcal{T}$ introduced in the previous section to an arbitrary number of subdomains and we generalize the ones proposed in ([2]) and ([8]). The interface operator (3.6) is preconditioned with a weighted sum of inverses based on a partition of unity argument:

$$T = \sum_{k=1}^{N} D_k^T (S_k)^{-1} D_k, \tag{3.7}$$

with $\sum_{k=1}^{N} D_k \bar{R}_k = Id_\Gamma$. For any $F_k \in \mathbb{V}_k'$ the action of the operator $(S_k)^{-1} F_k$ is equal to the trace on Γ_k of the solution w_k of the local variational problem $a_k(w_k, v_k) = \langle F_k, Tr_k v_k \rangle$, $\forall v_k \in \mathbb{H}(\Omega_k), w_k \in \mathbb{H}(\Omega_k)$, which is associated to the operator $\mathcal{L}_k = -\mathrm{div}(\nu_k \nabla w) + \vec{b} \cdot \nabla w + aw$ with Robin boundary condition on the interface $\nu_k \frac{\partial w}{\partial n_k} - \frac{1}{2} \vec{b} \cdot \vec{n}_k w = F_k$. In order to achieve good parallelization, the weights D_k are defined on each interface degree of freedom $\bar{u}(P)$ (with $P \in \Gamma_k$) as

$$D_k \, \bar{u}(P) \; = \; C_P \, \frac{\nu_k}{\sum_{P \in \Gamma_j} \nu_j} \, \bar{u}(P),$$

where the constant C_P is chosen to satisfy the partition of unity requirement, and depends only on the number of subdomains to which the point P belongs.

4. Numerical results in 3D. Problem (3.1) is discretized by means of GALS second order finite elements on hexaedral decomposition. The interface problem is solved by a GMRES algorithm preconditioned by the operator $\mathcal{T}$, which stops when the residual is less than 10^{-10}. We consider $\Omega = [0, 1]^3$, the unit cube, as costituted of two different materials with viscosity coefficients ν_1 and ν_2, we choose $a = 1$ and $f \equiv 0$ in the whole Ω, and we force the solution to have a boundary layer by imposing $u = 1$ on the bottom face of the cube as well as homogeneous Dirichlet conditions on the rest of the boundary $\partial\Omega$. We consider large jumps between the viscosity coefficients.

In Table 4.1 we report the number of iterations for a two-domain decomposition, where we choose different convective fields: perpendicular to the interface ($\vec{b} = \vec{e}_1$), parallel ($\vec{b} = \vec{e}_2 + \vec{e}_3$) and oblique ($\vec{b} = \vec{e}_1 + 3\,\vec{e}_2 + 5\,\vec{e}_3$). The preconditioner appears a little sensitive to the direction of the velocity but it is insensitive to the amplitude of the jumps in the viscosity coefficients.

In Table 4.2 we report the number of iterations for a eight domain decomposition. Each coefficient ν_j ($j = 1, 2$) refers to four subdomains which mutual position is varied: in Test 1 the two half cubes of the previous test are decoupled into four smaller subdomains, the configuration of Test 2 is given in Figure 4.1, while Test 3 is a black and white coloring where each subdomain of one kind is surrounded by subdomains of the other one. The convective

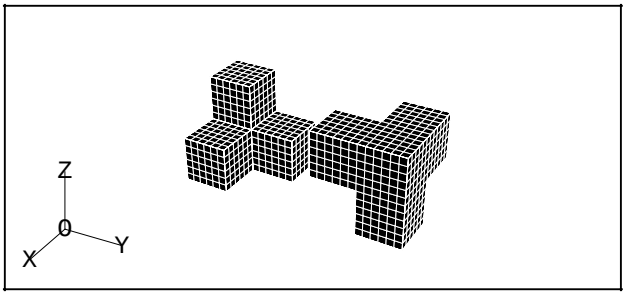


Figure 4.1: The subdomains Ω_1 (left) and Ω_2 (right) in Test 2.

ν_1, ν_2	ν_1/ν_2	Test 1	Test 2	Test 3
$10^{-1}, 10^{-5}$	10^4	33	33	34
$10^{-1}, 10^{-6}$	10^5	32	33	34
$10^{-1}, 10^{-7}$	10^6	32	33	34
$10^3, 10^{-3}$	10^6	29	28	21
$1, 10^{-7}$	10^7	29	31	29

Table 4.2: Number of iterations for the multidomain problem

field is $\vec{b} = -2\pi(y-0.5)\,\vec{e}_1 + 2\pi(x-0.5)\,\vec{e}_2 + \sin(2\pi(x-0.5))\,\vec{e}_3$. The preconditioner is again insensitive to the jumps and to the position of the subdomains.

A complete description of the tests will be given in ([6]).

5. Conclusions. The proposed preconditioner is a generalization of the Robin-Robin one to advection-diffusion problems with discontinuous coefficients. Numerical tests in 3D show, as we expected from the theoetical analysis of Section 2, that the preconditioner is fairly insensitive to the jumps in the viscosity coefficients as well as to the convective field, while it remains a little sensitive to the number of subdomains, but this seems unavoidable for advection-dominated problems. However, our knowledge of the preconditioner is not complete, and further work needs to be done: a convergence analysis in a more general setting is not yet available, the introduction of a coarse space to reduce the sensitivity to the number of subdomains should be analyzed and the algorithm should be tested on less academical situations.

REFERENCES

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

[2] Y. Achdou, P. L. Tallec, F. Nataf, and M. Vidrascu. A domain decoposition preconditioner for an advection-diffusion problem. *Comp. Meth. Appl. Mech. Engrg*, 184:145–170, 2000.

[3] J.-F. Bourgat, R. Glowinski, P. Le Tallec, and M. Vidrascu. Variational formulation and algorithm for trace operator in domain decomposition calculations. In T. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Domain Decomposition Methods*, pages 3–16, Philadelphia, PA, 1989. SIAM.

[4] J. H. Bramble, J. E. Pasciak, and A. H. Schatz. The construction of preconditioners for elliptic problems by substructuring, I. *Math. Comp.*, 47(175):103–134, 1986.

[5] M. Dryja and O. B. Widlund. Additive Schwarz methods for elliptic finite element problems in three dimensions. In D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs, and R. G.

Voigt, editors, *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*, pages 3–18, Philadelphia, PA, 1992. SIAM.

[6] L. Gerardo Giorda, P. Le Tallec, and F. Nataf. A Robin-Robin preconditioner for advection-diffusion equations with discontinuous coefficients. Technical report, CMAP, Ecole Polytechnique, 2002.

[7] P. Le Tallec. Domain decomposition methods in computational mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 1 (2), pages 121–220. North-Holland, 1994.

[8] P. L. Tallec and M. Vidrascu. Generalized Neumann-Neumann preconditioners for iterative substructuring. In P. E. Bjørstad, M. Espedal, and D. Keyes, editors, *Domain Decomposition Methods in Sciences and Engineering*. John Wiley & Sons, 1997. Proceedings from the Ninth International Conference, June 1996, Bergen, Norway.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

44. On a selective reuse of Krylov subspaces in Newton-Krylov approaches for nonlinear elasticity

P. Gosselet[1], C. Rey[2]

1. Introduction. We consider the resolution of large-scale nonlinear problems arising from the finite-element discretization of geometrically non-linear structural analysis problems. We use a classical Newton Raphson algorithm to handle the non-linearity which leads to the resolution of a sequence of linear systems with non-invariant matrices and right hand sides. The linear systems are solved using the FETI-2 algorithm. We show how the reuse, as a coarse problem, of a pertinent selection of the information generated during the resolution of previous linear systems, stored inside Krylov subspaces, leads to interesting acceleration of the convergence of the current system.

Nonlinear problems are a category of problems arising from various applications in mathematics, physics or mechanics. Solving these problems very often leads to a succession of linear problems the solution to which converges towards the solution to the considered problem. Within the framework of this study, all linear systems are solved using a conjugate gradient algorithm. It is well known that this algorithm is based on the construction of the so-called Krylov subspaces, on which depends its numerical efficiency and its convergence behaviour.

The purpose of this paper is to accelerate the convergence of linear systems by reusing information arising from previous resolution processes. Such an idea has already led to a classical algorithm for invariant matrices [8] which has been successfully extended to the case of non invariant matrices [6, 7]. We here propose, thanks to a spectral analysis of linear systems, to select the most significant part of the information generated during conjugate gradient iterations to accelerate the convergence via an augmented Krylov conjugate gradient algorithm.

The remainder of this paper is organized as follows: section 2 addresses characteristic properties of preconditioned conjugate gradient, section 3 exposes the acceleration strategies, section 4 gives numerical assessments and section 5 concludes the paper.

2. Basic properties of preconditioned conjugate gradient. We consider the linear system $Ax = b$ solved with a M-preconditioned conjugate gradient (A and M are $N \times N$ real symmetric positive definite matrices). We note x_i the ith estimation to $x = A^{-1}b$, $r_i = b - Ax_i = A(x - x_i)$ the associated residual and $z_i = M^{-1}r_i$ the preconditioned residual. In order to concentrate the notations, we also note with capital letters matrices built from set of vectors, e.g. $R_i = (r_0, \ldots, r_{i-1})$. Given initialization x_0, preconditioned conjugate gradient iteration consists in searching

$$\begin{array}{c} x_i \in \{x_0\} + \mathcal{K}_i(M^{-1}A, z_0) \text{ with } r_i \perp \mathcal{K}_i(M^{-1}A, z_0) \\ \text{where } \mathcal{K}_i(M^{-1}A, z_0) \text{ is the } i\text{th Krylov subspace} \\ \mathcal{K}_i(M^{-1}A, z_0) = \text{Span}(z_0, \ldots, (M^{-1}A)^{i-1}z_0) = \text{Range}(Z_i) \end{array} \tag{2.1}$$

2.1. Augmented conjugate gradient. The augmentation consists in defining full-ranked constraint matrix C and imposing $C^T r_i = 0$. It leads to the definition of a modified Krylov subspaces $\tilde{\mathcal{K}}_i(M^{-1}A, z_0, C)$ [2]:

$$\begin{array}{c} \tilde{\mathcal{K}}_i(M^{-1}A, z_0, C) = \mathcal{K}_i(M^{-1}A, z_0) \oplus \text{Range}(C) \\ x_i \in \{x_0\} + \tilde{\mathcal{K}}_i(M^{-1}A, z_0, C) \text{ with } r_i \perp \tilde{\mathcal{K}}_i(M^{-1}A, z_0, C) \end{array} \tag{2.2}$$

[1]Laboratoire de Modélisation et Mécanique des Structures, gosselet@ccr.jussieu.fr
[2]Laboratoire de Modélisation et Mécanique des Structures, rey@ccr.jussieu.fr

$Ax = b$ with $C^T r_i = 0$	Iterations $i = 0, \ldots, s$
$P_C = Id - C\left(C^T AC\right)^{-1} C^T A$	$z_i = P_C M^{-1} r_i \qquad (w_0 = z_0)$
Initialization (x_{00} is arbitrary)	$w_i = z_i + \sum_{j=0}^{i-1} \beta_i^j w_j \qquad \beta_i^j = -\dfrac{(z_i, Aw_j)}{w_j, Aw_j}$
$x_0 = C\left(C^T AC\right)^{-1} C^T b + P_C x_{00}$	$x_{i+1} = x_i + \alpha_i w_i \qquad \alpha_i = \dfrac{(r_i, z_i)}{(w_i, Aw_i)}$
$r_0 = b - Ax_0$	$r_{i+1} = r_i - \alpha_i A w_i$

Property: $R_i^T Z_i$ and $W_i^T A W_i$ are diagonal

Figure 2.1: Augmented Preconditioned Conjugate Gradient

The augmented preconditioned conjugate gradient can be implemented with a projected algorithm (fig. 2.1): initialization and projector P_C ensure orthogonality conditions.

Remark 2.1 *Although no optimality result holds anymore when matrix A is non-positive, conjugate gradient still proves good convergence behaviour [5].*

Remark 2.2 *As M is definite positive, it can be factorized under Cholevsky's form $M = LL^T$. Following [9] we prove that the M-preconditioned C-augmented conjugate gradient is equivalent to a non-preconditioned $\hat{C}$-augmented conjugate gradient $\hat{A}\hat{x} = \hat{b}$ with :*

$$\begin{array}{cccc} \hat{A} = L^{-1} A L^{-T} & \hat{x}_i = L^T x_i & \hat{b} = L^{-1} b & \hat{C} = L^T C \\ \hat{w}_i = L^T w_i \ \hat{z}_i = L^T z_i & \hat{r}_i = L^{-1} r_i & \beta_i^j = \hat{\beta}_i^j & \alpha_i = \hat{\alpha}_i \end{array} \tag{2.3}$$

2.2. Ritz's spectral analysis of symmetric system. Ritz's values and vectors $(\theta_i^j, \hat{y}_i^j)_{1 \leqslant j \leqslant i}$ defined in equation (2.4) are the eigenelements of the projection of matrix $\hat{A}$ onto $\tilde{\mathcal{K}}_i(\hat{A}, \hat{r}_0, \hat{C})$, they converge ($i \to N$) to eigenelements of matrix $\hat{A}$ [5].

$\hat{V}_i$ orthonormal basis of $\tilde{\mathcal{K}}_i(\hat{A}, \hat{r}_0, \hat{C})$	Diagonalization $B_i = Q_i^B \Theta_i {Q_i^B}^T$
$B_i = \hat{V}_i^T \hat{A} \hat{V}_i$ Rayleigh's matrix	$\Theta_i = \mathrm{Diag}(\theta_i^j)_{1 \leqslant j \leqslant i}$
	${Q_i^B}^T Q_i^B = Id, \quad \hat{Y}_i = \hat{V}_i Q_i^B$

(2.4)

Ritz's representation of conjugate gradient provides meaningful information. Especially, the convergence of Ritz's values is directly linked to the convergence of the conjugate gradient:

$$\hat{x} - \hat{x}_i = \pi(\hat{A})(\hat{x} - \hat{x}_0) \text{ with } \pi(\xi) = \prod_{j=1}^{i} \frac{\theta_i^j - \xi}{\theta_i^j} \tag{2.5}$$

3. Choice of optional constraints. The choice of matrix $\hat{C}$ is a very accurate problem which requires a study of the governing factors of the convergence of the conjugate gradient [11]. The condition number, which is proved to decrease [1] whatever the $\hat{C}$ matrix may be, is not sufficient for a relevant analysis. In the remainder of the paper, we will call "active" eigenelements that are excited by (i.e. non-orthogonal to) the initial residual and "effective" active eigenelements that are not yet properly estimated by Ritz's elements. Only effective condition number influences the convergence rate: when an eigenvalue is sufficiently

well approximated inside the Krylov subspace, the conjugate gradient acts as if it had been suppressed from the resolution process. This explains the superconvergent behaviour of the conjugate gradient: when highest eigenvalues are sufficiently well approximated by Ritz's values, the effective condition number is very low and the convergence rate very high. So a good way to ensure a decrease of the effective condition number is to put active eigenvectors of $\hat{A}$ inside matrix $\hat{C}$.

However, computing a priori active eigenvectors of a system is as expensive as solving it, hence in this section we first show, inspiring from [9], how a posteriori computation can be achieved costlessly when reusing information generated during the conjugate gradient process, then we propose within the framework of multiple systems resolution to use approximation of the eigenvectors of previous systems as constraints to accelerate the convergence of current system.

3.1. Efficient computation of Ritz's elements. Hessemberg matrix H_i arising from Lanczos' procedure is a specific tridiagonal Rayleigh matrix the coefficients of which can be recovered from the coefficients of the conjugate gradient:

$$\begin{gathered}
\underline{\hat{R}}_i = (\frac{\hat{r}_0}{\|\hat{r}_0\|}, \ldots, (-1)^{i-1}\frac{\hat{r}_{i-1}}{\|\hat{r}_{i-1}\|}) \text{ orthonormal basis of } \tilde{\mathcal{K}}_i(\hat{A}, \hat{r}_0, \hat{C}) \\
\underline{Z}_i = (\frac{z_0}{(z_0, r_0)}, \ldots, \frac{(-1)^{i-1} z_{i-1}}{\sqrt{(z_{i-1}, r_{i-1})}}) \text{ M-orthonormal basis of } \mathcal{K}_i(M^{-1}A, z_0, C) \\
H_i = \underline{\hat{R}}_i^T \hat{A} \underline{\hat{R}}_i = \underline{Z}_i^T A \underline{Z}_i \\
H_i = \text{Tridiag}(\eta_{j-1}, \delta_j, \eta_j) \text{ with } \eta_j = \frac{\sqrt{\beta_j^{j-1}}}{\alpha_j} \text{ and } \delta_j = \frac{1}{\alpha_j} + \frac{\beta_{j-1}^{j-2}}{\alpha_{j-1}}
\end{gathered} \tag{3.1}$$

So a tridiagonal Rayleigh matrix can be computed without vector manipulation, and a specific Lapack procedure can then be used to compute the eigenelements. To have an action on the non-symmetric preconditioned problem, we define "transported Ritz's vectors" $Y_i = L^{-T}\hat{Y}_i = \underline{Z}_i Q_i^H$, they verify the following orthonormalities:

$$Y_i^T A Y_i = \Theta_i \text{ and } Y_i^T M Y_i = Id_i \tag{3.2}$$

3.2. Selective reuse of Krylov subspaces. We focussed on the interest of reusing eigenvectors (or at least good estimations) as constraints. Our strategies are based on the simple equivalence $\hat{C} = (\hat{y}_i^j) \Leftrightarrow C = (y_i^j)$ which means that a spectral action can be achieved acting directly on the preconditioned problem.

We now consider the resolution of a sequence of linear systems $A^k x^k = b^k$ ($k \geqslant 1$ stands for the number of the linear system, matrices and right hand sides are non-invariant) with augmented conjugate gradient. We propose two strategies based on the reuse of spectral information.

The first strategy is a simple total reuse of Ritz's vectors which is equivalent, since $\text{Range}(Y_i) = \text{Range}(W_i) = \mathcal{K}_i(M^{-1}A, z_0, C)$, to a total reuse of Krylov subspaces: matrix C^k is built concatenating all previous Krylov subspaces $C^k = (W_1, \ldots, W_{k-1})$ ($C^1 = 0$). As all the information is reused without selection, this strategy gives the best decrease of the number of iterations of the conjugate gradient expectable from the reuse of Krylov subspaces. Of course it quickly leads to huge C^k matrices and expensive computations to handle the augmented algorithm. Note that when A^k is invariant ($\forall k,\ A^k = A$), ${C^k}^T A C^k$ is a diagonal matrix and this algorithm is equivalent to a multiple right hand side conjugate gradient [8].

The second strategy aims at reducing the dimension of matrices C^k concentrating the information stored inside Krylov subspaces into few vectors. It is managed through the spectral analysis exposed above and the selection of Ritz's vectors associated to converged

Ritz's values. The convergence of the values is estimated computing the values for the last two iterations and comparing them.

$$\text{for } j \leqslant (i+1),\ \theta_i^j \text{ is converged if } \left| \frac{\theta_i^j - \theta_{i-1}^{j-1}}{\theta_i^j} \right| \leqslant \varepsilon \tag{3.3}$$

4. Numerical assessment. We now assess the reuse of Krylov subspaces on the computation of the buckling of a clamped-free beam (fig. 4.1). The beam is a composite structure made up of Saint-Venant–Kirchoff materials, fibers are 1000 times stiffer than the matrix. It is decomposed into 32 substructures. We use Newton Raphson's algorithm [10] to linearize the problem, the resolution is then conducted in 28 linear systems with non-invariant matrices $K^k u^k = f^k$ (k is the linear system number). The linear systems are solved with FETI-2 method equipped with Dirichlet's preconditioner and superlumped projector.

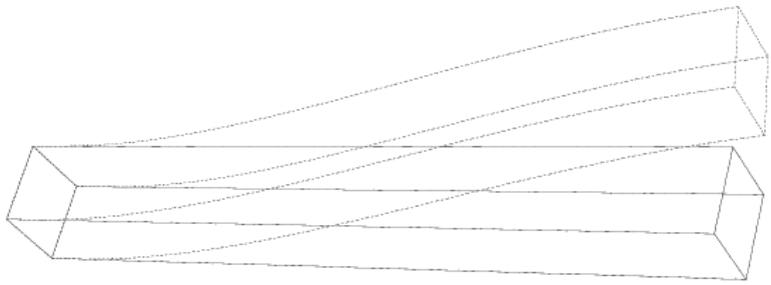


Figure 4.1: Buckling of the beam

4.1. Application of the reuse of Krylov subspaces to FETI-2. The Finite Elements Tearing and Interconnecting (FETI) method was first introduced by Farhat and Roux [4]. It consists in solving with a projected conjugate gradient the system arising from dual domain decomposition method. FETI-2 [3] solves the same problem with augmented conjugate gradient. Readers should refer to referenced papers for a complete description, we only show here the specificity of our strategies applied to FETI-2 method (fig. 4.2). With notations from [3], the system arising from the condensation writes:

$$\begin{pmatrix} F_I & -G_I \\ -G_I^T & 0 \end{pmatrix} \begin{pmatrix} \lambda \\ \alpha \end{pmatrix} = \begin{pmatrix} d \\ -e \end{pmatrix} \tag{4.1}$$

The first level projection P and initialization λ_{01} handle floating substructures, second level projector P_C and initialization λ_{02} handle the augmentation associated to matrix C. Note that constraints have to be made compatible with the first level projector $(PC)^T r_i = 0$. In the case of FETI algorithm, the augmentation possesses a mechanical interpretation: $P^T r_i$ represents the jump of the displacement field between substructures. Constraints matrix C then ensures a weak continuity of the displacement field. Forming and factorizing the so-called coarse problem matrix $\left((PC)^T F(PC)\right)$ is a complex operation requiring all-to-all exchanges between substructures, in a parallel processing context these operations are penalizing then matrix C has to be chosen as small as possible.

We checked that for this class of problem Dirichlet's preconditioner is positive for all the systems. We also verified the imbrication of the kernel of local matrices $\forall$ (substructure s, system k) $\text{Ker}({K^{(s)}}^{k+1}) \subset \text{Ker}({K^{(s)}}^{k})$ which implies that $\forall k$ $\text{Range}(G_I^{k+1}) \subset \text{Range}(G_I^k)$.

$$P = Id - QG_I \left(G_I^T Q G_I\right)^{-1} G_I^T$$
$$P_C = Id - (PC)\left((PC)^T F_I (PC)\right)^{-1} (PC)^T F_I$$

Initialization (λ_{00} is arbitrary)

$$\lambda_{01} = QG_I \left(G_I^T Q G_I\right)^{-1} e \qquad \lambda_{02} = (PC)\left((PC)^T F_I (PC)\right)^{-1} (PC)^T d$$
$$\lambda_0 = P_C(P\lambda_{00} + \lambda_{01}) + \lambda_{02}$$
$$r_0 = d - F_I \lambda_0$$

Iterations $i = 0, \ldots, s$

$$z_i = P_C P \tilde{F}_I^{-1} P^T r_i$$
$$w_i = z_i + \sum_{j=0}^{i-1} \beta_{ij} w_j \qquad (w_0 = z_0)$$
$$\lambda_{i+1} = \lambda_i + \alpha_i w_i \qquad \beta_{ij} = -\frac{(w_j, F_I z_i)}{(w_j, F_I w_j)}$$
$$r_{i+1} = r_i - \alpha_i F_I w_i \qquad \alpha_i = \frac{(w_i, r_i)}{(w_i, F_I w_i)}$$

Figure 4.2: Two-level FETI algorithm

So all previous Krylov subspaces are built orthogonally to the G_I^k matrix, hence when using vectors from Krylov subspaces as constraints we already have $P^k C^k = C^k$. Then the two projectors are decoupled which suppresses time consuming step of making constraints admissible.

4.2. Performance results. The first point concerns the choice of the ε parameter introduced in section 3.2 to determine whether Ritz's values are converged or not. Experiments (e.g. fig. 4.3) showed that the criterion is either very low ($> 10^{-14}$) or very high ($> 10^{-8}$), value ε can then be chosen inside a wide range without modifying the selection, typically we chose $\varepsilon = 10^{-13}$.

Figures 4.4, 4.5 and 4.6 summarize the action of the reuse of Krylov subspaces through the resolution of the linear systems. First figure 4.4 shows how effective the selection is: the number of constraints is quickly divided by a factor 2. Figure 4.5 presents the evolution of the number of iterations per linear system, the total reuse corresponds to the best result expectable from the reuse of Krylov subspaces, the number of iterations is divided by a factor 10, which proves the interest of the information stored inside Krylov subspaces. The selective reuse also proves interesting: with a two-time smaller constraints space, its performance results are quite near the total reuse. Figure 4.6 shows the performance results in terms of CPU time: the total reuse is already relevant, the selective reuse since its performance results in terms of iterations are almost equivalent with a lower number of constraints leads to impressive gain, it is 60% faster than the non accelerated method.

Figures 4.7 and 4.8 enable us to check the spectral action announced above, they represent the Ritz's spectrum for 4 linear systems (the 1st, 5th, 10th and 28th). The selective reuse filters the highest and the negative values, and suppresses part of mid-range values, giving better spectral properties for the resolution. Figures 4.9 and 4.10 show how the resolution process is improved by the selective reuse: two actions are combined, first a better

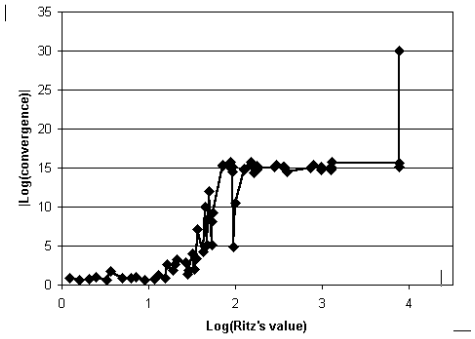


Figure 4.3: Convergence of Ritz's values

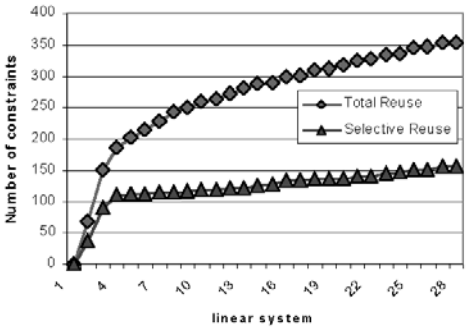


Figure 4.4: Action of Selective Reuse: number of constraints

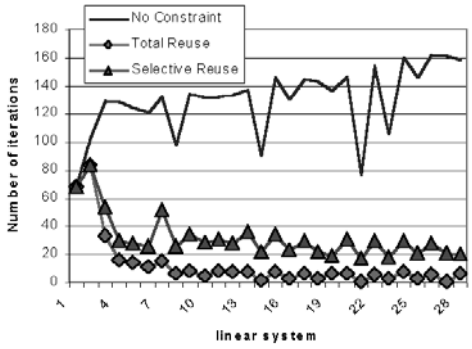


Figure 4.5: Action of Selective Reuse: number of iterations

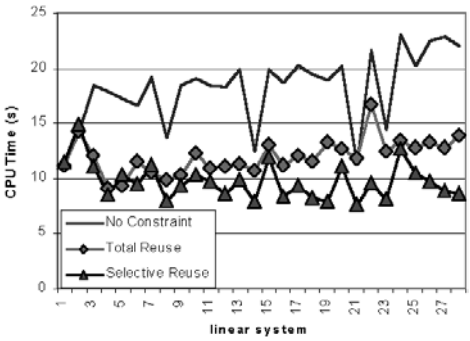


Figure 4.6: Action of Selective Reuse: CPU time

initialization is found, second the superconvergence is achieved from the beginning of the resolution.

5. Conclusion. In this paper we considered the resolution of a sequence of linear systems arising from geometrically nonlinear structural analysis, with a FETI-2 method. We proposed an algorithm to realize a spectral analysis of linear systems solved with a conjugate gradient algorithm with positive preconditioner. We showed that the complete reuse of former Krylov subspaces has already led to good performance results and that a selective reuse of Ritz vectors associated to Ritz's values giving good estimates of eigenvalues gave even better computational performance (up to 60% CPU time gain). Next studies will focuss on additional selection criteria for the Ritz vectors based on the activity of former vectors for the resolution of current system, the aim is to be even more selective and to suppress vectors containing information which is non-relevant for the current system.

We authors acknowledge support from the Centre Informatique National Enseignement Superieur (CINES) for computational resources.

REFERENCES

[1] Z. Dostal. Conjugate gradient method with preconditioning by projector. *INT. J. Comput.. Math.*, 23:315–323, 1988.

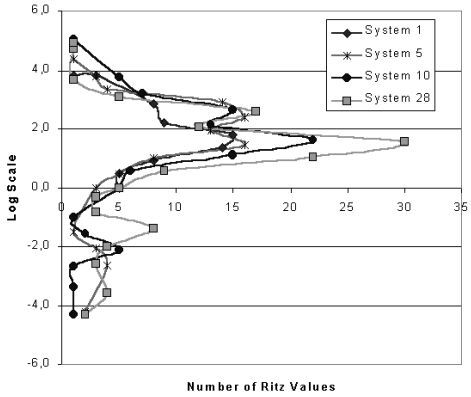


Figure 4.7: Ritz's spectrum, no constraint

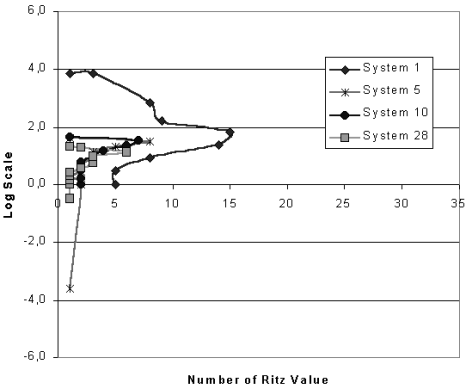


Figure 4.8: Ritz's spectrum, selective reuse

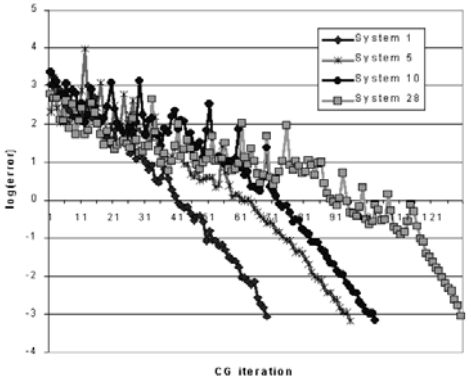


Figure 4.9: Evolution of the error, no constraint

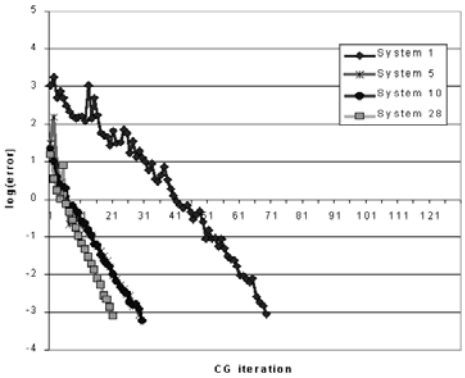


Figure 4.10: Evolution of the error, selective reuse

[2] J. Erhel and F. Guyomarc'h. An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems. *SIAM J. Matrix Anal. Appl.*, 21(4):1279–1299, 2000.

[3] C. Farhat, K. H. 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. *Computer Methods in Applied Mechanics and Engineering*, 184:333–374, 2000.

[4] C. Farhat and F.-X. Roux. A Method of Finite Element Tearing and Interconnecting and its Parallel Solution Algorithm. *Int. J. Numer. Meth. Engrg.*, 32:1205–1227, 1991.

[5] C. Paige, B. Parlett, and H. van der Vorst. Approximate solutions and eigenvalue bounds from Krylov subspaces. *Numerical Linear Algebra with Applications*, 2(2):115–133, 1995.

[6] C. Rey and F. Risler. A Rayleigh-Ritz preconditioner for the iterative solution to large scale nonlinear problems. *Numerical Algorithms*, 17:279–311, 1998.

[7] F. Risler and C. Rey. Iterative accelerating algorithms with Krylov subspaces for the solution to large-scale nonlinear problems. *Numerical Algorithms*, 23:1–30, 2000.

[8] Y. Saad. On the lanczos method for solving symmetric linear systems with several right hand sides. *Math. Comp.*, 48:651–662, 1987.

[9] Y. Saad. *Iterative Methods for Sparse Linear Systems*. PWS Publishing Company, 1996.

[10] P. L. Tallec. Numerical methods for non-linear three-dimensional elasticity. In J. L. Lions and P. Ciarlet, editors, *Handbook of numerical analysis vol.3*, pages 465–622. Elsevier, 1994.

[11] A. van der Sluis and H. van der Vorst. The rate of convergence of conjugate gradients. *Numer. Math.*, 48:543–560, 1986.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

45. Fast Solvers and Schwarz Preconditioners for Spectral Nédélec Elements for a Model Problem in H(curl)

Bernhard Hientzsch[1]

1. Introduction. In this paper, we present fast solvers and overlapping additive Schwarz methods for spectral Nédélec element discretizations for a model problem in $H(\text{CURL})$. Results of numerical experiments in two dimensions and arguments for a theoretical condition number bound in two and three dimensions are presented. We first show the derivation of the model problem in the implicit time discretization of one of the forms of Maxwell's equation and explain our discretization. We next present the main ideas of two fast direct solvers for such discretizations. We define the overlapping Schwarz methods considered in this paper, present numerical results in two dimensions and then, finally, explain the derivation of a condition number estimate. The estimate obtained for element-wise overlap is quadratic in the relative overlap and the number of colors. In our estimate for general overlap, we obtain an additional factor of N, the degree of the polynomials inside each spectral element.

The model problem is: Find $\mathbf{u} \in H_0(\text{CURL}, \Omega)$ such that for all $\mathbf{v} \in H_0(\text{CURL}, \Omega)$

$$a(\mathbf{u}, \mathbf{v}) := (\alpha \mathbf{u}, \mathbf{v}) + (\beta \text{ CURL } \mathbf{u}, \text{ CURL } \mathbf{v}) = (\mathbf{f}, \mathbf{v}) \tag{1.1}$$

Here, Ω is a bounded, open, connected polyhedron in $\mathbb{R}^3$ or polygon in $\mathbb{R}^2$, $H(\text{CURL}, \Omega)$ is the space of vectors in $(L^2(\Omega))^2$ or $(L^2(\Omega))^3$ with CURL in $L^2(\Omega)$ or $(L^2(\Omega))^3$, respectively; $H_0(\text{CURL}, \Omega)$ is its subspace of vectors with vanishing tangential components on $\partial\Omega$; $\mathbf{f} \in (L^2(\Omega))^d$ for $d = 2, 3$, and $(\cdot, \cdot)$ denotes the inner product in $L^2(\Omega)$ of functions or vector fields. For simplicity, we will assume that α and β are piecewise constant.

2. The model problem, its discretization. The model problem is obtained in several problems in mathematical physics, among them in the time-discretization of several formulations of Maxwell's equation [3, Chapter 3]. For instance, the second order evolution equation for the electric field reads:

$$\epsilon \partial_t^2 \mathbf{E} + \sigma \partial_t \mathbf{E} + \text{ CURL } \left(\frac{1}{\mu} \text{ CURL } \mathbf{E} \right) = \partial_t \mathbf{j}_i$$

If we use an implicit discretization in time, substituting finite differences for the time derivatives and evaluating the terms without time derivatives at the time levels t^k ($k = n+1, n, \cdots$), we obtain

$$\alpha \mathbf{E}^{n+1} + \text{ CURL } \left(\beta \text{ CURL } \mathbf{E}^{n+1} \right) = \mathbf{f}^n$$

where $\mathbf{f}^n$ is constructed from $\mathbf{j}_i$, $\mathbf{E}^k$ and CURL $\mathbf{E}^k$. For instance using leapfrog for the first derivative, a central difference for the second derivative, and Backward Euler for the right hand side, we obtain

$$\alpha = \epsilon + \frac{\sigma}{2}\Delta t \qquad \beta = \frac{1}{\mu}\Delta t^2 \qquad \mathbf{f}^n = \Delta t^2 (\partial_t \mathbf{j}_i)|_{t=t^{n+1}} + 2\epsilon \mathbf{E}^n + \left(\frac{\sigma}{2}\Delta t - \epsilon \right) \mathbf{E}^{n-1}.$$

To use spectral elements, we use the variational form

$$(\alpha \mathbf{E}^{n+1}, \mathbf{F}) + (\beta \text{ CURL } \mathbf{E}^{n+1}, \text{CURL } \mathbf{F}) = (\mathbf{f}^n, \mathbf{F})$$

[1]Courant Institute of Mathematical Sciences, 251 Mercer Street, New York, NY 10012, E-mail: hientzsc@cims.nyu.edu *or* Bernhard.Hientzsch@na-net.ornl.gov, Homepage: http://www.math.nyu.edu/∼hientzsc. This work was supported in part by the U.S. Department of Energy under Contract DE-FC02-01ER25482.

and we also approximate the L^2-inner products by Gauss-Lobatto-Legendre quadrature.

The problem is posed in $H(\text{CURL})$ and the use of H^1-conforming elements introduces spurious eigenvalues and unphysical continuity conditions. There are approaches to regularize the problem, so that H^1-conforming elements can be used. But in general, for non-convex domains, the solution is not in H^1; so that complicated, weighted formulations have to be used. Nédélec [5, 6] introduced $H(\text{CURL})$-conforming elements, with edge, face and interior moments as degree of freedom; and Ben Belgacem and Bernardi [1] studied methods with spectral element degrees of freedom for Maxwell's evolution equations in a theoretical paper. We will use spectral element type nodal degrees of freedom and enforce only the continuity of the tangential components across element interfaces. For the quadrature, we use variable order Gauss-Lobatto-Legendre formulae.

The discretization on one element results in a block tensor product matrix in both the two-dimensional and three-dimensional case. For simplicity, we consider the two-dimensional case, similar statements hold about the three-dimensional case. The system on one element corresponding to (1.1) reads:

$$K_E\mathbf{u} = \tilde{\mathbf{f}} \qquad \text{or} \qquad \begin{pmatrix} M_1^x \otimes A^y & B^x \otimes C^y \\ B^{x,T} \otimes C^{y,T} & A^x \otimes M_2^y \end{pmatrix} \begin{pmatrix} u_1 \\ u_2 \end{pmatrix} = \begin{pmatrix} \tilde{f}_1 \\ \tilde{f}_2 \end{pmatrix}$$

The M are mass matrices, the A are spectral discretizations of scaled Helmholtz operators, B and C are coupling terms between the components involving derivatives and mass matrices.

If we subassemble such elements on a rectangular arrangement with matching polynomial degrees enforcing only tangential continuity, we obtain a global matrix of the same structure, allowing the same solvers as in the element case. For further discussion of the discretization and subassembly procedures we refer to [3, Chapter 8].

The element-by-element (or rectangle-by-rectangle) computation of the matrix-vector product $K\mathbf{u}$ can be implemented by dense matrix-matrix multiplications of the factors of the tensor products with the vector $\mathbf{u}$ laid out in matrix form. It can be implemented by calls to a highly optimized BLAS 3 kernel, and will therefore run at close to peak performance on modern computer architectures.

3. Fast direct solvers. First we treat the case in which we work on one element or a rectangular arrangement of elements with matching polynomial degrees. We transform the block tensor product matrix system on all components into a generalized Sylvester equation for one of the components. Thus, in the two-dimensional case, we eliminate one of the components, say u_2, and obtain:

$$(M_1^x \otimes A^y + C_T \otimes D_T)u_1 = \tilde{f}_1 + (F_T \otimes G_T)\tilde{f}_2$$

with

$$F_T = -B^x(A^x)^{-1} \quad G_T = C^y(M_2^y)^{-1} \quad C_T = F_T C^x \quad D_T = G_T B^y$$

This generalized Sylvester matrix equation can be solved in several different ways. For one, we could solve generalized eigenvalue problems for the matrix pairs (M_1^x, C_T) and (A^y, D_T) and use the eigenvector matrices to diagonalize all factors in the tensor products, and the solution would reduce to a componentwise multiplication with a matrix of the same size as u_1. This is the so-called fast diagonalization method proposed by Lynch et al [4] and used for instance by Tufo and Fischer [9] for the Navier-Stokes equations. There are also Hessenberg and Schur reduction algorithms, see, e.g., Gardiner et al [2], that do not involve the possibly unstable use of transformations to and from the eigenbasis. In our implementation, we used a fast diagonalization method that performs well on most of the examples; we are in the process of testing the method of Gardiner et al.

Once A^x and M_2^y have been factored and the eigenbases and associated transformations have been found, the solution of the Sylvester matrix equation and the back-solve for u_2

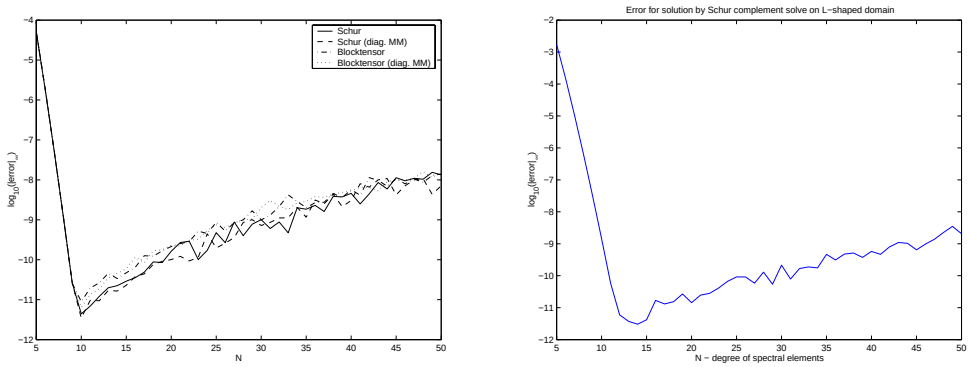


Figure 3.1: Direct solution of (1.1) with $\alpha = 1$, $\beta = 1$. Left: Fast diagonalization solver and Schur interface solver on 5×5 elements of degree $N \times N$, with an exact and with a diagonal mass matrix. Right: Schur interface solver on a L-shaped domain made from three spectral elements.

can be implemented by dense matrix-matrix multiplications and elementwise matrix-matrix multiplication, both kernels from BLAS that are usually available in highly optimized form for the common platforms and run at close to peak performance.

For the case that the above solver is not applicable because the domain is not a rectangle with appropriate polynomial degrees in the elements, we have implemented a direct substructuring method as a direct solver. The local Schur complements can be formed by solving local problems for each (tangential) degree of freedom on the element boundary. We can subassemble the Schur complement and the right hand side for the Schur complement system using the local Schur complements and local solves. The Schur complement system for the tangential components on the interface is then solved, and interior values are found by local (fast) interior solves.

The convergence for two examples is shown in figure 3.1. For further discussion of the implementation, timings, and experiments, we refer to [3, Chapter 9].

4. Overlapping Schwarz methods. To define Schwarz preconditioners in the standard abstract framework (see, e.g., [7]), we have to specify subspaces and solvers on them. We start with a collection of subdomains Ω_i that are either spectral elements themselves or rectangular arrangements thereof. Each subdomain is of size H, each spectral element has uniform degree N in all components. (The analysis goes through for more complicated settings, we chose this case here for simplicity and ease of presentation.) We also define overlapping subregions $\Omega'_{j,\delta} \subset \Omega$ with an overlap of δ. These subregions can be constructed in several ways, e.g., by extending subdomains by a fixed overlap δ in all directions, or by finding vertex centered subdomains that overlap by δ. Most of our early computations (and the numerical results that we show in this paper) were performed on 2×2 vertex centered assemblies of subdomains (taken as single spectral elements), but we will present numerical results for other layouts of $\Omega'_{j,\delta}$ and small overlap in a forthcoming paper.

The local spaces V_j are the linear span of the basis functions associated to Gauss-Lobatto-Legendre points in $\Omega'_{j,\delta}$. In general, the support of functions in V_j will be larger than $\Omega'_{j,\delta}$, but if one only considers the Gauss-Lobatto-Legendre grid, they vanish on grid points outside $\Omega'_{j,\delta}$. On the local spaces we use exact solvers which corresponds to inversion of a submatrix of K. In the 2×2 case the local solve corresponds to the solution of a standard tangential value problem on 2×2 element patches. In any case, the local solve can be implemented

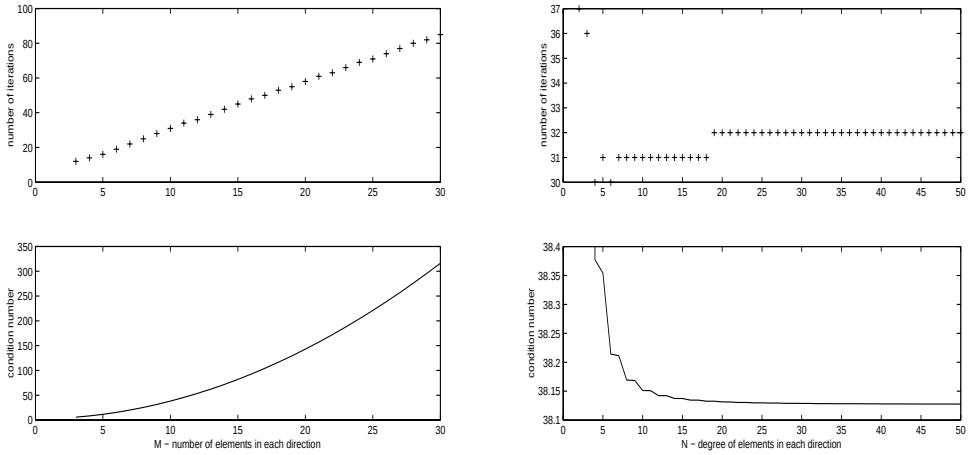


Figure 5.1: One-level method, $\alpha = 1$, $\beta = 1$. Left: Scaling with respect to M, the number of subdomains of degree 10×10. Right: Scaling with respect to N, the degree inside the 10×10 spectral elements.

using the direct fast solvers introduced in the previous section.

The coarse space V_0 is a low-order Nédélec spectral element space of uniform degree N_0 defined on the coarse (subdomain level) mesh. We use the direct solvers of the last section as exact solvers. In the standard way, the local and the coarse solve define local projections T_i and T_0 that can be used to implement different overlapping Schwarz methods. In this paper, we only consider two additive operators: a one-level operator T_{as1} and a two-level operator T_{as2}, defined by

$$T_{as1} = \sum_{i \geq 1} T_i \qquad T_{as2} = T_0 + T_{as1}$$

5. Numerical results in two dimensions. We solve the model problem (1.1) with $\alpha = 1$, $\beta = 1$ on the unit square, decomposed into $M \times M$ subdomains. These subdomains are single spectral Nédélec elements of degree $N \times N$ in all components. The overlapping subregions are patches of 2×2 spectral elements centered around each interior vertex of the spectral element mesh. (Therefore, $\delta = H$.) We have implemented a conjugate gradient method with the fast matrix-vector multiplication $K\mathbf{u}$ by tensor products with the additive Schwarz preconditioners defined by the fast local and coarse solves of section 3. We report the number of iterations needed to decrease the norm of the residual by 10^{-6}, and we also show estimates for the condition number of the preconditioned operator obtained, using the Lanczos connection, from the parameters computed in that conjugate gradient run.

In figure 5.1, we present results for the one-level operator T_{as1}. On the left, we work with degree $N = 10$, and vary the number of subdomains M^2. We see that the iteration numbers grow approximately linearly in M, and that the condition number is growing superlinearly in M. This behavior is to be expected from the absence of a coarse space; the one-level method is not scalable with respect to M. Having fixed the number of subdomains $M^2 = 10 \times 10$, increasing the polynomial degree N improves the condition number slightly, as seen on the right, and the iteration numbers also stay bounded. The condition number stabilizes slightly above 38.

In figure 5.2, we show the results for the two level method T_{as2}. On the left, we show the scaling with M^2 in the case of fixed $N = 10$. Both the iteration numbers and condition

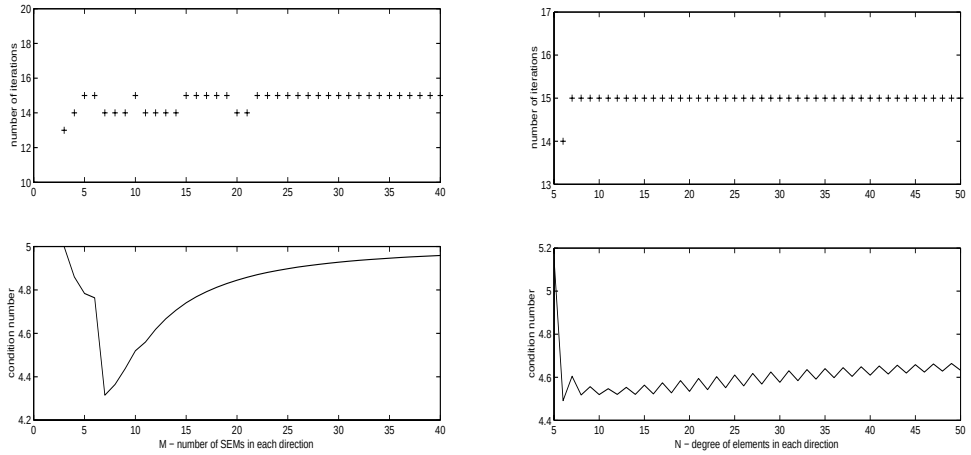


Figure 5.2: Two-level method, $\alpha = 1$, $\beta = 1$, $N_0 = 3$. Left: Scaling with respect to M, the number of subdomains of degree 10×10. Right: Scaling with respect to N, the degree inside the 10×10 spectral elements.

numbers are clearly bounded, the latter being below 5. On the right, we fix the number of subdomains to be 10×10, and vary the polynomial degree N. The condition number is clearly bounded, and less than 4.7 for $N = 50$. This example shows that the two-level additive Schwarz preconditioner T_{as2} is scalable with respect to M, and behaves uniformly in N. The condition numbers (and also the iteration numbers) of the two-level method are considerably smaller than those of the one-level method.

6. Condition number bound. We use the abstract Schwarz framework in which we obtain an estimate of the condition number in terms of N_C (number of colors), ω (norms of local solvers), and C_0^2 (splitting constant). For a general introduction to the abstract Schwarz framework in the context of preconditioners for the h-version, see Smith et al [7, chapter 5].

Since we use exact solvers, the parameter ω is equal to one. The largest eigenvalue is bounded by the number of colors N_C by a standard argument [7]. The smallest eigenvalue is bounded from below by C_0^{-2} where

$$\sum a(\mathbf{u}_i, \mathbf{u}_i) \leq C_0^2 a(\mathbf{u}, \mathbf{u}) \qquad \text{for } \mathbf{u} = \sum \mathbf{u}_i \qquad \text{and } \mathbf{u}_i \in V_i$$

We will estimate the smallest eigenvalue by a variant of the arguments in Toselli [8]. To complete the argument, we need to make explicit the N-dependence of the different bounds in Toselli's argument. The bounds were proven in the low-order case with a finite-dimensional space argument which only allows fixed N. For varying N the dimension of the space is related to N, and those arguments can not easily be extended to include the dependence on N.

We will sketch the analysis briefly and refer for complete details to a forthcoming paper.

Our variant of Toselli's analysis is expressed in the following theorem:

Theorem 6.1 (N-dependence of the condition number) *Given a Nédélec interpolation estimate on divergence-free functions from H(curl) with polynomial curl of the form:*

$$||(I - \mathbf{\Pi}_N^{ND,I})\mathbf{w}||_0 \leq Chf_1(N)||\mathrm{CURL}\mathbf{w}||_0 \tag{6.1}$$

the L^2-stability of the local splitting:

$$||\mathbf{\Pi}_N^{ND,I}(\chi_i \mathbf{u})||_0 \leq C f_2(N) ||\chi_i \mathbf{u}||_0 \tag{6.2}$$

the curl-stability of the local splitting:

$$||\mathrm{CURL}\left(\mathbf{\Pi}_N^{ND,I}(\chi_i \mathbf{u})\right)||_0 \leq C f_3(N) ||\mathrm{CURL}(\chi_i \mathbf{u})||_0 \tag{6.3}$$

and the standard conditions on a partition of unity χ_i:

$$||\chi_i||_\infty \leq C \qquad ||\nabla \chi_i||_\infty \leq \frac{C}{\delta} \tag{6.4}$$

then, the inverse of the smallest eigenvalue is bounded by

$$\max\left(CN_c\left(1+\frac{H}{\delta}\right), C\frac{\max(\alpha,\beta)}{\min(\alpha,\beta)}(1+N_c f_2^2(N)),\right.$$
$$\left. C\frac{\max(\alpha,\beta)}{\min(\alpha,\beta)}\left\{1+N_c f_3^2(N)\left(1+\left(\frac{H+hf_1(N)}{\delta}\right)^2\right)\right\}\right)$$

We will now discuss the different assumptions and estimates required and their proof.

The interpolation estimate (6.1) can be proven with $f_1(N) = 1 + C(\epsilon)N^{-1+\epsilon} \leq 1 + C(\epsilon)$. In two dimensions, $f_1(N) = C(\epsilon)N^{-1+f(\epsilon)}$. A similar bound should also hold in three dimensions, but is not yet proven.

We can reduce the L^2-stability of the local splitting (6.2) for polynomial χ_i to the L^2-stability of the Nédélec interpolation operator between polynomial spaces. Then we need to use χ_i that are polynomial interpolants of the standard piecewise linear ones. By the definition of the V_i, χ_i needs to be zero on the Gauss-Lobatto-Legendre grid points outside of $\Omega'_{i,\delta}$. We need to prove that they satisfy the conditions on the partition of unity (6.4). By a general theorem, we would obtain bounds involving the Lebesgue constant for the Gauss-Lobatto-Legendre interpolation, i.e., a logarithmic factor in N; numerical experiments and special arguments using the specific form of the Gauss-Lobatto-Legendre point values of the partition of unity result in the same bounds as for the standard piecewise linear partition of unity.

The L^2-stability of the Nédélec interpolant (6.2) can be proven by identifying the Nédélec interpolant as a componentwise tensor product operator having as factors discrete L^2- and modified H^1-projections. The L^2-projections are obviously stable, the modified H^1-projections have been proven, by a recent result of ours, to be stable for fixed differences in the degrees, and linear in the square root of the difference in the degree otherwise. We also confirmed the predicted behavior of the modified H^1-projections in extensive numerical experiments. Combining the results from the factors, we obtain a constant bound for $f_2(N)$ for χ_i of fixed degree (for instance for element-wise overlap), and a bound $f_2(N) = \sqrt{c_1 N + c_2}$ for χ_i of degree N (i.e., for general overlap).

The CURL-stability of the Nédélec interpolant (6.3) of the local splitting can be reduced to the L^2-stability of the Raviart-Thomas interpolant by the commuting diagram property. Therefore, for polynomial χ_i, it is enough to analyze the L^2-stability of the Raviart-Thomas interpolant between polynomial spaces of different degrees. One can identify the Raviart-Thomas interpolant as a componentwise tensor product operator having as factors discrete L^2- and modified H^1-projections. These factors are of the same form as for the Nédélec interpolant. Our recent results for the one-dimensional projections therefore imply, analogously to the results of the previous paragraph, a constant bound for $f_3(N)$ for χ_i of fixed degree and a bound $f_3(N) = \sqrt{c_3 N + c_4}$ for χ_i of degree N (i.e., for general overlap).

Combining the estimates with the above theorem, we obtain two corollaries:

Corollary 6.1 (Element-wise overlap) *In the case of element-wise overlap, the condition number of T_{as2} is bounded by*

$$\kappa(T_{as2}) \leq C(N_c+1)\frac{\max(\alpha,\beta)}{\min(\alpha,\beta)}\left(1+N_c\left(1+\left(\frac{H}{\delta}\right)^2\right)\right)$$

Corollary 6.2 (General overlap) *For general δ, an upper bound of the condition number of T_{as2} is given by*

$$\kappa(T_{as2}) \leq C(N_C+1)N\frac{\max(\alpha,\beta)}{\min(\alpha,\beta)}\left(1+N_C\left(1+\left(\frac{H}{\delta}\right)^2\right)\right)$$

It is not known if the powers of $\frac{H}{\delta}$ and N_C in both corollaries and of N in the second corollary are optimal, and it is not known if the estimates are sharp. We are performing a numerical study of the extreme eigenvalues for minimal and for fixed overlap which should give us some insight about sharpness and exponents.

For the limit cases $\alpha \to 0$ or $\beta \to 0$, one can find improved bounds by alternative splittings and proofs.

REFERENCES

[1] F. B. Belgacem and C. Bernardi. Spectral element discretization of the Maxwell equations. *Math. Comp.*, 68(228):1497–1520, 1999.

[2] J. D. Gardiner, A. J. Laub, J. J. Amato, and C. B. Moler. Solution of the Sylvester matrix equation $AXB^T + CXD^T = E$. *ACM Trans. Math. Software*, 18(2):223–231, 1992.

[3] B. Hientzsch. *Fast Solvers and Domain Decomposition Preconditioners for Spectral Element Discretizations of Problems in H(curl).* PhD thesis, Courant Institute of Mathematical Sciences, September 2001. also Technical Report TR2001-823, Department of Computer Science, Courant Institute.

[4] R. E. Lynch, J. R. Rice, and D. H. Thomas. Direct solution of partial difference equations by tensor product methods. *Numer. Math.*, 6:185–199, 1964.

[5] J.-C. Nédélec. Mixed finite elements in R^3. *Numer. Math.*, 35:315–341, 1980.

[6] J.-C. Nédélec. A new family of mixed finite elements in R^3. *Numer. Math.*, 50:57–81, 1986.

[7] B. F. Smith, P. E. Bjørstad, and W. Gropp. *Domain Decomposition: Parallel Multilevel Methods for Elliptic Partial Differential Equations.* Cambridge University Press, 1996.

[8] A. Toselli. Overlapping Schwarz methods for Maxwell's equations in three dimensions. *Numer. Math.*, 86(4):733–752, 2000.

[9] H. M. Tufo and P. F. Fischer. Terascale spectral element algorithms and implementations. In *Proceedings of the ACM/IEEE SC99 Conference*, 1999.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

46. A Dirichlet/Robin Iteration-by-Subdomain Domain Decomposition Method Applied to Advection-Diffusion Problems for Overlapping Subdomains

G. Houzeaux[1], R. Codina[2]

1. Introduction. We present a domain decomposition (DD) method to solve scalar advection-diffusion-reaction (ADR) equations which falls into the category of *iteration-by-subdomain* DD methods.

Domain decomposition methods are usually divided into two families, namely overlapping and non-overlapping methods. The former are based on the Schwarz method. At the differential level, they use alternatively the solution on one subdomain to update the Dirichlet data of the other. Contrary, non-overlapping DD methods use necessarily two different transmission conditions on the interface, in such a way that both the continuity of the unknown and its first derivatives are achieved on the interface (for ADR equations). Let us mention the Dirichlet/Neumann method introduced in [4, 8, 10]; the γ-Dirichlet/Robin method [2]; the Robin/Robin method [6, 9, 7]; the coercive γ-Robin/Robin method [2]; the Neumann/Neumann method [5, 3, 1], etc.

In the literature, all the mixed DD methods mentioned previously have been mainly studied in the context of disjoint partitioning. However, there exists no particular reason for restricting their application only to non-overlapping subdomains. This paper gives a possible line of study for the generalization of the mixed method to overlapping subdomains. We expect that the overlapping mixed DD methods will enjoy some properties of their disjoint brothers as well as some properties of the classical Schwarz method, as for example the dependence on the overlapping length.

Our motivation to study these types of methods has been to maintain the implementation advantages of the Schwarz method when used together with a numerical approximation of the problem. The possibility to have some overlapping simplifies enormously the discretization of the subdomains. However, very often this overlapping needs to be very small in practice, and thus the convergence rate of the Schwarz method becomes very small. Contrary to the Schwarz method, the limit case of zero overlapping will be possible using the formulation proposed herein. We have chosen to study an overlapping Dirichlet/Robin method, using the coercive bilinear form presented in [2] in the context of the γ-D/R and γ-R/R methods. This simplifies the analysis of the DD method as no assumption has to be made on the direction of the flow and its amplitude on the interfaces of the overlapping subdomains.

We would like to stress that our approach *is not* to view domain decomposition as a preconditioner for solving the linear systems of equations arising after the space discretization of the differential equations. In our case, the domain is decomposed at *the continuous level*. We are not concerned with the scaling properties with respect to the number of subdomains of the iteration-by-subdomain strategy we propose. For our purposes, it is enough to analyze *two* subdomains. More precisely, our final goal is to devise a Chimera type strategy taking Dirichlet/Robin(Neumann) transmission conditions rather than the classical Dirichlet/Dirichlet (Schwarz) approach. This paper must be understood as a theoretical basis for such a formulation. We recall briefly the Chimera method, of which we give an example in Figure 1.1. Firstly, independent meshes are generated for the background mesh and the mesh around the cylinder. Secondly, the mesh around the cylinder is placed on the background mesh. Then, according to some criteria (order of interpolation, geometrical overlap prescribed, etc.), we can impose in a simple way a Dirichlet condition on some nodes of the

[1]Universitat Politècnica de Catalunya - CIMNE, houzeaux@cimne.upc.es
[2]Universitat Politècnica de Catalunya - CIMNE, ramon.codina@upc.es

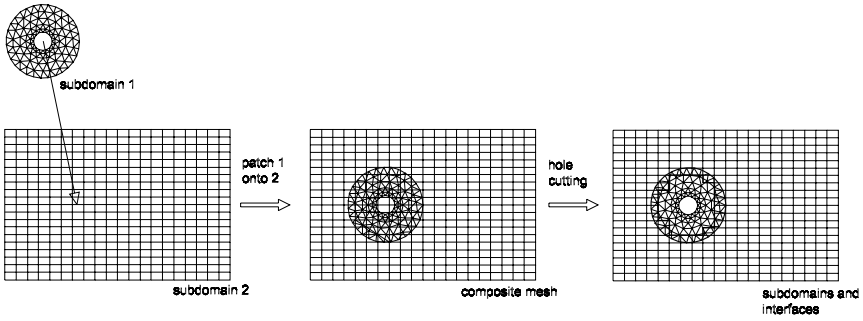


Figure 1.1: Chimera method.

background located inside the cylinder subdomain (this task is called hole cutting). Doing so, we form an apparent interface on the background subdomain to set up an iteration-by-subdomain method. Note that a natural condition of Neumann or Robin type is in general not possible as the apparent interface is irregular. Finally, by imposing a Dirichlet, Neumann or Robin condition on the outer boundary of the cylinder subdomain we can define completely an iteration-by-subdomain method to couple both subdomains. The Chimera method was first thought as a tool to simplify the meshing of complicated geometry. It is also a powerfull tool to treat subdomains in relative motion.

2. Problem statement. Let us consider the advection-diffusion-reaction problem of finding u such that:

$$\left\{ \begin{array}{ll} Lu := -\varepsilon \Delta u + \nabla \cdot (\boldsymbol{a} u) + \sigma u = f & \text{in } \Omega, \\ u = 0 & \text{on } \partial\Omega, \end{array} \right. \tag{2.1}$$

where Ω is a d-dimensional domain ($d = 1, 2, 3$) with boundary $\partial\Omega$, ε is the diffusion constant of the medium, f is the force term, $\boldsymbol{a}$ is the advection field (not necessarily solenoidal) and σ is a source (reaction) term.

We denote by $(\cdot, \cdot)$ the inner product in $L^2(\Omega)$, and by $V := H_0^1(\Omega)$ the space where u will be sought. Likewise, we use the notation

$$\langle \cdot, \cdot \rangle_\omega := \langle \cdot, \cdot \rangle_{H^s(\omega) \times H^{-s}(\omega)}, \tag{2.2}$$

for the duality pairing between the space $H^s(\omega)$ and its topological dual $H^{-s}(\omega)$, with $s = 1$ when ω is d-dimensional and with $s = 1/2$ when ω is $(d-1)$-dimensional.

Let us consider our differential problem 2.1. We restrict ourselves to solutions in V. To guarantee existence, we take $f \in H^{-1}(\Omega)$ and $\boldsymbol{a}, \sigma, \nabla \cdot \boldsymbol{a} \in L^\infty(\Omega)$. Since

$$\int_\Omega v \boldsymbol{a} \cdot \nabla u \, d\Omega = -\int_\Omega u \boldsymbol{a} \cdot \nabla v \, d\Omega - \int_\Omega u v \nabla \cdot \boldsymbol{a} \, d\Omega \quad \forall \, u, v \in V, \tag{2.3}$$

we transform the convective term into a skew symmetric operator, and we can enunciate our problem as follows: find $u \in V$ such that

$$a(u, v) = \langle f, v \rangle \quad \forall \, v \in V, \tag{2.4}$$

where the bilinear form is

$$a(w, v) \quad := \quad \varepsilon(\nabla w, \nabla v) + \frac{1}{2}(\boldsymbol{a} \cdot \nabla w, v) - \frac{1}{2}(w, \boldsymbol{a} \cdot \nabla v) + (\sigma_0 w, v), \tag{2.5}$$

with $\sigma_0 = \sigma + \frac{1}{2} \nabla \cdot \boldsymbol{a}$.

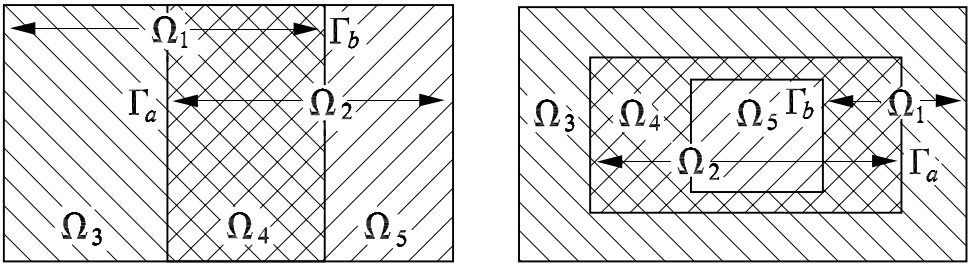


Figure 3.1: Examples of decomposition of a domain Ω into two overlapping subdomains Ω_1 and Ω_2.

3. Overlapping Dirichlet/Robin method.

3.1. Domain partitioning and definitions. We perform a geometrical decomposition of the original domain Ω into three disjoint and connected subdomains Ω_3, Ω_4 and Ω_5 such that

$$\Omega = \text{int}\left(\overline{\Omega_3 \cup \Omega_4 \cup \Omega_5}\right). \tag{3.1}$$

From this partition, we define Ω_1 and Ω_2 , as two overlapping subdomains:

$$\Omega_1 := \text{int}\left(\overline{\Omega_3 \cup \Omega_4}\right), \quad \Omega_2 := \text{int}\left(\overline{\Omega_5 \cup \Omega_4}\right). \tag{3.2}$$

Finally, we define Γ_a as the part of $\partial\Omega_2$ lying in Ω_1, and Γ_b as the part of $\partial\Omega_1$ lying in Ω_2. The geometrical nomenclature is shown in Figure 3.1. Γ_b and Γ_a are the *interfaces* of the domain decomposition method we now present. Ω_4 is the overlap zone. In the following, index i or j refers to a subdomain or an interface.

Let us introduce the following definitions:

$$(w,v)_{\Omega_i} := \int_{\Omega_i} wv\, d\Omega, \tag{3.3}$$

$$a_i(w,v) := \varepsilon(\nabla w, \nabla v)_{\Omega_i} + \frac{1}{2}(\boldsymbol{a}\cdot\nabla w, v)_{\Omega_i} - \frac{1}{2}(w, \boldsymbol{a}\cdot\nabla v)_{\Omega_i} + (\sigma_0 w, v)_{\Omega_i} \tag{3.4}$$

$$V_i := \{v \in H^1(\Omega_i) \mid v_{|\partial\Omega\cap\partial\Omega_i} = 0\}, \tag{3.5}$$

$$V_i^0 := H_0^1(\Omega_i), \tag{3.6}$$

where i can be any of the five subdomains introduced previously, i.e. $i = 1, 2, 3, 4$ or 5. Let us define the linear and continuous trace operators T_a and T_b on Γ_a and Γ_b, respectively. We explicitly define the trace space on Γ_a and Γ_b as $\Lambda_a := \{\mu_a \in H^{1/2}(\Gamma_a)\}$ and $\Lambda_b := \{\mu_b \in H^{1/2}(\Gamma_b)\}$, respectively.

3.2. Variational formulation. We propose to solve the following problem: find $u_1 \in V_1$ and $u_2 \in V_2$ such that

$$\begin{cases} a_1(u_1, v_1) = \langle f, v_1\rangle_{\Omega_1} & \forall\, v_1 \in V_1^0, \\ u_1 = u_2 & \text{on } \Gamma_b, \\ a_2(u_2, v_2) = \langle f, v_2\rangle_{\Omega_2} & \forall\, v_2 \in V_2^0, \\ a_3(u_1, E_3\mu_a) + a_2(u_2, E_2\mu_a) = \langle f, E_3\mu_a\rangle_{\Omega_3} + \langle f, E_2\mu_a\rangle_{\Omega_2} & \forall\, \mu_a \in \Lambda_a, \end{cases} \tag{3.7}$$

where E_i denotes any possible extension operator from Λ_a to $H^1(\Omega_i)$, that is to say,

$$E_i :\ \Lambda_a \longrightarrow H^1(\Omega_i), \quad T_a E_i \mu_a = \mu_a \quad \forall\, \mu_a \in \Lambda_a. \tag{3.8}$$

Equations 3.7_1 and 3.7_3 are the equations for the unknown in subdomains Ω_1 and Ω_2 respectively. Equation 3.7_2 is the condition that ensures continuity of the primary variable across Γ_b, and levels the solution in both subdomains. Finally, Eq. 3.7_4 is the equation for the primary variable on the interface Γ_a.

Theorem 3.1 *Problems 3.7 and 2.4 are equivalent.*

The proof can be obtained as in the case of the Dirichlet/Neumann method applied to disjoint subdomains. See for example [10].

3.3. Alternative formulation. We develop an alternative formulation for the domain decomposition method given by Eqs. 3.7_{1-4}.

Lemma 3.1 *The solution of the domain decomposition problem satisfies*

$$\frac{\partial u_1}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1 = \frac{\partial u_2}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_2 \quad \textit{on } \Gamma_a, \tag{3.9}$$

where $\partial(\cdot)/\partial n_2 = \boldsymbol{n}_2\cdot\nabla(\cdot)$, $\boldsymbol{n}_2$ *being the exterior normal to* Ω_2 *on* Γ_a.

In addition, we have the following result.

Theorem 3.2 *System of Eqs.* 3.7_{1-4} *can be reformulated as follows: find* $u_1 \in V_1$ *and* $u_2 \in V_2$ *such that*

$$\left\{\begin{array}{ll} a_1(u_1,v_1) = \langle f, v_1\rangle_{\Omega_1} & \forall\, v_1 \in V_1^0, \\ u_1 = u_2 & \textit{on } \Gamma_b, \\ a_2(u_2,v_2') = \langle f, v_2'\rangle_{\Omega_2} + \langle \varepsilon\dfrac{\partial u_1}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1, v_2'\rangle_{\Gamma_a} & \forall\, v_2' \in V_2. \end{array}\right. \tag{3.10}$$

The interpretation of the domain decomposition method now appears clearly. A Dirichlet problem is solved in Ω_1 using as Dirichlet data on the interface Γ_b the solution in Ω_2, whereas a mixed Dirichlet/Robin problem is solved in Ω_2 using as Robin data on Γ_a the solution in Ω_1. This formulation justifies the name *overlapping Dirichlet/Robin method* to designate this domain decomposition method.

Remark 3.1 *The system of Eqs.* 3.10_{1-3} *could have been derived directly from the following DD problem applied at the differential level:*

$$\left\{\begin{array}{ll} Lu_1 = f & \textit{in } \Omega_1, \\ u_1 = 0 & \textit{on } \partial\Omega_1 \cap \partial\Omega, \\ u_1 = u_2 & \textit{on } \Gamma_b, \\ Lu_2 = f & \textit{in } \Omega_2, \\ u_2 = 0 & \textit{on } \partial\Omega_2 \cap \partial\Omega, \\ \varepsilon\dfrac{\partial u_2}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_2 = & \varepsilon\dfrac{\partial u_1}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)u_1 \quad \textit{on } \Gamma_a. \end{array}\right. \tag{3.11}$$

3.4. Interface equations. A convenient way to study DD methods is to derive equations for the interface unknown(s). To do so, the problem is first rewritten into two purely Dirichlet problems for which the Dirichlet data are the unknowns on the interfaces. Starting form Eqs. 3.11_{1-6}, the problems to consider are:

$$\left\{\begin{array}{lcll} Lw_1 & = & f & \text{in } \Omega_1, \\ w_1 & = & 0 & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ w_1 & = & \lambda_b & \text{on } \Gamma_b, \end{array}\right. \quad \left\{\begin{array}{lcll} Lw_2 & = & f & \text{in } \Omega_2, \\ w_2 & = & 0 & \text{on } \partial\Omega_2 \cap \partial\Omega, \\ w_2 & = & \lambda_a & \text{on } \Gamma_a. \end{array}\right. \tag{3.12}$$

Now let us decompose w_1 and w_2 into L-homogeneous and Dirichlet-homogeneous parts,

$$w_1 = u_1^0 + u_1^*, \quad w_2 = u_2^0 + u_2^*, \tag{3.13}$$

where the L-homogeneous parts u_1^0 and u_2^0 are the solutions of

$$\left\{\begin{array}{lcll} Lu_1^0 & = & 0 & \text{in } \Omega_1, \\ u_1^0 & = & 0 & \text{on } \partial\Omega_1 \cap \partial\Omega, \\ u_1^0 & = & \lambda_b & \text{on } \Gamma_b, \end{array}\right. \quad \left\{\begin{array}{lcll} Lu_2^0 & = & 0 & \text{in } \Omega_2, \\ u_2^0 & = & 0 & \text{on } \partial\Omega_2 \cap \partial\Omega, \\ u_2^0 & = & \lambda_a & \text{on } \Gamma_a, \end{array}\right. \tag{3.14}$$

and the Dirichlet-homogeneous parts u_1^* and u_2^* are the solutions of

$$\left\{\begin{array}{lcll} Lu_i^* & = & f & \text{in } \Omega_i, \\ u_i^* & = & 0 & \text{on } \partial\Omega_i, \end{array}\right. \tag{3.15}$$

for $i = 1, 2$. We refer to u_1^0 as the L-homogeneous extension of λ_b into Ω_1, and we denote it by $\mathcal{L}_1\lambda_b$. Similarly, we call u_2^0 the L-homogeneous extension of λ_a into Ω_2, and we denote it by $\mathcal{L}_2\lambda_a$. In the case when $L = -\Delta$, $\mathcal{L}$ is the harmonic extension and is usually denoted by H. The Dirichlet-homogeneous parts u_1^* and u_2^* are rewritten as $\mathcal{G}_1 f$ and $\mathcal{G}_2 f$, respectively.

Comparing systems 3.12 with system 3.11, we have that $w_i = u_i$ for $i = 1, 2$ if and only if the following two conditions are satisfied:

$$\left\{\begin{array}{l} \varepsilon\dfrac{\partial w_2}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)w_2 = \varepsilon\dfrac{\partial w_1}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)w_1 \quad \text{on } \Gamma_a, \\ w_1 = w_2 \quad \text{on } \Gamma_b. \end{array}\right. \tag{3.16}$$

Using the previous definitions, conditions 3.16 can be rewritten as

$$\left\{\begin{array}{l} \varepsilon\dfrac{\partial \mathcal{L}_2\lambda_a}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_2\lambda_a = \varepsilon\dfrac{\partial \mathcal{L}_1\lambda_b}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_1\lambda_b \\ \qquad\qquad + \varepsilon\dfrac{\partial \mathcal{G}_1 f}{\partial n_2} - \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{G}_1 f - \varepsilon\dfrac{\partial \mathcal{G}_2 f}{\partial n_2} + \dfrac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{G}_2 f \quad \text{on } \Gamma_a, \\ \lambda_b = T_b\mathcal{L}_2\lambda_a + T_b\mathcal{G}_2 f \quad \text{on } \Gamma_b. \end{array}\right. \tag{3.17}$$

Let us clean up this system by introducing some definitions. In the first equation, we recognize the Steklov-Poincaré operator S_2 associated to subdomain Ω_2, defined as

$$S_2 : \ H^{1/2}(\Gamma_a) \longrightarrow H^{-1/2}(\Gamma_a), \tag{3.18}$$

$$S_2\lambda_a := \varepsilon\frac{\partial \mathcal{L}_2\lambda_a}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_2\lambda_a \quad \text{(evaluated on } \Gamma_a). \tag{3.19}$$

Note that $\mathcal{L}_2\lambda_a = \lambda_a$ on Γ_a. We define $\tilde{S}_b$, a Steklov-Poincaré-like operator acting on Γ_b, as

$$\tilde{S}_b : \ H^{1/2}(\Gamma_b) \longrightarrow H^{-1/2}(\Gamma_a), \tag{3.20}$$

$$\tilde{S}_b\lambda_b := -\varepsilon\frac{\partial \mathcal{L}_1\lambda_b}{\partial n_2} + \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{L}_1\lambda_b \quad \text{(evaluated on } \Gamma_a). \tag{3.21}$$

We also define $\tilde{T}_b$, the trace on Γ_b of the L-extension of λ_a into Ω_2:

$$\tilde{T}_b : \ H^{1/2}(\Gamma_a) \longrightarrow H^{1/2}(\Gamma_b), \tag{3.22}$$

$$\tilde{T}_b\lambda_a := T_b\mathcal{L}_2\lambda_a. \tag{3.23}$$

Finally, χ and χ' are defined as follows

$$\chi = \varepsilon\frac{\partial \mathcal{G}_1 f}{\partial n_2} - \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{G}_1 f - \varepsilon\frac{\partial \mathcal{G}_2 f}{\partial n_2} + \frac{1}{2}(\boldsymbol{a}\cdot\boldsymbol{n}_2)\mathcal{G}_2 f, \tag{3.24}$$

$$\chi' = T_b\mathcal{G}_2 f, \tag{3.25}$$

where we have $\chi \in H^{-1/2}(\Gamma_a)$ and $\chi' \in H^{1/2}(\Gamma_b)$. Owing to the previous definitions, the system of two equations for the interface unknowns reads

$$\left\{ \begin{array}{rcll} S_2\lambda_a & = & -\tilde{S}_b\lambda_b + \chi & \text{in } H^{-1/2}(\Gamma_a), \\ \lambda_b & = & \tilde{T}_b\lambda_a + \chi' & \text{in } H^{1/2}(\Gamma_b). \end{array} \right. \tag{3.26}$$

Let us introduce now the operator

$$\tilde{S}_1 : H^{1/2}(\Gamma_a) \longrightarrow H^{-1/2}(\Gamma_a), \tag{3.27}$$

$$\tilde{S}_1\lambda_a := \tilde{S}_b\tilde{T}_b\lambda_a, \tag{3.28}$$

and define S as

$$S = \tilde{S}_1 + S_2. \tag{3.29}$$

After substituting λ_b given by Eq. 3.26_2 into Eq. 3.26_1, we finally obtain the following system of equations for the interface unknowns

$$\left\{ \begin{array}{rcll} S\lambda_a & = & \chi - \tilde{S}_b\chi' & \text{in } H^{-1/2}(\Gamma_a), \\ \lambda_b & = & \tilde{T}_b\lambda_a + \chi' & \text{in } H^{1/2}(\Gamma_b). \end{array} \right. \tag{3.30}$$

Once λ_a and λ_b are obtained, we can solve the two Dirichlet problems 3.14 to obtain the L-homogeneous parts u_1^0 and u_2^0. The Dirichlet-homogeneous parts u_1^* and u_2^* are obtained by solving Eqs. 3.15 for $i = 1, 2$. Hence, the solutions u_1 and u_2 are calculated by adding up their respective L and Dirichlet-homogeneous contributions.

Let us go back to system 3.30. We can show that S_2 is both continuous (with constant M_{S_2}) and coercive (with constant N_{S_2}) and $\tilde{S}_1$ is continuous (with constant M_{S_1}) and non-negative. As a result we have the following theorem:

Theorem 3.3 *The operator S defined in 3.29 is invertible and system 3.30 has a unique solution $\{\lambda_a, \lambda_b\}$.*

The solutions of our interface problem can be written as

$$\left\{ \begin{array}{rcll} \lambda_a & = & S^{-1}(\chi - \tilde{S}_b\chi') & \text{in } H^{1/2}(\Gamma_a), \\ \lambda_b & = & \tilde{T}_bS^{-1}(\chi - \tilde{S}_b\chi') + \chi' & \text{in } H^{1/2}(\Gamma_b), \end{array} \right. \tag{3.31}$$

4. Iterative scheme.

4.1. Relaxed sequential algorithm. In this section, we derive an iterative procedure to solve the domain decomposition problem 3.7. The sequential version of the iterative overlapping D/R algorithm is defined solving first the Dirichlet problem, and then the Robin problem. Now we investigate the interface iterates produced by this relaxed iterative procedure. We enable relaxation of relaxation parameter $\theta > 0$ of one of the transmission condition at the same time. The Dirichlet-relaxed iterative scheme, denoted $\mathrm{D}_\theta/\mathrm{R}$, is given for any $k \geq 0$ by

$$\left\{ \begin{array}{rcl} S_2\lambda_a^{k+1} & = & -\tilde{S}_b\lambda_b^k + \chi, \\ \lambda_b^{k+1} & = & \theta(\tilde{T}_b\lambda_a^{k+1} + \chi') + (1-\theta)\lambda_b^k. \end{array} \right. \tag{4.1}$$

In terms of the interface unknowns, the Robin-relaxed iterative scheme, denoted $\mathrm{D}/\mathrm{R}_\theta$, produces the following iterates for any $k \geq 0$:

$$\left\{ \begin{array}{rcl} S_2\lambda_a^{k+1} & = & \theta(-\tilde{S}_b\lambda_b^k + \chi) + (1-\theta)S_2\lambda_a^k, \\ \lambda_b^{k+1} & = & \tilde{T}_b\lambda_a^{k+1} + \chi'. \end{array} \right. \tag{4.2}$$

Let us rewrite the Dirichlet and Robin-relaxed schemes as Richardson procedures. It can be shown that S_2 is invertible. We can therefore reformulate the system for the interface unknowns 3.26 as follows:

$$\begin{cases} Q_a\lambda_a = \chi_a, \\ Q_b\lambda_b = \chi_b, \end{cases} \tag{4.3}$$

where we have defined Q_a, Q_b, χ_a and χ_b by

$$Q_a = I_a + S_2^{-1}\tilde{S}_b\tilde{T}_b, \quad Q_b = I_b + \tilde{T}_b S_2^{-1}\tilde{S}_b, \tag{4.4}$$

$$\chi_a = S_2^{-1}\chi - S_2^{-1}\tilde{S}_b\chi', \quad \chi_b = \tilde{T}_b S_2^{-1}\chi + \chi'. \tag{4.5}$$

and where I_a is the identity on $H^{1/2}(\Gamma_a)$ and I_b is the identity on $H^{1/2}(\Gamma_b)$. By solving the Dirichlet-relaxed and Robin-relaxed systems for λ_a^{k+1} and λ_b^{k+1}, we can show that both schemes lead to the same following iterates for any $k \geq 1$:

$$\begin{cases} \lambda_a^{k+1} &= \theta(\chi_a - Q_a\lambda_a^k) + \lambda_a^k, \\ \lambda_b^{k+1} &= \theta(\chi_b - Q_b\lambda_b^k) + \lambda_b^k. \end{cases} \tag{4.6}$$

We recognize here two *stationary Richardson procedures* for solving Eqs. 4.3_1 and 4.3_2. We note that the Richardson procedure for solving λ_a is similar to that produced by the classical Dirichlet/Neumann method.

4.2. Convergence. This section studies the convergence of the DD algorithm, given by Eqs. 4.1_{1-2} for the D_θ/R method and Eqs. 4.2_{1-2} for the D/R_θ method. The result we can prove is:

Theorem 4.1 *Assume that ε is large enough so that*

$$\kappa^* := 2N_{S_2} - 2\|\boldsymbol{a}\|_{\infty,\Gamma_a}C_2^2\,\frac{M_{\tilde{S}_1} + M_{S_2}}{N_{S_2}} > 0, \tag{4.7}$$

where N_{S_2}, $M_{\tilde{S}_1}$ and M_{S_2} are the coercivity constant of S_2, and the continuity constants of $\tilde{S}_1$ and S_2, respectively. Then, there exists $\theta_{\max}$ such that for any given $\lambda_a^0 \in \Lambda_a$ and $\lambda_b^0 \in \Lambda_b$ and for all $\theta \in (0, \theta_{\max})$, the sequences $\{\lambda_a^k\}$ and $\{\lambda_b^k\}$ given by 4.6 converge in Λ_a and Λ_b, respectively. The upper bound of the relaxation parameter $\theta_{\max}$ can be estimated by

$$\theta_{\max} = \frac{\kappa^* N_{S_2}^2}{M_{S_2}(M_{\tilde{S}_1} + M_{S_2})^2} \tag{4.8}$$

More precisely, convergence is linear.

Remark 4.1 *This result carries over to the discrete variational problems provided the stability and continuity properties of the continuous case are inherited. In particular, the rate of convergence will be independent of the number of degrees of freedom.*

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*, 184:145–170, 2000.

[2] A. Alonso, R. L. Trotta, and A. Valli. Coercive domain decomposition algorithms for advection-diffusion equations and systems. *J. Comput. Appl. Math.*, 96:51–76, 1998.

[3] L. C. Berselli and F. Saleri. New substructuring domain decomposition methods for advection-diffusion equations. *J. Comput. Appl. Math.*, 116:201–220, 2000.

[4] P. E. Bjørstad and O. B. Widlund. Iterative methods for the solution of elliptic problems on regions partitioned into substructures. *SIAM J. Numer. Anal.*, 23(6):1093–1120, 1986.

[5] P. Le Tallec. Domain decomposition methods in computational mechanics. In J. T. Oden, editor, *Computational Mechanics Advances*, volume 1 (2), pages 121–220. North-Holland, 1994.

[6] P.-L. Lions. On the Schwarz alternating method. III: a variant for nonoverlapping subdomains. In T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund, editors, *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*, Philadelphia, PA, 1990. SIAM.

[7] G. Lube, L. Müller, and F.-C. Otto. A non-overlapping domain decomposition method for the advection-diffusion problem. *Computing*, 64:49–68, 2000.

[8] L. D. Marini and A. Quarteroni. A relaxation procedure for domain decomposition methods using finite elements. *Numer. Math*, (5):575–598, 1989.

[9] F. Nataf and F. Rogier. Factorization of the convection-diffusion operator and the Schwarz algorithm. M^3AS, 5(1):67–93, 1995.

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

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

47. Boundary Point Method in the Dynamic and Static Problems of Mathematical Physics

S. Kanaun, V. M. Romero[1]

1. Introduction. Boundary element method (BEM) is widely used for the numerical solution of integral equations of mathematical physics [1]. For the use of the BEM, the surface of the region should be divided into a finite number of subareas and the unknown functions are approximated by standard (as a rule polynomial) functions in every subarea. After applying the method of moments or the collocation method, the problem is reduced to the solution of a finite system of linear algebraic equations. The components of the matrix of this system are integrals over the subareas (boundary elements) of the surface. In many cases these integrals are singular and the complexity of their calculations depends on the type of approximating functions. In a standard BEM, a great portion of the computer time is spent in calculating these integrals. A non trivial auxiliary problem is dividing an arbitrary surface into a set of boundary elements.

In this work a new numerical method is used for the solution of boundary integral equations of some static and dynamic problems of mathematical physics. In this method actual distributions of unknown functions on the surface of the region is approximated by Gaussian functions located on the planes tangent to the boundary surface at some homogeneous set of surface nodes. The idea to use these functions for the solution of a wide class of integral equations of mathematical physics belongs to V. Maz'ya. The theory of approximation by Gaussian functions was developed in the works of V. Maz'ya [5] and V. Maz'ya and G. Shmidt [6].

In the method developed bellow we will use the following result of the mentioned authors. Let $u(x)$ be a scalar function in d-dimensional space $\mathbf{R}^d$. If $u(x)$ and its first derivative are bounded, $u(x)$ may be approximated by the following series

$$u(x) \approx u_h(x) = \sum_{m \in \mathbf{Z}^d} u_m \varphi(x - h\mathbf{m}), \quad \varphi(x) = \frac{1}{(\pi D)^{d/2}} \exp\left(-\frac{|x|^2}{Dh^2}\right). \tag{1.1}$$

Here $\mathbf{m} \in \mathbf{Z}^d$ is a d-dimensional vector with integer components, $h\mathbf{m}$ are the coordinates of the nodes of this approximation and h is the distance between the neighboring nodes, $u_m = u(h\mathbf{m})$ is the value of the function $u(x)$ at node $x = h\mathbf{m}$, D is a non-dimensional parameter. It is demonstrated in [2,3] that the following estimation holds

$$|u(x) - u_h(x)| \le ch \, \|\bigtriangledown u\| + |u(x)| R(D), \quad R(D) = O(\exp(-\pi^2 D)). \tag{1.2}$$

Here $||\nabla u||$ is the norm in the space of continuous functions, $c = O(1)$. If h is sufficiently small the error of the approximation (1.1) may be made negligible by the appropriate choosing of the parameter D ($D = O(1)$). The properties of this approximation were studied in detail in [5, 6].

The use of these functions for the solution of the integral equations of mathematical physics has two main advantages. First, the action of the integral operators of the problems on these functions in many cases is a combination of few standard functions. The latter may be simply tabulated, kept in the computer memory and then used for the solution of any similar problem for regions of arbitrary geometry. As a result, the time for the calculation of the matrix of the linear system obtained after the discretization of the problem, is essentially reduced in comparison with a standard BEM. It is also important that only the coordinates

[1]Instituto Tecnológico y de Estudios Superiores de Monterrey, Campus Estado de México, kanaoun@campus.cem.itesm.mx, vromero@campus.cem.itesm.mx

of the surface nodes and the surface orientations at the nodes are necessary for the surface description in the present method. The method was called by V. Maz'ya the Boundary Point Method (BPM) and in the latter the boundary points (nodes) play the role of boundary elements of the conventional BEM. Note that the problem of covering an arbitrary smooth surface by a homogeneous set of nodes is simpler than the detailed description of the geometry of all the boundary elements that is necessary for the application of any traditional BEM to the solution of surface integral equations.

Here we develope the method for the solution of 2D problems of elasticity and for 3D electromagnetic wave diffraction problems. The numerical results are compared with exact solutions existent in the literature.

2. Integral equations of the second boundary value problem of elasticity. Let an elastic body occupy region V in 3D or 2D-space with closed boundary S. The material of the body is homogeneous with elastic moduli tensor $\mathbf{C}$ (C_{ijkl}). The stress tensor in the body can be presented in the form

$$\sigma_{ij}(x) = \int_S S_{ijkl}(x - x')n_k(x')b_l(x')dS'. \tag{2.1}$$

Here $x(x_1, x_2, x_3)$ is a point of the medium with Cartesian coordinates x_1, x_2, x_3, summation with respect to repeated indexes is implied. The kernel of integral operators in Eq.(2.1) has form

$$S_{ijpq}(x) = -C_{ijkl} \bigtriangledown_k \bigtriangledown_m G_{ls}(x) C_{mspq} - C_{ijpq}\delta(x), \tag{2.2}$$

where $G_{ls}(x)$ is the Green function of the infinite medium with elastic moduli $\mathbf{C}$. Tensor $\sigma(x)$ in Eq.(2.1) satisfies the system of equations of continuum mechanics: $\nabla \cdot \sigma(x) = 0$, $\varepsilon^{(e)}(x) = \mathbf{C}^{-1} \cdot \sigma(x)$, Rot $\varepsilon^{(e)}(x) = \mathbf{0}$, $\bigtriangledown = \mathbf{e}_i \partial/\partial x_i$ is the vector gradient, $\mathbf{e}_i$ $(i = 1, 2, 3)$ are unit vectors of the axes x_i. A dot $(\cdot)$ is the scalar product, $\delta(x)$ is Dirac's delta-function. Thus, tensor $\sigma(x)$ in Eq.(2.1) gives us the solution of the second boundary value problem of elasticity if it satisfies the boundary conditions at the surface of the body

$$\sigma(x) \cdot \mathbf{n}(x)|_S = \mathbf{f}(x), \tag{2.3}$$

where $\mathbf{f}(x)$ is the vector of surface forces.

The integral equation for vector $\mathbf{b}(x)$ in Eq.(2.1) follows from the boundary condition (2.3) and takes the form

$$\int_S T_{ij}(x, x') \cdot b_j(x')dS' = f_i(x), \tag{2.4}$$

$$T_{ij}(x, x') = n_k(x) S_{kijl}(x - x') n_l(x).$$

The kernel of the integral operators in Eq.(2.4) has a high singularity and should be understood in the sense of some regularizations.

3. Numerical solution in 2D-case. Let us consider the plane problem of elasticity for homogeneous and isotropic elastic body. The body occupies a closed region Ω in 2D-space with the bourder Γ. The solution of this problem may be found in the form similar to Eq.(2.1)

$$\sigma_{ij}(x) = \int_\Gamma S_{ijkl}(x - x')n_k(x')b_l(x')\delta(\Gamma')dx'. \tag{3.1}$$

Here $\delta(\Gamma)$ is delta-function concentrated on the contour Γ, integration in this formula is spread over 2D-space. For the numerical solution of Eq.(2.3) let us chose a set of nodes $x^{(i)}$ on boundary Γ of the body with equal distances h between neigboring nodes.

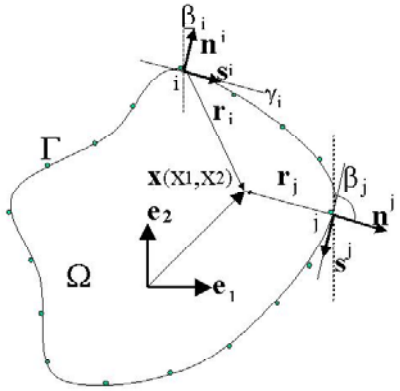


Figure 3.1: The local and global coordinate systems

Then let us change the potential (3.1) concentrated on Γ for the sum of potentials concentrated on the tangent lines γ^i at every node i (see Fig.(3.1)). Thus, density $n_k(x)b_l(x)\delta(\Gamma)$ of the potential in Eq.(3.1) is approximated by the equation

$$n_k(x)b_l(x)\delta(\Gamma) = \sum_i n_k^{(i)} b_l^{(i)} \varphi^i(x)\gamma^i(x), \quad \varphi^i(x) = \frac{1}{\sqrt{\pi D}} \exp\left(-\frac{|x-x^{(i)}|^2}{Dh^2}\right). \tag{3.2}$$

Here $\mathbf{n}^{(i)}$ is the external normal vector to Γ at the node $x^{(i)}$, $\gamma^i(x)$ is delta function concentrated in the tangent line γ^i to Γ in the node $x^{(i)}$, h is the distance between neighboring nodes, $D = 2$. The vectors $\mathbf{b}^{(i)}$ in every node should be found from the solution of the problem and are the main unknowns of the method. If we substitute Eq.(3.2) into Eq.(3.1) the latter is converted into the sum of potentials concentrated in the tangent lines γ^i

$$\begin{aligned} \sigma_{ij}(x) &= \int_\Gamma S_{ijkl}(x-x')n_k(x')b_l(x')\delta(\Gamma')dx' \approx \sum_i I_{ij}^{(i)}(x)\cdot b_j^{(i)}, \\ I_{ijl}^{(i)}(x) &= \int_{\gamma_i} S_{ijkl}(x-x')n_k(x')\varphi^s(x')\gamma^i(x')dx'. \end{aligned} \tag{3.3}$$

Let us introduce the local coordinate systems (s, z) connected with the nodes; $(\mathbf{s}^{(i)}, \mathbf{n}^{(i)})$ are the unit vectors of axis s directed along tangent line γ^i and of axis z directed along normal to Γ at the node $x^{(i)}$ (Fig.(3.1)). In this basis vector $\mathbf{b}^{(i)}$ in Eq.(3.1) has the form

$$\mathbf{b}^{(i)} = b_s^{(i)}\mathbf{s}^{(i)} + b_n^{(i)}\mathbf{n}^{(i)}. \tag{3.4}$$

After substituting Eq.(3.2) into Eq.(3.3) and calculating the integrals we go to the fol-

lowing expression of the tensor $\mathbf{I}^{(i)}$ in the local basis of the i-th node (see [4] for details)

$$\begin{aligned}
\mathbf{I}^{(i)}(s,z) &= -4\mu_0\kappa_0\left[J_{11}(s,z)\mathbf{t}_{11}^{(i)}+J_{12}(s,z)\mathbf{t}_{12}^{(i)}+J_{21}(s,z)\mathbf{t}_{21}^{(i)}+J_{22}(s,z)\mathbf{t}_{22}^{(i)}\right],\\
J_{11}(s,z) &= \frac{h}{4p^2}\,\mathrm{sign}(\zeta)\left[2\mathrm{sign}(\eta)j_1(|\zeta|,|\eta|)-\eta j_2(|\zeta|,|\eta|)\right],\\
J_{12}(s,z) &= \frac{h}{4p^2}\left[j_3(|\zeta|,|\eta|)-|\eta|j_4(|\zeta|,|\eta|)\right],\\
J_{21}(s,z) &= \frac{h}{4p^2}\eta j_2(|\zeta|,|\eta|),\quad J_{22}(x,y)=\frac{h}{4p^2}\left[j_3(|\zeta|,|\eta|)+|\eta|j_4(|\zeta|,|\eta|)\right],\\
\mathbf{t}_{11}^{(i)} &= \mathbf{s}^{(i)}\otimes\mathbf{s}^{(i)}\otimes\mathbf{s}^{(i)},\quad \mathbf{t}_{22}^{(i)}=\mathbf{n}^{(i)}\otimes\mathbf{n}^{(i)}\otimes\mathbf{n}^{(i)}\\
\mathbf{t}_{12}^{(i)} &= \mathbf{s}^{(i)}\otimes\mathbf{s}^{(i)}\otimes\mathbf{n}^{(i)}+\mathbf{s}^{(i)}\otimes\mathbf{n}^{(i)}\otimes\mathbf{s}^{(i)}+\mathbf{n}^{(i)}\otimes\mathbf{s}^{(i)}\otimes\mathbf{s}^{(i)},\\
\mathbf{t}_{21}^{(i)} &= \mathbf{s}^{(i)}\otimes\mathbf{n}^{(i)}\otimes\mathbf{n}^{(i)}+\mathbf{n}^{(i)}\otimes\mathbf{s}^{(i)}\otimes\mathbf{n}^{(i)}+\mathbf{n}^{(i)}\otimes\mathbf{n}^{(i)}\otimes\mathbf{s}^{(\mathbf{i})}.
\end{aligned}$$

Here $p^2=(h^2D)/4$, $\zeta=s/p$, $\eta=z/p$, $\kappa_0=(\lambda_0+\mu_0)/((\lambda_0+2\mu_0),\lambda_0,\mu_0$ are Lamé parameters of the material.

The four functions $j_i(\zeta,\eta)$ in these equations are connected with function Erf(ξ) (the error function) by the equations

$$j_1(\zeta,\eta)+ij_3(\zeta,\eta)=iF_1\left(\frac{\eta+i\zeta}{2}\right),\qquad j_2(\zeta,\eta)+ij_4(\zeta,\eta)=iF_2\left(\frac{\eta+i\zeta}{2}\right),\tag{3.5}$$

$$\begin{aligned}
F_1(z) &= \frac{1}{2\pi}\left[1-\sqrt{\pi}z\exp(z^2)\,(1-\ \mathrm{Erf}(z))\right],\\
F_2(z) &= \frac{1}{2\pi}\left[-z+\frac{\sqrt{\pi}}{2}(1+2z^2)\exp(z^2)\,(1-\ \mathrm{Erf}(z))\right].
\end{aligned}\tag{3.6}$$

The system of linear algebraic equations for the components $(b_s^{(i)},b_n^{(i)})$ of the vectors $\mathbf{b}^{(i)}$ in the local bases can be obtained from the boundary conditions (2.3) that will be satisfied in all the nodes (the collocation method). Let us introduce vector $\mathbf{X}$ of the unknowns that is connected with the components $b_s^{(m)},b_n^{(m)}$ by the relations

$$\begin{aligned}
\mathbf{X} &= \|X_j\|,\qquad j=1,2,3,...,2M,\\
X_{2m-1} &= b_s^{(m)},\qquad X_{2m}=b_n^{(m)},\qquad m=1,2,3,...,M
\end{aligned}\tag{3.7}$$

Here M is the total number of nodes. The vector-column $\mathbf{F}$ defines the forces that act in the nodes

$$\begin{aligned}
\mathbf{F} &= \|F_j\|,\qquad j=1,2,3,...,2M,\\
F_{2m-1} &= -\frac{f_s^{(m)}}{4\mu_0\kappa_0},\qquad F_{2m}=-\frac{f_n^{(m)}}{4\mu_0\kappa_0},\qquad m=1,2,3,...,M.
\end{aligned}\tag{3.8}$$

Here $f_s^{(m)}$, $f_n^{(m)}$ are the values of the forces applied at the nodes that are known from the boundary conditions.

The equation for the vector $\mathbf{X}$ follows from the boundary conditions at the nodes and takes the form

$$\sum_{j=1}^{2N}B_{ij}X_j=F_i,\qquad j=1,2,3,...,2M.\tag{3.9}$$

Here the components of the matrix $\mathbf{B}=\|B_{ij}\|$ are defined in [3] and expressed via the standard functions $j_1,...,j_4$ defined in Eqs.(3.5, 3.6). The computer time of calculation of these functions is very small.

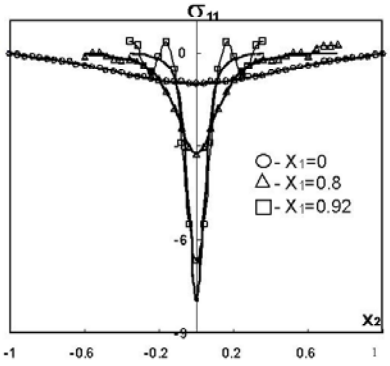


Figure 3.2: Distribution of stresses in a disk subjected with two concentrated forces.

Let us consider a numerical example. An elastic disk of unit radius, $R = 1$, subjected by two concentrated forces applied along its diameter. The distributions of normal stress $\sigma(x_1, x_2)$ in various intersections orthogonal to the direction of the force application are presented in Fig.3.2. Solid lines are exact solutions; dashed lines correspond to 60 nodes homogeneously distributed along the boundary. It is seen that the error of the numerical solution is essential only in a small vicinity of the points of application of the forces.

4. The integral equation of the problem of electromagnetic wave diffraction on a perfectly conducting screen. Let a monochromatic electromagnetic wave of frequency ω propagate through a homogeneous and isotropic medium, and Ω be a smooth, perfectly conducting surface embedded in this medium. The electric field $\mathbf{E}(x)$ in the medium with such a surface may be presented in the form

$$\begin{aligned} \mathbf{E}(x) &= \mathbf{E}^0(x) + \mathbf{E}^s(x), \quad \mathbf{E}^s(x) = -i\frac{4\pi c}{k_0}\int_\Omega \mathbf{K}(x-x')\cdot\mathbf{J}(x')d\Omega', \\ \mathbf{K}(x) &= \bigtriangledown\otimes\bigtriangledown g(x) + k_0^2 g(x)\mathbf{1}. \end{aligned} \tag{4.1}$$

Here $\mathbf{1}$ is the second-rank unit tensor, c is the wave velocity, $k_0 = \omega/c$. $\mathbf{E}^0(x)$ is an incident field that is assumed to be a plane monochromatic wave: $\mathbf{E}^0(x) = \mathbf{e}\exp(-i\mathbf{k}_0\cdot x)$, $\mathbf{k}_0 = k_0\mathbf{m}$, $|\mathbf{m}| = 1$, $\mathbf{k}_0$ is the wave vector, and $\mathbf{e}$ is the polarization vector of this wave. The kernel $\mathbf{K}(x)$ of the integral operator in Eq.(4.1) is the second derivative of Green function $g(x)$ of Helmholtz's operator, and in the 3D-case $g(x)$ takes the form $g(x) = (e^{-ik_0 r})/(4\pi r)$, $r = |x|$.

The density $\mathbf{J}(x)$ of the potential in Eq.(4.1) is the surface current generated on Ω by incident field $\mathbf{E}^0(x)$. Vector $\mathbf{J}(x)$ belongs to Ω and satisfies the integral equation

$$i\frac{4\pi c}{k_0}\int_\Omega \mathbf{U}(x,x')\cdot\mathbf{J}(x')d\Omega' = \theta(x)\cdot\mathbf{E}^0(x), \qquad \theta(x) = \mathbf{1} - \mathbf{n}(x)\otimes\mathbf{n}(x), \qquad x\in\Omega. \tag{4.2}$$

$$\mathbf{U}(x,x') = \theta(x)\cdot\mathbf{K}(x-x')\cdot\theta(x').$$

The integral on the left hand side of Eq.(4.2) has a strong singularity and should be understood in terms of some regularization (see [3]).

5. Numerical solution of the diffraction problems. Integral equation (4.2) may be presented in the form

$$i\frac{4\pi c}{k_0}\int \mathbf{U}(x,x')\cdot\mathbf{J}(x')\Omega(x')dx' = \theta(x)\cdot\mathbf{E}^0(x), \qquad x\in\Omega, \tag{5.1}$$

where $\Omega(x)$ is delta-function concentrated on the surface Ω and integration in this equation is spread over 3D-space. Let us cover the scattering surface by a set of nodes $x^{(i)}$ $(i = 1, 2, ..., M)$ with approximately the same distances between neighboring nodes, and ω_i be the tangent plane to Ω at the i-th node. For the application of the BPM the actual current distribution on Ω is changed for the following sum

$$\mathbf{J}(x)\Omega(x) \approx \sum_i \mathbf{J}^{(i)}\varphi_i(x)\omega_i(x), \quad \varphi_i(x) = \frac{1}{\pi D}\exp\left(-\frac{|x-x^{(i)}|}{Dh^2}\right). \tag{5.2}$$

Here $\omega_i(x)$ is delta-function concentrated in the plane ω_i and $\mathbf{J}^{(i)}$ is the vector of this plane. The approximation of the scattered field $\mathbf{E}^s(x)$ in Eq.(4.1) takes the form

$$\mathbf{E}^s(x) = -i\frac{4\pi c}{k_0}\int \mathbf{K}(x-x')\cdot\mathbf{J}(x')\Omega(x')dx' \approx -i4\pi c\sum_i \mathbf{I}^{(i)}(x)\cdot\mathbf{J}^{(i)}, \tag{5.3}$$

$$\mathbf{I}^{(i)}(x) = \frac{1}{k_0}\int \mathbf{K}(x-x')\varphi_i(x')\omega_i(x')dx'. \tag{5.4}$$

Let us introduce a local Cartesian basis $\left(\mathbf{e}_1^{(i)}, \mathbf{e}_2^{(i)}, \mathbf{e}_3^{(i)}\right)$ with the origin at the i-th node (the unit vector $\mathbf{e}_3^{(i)}$ coincides with the normal $\mathbf{n}^{(i)}$ to ω_i at point $x^{(i)}$). In this local coordinate system the scalar product $\mathbf{I}^{(i)}(x)\cdot\mathbf{J}^{(i)}$ in Eq.(5.3) in the local basis of the i-th node takes the form

$$\mathbf{I}^{(i)}(x)\cdot\mathbf{J}^{(i)} = \frac{4}{D\kappa_0}\left[F_1(\kappa_0,\eta,\zeta)\mathbf{1} + 2F_2(\kappa_0,\eta,\zeta)\mu\otimes\mu + \right.$$
$$\left. +2\ \mathrm{sign}(\zeta)F_3(\kappa_0,\eta,\zeta)\mathbf{n}\otimes\mu\right]\cdot\mathbf{J}^{(i)}, \qquad \mathbf{n} = \mathbf{e}_3^{(i)}, \quad \kappa_0 = k_0 h_1,$$

$$\eta = \frac{1}{h_1}\left(x_1^2 + x_2^2\right)^{1/2}, \quad \zeta = \frac{x_3}{h_1}, \quad \mu = \frac{x_1\mathbf{e}_1^{(i)} + x_2\mathbf{e}_2^{(i)}}{h_1\eta}, \quad h_1 = \frac{D^{1/2}}{2}h.$$

Here the three functions $F_i(\kappa_0,\eta,\zeta)$ are the following one dimensional integrals

$$\begin{aligned}
F_1(\kappa_0,\eta,\zeta) &= \frac{1}{8\pi}\int_0^\infty \left[(2\kappa_0^2-\kappa^2)J_0(\kappa\eta) - \right. \\
&\quad \left. -\kappa^2 J_2(\kappa\eta)\right]\exp\left[-k^2 - |\zeta|\beta(\kappa,\kappa_0)\right]\frac{\kappa d\kappa}{\beta(\kappa,\kappa_0)}, \\
F_2(\kappa_0,\eta,\zeta) &= \frac{1}{8\pi}\int_0^\infty J_2(\kappa\eta)\exp\left[-k^2 - |\zeta|\beta(\kappa,\kappa_0)\right]\frac{\kappa^3 d\kappa}{\beta(\kappa,\kappa_0)}, \\
F_3(\kappa_0,\eta,\zeta) &= \frac{1}{8\pi}\int_0^\infty J_1(\kappa\eta)\exp\left[-k^2 - |\zeta|\beta(\kappa,\kappa_0)\right]\kappa^2 d\kappa,
\end{aligned} \tag{5.5}$$

where $\beta(\kappa,\kappa_0) = \sqrt{\kappa^2-\kappa_0^2}$ if $\kappa > \kappa_0$, and $\beta(\kappa,\kappa_0) = i\sqrt{\kappa_0^2-\kappa^2}$ if $\kappa < \kappa_0$; $J_n(z)$ is the Bessel function of order n. For small values of the arguments η, ζ $(\rho = \left(\eta^2+\zeta^2\right)^{1/2} \le 10)$ these integrals may be simply tabulated and kept in the computer memory. For $\rho > 10$ these integrals have simple asymptotic expressions presented in [2].

Using approximation (5.3) one can reduce the integral equation (5.1) to the system of linear algebraic equations which unknowns are the components of the vector $\mathbf{J}$ in the local bases connected with the nodes. This linear system may be written in the canonical form

$$BX = F, \tag{5.6}$$

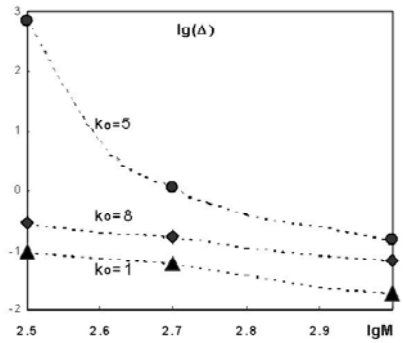


Figure 5.1: Error of the numerical solution for a sphere.

were the elements of the square matrix B of dimension $(2M \times 2M)$ are defined via standard functions F_1, F_2, F_3 in Eq.(5.5) and the components of vectors X and F are

$$\begin{aligned} X_i &= J_1^{(i)}, \quad i \le M; \quad X_i = J_2^{(i-M)}, \quad i > M; \\ F_i &= E_1^0(x^{(i)}), \quad i \le M; \quad F_i = E_2^0(x^{(i-M)}), \quad i > M. \end{aligned}$$

Here M is the total number of nodes. B in Eq.(5.6) is a dense matrix with maximal terms concentrated near the main diagonal. For a homogeneous distribution of the nodes on Ω matrix B is symmetric with the same elements in the main diagonal.

Let us consider a spherical surface Ω of unit radius ($a = 1$) when an analytical solution of the problem may be constructed. For the application of the BPM, a homogeneous set of nodes on Ω was generated by the algorithm described in [2]. In Fig.5.1 the dependences of relative error Δ of the numerical solutions on the number of surface nodes M are presented for $k_0 a = 1; 5; 8$.

$$\Delta = \frac{\int_\Omega (|\mathbf{J}_*| - |\mathbf{J}|)^2 d\Omega}{\int_\Omega |\mathbf{J}|^2 d\Omega}. \tag{5.7}$$

Here $\mathbf{J}_*$ is a numerical solution of Eq.(5.6), $\mathbf{J}$ is an exact current distribution.

6. Conclusion. The use of Gaussian approximating functions proposed in [5, 6] for the solution of boundary integral equations has two main advantages: the simplicity of preparation of the initial data (the coordinates of surface nodes and surface orientations at the nodes), and a short time for the construction of the matrix of the linear system obtained after the discretization of the problem. The accuracy of the method depends on the density of surface nodes. The method may be applied to a wide class of the problems of mathematical physics that are reduced to surface pseudo-differential equations. In particular, the problems of electrostatics, static elasticity and elasto-plasticity, the problems of elastic wave diffraction on inclusions and cracks [3], etc., may be successfully solved with the help of the developed version of BPM.

REFERENCES

[1] P. K. Banerjee and R. Batterfield. *Boundary Elements Methods in Engineering Sciences*. McGraw-Hill, 1981.

[2] S. K. Kanaun. A numerical method for the solution of electromagnetic wave diffraction problems on perfectly conducting screens. *Journal of Computational Physics*, 196(1):170–195, 2002.

[3] S. K. Kanaun and V. Romero. Boundary point method in the dynamic problems of elasticity for plane areas with cracks. *International Journal of Fracture*, 111(1):L3–L8, 2001.

[4] S. K. Kanaun, V. Romero, and J. Bernal. A new numerical method for the solution of the second boundary value problem of elasticity for bodies with cracks. *Revista Mexicana de Física*, 47(4):309–323, 2001.

[5] V. Maz'ya. Approximate approximations. In J. R. Whitman, editor, *The mathematics of Finite Elements and Applications, Highlights 1993*, pages 77–98. Wiley, Chichester, 1994.

[6] V. Maz'ya and G. Shmidt. On approximate approximations using Gaussian kernels. *IMA J. Numer. Anal.*, 16(1):13–29, 1996.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

48. V–cycle Multigrid Convergence for Cell Centered Finite Difference Method, 3-D case.

D. Y. Kwak[1]

1. Introduction. In this paper, we study a multigrid algorithm for cell centered finite difference method for elliptic problems in three dimensions.

Cell centered finite difference methods are very popular among engineering circle working on various fluid computations such as oil reservoir simulation, underground water flow, or steady Euler equations, etc. It seems mainly due to the conservation property and simplicity of the scheme. On the other hand, as a solution process of the corresponding linear system, multigrid methods have been known fast for many class of problems[1],[7], [9],[4], [5],[2]. The performance of multigrid algorithms for two dimensional cell-centered finite difference method have been investigated in [10],[6] and W–cycle convergence has been analyzed in [3]. Recently V–cycle convergence has been shown with certain weighted prolongation operator[8]. This paper is a continuation of [8] dealing with three dimensional aspect of multigrid algorithm for cell-centered finite difference methods.

One of the main ingredient of multigrid algorithms in the nonstandard discretization is the design of prolongation operators between two consecutive levels, since for the cell centered finite difference case, the natural injection increases the energy norm even in two dimensional problems as shown in [3, 8]. Hence we consider a certain weighted prolongation and show its energy norm is bounded by one. Another natural operator is trilinear based operator. In this case, we also show the energy norm is less than equal to 1. Finally, we consider prolongation with different weight. This is motivated by the geometric configuration: when a box element is subdivided by 8 subboxes, one of the subbox shares three faces with its mother box, while it shares just one face with three neighboring box, thus the weights $\{3, 1, 1, 1\}$. In this last case, one can only show the energy norm is bounded by $\sqrt{10/9}$, but the multigrid performance is better than any other operator(either as an iterative solver or as a preconditioner). The rest of the paper is organized as follows. In section 2, we derive cell-centered FDM for a model 3-dimensional problem through the use of Raviart-Thomas-Nedelec element for the mixed formulation. In section 3, we describe the multigrid algorithm and some convergence theory. In section 4, we consider various prolongation operators together with their energy norm estimates. Finally in section 5, we present numerical experiments.

2. Derivation of Cell Centered FDM from RTN. Consider a model problem

$$\begin{aligned} -\nabla \cdot \mathcal{K}\nabla p &= f \text{ in } \Omega \\ p &= 0 \text{ on } \partial\Omega \end{aligned} \tag{2.1}$$

where Ω is a unit cube, $\mathcal{K}$ is a diagonal tensor whose entries are piecewise smooth. Let $h := h_k = 2^{-k}$ for some positive integer k. Assuming the domain has been subdivided by axis parallel planes into small cubes of equal size h with index (i, j, l), we consider the Raviart-Thomas-Nedelec (RTN) mixed finite element space. Let

$$\vec{V}_h = \{\mathbf{u}_h = (a_1 + a_2x, b_1 + b_2y, c_1 + c_2z) \text{ on each element } \} \cap H(\operatorname{div}\Omega) \tag{2.2}$$

$$L_h = \{p_h : \text{ piecewise constant on each element}\}. \tag{2.3}$$

The RTN mixed method is to find $(\mathbf{u}_h, p_h) \in \vec{V}_h \times M_h$ such that

$$(\mathcal{K}^{-1}\mathbf{u}_h, \mathbf{v}) - (\operatorname{div}\mathbf{v}, p_h) = 0, \quad \mathbf{v} \in \vec{V}_h \tag{2.4}$$

$$(\operatorname{div}\mathbf{u}_h, q) = (f, q), \quad q \in L_h. \tag{2.5}$$

[1]Korea Advanced Institute of Science and Technology, Taejon, Korea, 305-701, Partially supported by Korea Science and Engineering Foundation. dykwak@math.kaist.ac.kr

Let $\mathbf{v}$ be a test function in $\vec{V}_h$ whose only nonzero component, the x component, is one at a vertex $i+1/2$ and zero at all others. One uses the trapezoidal rule to evaluate first integral. Then we get

$$u^1_{i+1/2,j,l} Fac = h^2(p_{i,j,l} - p_{i+1,j,l}), \tag{2.6}$$

where

$$Fac = \frac{h}{2}\left[\int_{j-\frac{1}{2}}^{j+\frac{1}{2}}\int_{l-\frac{1}{2}}^{l+\frac{1}{2}} \mathcal{K}_L^{-1}(x_{i+\frac{1}{2}}, y, z)dydz + \int_{j-\frac{1}{2}}^{j+\frac{1}{2}}\int_{l-\frac{1}{2}}^{l+\frac{1}{2}} \mathcal{K}_R^{-1}(x_{i+\frac{1}{2}}, y, z)dydz\right].$$

Similarly we integrate along y-directional and z-directional volumes using y, z directional test functions to get difference equations along the y and z axes. The second equation of mixed formulations reads:

$$h^2(u^1_{i+\frac{1}{2},j,l} - u^1_{i-\frac{1}{2},j,l} + u^2_{i,j+\frac{1}{2},l} - u^2_{i,j-\frac{1}{2},l} + u^3_{i,j,l+\frac{1}{2}} - u^3_{i,j,l-\frac{1}{2}}) = h^3 f_{i,j,l}, \tag{2.7}$$

where we assumed f is piecewise constant for simplicity. By substituting the expressions for $u^1_{i+1/2,j,l}$ etc, if we denote the integral of $\mathcal{K}_L^{-1}$ simply as $\mathcal{K}_L^{-1}$, we have

$$2\left[\frac{-p_{i-1,j,l} - p_{i+1,j,l} - p_{i,j-1,l} + 6p_{i,j,l} - p_{i,j+1,l} - p_{i,j,l-1} - p_{i,j,l+1}}{\mathcal{K}_L^{-1} + \mathcal{K}_R^{-1}}\right] = f_{i,j,l}h^2. \tag{2.8}$$

When $\mathcal{K} = 1$ the stencil for interior is(without h-factor) $-1,-1,-1,6,-1,-1,-1$ while on the boundary face $-1,-1,-1,\ 7,-1,-1,0$ and on the boundary edge $0,-1,-1,\ 8,-1,-1,0$ and on the corner $-1,-1,-1,\ 9,0,0,0$. This is the cell-centered finite difference method. If we denote by M_k the space of functions which are piecewise constant on each cell, the problem can be viewed as seeking a solution $x \in M_k$ satisfying an algebraic equation of the form

$$A_k x = b, \tag{2.9}$$

where x is identified as the vector representation of p_h.

2.1. Multgrid Method. Now we describe a V–cycle multigrid algorithm (with one smoothing R_k, e.g, Gauss-Seidel) for solving (2.9) for $k = J$. First consider the sequence of spaces

$$M_1, \cdots, M_J.$$

One can view this sequence of space nested with obvious injection. But as we shall see other types of operator to be considered in this paper work better for multigrid.

ALGORITHM. If $k = 1$, set $B_1 b = A_1^{-1} b$. Otherwise define B_k recursively as follows:

1. Pre-smooth
$$x^1_k := R_k b.$$
2. Set
$$q = B_{k-1} P^0_{k-1}(b - A_k x^1_k).$$
3. Correct
$$x^2_k := x^1_k + I_k q.$$
4. Post-smooth
$$B_k b := x^2 + R^t_k(b - A_k x^2_k).$$

For the convergence analysis, we need two conditions to verify: One is the so-called regularity and approximation property: There exist constants $\alpha \in (0,1]$ and C_α such that, for all $k = 1, \cdots, J$,

$$A_k((I - I_k P_{k-1})u, u) \le C_\alpha^2 \left(\frac{\|A_k u\|^2}{\lambda_k} \right)^\alpha A_k(u,u)^{1-\alpha}, \quad \forall u \in M_k. \tag{2.10}$$

Here, λ_k is the largest eigenvalue of A_k, and P_{k-1} is the elliptic projection defined by

$$A_{k-1}(P_{k-1}u, v) = A_k(u, I_{k-1}^k v), \quad \forall u \in V_k, v \in M_{k-1}. \tag{2.11}$$

The next is

$$A_k(I_k v, I_k v) \le C_I A_{k-1}(v,v), \quad \forall v \in V_{k-1}. \tag{2.12}$$

With these verified one can prove the following result[5].

Theorem 2.1 *We have*
1. If $C_I \le 1$, then V–cycle multigrid algorithm satisfies

$$0 \le A_k(E_k u, u) \le \delta_k A_k(u,u), \quad \forall u \in M_k, \tag{2.13}$$

where $E_k = I - B_k A_k$ and $\delta_k = \frac{Ck}{Ck+1}$.
2. If $C_I \le 1 + Ch_k$, then B_k is a good preconditioner in the sense that

$$\eta_0 A_k(u,u) \le A_k(B_k A_k u, u) \le \eta_1 A_k(u,u), \quad \forall u \in M_k, \tag{2.14}$$

where η_1 is independent of k and $\eta_0 \le 1 - \delta_k$.

3. Energy norm estimate of various prolongations. For all the prolongation operators to be considered below, this regularity and approximation property holds(see [8] for details). Hence we concentrate (2.12) only. To make things easier we summarize 2-D result briefly first and extend it to 3-D. Referring to figure 2.1, we shall use the notation (i,j) to denote a coarse grid cell center, while we use (I_1, J_1) etc, to denote the fine grid center obtained by halving the coarse cell. We define the prolongation operator $I_k : M_{k-1} \to M_k$ as follows: With any positive number w let

$$(I_k v)_{I-1,J-1} = \frac{1}{w}((w-2)v_{i,j} + v_{i-1,j} + v_{i,j-1}) \tag{3.1}$$

$$(I_k v)_{I-1,J} = \frac{1}{w}((w-2)v_{i,j} + v_{i,j+1} + v_{i-1,j}) \tag{3.2}$$

$$(I_k v)_{I,J-1} = \frac{1}{w}((w-2)v_{i,j} + v_{i+1,j} + v_{i,j-1}) \tag{3.3}$$

$$(I_k v)_{I,J} = \frac{1}{w}((w-2)v_{i,j} + v_{i,j+1} + v_{i+1,j}) \tag{3.4}$$

One can show that in this case (2.12) holds with $C_I = (2(w-2)^2+8)/w^2$ whose minimum is obtained when $w = 4$. Thus we have weight $\{1/2, 1/4, 1/4\}$. Hence the analysis in [8] can be carried out to show that symmetric V–cycle with one smoothing yields a convergence factor $\delta < 1$. For 3D, the situation is different. The weight has to be changed to get suitable operator. We use similar notations as in 2-D. Fix a box element (i,j,l) in $k-1$ level and divided it by 8 axi-parallel subboxes, denoted by $(I,J,L), (I_1,J,L), (I,J_1,L), (I_1,J_1,L)$ and $(I,J,L_1), (I_1,J,L_1), (I,J_1,L_1), (I_1,J_1,L_1)$, etc. It is natural to define $I_k v$ on each subbox as a linear combination of values of v on (i,j,l) and its adjacent boxes. Referring to figure

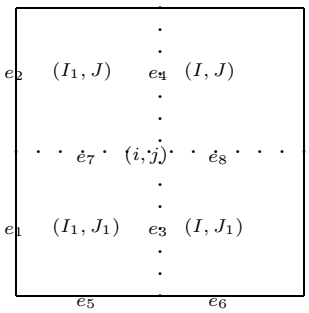


Figure 3.1: Numbering of (i, j) element and its subdivision

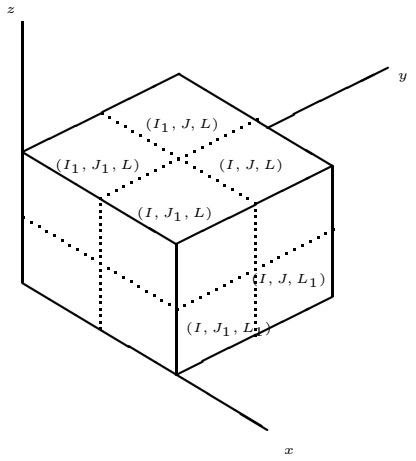


Figure 3.2: A box element and its subdivision

2.1 and 2.2, let $u^U_{I,J,L}, u^U_{I_1,J,L}$, etc., denote $I_k v$ on the upper part of the box (i, j, l), and let $u^L_{I,J,L}, u^L_{I_1,J,L}$ etc., denote its lower part. We define

$$u^U_{I_1,J_1,L} = \frac{1}{w}((w-3)v_{i,j,l} + v_{i-1,j,l} + v_{i,j-1,l} + v_{i,j,l+1}) \tag{3.5}$$

$$u^U_{I_1,J,L} = \frac{1}{w}((w-3)v_{i,j,l} + v_{i,j+1,l} + v_{i-1,j,l} + v_{i,j,l+1}) \tag{3.6}$$

$$u^U_{I,J_1,L} = \frac{1}{w}((w-3)v_{i,j,l} + v_{i+1,j,l} + v_{i,j-1,l} + v_{i,j,l+1}) \tag{3.7}$$

$$u^U_{I,J,L} = \frac{1}{w}((w-3)v_{i,j,l} + v_{i,j+1,l} + v_{i+1,j,l} + v_{i,j,l+1}) \tag{3.8}$$

and u^L are defined similarly with $l+1$ replaced by $l-1$.

This choice of weight reflects that the prolongation operator must have a certain approximation property, i.e, $\|I_k v - v\| \le Ch\|v\|_{1,h}$ for all piecewise constant functions. Here $\|\cdot\|_{1,h}$ denotes the discrete energy norm $A_k(v, v)^{1/2}$. By considering the differences between two

cell centers, it is easy to see that for $v \in M_{k-1}$,

$$\begin{aligned}(A_{k-1}v, v)_{k-1} &= -h_{k-1}\sum_{i,j,l} v_{i,j,l}\big[(v_{i,j,l+1} - v_{i,j,l}) + (v_{i,j,l-1} - v_{i,j,l}) \\ &\quad +(v_{i,j+1,l} - v_{i,j,l}) + (v_{i,j-1,l} - v_{i,j,l}) \\ &\quad +(v_{i+1,j,l} - v_{i,j,l}) + (v_{i-1,j,l} - v_{i,j,l})\big] \\ &= h_{k-1}\sum_{i,j,l}(v_{i,j,l} - v_{i-1,j,l})^2 + (v_{i,j,l} - v_{i,j-1,l})^2 + (v_{i,j,l} - v_{i,j,l-1})^2.\end{aligned}$$

Let $u = I_k v$. Then

$$(A_k u, u)_k = h_k \sum_{i,j,l}(D_i^2 + D_j^2 + D_l^2), \tag{3.9}$$

where D_i, D_j and D_l are the differences along the x, y, z directions respectively, i.e,

$$D_i = (u_{i,j,l} - u_{i-1,j,l}), \quad D_j = (u_{i,j,l} - u_{i,j-1,l}), \quad D_l = (u_{i,j,l} - u_{i,j,l-1}).$$

First fix L and consider square differences along the x direction of the upper part of subdivisions. Across e_1, the square is $(u_{I_1,J_1,L} - u_{I_2,J_1,L})^2$. Similarly, across e_2, the square difference is $(u_{I_1,J_2,L} - u_{I_2,J_2,L})^2$. If we let E_i denote the contribution from edge e_i, then we see that, ignoring the $\frac{1}{w^2}$ factor,

$$\begin{aligned}E_1 &= \big[(w-3)v_{i,j,l} + v_{i-1,j,l} + v_{i,j-1,l} + v_{i,j,l+1} \\ &\quad -((w-3)v_{i-1,j,l} + v_{i,j,l} + v_{i-1,j-1,l} + v_{i-1,j,l+1})\big]^2 \\ &= \big[(w-4)(v_{i,j,l} - v_{i-1,j,l}) + (v_{i,j-1,l} - v_{i-1,j-1,l}) + (v_{i,j,l+1} - v_{i-1,j,l+1})\big]^2 \\ &\le (w-2)[(w-4)(v_{i,j,l} - v_{i-1,j,l})^2 + (v_{i,j-1,l} - v_{i-1,j-1,l})^2 \\ &\quad +(v_{i,j,l+1} - v_{i-1,j,l+1})^2]\end{aligned}$$

where general Cauchy-Schwarz inequality

$$(\sum w_i\alpha_i)^2 \le (\sum w_i)(\sum w_i\alpha_i^2)$$

has been used. Similarly, the contributions from edges $e_2, \cdots, e_8$ are estimated.

$$\begin{aligned}E_2 &\le (w-2)[(w-4)(v_{i,j,l} - v_{i-1,j,l})^2 + (v_{i,j+1,l} - v_{i-1,j+1,l})^2 \\ &\quad +(v_{i,j,l+1} - v_{i-1,j,l+1})^2] \\ E_3 &= \big[(w-3)v_{i,j,l} + v_{i+1,j,l} + v_{i,j-1,l} + v_{i,j,l+1} \\ &\quad -((w-3)v_{i,j,l} + v_{i-1,j,l} + v_{i,j-1,l} + v_{i,j,l+1})\big]^2 \\ &\le 2(v_{i+1,j,l} - v_{i,j,l})^2 + 2(v_{i,j,l} - v_{i-1,j,l})^2 \\ E_4 &\le 2(v_{i+1,j,l} - v_{i,j,l})^2 + 2(v_{i,j,l} - v_{i-1,j,l})^2.\end{aligned}$$

The contribution E_5, E_6 are obtained from E_1, E_2 by interchanging the role of i, j. Thus

$$\begin{aligned}E_5 &\le (w-2)[(w-4)(v_{i,j,l} - v_{i,j-1,l})^2 + (v_{i-1,j,l} - v_{i-1,j-1,l})^2 \\ &\quad +(v_{i,j,l+1} - v_{i,j-1,l+1})^2]\end{aligned}$$

$$\begin{aligned}E_6 &\le (w-2)[(w-4)(v_{i,j,l} - v_{i,j-1,l})^2 + (v_{i+1,j,l} - v_{i+1,j-1,l})^2 \\ &\quad +(v_{i,j,l+1} - v_{i,j-1,l+1})^2]\end{aligned}$$

Also, E_7, E_8 are obtained from E_3, E_4 by interchanging the role of i, j. Thus

$$E_7 \leq 2(v_{i,j+1,l} - v_{i,j,l})^2 + 2(v_{i,j,l} - v_{i,j-1,l})^2 \tag{3.10}$$
$$E_8 \leq 2(v_{i,j+1,l} - v_{i,j,l})^2 + 2(v_{i,j,l} - v_{i,j-1,l})^2. \tag{3.11}$$

Now let us count the terms of the form $(v_{i,j,l} - v_{i-1,j,l})^2$. From E_1, we see that the coefficient is $(w-2)(w-4)$, while E_1 contributes $w-2$ to the neighboring boxes ($l+1$ and $j-1$) respectively. Thus the same amount come from those boxes. All together, the contribution to $(v_{i,j,l} - v_{i-1,j,l})^2$ is $(w-2)(w-4)+2(w-2) = (w-2)^2$. By the same reasoning the contributions from e_2, e_3 and e_4 are $(w-2)^2$, 4, and 4. The lower part of the subdivision has the same form except $l+1$ is replaced by $l-1$. Thus the sum of the coefficient for $(v_{i,j,l} - v_{i-1,j,l})^2$ is $2\frac{2(w-2)^2+8}{w^2}$. The same reasoning shows that the coefficients for $(v_{i,j+1,l} - v_{i,j,l})^2$ and $(v_{i,j,l+1} - v_{i,j,l})^2$ are shown to be the same. It is an elementary calculus to see $2\frac{2(w-2)^2+8}{w^2}$ has minimum 2 when $w = 4$. Considering h_k factor in A_k form, we have proved (2.12) with $C_I = 1$. Thus we obtain $\{1, 1, 1, 1\}$ as a good choice for weight.

3.1. Trilinear case. The prolongation is defined as (with $w = 64$)

$$u_{I_1,J_1,L} = \frac{3}{w}(9v_{i,j,l} + 3v_{i-1,j,l} + 3v_{i,j-1,l} + v_{i-1,j-1,l}) \tag{3.12}$$
$$+ \frac{1}{w}(9v_{i,j,l+1} + 3v_{i-1,j,l+1} + 3v_{i,j-1,l+1} + v_{i-1,j-1,l+1}) \tag{3.13}$$

where other terms are similarly defined. By the same argument as above, we can show (2.12) holds with $C_I = 1$ for trilinear prolongation also. Hence the V-cycle convergence theory follows.

3.2. Different weight. Finally consider weight $\{3, 1, 1, 1\}$. We can follow the same line of argument but we could only show $C_I \leq 10/9$. However, the numerical result shows this one performs best. This phenomenon is subject to further investigation.

4. Numerical experiment. We set $\mathcal{K} = 1$ and compare all three prolongation with natural injection whose weight can be viewed as $\{1, 0, 0, 0\}$. All three weighted operators perform well and the reduction factor seems to be independent of the number of levels. We note that the weight $\{3, 1, 1, 1\}$ works best. As a reference, we give numerical estimate on the size of prolongation operators in Table 5.

h_J	λ_{min}	λ_{max}	K	δ
1/8	0.734	1.283	1.749	0.279
1/16	0.702	1.475	2.102	0.467
1/32	0.684	1.678	2.452	0.666
1/64	0.673	1.880	2.794	0.863

Table 1. Natural injection $\{1, 0, 0, 0\}$

h_J	λ_{min}	λ_{max}	K	δ
1/8	0.615	0.999	1.626	0.378
1/16	0.581	0.999	1.721	0.410
1/32	0.556	0.999	1.797	0.432
1/64	0.536	0.999	1.864	0.450

Table 2. Weight $\{1, 1, 1, 1\}$

h_J	λ_{min}	λ_{max}	K	δ
1/8	0.684	0.999	1.460	0.308
1/16	0.661	0.999	1.514	0.326
1/32	0.644	0.999	1.553	0.333
1/64	0.634	0.999	1.578	0.336

Table 3. $\{3, 1, 1, 1\}$

h_J	λ_{min}	λ_{max}	K	δ
1/8	0.641	0.999	1.560	0.353
1/16	0.616	0.999	1.624	0.374
1/32	0.599	0.999	1.669	0.383
1/64	0.589	0.999	1.698	0.389

Table 4. Trilinear

$\{1, 0, 0, 0\}$	$\{1, 1, 1, 1\}$	$\{3, 1, 1, 1\}$	Trilinear
2	0.59	0.67	0.49
2	0.65	0.78	0.60
2	0.69	0.84	0.66
2	0.71	0.86	0.69

Table 5. Estimate of energy of I_k

Concluding remarks: We proved V-cycle multigrid convergence for the cell-centered FDM for 3-dimensional problem for two kinds of weighted prolongation operators. A third weight, $\{3, 1, 1, 1\}$, works slightly better even though the energy norm seems larger than the other two. Thus, we guess that an operator with smaller energy norm (although they guarantee convergence) does not always work better.

REFERENCES

[1] R. Bank and T. Dupont. An optimal order process for solving finite element equations. *Math. Comp.*, 36:35–51, 1981.

[2] J. Bramble. *Multigrid methods*, volume 294 of *Pitman Research Notes in Mathematics Series*. Longman Scientific, 1993.

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

[4] J. Bramble and J. Pasciak. New convergence estimates for multigrid algorithms. *Math. Comp.*, 49:311–329, 1987.

[5] J. Bramble, J. Pasciak, and J. Xu. The analysis of multigrid algorithms with non-nested spaces or non-inherited quadratic forms. *Math. Comp.*, 56:1–34, 1991.

[6] R. Ewing and J. Shen. A multigrid algorithm for the cell-centered finite difference scheme. In *The Proceedings of 6-th Copper Mountain Conference on Multigrid Methods*. NASA Conference Publication, April 1993.

[7] W. Hackbusch. *Multi-Grid Methods and Applications*. Springer-Verlag, Berlin – Heidelberg – New York, 1985.

[8] D. Y. Kwak. V-cycle multigrid for cell centered finite. *SIAM J. Sci. Comput.*, 21:552–564, 1999.

[9] S. McCormick. *Multigrid methods*. SIAM, 1987.

[10] P. Wesseling. Cell centered multigrid for interface problems. *J. Comp. Physics*, 79:85–91, 1988.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

49. Asynchronous domain decomposition methods for solving continuous casting problem

E. Laitinen[1], A. Lapin[2], J. Pieskä[3]

1. Introduction. The general idea of the Schwarz alternating methods is to solve the boundary value problem restricted to each subdomain, using as the boundary conditions the function values of the approximative solution of the neighboring subdomains. One of the advantages of the additive Schwarz is that the solutions in the subdomains can be handled by the different processors of a parallel computer. However, due to the mutual waits among the processors when a synchronous method is applied, it leads to a substantial loss of computing time. To exploit the asynchronous parallel computing capacity of a multiprocessor system, we propose and study theoretically and numerically the asynchronous algorithms [1] for solving nonlinear finite-dimensional problem.

2. Continuous casting problem. A continuous casting problem can be stated mathematically as follows. Let $\Omega = \{0 < x_1 < L_{x_1}, 0 < x_2 < L_{x_2}\}$ be the rectangular domain with the boundary $\Gamma = \partial\Omega$ consisting of two parts: $\Gamma_1 = \{x \in \partial\Omega : x_2 = 0 \vee x_2 = L_{x_2}\}$ and $\Gamma_2 = \{x \in \partial\Omega \setminus \Gamma_1\}$. We assume that the domain $\Omega \subset \mathbb{R}^2$ is occupied by a thermodynamically homogeneous and isotropic steel. We denote by $H(x,t)$ the enthalpy related to the unit mass and by $u(x,t)$ the temperature for $(x,t) \in \Omega\times]0,T[$. We have a constitutive law

$$H = H(u) = \rho \int_0^u c(\Theta)d\Theta + \rho L(1 - f_s(u)) \text{ in } \Omega\times]0,T[,$$

where ρ is density, $c(u)$ is specific heat, L is latent heat and $f_s(u)$ is solid fraction. For a steel casting process the graph $H(u)$ is an increasing function $\mathbb{R} \to \mathbb{R}$, involving nearly vertical segments, which correspond to a phase transition states, namely, for $u \in [T_L, T_S]$ where $0 < T_L < T_S$ are melting and solidification temperatures. When a copper casting problem is studied, the graph $H(u)$ has a vertical segment for $u = T_L = T_S$. We denote by $H(u),\ u \in \mathbb{R}$, a maximal monotone, generally multivalued, graph.

We also suppose, that the graph $H(u)$ is uniformly monotone: there exists a positive constant α such that

$$(\gamma_1 - \gamma_2, u_1 - u_2) \geq \alpha(u_1 - u_2, u_1 - u_2) \forall u_1, u_2 \forall \gamma_i \in H(u_i). \tag{2.1}$$

Now a continuous casting process can be described by a boundary-value problem, formally written in the following pointwise form: find $u(x,t)$ and $\gamma(x,t)$ such that

$$\text{(P)} \begin{cases} \dfrac{\partial\gamma}{\partial t} + v\dfrac{\partial\gamma}{\partial x_2} - \Delta u = 0 \text{ for } x \in \Omega, t > 0, \\ u = z(x_1,t) \text{ for } x \in \Gamma_1, t > 0, \\ \dfrac{\partial u}{\partial n} + au + b|u|^3 u = g, a \geq 0, b \geq 0, \text{ for } x \in \Gamma_2, t > 0, \\ \gamma = H_0(x) \text{ for } x \in \bar{\Omega}, t = 0, \\ \gamma(x,t) \in H(u(x,t)) \text{ for } x \in \Omega, t > 0. \end{cases}$$

Below we suppose, that the boundary temperature $z(x_1,t)$ at any point of Γ_1 and for all $t \geq 0$ does not coincide with the phase transition temperature $T_L = T_S$, in other words, the enthalpy function H has a single values at all these points. This corresponds to the physical

[1]University of Oulu, erkki.laitinen@oulu.fi
[2]Kazan State University, alapin@ksu.ru
[3]University of Oulu, jpieska@cc.oulu.fi

meaning of the problem, because the incoming material (points $x \in \Gamma_1 : x_2 = 0$) is in liquid state, while the outcoming material (points $x \in \Gamma_1 : x_2 = L_{\times_2}$) is in solid state. The existence and uniqueness of a weak solution for problem (P) are proved in [6].

We approximate problem (P) by an implicit in time finite difference scheme and by a semi-implicit finite difference scheme, using for the approximation in the space variables a finite element method with the quadrature rules.

Let T_h be a triangulation of Ω in the rectangular elements δ of dimensions $h_1 \times h_2$ and $V_\mathrm{h} = \{u_\mathrm{h}(x) \in H^1(\Omega) : u_\mathrm{h}(x) \in Q_1 \text{ for all } \delta \in T_\mathrm{h}\}$, where Q_1 is the space of bilinear functions. By $\Pi_\mathrm{h} v(x)$ we denote the V_h-interpolant of a continuous function $v(x)$, i.e. $\Pi_\mathrm{h} v(x) \in V_\mathrm{h}$ and coincides with $v(x)$ in the mesh nodes – vertices of all $\delta \in T_\mathrm{h}$. We also use an interpolation operator P_h, which is defined as follows: for any continuous function $v(x)$ the function $P_\mathrm{h} v(x)$ is piecewise linear in x_1, piecewise constant in x_2 and on $\delta = [x_1, x_1 + h_1] \times [x_2, x_2 + h_2]$ it coincides with $v(x)$ at $(x_1, x_2 + h_2)$ and $(x_1 + h_1, x_2 + h_2)$.

Let further $V_\mathrm{h}^0 = \{u_\mathrm{h}(x) \in V_\mathrm{h} : u_\mathrm{h}(x) = 0 \text{ for all } x \in \Gamma_1\}$, $V_\mathrm{h}^\mathrm{z} = \{u_\mathrm{h}(x) \in V_\mathrm{h} : u_\mathrm{h}(x) = z_\mathrm{h} \text{ for all } x \in \Gamma_1\}$. Here z_h is the bilinear interpolation of z on the boundary Γ_1. For any continuous function $v(x)$ we define the quadrature formulas:

$$S_\delta(v) = \int_\delta \Pi_\mathrm{h} v dx, S_\Omega(v) = \sum_{\delta \in T_\mathrm{h}} S_\delta v,$$

$$S_{\partial\delta}(v) = \int_{\partial\delta} \Pi_\mathrm{h} v dx, S_{\Gamma_2}(v) = \sum_{\partial\delta \in T_\mathrm{h} \cap \bar{\Gamma}_2} S_{\partial\delta}(v);$$

$$E_\delta(v) = \int_\delta P_\mathrm{h} v dx, E_\Omega(v) = \sum_{\delta \in T_\mathrm{h}} E_\delta(v).$$

Let also $\omega_\tau = \{t_\mathrm{k} = k\tau, 0 \le k \le M, M\tau = T\}$ be a uniform mesh in time on the segment $[0, T]$ and $\partial_{\bar{\mathrm{t}}}\gamma = \frac{1}{\tau}(\gamma(x,t) - \gamma(x, t-\tau))$. Then the implicit in time finite difference scheme with up-wind approximation of the convective term $v\partial\gamma/\partial x_2$ can be written as follows: for all $t \in \omega_\tau$, $t > 0$, find $u_\mathrm{h} \in V_\mathrm{h}^\mathrm{z}$ and $\gamma_\mathrm{h} \in V_\mathrm{h}$ such that

$$\begin{cases} S_\Omega(\partial_{\bar{\mathrm{t}}}\gamma_\mathrm{h}\eta_\mathrm{h}) + E_\Omega(v(t)\dfrac{\partial\gamma_\mathrm{h}}{\partial x_2}\eta_\mathrm{h}) + S_\Omega(\nabla u_\mathrm{h}\nabla\eta_\mathrm{h}) \\ +S_{\Gamma_2}((au_\mathrm{h} + b|u_\mathrm{h}|^3 u_\mathrm{h})\eta_\mathrm{h}) = S_{\Gamma_2}(g\eta_\mathrm{h}) \text{ for all } \eta_\mathrm{h} \in V_\mathrm{h}^0, \\ \gamma_\mathrm{h}(x,t) \in H(u_\mathrm{h}(x,t)) \text{ for all mesh nodes } x. \end{cases} \tag{2.2}$$

When constructing the semi-implicit mesh scheme the term $\left(\frac{\partial}{\partial t} + v(t)\frac{\partial}{\partial x_2}\right)\gamma$ is approximate by using the characteristics of the first order differential operator (similar to [2], [3]). Namely, if (x_1, x_2, t) is the mesh point on the time level t we choose $\tilde{x}_2 = x_2 - \int_{t-\tau}^{t} v(\xi)d\xi$ and approximate this term by: $\left(\dfrac{\partial}{\partial t} + v(t)\dfrac{\partial}{\partial x_2}\right)\gamma \approx \dfrac{1}{\tau}\left(\gamma(x_1, x_2, t) - \gamma(x_1, \tilde{x}_2, t-\tau)\right)$. We denote $\tilde{\gamma}(x, t-\tau) = \gamma(x_1, \tilde{x}_2, t-\tau)$. If $\tilde{x}_2 < 0$ then we put $\tilde{\gamma}(x, t-\tau) = \gamma(x_1, 0, t-\tau)$. Note, that $\gamma(x_1, 0, t-\tau) = H(z(x_1, t-\tau))$ with single values $H(z(x_1, t-\tau))$ of H at these points, as it was mentioned above. In what follows we use the notation $d_{\bar{\mathrm{t}}}\gamma = \frac{1}{\tau}(\gamma(x,t) - \tilde{\gamma}(x, t-\tau))$ for the difference quotient in each mesh point on time level t.

Now, the semi-implicit finite difference scheme for problem (P) is: for all $t \in \omega_\tau$, $t > 0$, find $u_\mathrm{h} \in V_\mathrm{h}^\mathrm{z}$ and $\gamma_\mathrm{h} \in V_\mathrm{h}$ such that

$$\begin{cases} S_\Omega(d_{\bar{\mathrm{t}}}\gamma_\mathrm{h}\eta_\mathrm{h}) + S_\Omega(\nabla u_\mathrm{h}\nabla\eta_\mathrm{h}) + S_{\Gamma_2}((au_\mathrm{h} + b|u_\mathrm{h}|^3 u_\mathrm{h})\eta_\mathrm{h}) \\ = S_{\Gamma_2}(g\eta_\mathrm{h}) \text{ for all } \eta_\mathrm{h} \in V_\mathrm{h}^0, \\ \gamma_\mathrm{h}(x,t) \in H(u_\mathrm{h}(x,t)) \text{ for all mesh nodes } x. \end{cases} \tag{2.3}$$

Let $N_0 = \text{card } V_{\text{h}}^0$ and $u \in \mathbb{R}^{\text{N}_0}$ be the vector of nodal values for $u_{\text{h}} \in V_{\text{h}}^0$. We use the writing $u_{\text{h}} \Leftrightarrow u$ for this bijection. We define $N_0 \times N_0$ matrices A and B and nonlinear operator C by the following relations: for all $V_{\text{h}}^0 \ni u_{\text{h}} \Leftrightarrow u \in \mathbb{R}^{\text{N}_0}$ and $V_{\text{h}}^0 \ni \eta_{\text{h}} \Leftrightarrow \eta \in \mathbb{R}^{\text{N}_0}$

$$(Au, \eta) = S_\Omega(\nabla u_{\text{h}} \nabla \eta_{\text{h}}) + S_{\Gamma_2}(a u_{\text{h}} \eta_{\text{h}}),$$
$$(Bu, \eta) = S_\Omega(\frac{1}{\tau} u_{\text{h}} \eta_{\text{h}}) + E_\Omega(v(t) \frac{\partial u_{\text{h}}}{\partial x_2} \eta_{\text{h}}),$$
$$(Cu, \eta) = S_{\Gamma_2}(b|u_{\text{h}}|^3 u_{\text{h}} \eta_{\text{h}}).$$

Further we define a vector f: $(f, \eta) = S_{\Gamma_2}(g\eta_{\text{h}}) + S_\Omega(\frac{1}{\tau}\gamma(u_{\text{h}}(x, t-\tau))\eta_{\text{h}})$. Let now $\tilde{z}_{\text{h}}(x) \in V_{\text{h}}$ be the function which is equal to z_{h} on $\bar{\Gamma}_1$ and 0 for all nodes in $\Omega \cup \Gamma_2$, then f_0 is defined by the equality:

$$(f_0, \eta) = S_\Omega(\nabla \tilde{z}_{\text{h}}, \nabla \eta_{\text{h}}) + E_\Omega(v(t) \frac{\partial \Pi_{\text{h}}(H(\tilde{z}_{\text{h}}))}{\partial x_2} \eta_{\text{h}}) \text{ for all } \eta_{\text{h}} \in V_{\text{h}}^0.$$

(Here we use again the fact, that the graph $H(u)$ is single-valued for $u = \tilde{z}_{\text{h}}(x)$, when x is a mesh point). Finally we get $F = f - f_0$.

In these notations the algebraic form for the implicit mesh scheme (2.2) at fixed time level is:

$$Au + B\gamma + Cu = F, \gamma \in H(u). \tag{2.4}$$

If we set $(Bu, \eta) = S_\Omega(\frac{1}{\tau} u_{\text{h}} \eta_{\text{h}})$ and $(f, \eta) = S_{\Gamma_2}(g\eta_{\text{h}}) + S_\Omega(\frac{1}{\tau}\tilde{\gamma}_{\text{h}}\eta_{\text{h}})$, then the semi-implicit mesh scheme (2.3) has also the algebraic form (2.5).

$$Au + B\gamma + Cu = F, \ \ \gamma \in H(u). \tag{2.5}$$

The matrices A, and B and the operators C, and H have the following properties:

$$A \text{ and } B \text{ are } M - \text{matrices}, \tag{2.6}$$

$$A \text{ is weakly diagonally dominant in columns: } \sum_{j \neq i}^{N_0} |a_{\text{ji}}|/a_{\text{ii}} \leq 1 \forall i; \tag{2.7}$$

$$B \text{ is strictly diagonally dominant in columns: } \sum_{j \neq i}^{N_0} |b_{\text{ji}}|/b_{\text{ii}} \leq \beta < 1 \forall i; \tag{2.8}$$

(in fact, for the semi-implicit scheme matrix B is diagonal); the operators γ and C have the diagonal forms:

$$\gamma(u) = (\gamma(u_1), \gamma(u_2), ..., \gamma(u_{N_0}))^{\text{t}}, Cu = (c_1(u_1), c_2(u_2), ..., c_N(u_{N_0}))^{\text{t}}, \tag{2.9}$$

where c_{i} are continuous non-decreasing functions and $\gamma(.)$ is maximal monotone and uniformly monotone graph (see (2.1)). Note, that $\beta = \dfrac{\tau}{\tau + h_2}$ for the case of the implicit finite difference scheme, while $\beta = 0$ for the semi-implicit scheme.

Below we use the following notations: $u \gg 0 \Leftrightarrow u_{\text{i}} \geq 0 \quad \forall i, \quad A \gg 0 \Leftrightarrow a_{\text{ij}} \geq 0 \quad \forall i, j$. There exist a subsolution $(\underline{u}, \underline{\gamma})$:

$$A\underline{u} + B\underline{\gamma} + C\underline{u} \leq F, \ \underline{\gamma} \in H(\underline{u}), \tag{2.10}$$

and a supersolution $(\bar{u}, \bar{\gamma})$:

$$A\bar{u} + B\bar{\gamma} + C\bar{u} \geq F, \ \bar{\gamma} \in H(\bar{u}) \tag{2.11}$$

for form (2.4). Under above assumptions, the following theorem can be proved [4], [5], .

Theorem 2.1 *The implicit mesh scheme (2.2) and the semi-implicit mesh scheme (2.3) have unique solutions.*

3. Asynchronous algorithms. In this section we present the asynchronous additive Schwarz alternating algorithms.

Algorithm 1 (ASM1)

1. Divide the domain Ω into p overlapping subdomains and construct approximative subproblems in these subdomains.
2. Solve simultaneously the subproblems in the slave processors.
3. When the local stopping criterion in a slave processor is reached, send information about this to the master processor and keep calculating further.
4. When all slaves have finished the calculations, send the subsolutions to the master processor for updating the information for all slave processors.
5. If the accuracy is reached, then **STOP**, else goto 2.

Algorithm 2 (ASM2)

1. Divide the domain Ω into p overlapping subdomains and construct approximative subproblems in these subdomains.
2. Solve simultaneously the subproblems in the slave processors.
3. When the local stopping criterion in a slave processor is reached, send subsolution to the master processor and check if there is a new information from the neighboring subdomains. If yes, then update it and restart the calculations, otherwise keep calculating further.
4. When all slaves have finished the calculations, send the subsolutions to the master processor for updating the information for all slave processors.
5. If the accuracy is reached, then **STOP**, else goto 2.

In **Algorithm 1** we do not use the newest available information. This slows convergence. Although it is much faster to just send a signal to the master that the processor is ready than send the whole subsolution.

In **Algorithm 2** we send the subsolution to the master whenever there is an improvement. This increase the total calculation time. On the other hand we use the newest available information which decreases the calculation time.

Intuitively if there is a large load imbalance, i.e. if some processors have substantially more work than others, one can expect the asynchronous versions to converge faster than the synchronous one. It is also expected that ASM2 would be faster than ASM1.

4. Iterative methods. In this section we study the convergence of asynchronous iterative methods. For simplicity but without loss of generality we suppose that the domain Ω is decomposed into two overlapping subdomains Ω_1 and Ω_2, consisting of the elements of a triangulation T_h. We arrange the nodes of the mesh as follows. First, we enumerate the nodes lying in the non-overlapping part of the first subdomain, namely $x \in (\bar{\Omega}_1 \setminus \bar{\Gamma}_1) \setminus \overline{\Omega_1 \cap \Omega_2}$, then the nodes in the overlapping zone $x \in \overline{\Omega_1 \cap \Omega_2} \setminus \bar{\Gamma}_1$ and at last the nodes in the non-overlappping part of the second subdomain. A vector $u \in \mathbb{R}^{\mathrm{N}_0}$, $u \Leftrightarrow u_\mathrm{h}(x)$, takes the form $u = (u_{11}, u_{12}, u_{22})^\mathrm{t}$ with the subvectors u_ij corresponding to enumeration of the nodes.

This decomposition implies also the partitioning of the matrices and nonlinear operators: $A = (A_\mathrm{ij})_{\mathrm{ij}=1}^3$, $B = (B_\mathrm{ij})_{\mathrm{ij}=1}^3$, $C = \mathrm{diag}(C_1, C_2, C_3)$. Note, that $A_\mathrm{ij} \ll 0$, $B_\mathrm{ij} \ll 0$ for $i \neq j$ and the blocks A_{13}, A_{31}, B_{13}, B_{31} are equal to zero.

We use also the following notations:

$$A_0^1 = \begin{pmatrix} A_{11} & A_{12} \\ A_{21} & A_{22} \end{pmatrix}, B_0^1 = \begin{pmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{pmatrix}, A_1^1 = \mathrm{diag}(0, A_{23}),\ B_1^1 = \mathrm{diag}(0, B_{23});$$

$$A_0^2 = \begin{pmatrix} A_{22} & A_{23} \\ A_{32} & A_{33} \end{pmatrix}, B_0^2 = \begin{pmatrix} B_{22} & B_{23} \\ B_{32} & B_{33} \end{pmatrix}, A_1^2 = \mathrm{diag}(A_{21}, 0),\ B_1^2 = \mathrm{diag}(B_{21}, 0);$$

$$C^1 = \mathrm{diag}(C_1, C_2),\ C^2 = \mathrm{diag}(C_2, C_3).$$

Let further $u_1 = (u_{11}, u_{12})^{\mathrm{t}},\ u_2 = (u_{12}, u_{22})^{\mathrm{t}}$ and similar for all other vectors.

Then ASAM has the form (4.1), (4.2):

$$\begin{cases} A_0^1 v_1^{\mathrm{k}+1} + B_0^1 \eta_1^{\mathrm{k}+1} + C^1 v_1^{\mathrm{k}+1} = F_1 - A_1^1 u_2^{\mathrm{k}} - B_1^1 \gamma_2^{\mathrm{k}}; \eta_1^{\mathrm{k}+1} \in H(v_1^{\mathrm{k}+1}), \\ A_0^2 w_2^{\mathrm{k}+1} + B_0^2 \xi_2^{\mathrm{k}+1} + C^2 w_2^{\mathrm{k}+1} = F_2 - A_1^2 u_1^{\mathrm{k}} - B_1^2 \gamma_1^{\mathrm{k}}; \xi_2^{\mathrm{k}+1} \in H(w_2^{\mathrm{k}+1}), \end{cases} \tag{4.1}$$

$$\begin{cases} u_{11}^{\mathrm{k}+1} = v_{11}^{\mathrm{k}+1},\ u_{22}^{\mathrm{k}+1} = w_{22}^{\mathrm{k}+1},\ u_{12}^{\mathrm{k}+1} = \alpha v_{12}^{\mathrm{k}+1} + (1-\alpha) w_{12}^{\mathrm{k}+1}, \\ \gamma_{11}^{\mathrm{k}+1} = \eta_{11}^{\mathrm{k}+1},\ \gamma_{22}^{\mathrm{k}+1} = \xi_{22}^{\mathrm{k}+1},\ \gamma_{12}^{\mathrm{k}+1} = \alpha \eta_{12}^{\mathrm{k}+1} + (1-\alpha) \xi_{12}^{\mathrm{k}+1}, \end{cases} \tag{4.2}$$

with an initial guess (u^0, γ^0) and $\alpha \in (0, 1)$.

Let now every subproblem in (4.1) be solved by using a finite number of iterations of an inner iterative algorithm. Then we derive a two-stage additive Schwarz alternating method.

Let for $i = 1, 2\ A_0^{\mathrm{i}} = M_{\mathrm{i}} + N_{\mathrm{i}}, B_0^{\mathrm{i}} = K_{\mathrm{i}} + L_{\mathrm{i}}$ be regular splittings of A and B with $\mathrm{diag}(A_0^{\mathrm{i}}) \subseteq M_{\mathrm{i}},\ \mathrm{diag}(B_0^{\mathrm{i}}) \subseteq K_{\mathrm{i}}$ and $N_{\mathrm{i}} \ll 0, L_{\mathrm{i}} \ll 0$. Starting from the initial guess $z_{1,0} = u_1^{\mathrm{k}},\ z_{2,0} = u_2^{\mathrm{k}},\ \epsilon_{1,0} = \gamma_1^{\mathrm{k}},\ \epsilon_{2,0} = \gamma_2^{\mathrm{k}}$, we solve the subproblems in (4.1) by the iterative methods:

$$\begin{cases} M_1 z_{1,\mathrm{i}} + K_1 \epsilon_{1,\mathrm{i}} + C^1 z_{1,\mathrm{i}} = \varphi_1^{\mathrm{k}} - N_1 z_{1,\mathrm{i}-1} - L_1 \epsilon_{1,\mathrm{i}-1}, \\ \epsilon_{1,\mathrm{i}} \in H(z_{1,\mathrm{i}}),\ i = 1, \ldots, p_1, \end{cases} \tag{4.3}$$

$$\begin{cases} M_2 z_{2,\mathrm{i}} + K_2 \epsilon_{2,\mathrm{i}} + C^2 z_{2,\mathrm{i}} = \varphi_2^{\mathrm{k}} - N_1 z_{2,\mathrm{i}-1} - L_1 \epsilon_{2,\mathrm{i}-1}, \\ \epsilon_{2,\mathrm{i}} \in H(z_{2,\mathrm{i}}),\ i = 1, \ldots, p_2, \end{cases} \tag{4.4}$$

set $v_1^{\mathrm{k}+1} \equiv z_{1,\mathrm{p}_1},\ \eta_1^{\mathrm{k}+1} \equiv \epsilon_{1,\mathrm{p}_1};\ w_2^{\mathrm{k}+1} \equiv z_{2,\mathrm{p}_2},\ \xi_2^{\mathrm{k}+1} \equiv \epsilon_{2,\mathrm{p}_2}$ and then update the outer iterations using formulas (4.2).

Here $\varphi_1^{\mathrm{k}} = F_1 - A_1^1 u_2^{\mathrm{k}} - B_1^1 \gamma_2^{\mathrm{k}},\ \varphi_2^{\mathrm{k}} = F_2 - A_1^2 u_1^{\mathrm{k}} - B_1^2 \gamma_1^{\mathrm{k}}$ for method ASM1, when we calculate all subproblems by using inner iterative methods until we reach the desired accuracy in all subproblems and after that send the calculated $v_1^{\mathrm{k}+1},\ w_2^{\mathrm{k}+1},\ \eta_1^{\mathrm{k}+1},\ \xi_2^{\mathrm{k}+1}$ to the master processor to update the outer iterations to using formulas (4.2). On the other hand, for method ASM2 the formulas for φ_i^{k} are changed to $\varphi_1^{\mathrm{k}} = F_1 - A_1^1 w_2^{\mathrm{k}+1} - B_1^1 \xi_2^{\mathrm{k}+1}$ or to $\varphi_2^{\mathrm{k}} = F_2 - A_1^2 v_1^{\mathrm{k}+1} - B_1^2 \eta_1^{\mathrm{k}+1}$, depending on which of subproblems was solved faster.

Theorem 4.1 *Iterative method (4.3)– (4.4), (4.2) with an initial guess* $(u^0, \gamma^0) \in \langle (\underline{u}, \underline{\gamma}), (\bar{u}, \bar{\gamma}) \rangle$ *converges with geometric rate of convergence:*

$$||A^0(u^{\mathrm{k}+1} - u) + B^0(\gamma^{\mathrm{k}+1} - \gamma)||_1 \le q ||A^0(u^{\mathrm{k}} - u) + B^0(\gamma^{\mathrm{k}} - \gamma)||_1, \tag{4.5}$$

with $q = \dfrac{c_{\mathrm{AB}} + \alpha\beta}{c_{\mathrm{AB}} + \alpha} < 1,\ c_{\mathrm{AB}} = \max\limits_{1 \le i \le N_0} \dfrac{a_{\mathrm{ii}}}{b_{\mathrm{ii}}}$. *Here* $||v||_1 = \sum\limits_{i=1}^{N_0} |v_{\mathrm{i}}|$ *and* $c_{\mathrm{AB}} = \dfrac{2\tau(1 + h_2^2/h_1^2)}{h_2(\tau + h_2)}$ *for the implicit scheme, while* $c_{\mathrm{AB}} = \dfrac{2\tau(1 + h_2^2/h_1^2)}{h_2^2}$ *for the semi-implicit scheme. The parameter* α *is from equation (2.1) and* β *from (2.8).*

5. Numerical results. To validate theoretical results the following numerical example was considered. Let $\Omega =]0,1[\times]0,1[$ with the boundary Γ divided in two parts such that $\Gamma_{\mathrm{D}} = \{x \in \partial\Omega : x_2 = 0 \vee x_2 = 1\}$ and $\Gamma_{\mathrm{N}} = \Gamma \setminus \Gamma_{\mathrm{D}}$, moreover let $T = 1$. Let us consider the case where the phase change temperature $u_{\mathrm{SL}} = 1$ and the latent heat $L = 1$ and the density $\rho = 1$. Let the velocity be $v(t) = \frac{1}{5}$. Our numerical example is

$$\begin{aligned}
\frac{\partial H}{\partial t} - \Delta K + v(t)\frac{\partial H}{\partial x_2} &= f(x;t) && \text{on } \Omega,\\
u(x_1,x_2;t) &= (x_1-\tfrac{1}{2})^2 + (x_2-\tfrac{1}{2})^2 - \tfrac{1}{2}e^{-4t} + \tfrac{5}{4} && \text{on } \Gamma_{\mathrm{D}},\\
\frac{\partial u}{\partial n} &= 1 && \text{on } \Gamma_{\mathrm{N}},\\
u(x_1,x_2;0) &= (x_1-\tfrac{1}{2})^2 + (x_2-\tfrac{1}{2})^2 + \tfrac{1}{2} && \text{on } \Omega,
\end{aligned}$$

where the Kirchoff's temperature $K(u)$ and the enthalpy $H(u)$ are according to their definitions

$$K(u) = \begin{cases} u & \text{if } u < u_{\mathrm{SL}}, \\ 2u-1 & \text{if } u \geq u_{\mathrm{SL}}, \end{cases} \quad \text{and } H(u) = \begin{cases} 2u & \text{if } u < u_{\mathrm{SL}}, \\ [2u_{SL}, 2u_{SL}+\rho L] & \text{if } u = u_{\mathrm{SL}}, \\ 6u - 4u_{SL} + \rho L & \text{if } u > u_{\mathrm{SL}}. \end{cases}$$

Furthermore the known right-hand side is

$$f(x;t) = \begin{cases} 4e^{-4t} + \frac{1}{5}(4x_2-2) - 4 & \text{if } u < u_{\mathrm{SL}}, \\ 12e^{-4t} + \frac{1}{5}(12x_2-6) - 8 & \text{if } u \geq u_{\mathrm{SL}}. \end{cases}$$

The stopping criterion of the outer iterations was the value of the L_2-norm of residual $\|r\|_{\mathrm{L}_2} = \|Au + B\gamma + \delta - f\|_{\mathrm{L}_2} \leq 10^{-3}$. We use through all the calculations the decomposition presented on the figure 5.1. The subdomain Ω_1 is roughly twice as big as other subdomains.

5.1. Implicit scheme. In our first test case we changed the number of grid points both in time and in space. We solved the problem by using the implicit scheme (2.2). The results can be seen in table 5.1. The over is the number of grid lines in the overlapping area. The inner iterations was performed till all of the processors have reached the desired accuracy $\|r\|_{\mathrm{L}_2} \leq 10^{-3}$. Due to this the number of inner iterations can be different for different processors. The synchronous Schwarz alternating method is denoted by (SASM)

Grid	over	ASM1 iterations	ASM1 T[s]	ASM2 iterations	ASM2 T[s]	SASM iterations	SASM T[s]
$65 \times 65 \times 128$	4	17	14.8	8	11.4	16	16.0
$129 \times 129 \times 256$	8	16	92.1	11	73.0	11	146
$257 \times 257 \times 512$	16	19	1184	17	1120	9	2776

Table 5.1: The number of outer iterations and calculation times in seconds for different grids for 4 processors; Implicit scheme.

5.2. Semi-Implicit scheme. We solve the same problem as for the implicit scheme to compare these methods against each other. The results can be seen in table 5.2.

6. Conclusions. Two mesh schemes with two different kind of discretizations for the convection term were considered, an implicit and a semi-implicit scheme. A model problem was solved by using both asynchronous methods ASM1 and ASM2. It can be seen from tables 5.1 and 5.2 that ASM2 takes fewer outer iterations than ASM1 and is thus the faster of the these two methods.

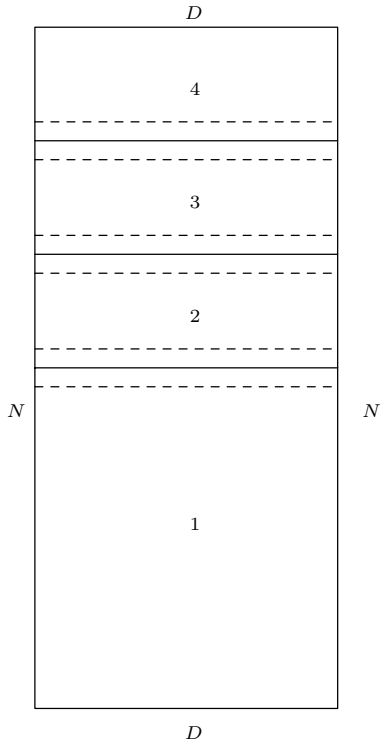


Figure 5.1: The decomposition used in model continuous casting problem.

Grid	over	ASM1 iterations	ASM1 T[s]	ASM2 iterations	ASM2 T[s]	SASM iterations	SASM T[s]
$65 \times 65 \times 128$	4	17	12.2	13	10.6	16	15.8
$129 \times 129 \times 256$	8	16	84.5	14	65.9	11	128
$257 \times 257 \times 512$	16	19	1171	17	1056	9	2528

Table 5.2: The number of outer iterations and calculation times in seconds for different grids for 4 processors: Semi-Implicit scheme.

Numerical results confirm the theoretical results. Our numerical results show that the calculation times of the asynchronous methods ASM1 and ASM2 are smaller than for the synchronous method SASM. In our opinion, ASM1 and ASM2 are faster for this kind of decomposition. We could also gain some advantage with asynchronous methods if the processors differ from each other.

REFERENCES

[1] Z.-Z. Bai. A class of asynchronous parallel iterations for the system of nonlinear algebraic equations. *Computers and Mathematics with Applications*, 39(7-8):81–94, 2000.

[2] Z. Chen. Numerical solutions of a two-phase continuous casting problem. In P. Neittaanmaki, editor, *Numerical Methods for Free Boundary Problem*, pages 103–121, Basel, 1991. International Series of Numerical Mathematics, Birkhuser.

[3] J. D. Jr. and T. F. Russel. Numerical methods for convection-dominated diffusion problem based on combining the method of characteristic with finite element or finite difference procedure. *Siam J. Numer Anal.*, 19:871–885, 1982.

[4] E. Laitinen, A. Lapin, and J. Pieska. Mesh approximation and iterative solution of the continuous casting problem. In P. Neittaanmaki, T. Tiihinen, and P. Tarvainen, editors, *ENUMATH 99 - Proceedings of the 3rd European Conference on Numerical Mathematics and Advanced Applications*, pages 601–617. World Scientific, Singapore, 1999.

[5] A. Lapin. Finite dimensional inclusions with several m-matricies and maximal monotone operators. Preprint, Department of Mathematical Sciences, University of Oulu, 2000.

[6] J. F. Rodrigues and F. Yi. On a two-phase continuous casting stefan problem with nonlinear flux. *Euro. of Applied Mathematics*, 1:259–278, 1990.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

50. A domain decomposition algorithm for nonlinear interface problem

T. Sassi[1]

1. Introduction. In this paper, we are interested in the numerical solution of a nonlinear elliptic problem by a nonoverlapping domain decomposition technique. The model problem under consideration takes the standard form

For a given $f \in L^2(\Omega)$, find $u \in V$ such that

$$\sum_i \Big\{ \int_{\Omega_i} \nabla u \cdot \nabla v \, dx \; + \int_{\Omega_i} (u_i^3 - f) v \, dx \Big\} \; = \; 0 \quad , \quad \forall \; v \in V \; , \tag{1.1}$$

where V is the usual Sobolev space

$$V = \big\{ v \in H^1(\Omega) \; , \; v = 0 \text{ on } \partial\Omega_D \big\} \; ,$$

defined over a given domain $\Omega = \cup_{i=1}^N \Omega_i$ of $\mathbb{R}^2$.
In any case, even if Ω is partitioned into nonoverlapping subdomains Ω_i (see Figure 4.1), the nonlinear problem (1.1) is not reduced to independent subproblems set on each subdomain Ω_i because elements of the space V are constrained to be continuous across the different interfaces $\partial\Omega_i \cap \partial\Omega_j$. Most nonoverlapping domain decomposition techniques handle this constraint by a standard Newton's algorithm in which all linearized subproblems are solved by iterative substructuring methods (see [4], [5]).

The purpose of this paper is to propose and study another numerical strategy well adapted to nonlinear problems. The resulting discrete problem of (1.1) is reduced to an interface problem via a nonlinear Steklov-Poincaré operator [8]. Modified Newton iterations are used to treat the nonlinear aspect of the interface problem. We extend the results obtained in [7] to the case of multidomain decomposition. We prove that this algorithm converges independently of the discretization step h. Numerical results are given to illustrate the efficiency of this approach. Moreover, the proposed algorithm is compared to the so called Newton conjugate gradient algorithm introduced in [4].

2. A generalized nonlinear interface problem. We begin with some notation used hereafter. Let us introduce the boundaries (see Figure 4.1)

$$\begin{aligned}
\partial\Omega &= \Gamma_D \cup \Gamma_N \; , \quad \text{external Dirichlet and Neumann boundaries,} \\
\partial\Omega_{D_i} &= \Gamma_D \cap \partial\Omega_i \; , \quad \text{local Dirichlet boundary,} \\
\Gamma_i &= \partial\Omega_i \setminus \partial\Omega, \quad \text{local interface,} \\
\Gamma &= \cup_i \Gamma_i, \quad \text{global interface,}
\end{aligned}$$

with $\Gamma_D \neq \{\emptyset\}$. The global interface Γ is made of N_f faces Γ_{ij} separating the domain Ω_i from the domain Ω_j. In this decomposition, we neglect corners. This is ligitimate if there are no corners (partition in strip) or if the interfaces are discretized by *mortar* elements which define discrete traces on faces and not on corners [9].
On this geometry, we introduce the spaces

$$V_i = \big\{ v \in H^1(\Omega_i) \; , \; v = 0 \text{ on } \partial\Omega_{D_i} \big\} \; , \quad V_i^0 \;\; = \;\; \big\{ v \in H^1(\Omega_i) \; , \; v = 0 \text{ on } \partial\Omega_{D_i} \cup \Gamma_i \big\} \; .$$

In a domain decomposition framework, the variational problem (1.1) is reduced to an interface problem whose unknown is the trace φ of u on the interface Γ. Indeed, if we knew φ on Γ

[1]Institut National des Sciences Appliquées de Lyon, sassi@insa-lyon.fr

and if we restrict ourselves to the test functions v in spaces V_i^0, then we observe that u_i is the solution of the following variational problem

$$\left\{ \begin{array}{l} \displaystyle\int_{\Omega_i} (\nabla u_i \cdot \nabla v_i + u_i^3) v \, dx \; = \; \int_{\Omega_i} f_i v \, dx \quad , \quad \forall \, v \in \; V_i^0 \; , \\ u_i - \varphi \; \in \; V_i^0 \; . \end{array} \right. \tag{2.1}$$

Introducing the Lagrange multiplier of the constraint $v_i \in V_i^0$, (2.1) can be written as

$$\left\{ \begin{array}{l} \displaystyle\int_{\Omega_i} (\nabla u_i \cdot \nabla v_i + u_i^3 \, v) \, dx \; = \; \int_{\Gamma_i} \lambda_i v \, ds + \int_{\Omega_i} f_i v \, dx \; , \quad \forall \, v \in V_i \; , \\ u_i = \; \varphi \quad \text{on} \quad \Gamma_i, \quad \lambda_i \in H^{-1/2}(\Gamma_i) \; . \end{array} \right. \tag{2.2}$$

To each $\varphi \in TrV$ (Tr is the operator mapping functions in V to their traces on Γ), we can then associate these multipliers $\lambda_i(\varphi, f)$, the corresponding map being the so-called Steklov-Poincaré operator. Then, by addition, the correct value of φ is the solution of the following interface problem.

$$\sum_i \int_{\Gamma_i} \lambda_i(\varphi, f) ds = 0, \quad \forall v \in V. \tag{2.3}$$

We want to approximate problem (2.3) with a *mortar* finite element method (see [3]). For this purpose, for each face Γ_{ij}, we introduce an approximation space W_{ijh}. We then define the trace space W_h and the local interface scalar product $< \cdot, \cdot >_{\Gamma_i}$:

$$W_h = \prod_{ij=1,N_f} W_{ijh}, \qquad < v, w >_{\Gamma_i} = \sum_{ij \ni i} \int_{\Gamma_i} vw ds.$$

We denote by Tr_{ih} the discrete trace operator defined from V_{ih} into W_h and which to a given $v_{ih} \in V_{ih}$ associates its L^2 projection $Tr_{ih} v_{ih}$ onto W_h. With this notation, the definition of the global approximation space V_h is

$$V_h \; = \; \{ v_h = (v_{ih})_i \in \prod_i V_{ih} \; , \text{ s.t.} \quad Tr_{ih} v_{ih} = Tr_{jh} v_{jh} \; , \; \forall \, i < j \} \; .$$

The generalisation to the discrete level of problem (2.3) is then immediate. Let $\varphi \in W_h$, be given, for $1 \leq i \leq N$, find $u_{ih}(\varphi, f) \in V_{ih}$, $\lambda_{ih} \in W_h|_{\Gamma_i}$ solution to

$$\int_{\Omega_i} \nabla u_{ih}(\varphi, f) \nabla v_{ih} + u_{ih}^3(\varphi, f) v_{ih} \, dx \quad = \quad < \lambda_{ih}, Tr v_{ih} >_{\Gamma_i} + \int_{\Omega_i} f v_{ih} \, dx, \tag{2.4}$$

$$< Tr u_{ih}(\varphi, f) - \varphi) q_h >_{\Gamma_i} \quad = \quad 0 \; , \; \forall \, q_h \in M_h \; ; \forall \, v_{ih} \in V_{ih}, \tag{2.5}$$

where M_h is the approximation space of $H^{-1/2}(\Gamma)$ (see [2] for the definition of M_h). The discrete Steklov-Poincaré operator (see [7]) which to φ associates λ_{ih} the generalized normal derivative of $u_{ih}(\varphi, f)$ on Γ_i is defined as follows:

$$\begin{array}{rcl} S_{ih} : W_h|_{\Gamma_i} & \longrightarrow & M_h|_{\Gamma_i} \; , \\ Tr_{ih} u_{ih}(\varphi, f) & \mapsto & \lambda_{ih} \; , \end{array}$$

where $(u_{ih}(\varphi, f), \lambda_{ih})$ is the solution of (2.4)-(2.5).

Theorem 2.1 *Assume $h \leq h_0$, if $\varphi \in W_h$ is a solution of the following interface problem*

$$\sum_{i=1}^{N} < S_{ih} \varphi, Tr v_{ih} >_{\Gamma_i} = 0, \; \forall \, v_h \in V_h, \tag{2.6}$$

then Problem (1.1) and the interface Problem (2.6) are equivalent.

Proof. For the proof, the reader is referred to [7] when Ω is decomposed into two nonoverlapping subdomains. The extension to the case of multidomain is straightforward. ■

Let us notice that S_{ih} is a C^1 mapping from the Banach space $(W_h|_{\Gamma_i}\ ;\ \|\cdot\|_{\frac{1}{2};\Gamma})$ with values in the Banach space $(M_h|_{\Gamma_i};\ \|\cdot\|_{-\frac{1}{2};\Gamma})$, with $DS_{ih}(\varphi, f) \in \mathcal{L}(W_h|_{\Gamma_i}; M_h|_{\Gamma_i})$ defined by: $DS_{ih}(\varphi, f)\psi = \mu_{ih}$ where $(v_{ih}, \mu_{ih}) \in V_{ih} \times M_h|_{\Gamma_i}$ verifies

$$\left\{ \begin{array}{l} \displaystyle\int_{\Omega_i} \nabla v_{ih}\nabla\eta_{ih} + 3u_{ih}^2(\varphi, f)v_{ih}\eta_{ih}\, dx =< \mu_{ih}, Tr\eta_{ih} >_{\Gamma_i},\ \forall \eta_{ih} \in V_{ih}, \\ < Trv_{ih} - \psi, q_h >_{\Gamma_i} = 0,\ \forall q_h \in M_h. \end{array} \right. \tag{2.7}$$

Here $(u_{ih}(\varphi, f), \lambda_{ih})$ is solution to Problem (2.4)-(2.5). Please remark that $DS_{ih}(0, 0)$ is the classical discrete Steklov-Poincaré operator (see [1]) and that

$$\begin{array}{rcl} DS_{ih}^{-1}(0,0) : M_h|_{\Gamma_i} & \longrightarrow & W_h|_{\Gamma_i}\ , \\ \mu_{ih} & \mapsto & Tr_{ih}v_{ih}\ , \end{array}$$

where v_{ih} verifies the first equation of (2.7) with $u_{ih} = 0$.

3. A modified Newton algorithm for interface problem. The solution algorithm that we propose for solving (2.6) is a Modified Newton method, with preconditioner M. It writes

- for $\varphi^0 \in W_h$ given and φ^n known, define φ^{n+1} as the solution of
- $\varphi^{n+1} = \varphi^n - \rho MS\varphi^n$

where

$$M\ =\ \sum_i (\alpha_i\, Id|_\Gamma)\, S_{ih}^{-1}(0,0)\, (\alpha_i\, Id|_\Gamma)^t \qquad \text{and } S_h = \Big(\sum_{i=1}^N S_{ih}\varphi^n\Big).$$

Above, α_i defined face by face and such that

$$\left\{ \begin{array}{lll} \alpha_l|_{\Gamma_{ij}} & = & 0 \qquad \text{if } l \neq i \text{ and } l \neq j, \\ (\alpha_i\ +\ \alpha_j)|_{\Gamma_{ij}} & = & 1, \end{array} \right.$$

and ρ is a positive parameter which will be specified later.

Modified Newton iterations can be rephrased in a parallel way as follows:

• Let φ^n be given on Γ. Then on each subdomain solve in parallel (2.4)-(2.5), with $\varphi = \varphi^n$ in order to compute

$$L_i = S_{ih}\varphi^n, \qquad \text{and} \quad \text{set } L(\varphi^n) = \sum_i L_i. \tag{3.1}$$

- On each subdomain, compute $Tr_{ih}v_{h_i}$ where v_{ih} is the solution of

$$\int_{\Omega_i} \nabla v_{ih}\nabla\eta_{ih}dx =< L(\varphi^n), \alpha_i Tr\eta_{ih} >_{\Gamma_i}, \forall\, \eta_{ih} \in V_{ih}. \tag{3.2}$$

$$\bullet \quad \text{set } \varphi^{n+1} = \varphi^n - \rho\sum_{i=1}^N \alpha_i Tr_{ih}v_{ih}. \tag{3.3}$$

Please remark that the linear preconditioner in the above algorithm (3.1)-(3.3) is determined for the nonlinear interface problem obtained after elimination of interior unknowns. Another approach, the so called Newton Preconditioned Conjugate Gradient method, is to use the Newton algorithm on the global problem (1.1) in which all linearized subproblems are solved by a domain decomposition solver based on a Preconditioned Conjugate Gradient algorithm on the interface Γ (see [4]). Concerning the Modified Newton algorithm the main result of this section is:

Theorem 3.1 *For all $h \leq h_0$, let φ be the solution to Problem (2.6). There exists a neighborhood $\mathcal{V}(\varphi) \subset W_h$ and a parameter $0 < \rho$ independent of h such that for all $\varphi^0 \in \mathcal{V}(\varphi)$ Modified Newton iterations (3.1)-(3.3) converge towards φ.*

The proof of Theorem 3.1 is classical. Define the iteration mapping $G_\rho : W_h \to W_h$ which to ψ associates $\psi - \frac{\rho}{2} MS\psi$. We want to show that for a certain norm on the finite dimensional space W_h, the mapping G_ρ is locally a contraction. The key property to be established is that the eigenvalues of the derivative of G_ρ are non negative, thus it will be possible to choose ρ such that G_ρ is a contracting mapping. Now, let us give some intermediate results useful for the proof of Theorem 3.1.

Lemma 3.1 *The trace operators Tr_{ih} are linear uniformly with respect to h, surjective and continuous from V_{ih} into $W_h|_{\Gamma_i}$. For all $\psi \in W_h|_{\Gamma_i}$, there exists at least an element $Tr_{ih}^{-1}\psi$ in V_{ih} and a constant $C > 0$ independent of h verifying: $Tr_{ih}\left(Tr_{ih}^{-1}\psi\right) = \psi$ and $\|Tr_{ih}^{-1}\psi\|_{V_i} \leq C\|\psi\|_{\frac{1}{2},\Gamma}$.*

Our motivation now is to define a discrete scalar product on W_h such that the operator MS is positive. So let us set

$$V_{ih}^0 = \{v_{ih} \in V_{ih}; Tr_{ih} v_{ih} = 0\}$$

and define for all $\psi \in W_h$ the function $\theta_{ih}(\psi) \in V_{ih}$ solution to

$$\begin{cases} \int_{\Omega_i} \nabla\theta_{h_i}(\psi)\nabla\phi_{h_i}\, dx = 0 \quad \forall \phi_{h_i} \in V_{ih}^0 \\ Tr_{ih}\theta_{ih}(\psi) = \psi \text{ on } \Gamma_i. \end{cases} \tag{3.4}$$

We then define the discrete scalar products $(\cdot,\cdot)_h$ on $W_h \subset H^{\frac{1}{2}}(\Gamma)$ by:

$$(\psi,\varphi)_h = \sum_i \int_{\Omega_i} \nabla\theta_{ih}(\psi)\nabla Tr_{ih}^{-1}\varphi\, dx = \sum_i \int_{\Omega_i} \nabla\theta_{ih}(\psi)\nabla\theta_{ih}(\varphi)\, dx\ , \tag{3.5}$$

since $Tr_{ih}^{-1}\varphi - \theta_{ih}(\psi) \in V_{ih}^0$.

Lemma 3.2 *The discrete scalar products $(\cdot,\cdot)_h$ are uniformly with respect to h equivalent to the standard scalar product of $H^{\frac{1}{2}}(\Gamma)$ in W_h.*

For the proof, of Lemma 3.1 and Lemma 3.2, the reader is referred to [6].

Lemma 3.3 *There exists $\beta > 0$ independent of h such that $\forall \varphi \in W_h$ we have*

$$(M\, DS_h(\varphi, f)\psi, \psi)_h \geq \beta \quad \forall\, \psi \in W_h,$$

with

$$DS_h(\varphi, f) = \sum_i DS_{ih}(\varphi, f).$$

Proof. Let $\mu_{ih} \in M_h|_{\Gamma_i}$ and $\mu \in M_h$ be defined by $\mu_{ih} = DS_{ih}(\varphi, f)\psi$, $\mu = DS_h(\varphi, f)\psi$ respectively, and let $w_{ih} \in X_{ih}$ be solution to

$$\int_{\Omega_i} \nabla w_{ih}\nabla\eta_{ih}\, dx = < \mu, Tr_{ih}\eta_{ih} >_{\Gamma_i} \ \forall \eta_{ih} \in V_{ih}. \tag{3.6}$$

We have

$$\begin{aligned}(M\,DS_h(\varphi,f)\psi,\psi)_h &= \sum_{i=1}^{N}\int_{\Omega_i}\nabla\theta_{ih}(\psi)\nabla Tr_{ih}^{-1}(M\mu)\,dx\\ &= \sum_{i=1}^{N}\int_{\Omega_i}\nabla\theta_{ih}(\psi)\nabla Tr_{ih}^{-1}\Big(\sum_j \alpha_j Tr_{jh}w_{jh}\Big)dx\\ &= \sum_{i=1}^{N} <\mu_{ih}, Tr\theta_{ih}(\psi)>_{\Gamma_i}\end{aligned}$$

Finally, Problem (2.7) provides

$$(M\,DS_h(\varphi,f)\psi,\psi)_{h_i} = \sum_{i=1}^{N}\int_{\Omega_i}\nabla v_{ih}\nabla\theta_{ih}(\psi) + 3u_{ih}^2(\varphi,f)v_{ih}\theta_{ih}(\psi)\,dx. \tag{3.7}$$

From the identity

$$\sum_{i=1}^{N}\int_{\Omega_i}\nabla\Big(v_{ih}-\theta_{ih}(\psi)\Big)\nabla\eta_{ih} + 3u_{ih}^2(\varphi,f)v_{ih}\eta_{ih}\,dx = 0\,,\forall\eta_{ih}\in V_{ih}^0, \tag{3.8}$$

written with $\eta_{ih} = v_{ih} - \theta_{ih}(\psi) \in V_{ih}^0$ we get

$$\begin{array}{l}\sum_{i=1}^{N}\int_{\Omega_i}\nabla v_{ih}\nabla\theta_{ih}(\psi) + 3u_{ih}^2(\varphi,f)v_{ih}\theta_{ih}(\psi)\,dx =\\ \sum_{i=1}^{N}\int_{\Omega_i}|\nabla v_{ih}|^2 + |\nabla\theta_{ih}(\psi)|^2 - \nabla v_{ih}\nabla\theta_{ih}(\psi) + 3u_{ih}^2(\varphi,f)v_{ih}^2\,dx.\end{array} \tag{3.9}$$

The identity $0 \le \frac{1}{2}a^2 + \frac{1}{2}b^2 + \frac{1}{2}(a-b)^2 = a^2 + b^2 - ab$ implies that the right hand side of (3.9) is bounded from below by $\sum_{i=1}^{N}\int_{\Omega_i}|\nabla\theta_{ih}(\psi)|^2\,dx$. From Lemma 3.1 we have the desired estimate. Lemma 3.3 is proved. ∎

Proof. **of Theorem 3.1** We show that $DG_\rho$the derivative of G_ρ is bounded by a constant less than one in a neighborhood of φ. It is well known that for an $0 < \delta$ given, there exists a norm $|||\cdot|||$ on W_h such that for the induced norm for the operators we have $|||DG_\rho(\varphi)||| \le \sigma\big(DG_\rho(\varphi)\big) + \delta$, where σ denotes the spectral radius. Lemma 3.2 and Lemma 3.3 imply that $M\,DS_h(\varphi,f)$ has positive eigenvalues in W_h uniformly bounded from below with respect to h. Thus we have

$$k = \sigma\Big(I - \rho M\,DS_h(\varphi,f)\Big) = 1 - \rho\sigma\Big(M\,DS_h(\varphi,f)\Big).$$

The stability of $DS_{ih}(\varphi,f)$ and $DS_{ih}^{-1}(0,0)$ provides the existence of $0 < \rho$ independent of h such that $k < 1$. Then a classical Banach fixed point theorem applies and thus Theorem 3.1 is proved. ∎

4. Numerical results. In this section we describe some numerical results obtained with the Preconditioned Modefied Newton (PMN) algorithm (3.1)-(3.3). This resulst are done for various mesh sizes and various numbers of subdomains in the case of nonmatching grids. the corresponding physical problem is the nonlinear elliptic problem $-\Delta u + u^3 = f \quad \text{in}\,\Omega = (0,1)\times(0,1)$ where the source term f is a Gaussian function centred at the point (1,1) and the Dirichlet boundary conditions are prescribed on the side $x_2 = 0$ (see Figure 4.1).

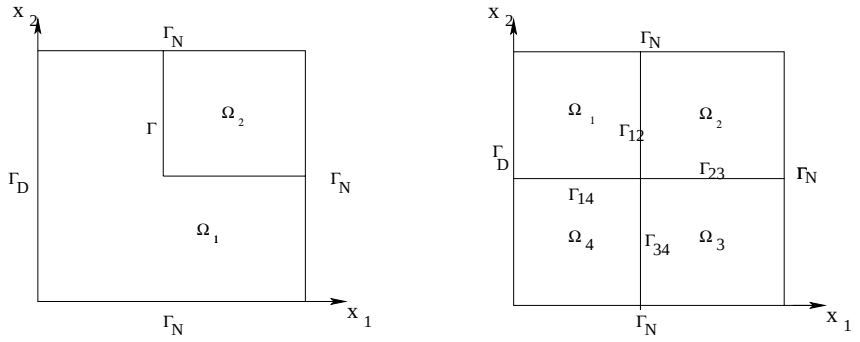


Figure 4.1: Decomposition in 2 and 4 subdomains

Remark 4.1 *The modified Newton algorithm requires on each subdomain the successive solution of a Dirichlet and of a Neumann problem (preconditioner). In the abscence of a Dirichlet boundary conditions on $\partial\Omega_i \setminus \Gamma$, the Neumann problem is not well-posed. In such situations, we replace in the factorization of the finite-element matrix of problem (3.2), all the singular pivots by an averaged strictly positive pivot.*

First we present some numerical results obtained with the PMN algorithm (3.1)-(3.3) in two-domains case with a fixed value of the relaxation parameter ρ . For the optimal value of the relaxation parameter ρ, the results could be different. Next, the obtained results with Newton Preconditioned Conjugate Gradient (Newton-PCG) algorithm for the same test case are given.

In Table 1 the number of iterations necessary for Modified Newton iterations to converge (with a level of precision of 10^{-6}), and the values of parameter ρ are reported as functions of degrees of freedom. Please remark that the number of iterations for reaching convergence with a constant ρ are independent of h.

d.o.f in $\Omega_1 \cup \Omega_2$	ρ	*number of iter.*
102	0.16	32
354	0.155	34
1314	0.16	34

Table 1: evolution of ρ and the number of iterations of Modified Newton algorithm

d.o.f in $\Omega_1 \cup \Omega_2$	*Newton iter.*	*PCG iter. on* Γ	*total nb. of iter.*
102	6	6	36
354	6	6	36
1314	7	6	42

Table 2: evolution of the number of iterations of Newton-PCG algorithm.

Table 2 shows that the Newton-PCG algorithm converges (with a level of precision of 10^{-6} for the Newton algorithm and for the PCG algorithm) at a rate which is independent of the mesh size h.

Modified Newton algorithm is proved to converge independently of the discretization step, which is confirmed by our numerical tests. Moreover, the potential parallelism offered by this algorithm is easy to exploit on the contrary of the Newton-PCG. Nevertheless, its practical implementation still faces the problem of the optimal choice of the parameter ρ.

We have tested the dependency over h in the case where Ω is decomposed into four geometrically identical subdomains (see Figure 4.1). There is a slight dependence on h due to the presence of cross points in our decomposition (see Table 3).

step	*nb. of iter.*	ρ
h	64	0.15
h/2	68	0.14

Table 3: Test over the mesh size h (**p=4**)

Here, we study the convergence rate of the PMN algorithm (3.1)-(3.3) with respect to the number of subdomains p. We consider the case where the domain Ω has been decomposed into two and four subdomains (see Figure 4.1). The number of degrees of freedom in Ω varies with p because each interface node is treated in our approach as two independent nodes.

p	*number of iter.*	*d.o.f in* Ω
2	34	1314
4	60	1350

Table 4: Test over the number of subdomains **p**.

In terms of iteration count, Table 4 and show that the smaller the number of subdmains the faster the PMN convergence. Indeed, the diameter d of each subdomain has a direct influence on the condition number of our operator.

5. Conclusion. A Modified Newton method for a domain decomposed nonlinear elliptic problem has been introduced and studied. For a small number of subdomains and very fine grids, this approach leads to efficient numerical algorithm even in the case of nonmatching grids. Indeed with the choice of adequate preconditioners such as the one introduced in §3, the method is proved to converge independently of the discretization step, which is confirmed by our numerical tests. Nevertheless, the Preconditioned Modified Newton algorithm does not scale well with the diameter of the subdomains. The addition of an unstructured coarse grid solver when using decompositions with a large number of subdomains is actually under consideration.

REFERENCES

[1] V. I. Agoshkov. Poincaré-Steklov operators and domain decomposition methods in finite dimensional spaces. In R. Glowinski, G. H. Golub, G. A. Meurant, and J. Périaux, editors, *First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Philadelphia, PA, 1988. SIAM.

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

[3] C. Bernardi, Y. Maday, and A. T. Patera. A new non conforming approach to domain decomposition: The mortar element method. In H. Brezis and J.-L. Lions, editors, *Collège de France Seminar*. Pitman, 1994. This paper appeared as a technical report about five years earlier.

[4] P. De Roeck Yann-Hervé, Le Tallec and V. M. Marina. A domain-decomposition solver for nonlinear elasticity. *Comp. Meth. Appli. Mech. Eng.*, 99:187–207, 1992.

[5] M. Dryja and W. Hackbusch. On the nonlinear domain decomposition method. *BIT*, pages 296–311, 1997.

[6] P. Le Tallec and T. Sassi. Domain decomposition with nonmatching grids: Augmented Lagrangian approach. *Math. Comp.*, 64(212):1367–1396, 1995.

[7] J. Pousin and T. Sassi. Domain decomposition with non matching grids and adaptive finite element techniques. *East-West J. Numer. Math.*, 8:243–246, 2000.

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

[9] P. L. Tallec, T. Sassi, and M. Vidrascu. Three-dimensional domain decomposition methods with nonmatching grids and unstructured coarse solvers. In D. Keyes and J. Xu, editors, *the seventh international symposium on Domain Decomposition Methods for Partial Differential Equations.*, pages 133–139. AMS, 1994.

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

51. Singular Function Enhanced Mortar Finite Element

Xuemin Tu[1], Marcus Sarkis [2]

1. Introduction. We are interested in solving the following elliptic variational problem: Find $u^* \in H^1(\Omega)$, such that

$$\left\{ \begin{array}{lcll} a(u^*, v) & = & f(v) & \forall v \in H_0^1(\Omega) \\ u^* & = & u_0^* & \text{on } \partial\Omega \end{array} \right. , \tag{1.1}$$

where

$$a(u^*, v) = \int_\Omega \nabla u^* \cdot \nabla v \, dx \quad \text{and} \quad f(v) = \int_\Omega f v \, dx.$$

We assume the function $f \in L^2(\Omega)$. We also assume the function u_0^* has an extension $H^2(\Omega)$, which we denote also by u_0^*. We let the domain Ω to be the L-shaped domain in $\Re^2$ with vertices $V_1 = \{0, 0\}, V_2 = \{1, 0\}, V_3 = \{1, 1\}, V_4 = \{-1, 1\}, V_5 = \{-1, -1\}$, and $V_6 = \{0, -1\}$. It is well-known that the solution u^* of (1.1) does not necessarily belong to $H^2(\Omega)$ due to the nonconvexity of the domain Ω at the corner V_1, and therefore, standard finite element discretizations do not give second order accurate schemes. Theoretical and numerical work on corner singularity are very well-known and several different approaches were proposed [4, 2, 5, 6, 7, 8, 9, 10]; see the references therein. The main goal of the paper is to design and analyze optimal accurate finite element discretizations based on mortar techniques [1, 11] and singular functions [8, 7]. The proposed methods are variation of the methods described in Chapter 8 of [10] where a smoothed cut-off singular function is added to the space of finite elements. There, a smoothed cut-off function is applied to make the singular function to satisfy the zero Dirichlet boundary condition. Here, instead, we use mortar finite element techniques on the boundary of $\partial\Omega$ to force, in a weak sense, the boundary condition. As a result, accurate and general schemes can be obtained for which they do not rely on costly numerical integrations and linear solvers.

2. Notations. We next introduce some notations and tools.

2.1. Triangulation. Let $\mathcal{T}^h(\Omega)$ be a standard finite element triangulation of $\overline{\Omega}$. We assume the triangulation $\mathcal{T}^h(\Omega)$ to be shape regular and quasi-uniform with grid size of $O(h)$. Let $V^h(\Omega)$, also denoted by V^h, be the space of continuous piecewise linear functions on $\mathcal{T}^h(\Omega)$; note that we have not assumed the functions of V^h to vanish on $\partial\Omega$.

2.2. Singular Functions and Regularity Results. We note that the solution u^* of (1.1) does not necessarily belong to $H^2(\Omega)$ even if f and u_0^* are very smooth. For instance, consider the primal singular function defined by $\psi^+(r, \theta) = r^{\frac{2}{3}} \sin(\frac{2}{3}\theta)$. The function ψ^+ is smooth everywhere in Ω except near the non-convex corner V_1. It is easy to check that $\psi^+ \in H^{5/6-\epsilon}(\Omega)$ if, and only if, ϵ is positive and $-\triangle\psi^+ \equiv 0$ on Ω. We note that ψ^+ vanishes on the intervals $[V_1, V_2]$ and $[V_6, V_1]$, plus it is smooth on the remaining boundary of $\partial\Omega$.

Another function that will play an important role in our studies here is the dual singular function ψ^- defined as $\psi^-(r, \theta) = r^{-\frac{2}{3}} \sin(\frac{2}{3}\theta)$. We note that $-\Delta\psi^- \equiv 0$ and ψ^- vanishes on the intervals $[V_1, V_2]$ and $[V_6, V_1]$, and it is easy to check that $\psi^- \in H^{1/3-\epsilon}$ if, and only if, ϵ is positive.

[1]Graduate Student of the Mathematical Sciences Department, Worcester Polytechnic Institute, 100 Institute Rd, Worcester, MA 01619
[2]Instituto de Matemática Pura e Aplicada, Est. Dona Castorina, 110, Rio de Janeiro, RJ, CEP 22420-320, Brazil, sarkis@impa.edu

It is well-known [8, 7] that the solution of (1.1) has a unique representation

$$u^* = w_{u^*} + \lambda_{u^*}\psi^+, \tag{2.1}$$

where $w_{u^*} \in H^2(\Omega)$ and $\lambda_{u^*} \in \Re$, and the following regularity estimates hold:

$$\|w_{u^*}\|_{H^2(\Omega)} \leq C\left(\|f + \Delta u_0^*\|_{L^2(\Omega)} + \|u_0^*\|_{H^2(\Omega)}\right), \tag{2.2}$$

and

$$|\lambda_{u^*}| \leq C\|f\|_{L^2(\Omega)}. \tag{2.3}$$

2.3. Mortar Functions on the Boundary. The boundary of our domain is given by $\partial\Omega = \bigcup_{m=1}^{6} \overline{D}_m$, where the open segments D_m are given by the intervals $D_1 = (V_1, V_2)$, $D_2 = (V_2, V_3)$, $D_3 = (V_3, V_4)$, $D_4 = (V_4, V_5)$, $D_5 = (V_5, V_6)$, and $D_6 = (V_6, V_1)$. For each interval $\overline{D}_m$, the triangulation $\mathcal{T}_h(D_m)$ is inherited from the triangulation $\mathcal{T}_h(\Omega)$. Let us denote the space $W_h(D_m)$ as the trace of V_h to $\overline{D}_m$; i.e.

$$W_h(D_m) = \{v \in C(\overline{D}_m) : v = w(\overline{D}_m), w \in V_h\}.$$

We also denote the space $W_h^0(D_m)$ as the functions of $W_h(D_m)$ which vanish at the two end points of $\overline{D}_m$. Thus, $W_h^0(D_m) = W_h(D_m) \cap H_0^1(D_m)$. The number of degrees of freedom of $W_h^0(D_m)$ are the number of interior nodes of $\mathcal{T}_h(D_m)$ which are equal aslo the number of degrees of freedom of the Lagrange multiplier spaces $M_h(D_m)$. In this paper, in the numerical experiments, we adopt the dual biorthogonal functions introduced in [11]. We note that the theory presented here also holds for the old mortars [1]. For each edge D_m, the mortar projection operator $\Pi_m : C(\overline{D}_m) \longrightarrow W_h(D_m)$ is defined by

$$v - \Pi_m v \in C_0(D_m), \quad \text{and} \quad \int_{D_m} (v - \Pi_m v)\mu_m ds = 0, \quad \forall \mu_m \in M_h(D_m). \tag{2.4}$$

It can be shown [1, 11] that

$$\|v - \Pi_m v\|_{H_{00}^{1/2}(D_m)} \leq Ch\|v\|_{H^{3/2}(D_m)}, \ \forall v \in H^{3/2}(D_m), \tag{2.5}$$

and

$$\inf_{\mu_m \in M_h(D_m)} \|v - \mu_m\|_{(H^{1/2}(D_m))'} \leq Ch\|v\|_{H^{1/2}(D_m)}, \ \forall v \in H^{1/2}(D_m). \tag{2.6}$$

3. Singular Function Enhanced Mortar Finite Element. We define the discrete global space V_h^+ as follows:

$$V_h^+ = \{v = w + \lambda\psi^+ : w \in V_h, \quad \lambda \in \Re, \text{ and } \ \Pi_m v = 0, m = 1, \cdots, 6\}.$$

Functions of the space V_h^+ vanish at the vertices $V_k, k = 1, \cdots, 6$ and satisfy zero Dirichlet boundary condition (in the weak discrete sense) on the intervals $\overline{D}_m$. It is easy to see that the degrees of freedom of the space V_h^+ are the λ and the nodal values of w at the interior nodes of $\mathcal{T}_h(\Omega)$; the values of w on the $\overline{D}_m$ are obtained via $w = -\lambda\Pi_m\psi^+$.

We next introduce the new finite element formulation using the primal singular function ψ^+ and mortar techniques in order to obtain an approximation for u^*. We then introduce two second order accurate approximations for the stress intensive factor (SIF) λ_{u^*} based on the dual singular function ψ^-.

3.1. Finite Element Formulation. Let us define $u_0 \in V_h$ as $u_0 = \Pi_m u_0^*$ on D_m, $m = 1, \cdots, 6$, and zero nodal values at the interior nodes of $\mathcal{T}_h(\Omega)$. We define the singular function enhanced mortar finite element method as follows:

Find $u = w_u + \lambda_u \psi^+$ such that $u - u_0 \in V_h^+$ and

$$a(u, v) = f(v), \quad \forall v \in V_h^+. \tag{3.1}$$

We prove later in this paper that u is a second order approximation to u^*. We note however that the λ_u and w_u separately are not second order approximations of λ_{u^*} and w_{u^*}, respectively. So, in the next subsections, we introduce two algorithms for obtaining second order approximations for the stress intensive factor (SIF) λ_{u^*}.

3.2. Extraction of SIF through a Smoothed Cut-off Function. Define $f = -\triangle u^*$ and $f^- = -\triangle s^-$, where $s^- = \rho\psi^-$. Here, the smoothed cut-off function $\rho(r)$ is defined in the polar coordinate system as

$$\rho(r) = \begin{cases} 1 & 0 \le r \le \frac{1}{4} \\ -192r^5 + 480r^4 - 440r^3 + 180r^2 - \frac{135}{4}r + \frac{27}{8} & \frac{1}{4} \le r \le \frac{3}{4} \\ 0 & \frac{3}{4} \le r \end{cases}.$$

It is easy to check that the function ρ has two continuous derivatives. By applying Green's formula twice [9], we obtain

$$\lambda_{u^*} = \frac{\int_\Omega (fs^- - f^- u^*) + \int_{\partial\Omega} s^- \partial_n s^- - u_0^* \partial_n s^-}{\pi},$$

and by using that s^- vanishes on $\partial\Omega$ we have

$$\lambda_{u^*} = \frac{\int_\Omega (fs^- - f^- u^*) - \int_{\partial\Omega} s^- u_0^* \partial_n s^-}{\pi}. \tag{3.2}$$

The discrete stress intensity factor is obtained as follows. We first solve (3.1) to obtain $u = w_u + \lambda_u \psi^+$, and then we plug this u as u^* in (3.2) to define the discrete stress intensity factor as

$$\lambda_u^h = \frac{\int_\Omega (fs^- - f^- u) - \int_{\partial\Omega} u_0^* \partial_n s^-}{\pi}. \tag{3.3}$$

3.3. Extraction of SIF without a Smoothed Cut-off Function. Similarly, we can use the same approach above for ψ^- as s^-. Using $-\triangle\psi^- \equiv 0$, we obtain

$$\lambda_{u^*} = \frac{\int_\Omega f\psi^- - \int_{\partial\Omega} (u_0^* \partial_n \psi^- - \psi^- \partial_n u^*)}{\pi}. \tag{3.4}$$

We note that we do not know the value of $\partial_n u^*$ and therefore, the formula (3.4) is not applicable for defining the discrete stress intensity factor. We remark that an approximation of $\partial_n u^*$ can be obtained via the saddle point formulation [11] of (3.1) but unfortunately we cannot prove that this approximation is of second order. We next introduce a new method that does not require the knowledge of $\partial_n u^*$.

We modify ψ^- to $\tilde{\psi}^-$, where $\tilde{\psi}^-$ vanishes on the whole $\partial\Omega$, $\tilde{\psi}^-$ and ψ^- have the same singular behavior in a neighbourhood of the origin, and $-\Delta\tilde{\psi}^- \equiv 0$. This is done as follows. We first solve $\delta\psi^- \in H^1(\Omega)$ such that

$$\begin{cases} a(\delta\psi^-, v) = 0 & \forall v \in H_0^1(\Omega) \\ \delta\psi^- = \psi^- & \text{on } \partial\Omega. \end{cases} \tag{3.5}$$

Then, we define $\tilde{\psi}^- = \psi^- - \delta\psi^-$. We note that ψ^- has a H^2 extension to Ω and therefore, the solution of (3.5) is in the form of $\delta\psi^- = w_{\delta\psi^-} + \lambda_{\delta\psi^-}\psi^+$, where $w_{\delta\psi^-} \in H^2(\Omega)$. Hence, the singular behavior of $\tilde{\psi}^-$ near the origin is the same as of ψ^-, and we obtain

$$\lambda_{u^*} = \frac{\int_\Omega f\tilde{\psi}^- - \int_{\partial\Omega} u_0^* \partial_n \tilde{\psi}^-}{\pi}.$$

In the case the boundary value u_0^* vanishes on $\partial\Omega$, we have

$$\lambda_{u^*} = \frac{\int_\Omega f\tilde{\psi}^-}{\pi}. \tag{3.6}$$

We note that we do not know $\tilde{\psi}^-$ and therefore, a numerical approximation for $\tilde{\psi}^-$ must be obtained. We first define $\delta\psi_0^- \in V_h$ as $\delta\psi_0^- = \Pi_m\psi^-$ on the D_m and zero nodal values at the interior nodes of $\mathcal{T}_h(\Omega)$. We solve $\delta\psi_h^- - \delta\psi_0^- \in V_h^+$ such that

$$a(\delta\psi_h^-, v) = 0, \forall v \in V_h^+.$$

We let $\tilde{\psi}_h^- = \psi^- - \delta\psi_h^-$, and define the discrete stress intensity factor by

$$\hat{\lambda}_u^h = \frac{\int_\Omega f\tilde{\psi}_h^-}{\pi} = \frac{\int_\Omega f\psi^- - f\delta\psi_h^-}{\pi}. \tag{3.7}$$

We remark that $\hat{\lambda}_u^h$ can be obtained without computing the discrete solution u and can be used only if u_0^* vanishes on $\partial\Omega$.

4. Analysis. In this section we analyze the proposed methods. We will prove optimality accuracy errors of the discrete solution u on the L_2 and H_1 norms. We also show that the two proposed discrete stress intensive factor formulas given by (3.3) and (3.7) are both second order approximations for λ_{u^*}.

4.1. Uniform ellipticity. We note that $v \in V_h^+$ implies that v vanishes on D_1 and D_6. Therefore, using a standard Poincaré inequality, we have:

Lemma 4.1 *There exists a constant C that does not depend on h and v such that*

$$\|v\|_{H^1(\Omega)} \leq C|v|_{H^1(\Omega)}, \quad \forall v \in V_h^+. \tag{4.1}$$

4.2. Energy Discrete Error. We note that proposed discretization (3.1) is non-conforming since the space V_h^+ is not included in $H_0^1(\Omega)$; functions in V_h^+ vanishes on D_m, $m = 2, \cdots 5$ only in a weak sense. To establish H_1 apriori error estimate, we use the Cea's lemma (the second Strang lemma) for non-conforming discretization [3]. We obtain

$$\|u^* - u\|_{H^1(\Omega)} \leq \inf_{v \in u_0 + V_h^+} \|u^* - v\|_{H^1(\Omega)} + \sup_{z \in V_h^+} \frac{|a(u^*, z) - f(z)|}{\|z\|_{H^1(\Omega)}} =$$

$$\inf_{v \in u_0 + V_h^+} \|u^* - v\|_{H^1(\Omega)} + \sup_{z \in V_h^+} \frac{|\int_{\partial\Omega} z\partial_n u^* ds|}{\|z\|_{H^1(\Omega)}}. \tag{4.2}$$

The first term of (4.2) is the **best aproximation error** and the second term is the **consistency error**.

4.2.1. Best Approximation Error. We next establish that the best approximation error in the energy norm is of optimal order.

Lemma 4.2 *The best approximation error is of order h,*

$$\inf_{v\in u_0+V_h^+} \|u^*-v\|_{H^1(\Omega)} \le Ch\left(\|f+\Delta u_0^*\|_{L^2(\Omega)}+\|u_0^*\|_{H^2(\Omega)}\right). \tag{4.3}$$

Proof. Let $\tilde{v}$ be defined as

$$\tilde{v} = I_h(u^*-\lambda_{u^*}\psi^+)+\lambda_{u^*}\psi^+,$$

where I_h is the standard pointwise interpolator on V_h. Note that the interpolation is well defined since the function $w_{u^*} = u^*-\lambda_{u^*}\psi^+$ belongs to $H^2(\Omega)$ and therefore, w_{u^*} is a continuous function. The function $\tilde{v}-u^*$ belongs to $H_0^1(D_m)$ and does not satisfy the mortar condition. We next modify $\tilde{v}$ to v to make u^*-v to satisfy the mortar condition (2.4). This is done by $v=\tilde{v}+\sum_{m=1}^6 \mathcal{H}_m\Pi_m(u^*-\tilde{v})$, where the operator $\mathcal{H}_m$ denote the V_h-discrete harmonic extension function with boundary values given on $\overline{D}_m$ and zero on $\partial\Omega\backslash D_m$. In addition, it is easy to check that $v\in u_0+V_h^+$. We have

$$\|u^*-v\|_{H^1(\Omega)} = \|w_{u^*}-I_hw_{u^*}\|_{H^1(\Omega)}+\|\sum_{m=1}^6 \mathcal{H}_mQ_m(u^*-\tilde{v})\|_{H^1(\Omega)}. \tag{4.4}$$

For the first term of (4.4), we use a standard approximation result on pointwise interpolation and (2.2) to obtain

$$\|w_{u^*}-I_hw_{u^*}\|_{H^1(\Omega)} \le Ch\|w_{u^*}\|_{H^2(\Omega)} \le Ch\left(\|f+\Delta u_0^*\|_{L^2(\Omega)}+\|u_0^*\|_{H^2(\Omega)}\right).$$

For the second term of (4.4), we use properties of discrete harmonic extensions and $H_{00}^{1/2}$-norm, and the approximation result (2.5) to obtain

$$\begin{aligned}\|\sum_{m=1}^6 \mathcal{H}_m\Pi_m(u^*-\tilde{v})\|_{H^1(\Omega)} &\le C\sum_{m=1}^6 \|\mathcal{H}_m\Pi_m(u^*-\tilde{v})\|_{H^1(\Omega)}\\ &\le C\sum_{m=1}^6 \|\Pi_m(u^*-\tilde{v})\|_{H_{00}^{1/2}(D_m)} \le C\sum_{m=1}^6 \|u^*-\tilde{v}\|_{H_{00}^{1/2}(D_m)}\\ &\le Ch\|u_0^*\|_{H^{3/2}(D_m)} \le Ch\|u_0^*\|_{H^2(\Omega)}.\end{aligned}$$

■

4.2.2. Consistency Error. We next establish that the consistency error is of optimal order.

Lemma 4.3 *The consistency error is of order h*

$$\sup_{z\in V_h^+} \frac{|\int_{\partial\Omega}\partial_n u^* z ds|}{\|z\|_{H^1(\Omega)}} \le Ch\left(\|f\|_{L^2(\Omega)}+\|f+\Delta u_0^*\|_{L^2(\Omega)}\right). \tag{4.5}$$

Proof. We remark that $z\in V_h^+$ implies that z vanishes on $\overline{D}_1$ and $\overline{D}_6$. Therefore,

$$\int_{\partial\Omega} z\partial_n u^* ds = \sum_{m=2}^5 \int_{D_m} z\partial_n u^* ds.$$

By the definition of V_h^+, we have $\int_{D_m} z\mu_m ds = 0, \quad \mu_m \in M_h(D_m)$. Thus,

$$\sum_{m=2}^{5} \int_{D_m} z\partial_n u^* ds = \sum_{m=2}^{5} \int_{D_m} z(\partial_n u^* - \mu_m) ds, \quad \forall \mu_m \in M_h(D_m),$$

and using duality arguments we obtain

$$\sum_{m=2}^{5} |\int_{D_m} z\partial_n u^* ds| \leq C\|z\|_{H^{1/2}(D_m)} \inf_{\mu_m \in M_h(D_m)} \|\partial_n u^* - \mu_m\|_{(H^{1/2})'(D_m)}.$$

Let us denote $\Omega_{1/4} = \Omega \cap \{r^2 = x^2 + y^2 \leq 1/16\}$, and $\Omega^c_{1/4} = \Omega \backslash \Omega_{1/4}$. Since $\psi^+ \in H^2(\Omega^c_{1/4})$, we have $u^* \in H^2(\Omega^c_{1/4})$, and therefore we can use a trace theorem to obtain $\partial_n u^* \in H^{1/2}(D_m), m = 2, \cdots, 5$. We then use approximation property (2.6), a trace result, and the regularity estimates (2.2) and (2.3) to obtain

$$\inf_{\mu_m \in M_h(D_m)} \|\partial_n u^* - \mu_m\|_{(H^{1/2})'(D_m)} \leq Ch\|\partial_n u^*\|_{H^{1/2}(D_m)} \leq Ch\|u^*\|_{H^2(\Omega^c_{1/4})}$$

$$\leq Ch(|\lambda_{u^*}|\|\psi^+\|_{H^2(\Omega^c_{1/4})} + \|w_{u^*}\|_{H^2(\Omega)}) \leq Ch\left(\|f + \Delta u_0^*\|_{L^2(\Omega)} + \|u_0^*\|_{H^2(\Omega)}\right).$$

We finally use that $\|z\|_{H^{1/2}(D_m)} \leq C\|z\|_{H^1(\Omega)}$ to obtain (4.5). ∎

4.3. Error in the L^2-norm. We also obtain an optimal error estimates in $L^2(\Omega)$-norm for the problem (1.1).

Lemma 4.4 *The L_2 discrete error is of order h^2*

$$\|u^* - u\|_{L^2(\Omega)} \leq Ch^2\left(\|f + \Delta u_0^*\|_{L^2(\Omega)} + \|u_0^*\|_{H^2(\Omega)}\right). \tag{4.6}$$

Proof. The proof follows easily from an Aubin-Nitche trick argument and by the fact that the enhanced space V_h^+ is used both as the solution space as well as the test function space for (3.1). ∎

4.4. Stress Intensive Factor Error. The apriori error estimate for stress intensive factor errors $|\lambda_{u^*} - \lambda_u^h|$ with λ_u^h defined on (3.3), and $|\lambda_{u^*} - \hat{\lambda}_u^h|$ with $\hat{\lambda}_u^h$ defined on (3.7) for the case $u_0^* \equiv 0$, will follow easily from the L_2-error estimates.

Lemma 4.5 *If $f \in L^2(\Omega)$, then the recovering formula (3.3) gives h^2 accuracy*

$$|\lambda_{u^*} - \lambda_u| \leq Ch^2\left(\|f + \Delta u_0^*\|_{L^2(\Omega)} + \|u_0^*\|_{H^2(\Omega)}\right).$$

Proof. We subtract (3.3) from (3.2) and we obtain

$$|\lambda_{u^*} - \lambda_u| = |\frac{\int_\Omega f^-(u - u^*)}{\pi}| \leq \|f^-\|_{L^2(\Omega)}\|u - u^*\|_{L^2(\Omega)}.$$

The lemma follows from the Lemma 4.4 and the smoothing properties of the smoothed cut-off function ρ. ∎

Using similar arguments we obtain:

Lemma 4.6 *If $f \in L^2(\Omega)$ and $u_0^* \equiv 0$, then the recovering formula (3.7) gives h^2 accuracy*

$$|\lambda_{u^*} - \lambda_{u^*}^h| \leq Ch^2\|f\|_{L^2(\Omega)}.$$

Table 5.1: Results with $f = -\triangle s^+ - \triangle s_2^+ + 6x(y^2 - y^4) + (x - x^3)(12y^2 - 2)$

k	$\lambda^k - 1$	σ^k	$1 - \hat{\lambda}^k$	$\hat{\sigma}^k$	e_2^k	ϵ_2^k	e_1^k	ϵ_1^k
2	2.967e-1	–	2.698e-3	–	7.512e-2	–	9.032e-1	–
3	9.457e-2	1.6497	6.914e-4	1.9642	2.415e-2	1.6380	5.027e-1	0.8454
4	2.651e-2	1.8349	1.673e-4	2.0474	6.805e-3	1.8275	2.673e-1	0.9115
5	6.862e-3	1.9497	4.083e-5	2.0152	1.764e-3	1.9475	1.361e-1	0.9736
6	1.730e-3	1.9873	1.006e-5	2.0216	4.454e-4	1.9858	6.839e-2	0.9928
7	4.341e-4	1.9952	2.550e-6	1.9832	1.116e-4	1.9958	3.424e-2	0.9980
8	1.085e-5	1.9996	6.290e-7	2.0154	2.794e-5	1.9991	1.713e-2	0.9994

5. Numerical Experiments. An advantage of the proposed methods is in the construction of the stiffness matrix of (3.1). Its construction requires few work on numerical integrations since we do integrations by parts on $a(\psi^+, \varphi_i)$ or $a(\psi^+, \psi^+)$. Here the function φ_i stands for a nodal basis function of V_h. The only integrations that cannot be done exact are on $D_m, m = 2, \cdots, 5$. There, the singular function is very smooth and therefore easy in in numerical integrations.

In the set of experiments, we solve the discrete Poisson equation (3.1) with $f = -\triangle s^+ - \triangle s_2^+ + 6x(y^2 - y^4) + (x - x^3)(12y^2 - 2)$. Hence, the exact solution is $u = s^+ + s_2^+ + (x - x^3)(y^2 - y^4)$. Here, $s^+ = \rho(r)\psi^+$ and $s_2^+ = \rho(r)\psi_2^+$, where ψ_2^+ is the next singular function associated to the problem (1.1); i.e. $\psi_2^+ = r^{4/3}\sin(4/3\theta)$. The integer k is the level of refinement of the mesh; $k = 0$ is a mesh with 2 triangles per quadrant. The L^2 norm (H^1 semi-norm) discretization error on the kth level mesh is given by $e_2^k = \|u - u^*\|_{L^2(\Omega)}$ ($e_1^k = |u - u^*|_{H^1(\Omega)}$). The discrete stress intensity factor are given by $\lambda^k = \lambda_u^h$ and $\hat{\lambda}^k = \hat{\lambda}_u^h$. In our example, $\lambda_{u^*} = 1$. We also measure the rate of convergences for the four discrete errors given by

$$\sigma^k = \log_2(\frac{|\lambda^{k-1} - 1|}{|\lambda^k - 1|}),\ \hat{\sigma}^k = \log_2(\frac{|\hat{\lambda}^{k-1} - 1|}{|\hat{\lambda}^k - 1|})\ \epsilon_2^k = \log_2(\frac{e_2^{k-1}}{e_2^k}),\ \text{and}\ \epsilon_1^k = \log_2(\frac{e_1^{k-1}}{e_1^k}).$$

The numerical experiments confirm the theory showing optimality of the proposed algorithms and show that the recovering formula (3.7) is very accurate.

Acknowledgements: The work was supported in part also by the NSF grant CCR-9984404 and PRH-ANP/MME/MCT 32.

REFERENCES

[1] C. Bernardi, Y. Maday, and A. T. Patera. A new non conforming approach to domain decomposition: The mortar element method. In H. Brezis and J.-L. Lions, editors, *Collège de France Seminar*. Pitman, 1994. This paper appeared as a technical report about five years earlier.

[2] H. Blum and M. Dobrowolski. On finite element methods for elliptic equations on domains with corners. *Computing*, 28:53–63, 1982.

[3] D. Braess. *Finite Elements: Theory, Fast Solvers, and Applications in Solid Mechanics*. Cambridge University Press, Cambridge, 1997.

[4] S. Brenner. Overcoming corner singularities using multigrid methods. *SIAM J. Numer. Anal.*, 35:1883–1892, 1998.

[5] Z. Cai and S. Kim. A finite element method using singular function for the Poisson equation with corner singularities. *Siam J. Numer. Anal.*, 39:286–299, 2001.

[6] M. Dauge. *Elliptic boundary value problems on corner domains*. Lecture Notes in Mathematics. Springer Verlag, Berlin, 1988.

[7] P. Grisvard. *Elliptic problems in nonsmooth domains*. Pitman Publishing, Boston, 1985.

[8] V. Kondratiev. Boundary value problem for elliptic equations in domains with conical or angular points. *Transact. Moscow Math. Soc.*, 16:227–313, 1967.

[9] D. Leguillon and E. Sanchez-Palencia. *Computation of Singular Solution in Elliptic Problems and Elasticity*. John Wiley and Sons and Masson, 1987.

[10] G. Strang and G. J. Fix. *An Analysis of the Finite Element Method*. Prentice-Hall, Englewood Cliffs, N.J., 1973.

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

Fourteenth International Conference on Domain Decomposition Methods
Editors: Ismael Herrera , David E. Keyes, Olof B. Widlund, Robert Yates

52. A domain decomposition strategy for the numerical simulation of contaminant transport in pipe networks

V.G. Tzatchkov[1], A.A. Aldama[2], F.I. Arreguin[3]

1. Introduction. In this paper, a very efficient domain decomposition strategy is proposed for the numerical simulation of advective-dispersive-reactive (ADR) processes in pipe networks. The problem is modeled by applying the ADR equation in each pipe, as well as boundary conditions at each of the network nodes. In order to numerically solve the ADR equation, each pipe is discretized by means of a finite difference scheme. The presence of the dispersion term, and the inclusion of the boundary conditions at each network node, often produces large, unsymmetric and unstructured systems of linear equations. These systems of equations must be solved at each time step considered in the simulation, leading to significant computational costs. The proposed domain decomposition technique is based on the use of numerically computed Green functions and nodal mass balance considerations. Thus, the large system of equations that represents the discretized network is decomposed exactly in three easy-to-solve tridiagonal systems that represent the ADR processes for each pipe, and one low order system for the concentration at the pipe junctions. In each pipe the sought solution is represented by the superposition of three numerically obtained auxiliary solutions: a homogeneous (zero boundary conditions) solution, and two Green function solutions (one for each reach end) multiplied by the unknown values of the constituent concentration at the two reach ends. To obtain the Green functions corresponding to each reach end, a unit value for the concentration is imposed at one boundary and a value of zero at the other, and the resulting tridiagonal system is numerically solved. The fluxes at each of the pipe ends are expressed in terms of the values of the concentration there. Henceforth, continuity balance relations are used to construct a system of linear equations for the values of the unknown quantities at the network nodes. The method is applicable to any type of network, branched or looped.

Computer-based mathematical models able to predict the time history and the spatial distribution of constituents in water distribution networks are useful in network design and operation. Such models can be used to analyze water quality degradation problems, to assess alternative operational and control strategies for improving and maintaining water quality, to design water-quality-sampling programs, to optimize disinfection processes and to evaluate water quality aspects of distribution network improvement projects. Several authors have proposed models of this type that consider advection and reaction and neglect dispersion [4], [3]; and several computer programs that implement such models are available [7], [3]. Field observations conducted in distribution networks [8], [3], have shown that the advection-reaction model predictions are in good agreement with the observed concentrations in pipes with medium and high flow velocities, but fail in dead-end pipes where low velocities prevail.

While relatively simple Lagrangian tracking explicit-type numerical algorithms are used in the network advection-reaction models, more complicated numerical solutions have to be applied when dispersion is to be considered. The numerical solution for advection-dispersion-reaction in networks poses three main problems:

a) Boundary conditions at the nodes common to several domains have to be formulated and considered in the numerical solution.

b) The direct application of the numerical schemes produces large non-banded, unsymmetric and unstructured systems of equations to be solved, especially when the network is

[1]Mexican Institute of Water Technology, velitchk@tlaloc.imta.mx
[2]Mexican Institute of Water Technology, aaldama@tlaloc.imta.mx
[3]Mexican Institute of Water Technology, farreguin@sgt.cna.gob.mx

large.

c) The computational difficulties increase when advection dominates over dispersion. Sharp concentration gradients are expected in this case, and a very fine discretization would be needed if Eulerian methods are to be applied, which makes them impractical. Because of the small values of the dispersion coefficient, contaminant transport in water distribution networks falls exactly in this category of advection-dominated problems.

In this paper a numerical solution for the advection-dispersion-reaction equation in pipe networks is presented with special emphasis on the domain decomposition strategy used to efficiently solve the resulting finite difference equations. More information about the rest of the solution procedure can be found in [2], [10], [10], [12] and [11]. An Eulerian-Lagrangian numerical scheme is applied. The solution is applicable to advection-dominated and dispersion-dominated transport and is stable for a broad range of flow velocities that can be met in real distribution networks. The model is applied to simulate the variation of fluoride and chlorine concentration in a real distribution network and its predictions are compared to field observations and to the EPANET computer program that considers advection-reaction only.

2. Problem statement. The non-steady advection-dispersion-reaction process in a pipe flowing full is described by the following partial differential equation:

$$\frac{\partial C}{\partial t} + u\frac{\partial C}{\partial x} = D\frac{\partial^2 C}{\partial x^2} - KC \tag{2.1}$$

where C = constituent concentration; u = cross-sectional average flow velocity; D = dispersion coefficient; K = first order decay constant; x= distance along pipe; and t = time.

The following boundary conditions hold at the network nodes:

a) At some nodes, as constituent sources, the concentration C is given as a prescribed function of time.

b) Mixing at the network nodes. Two or more pipes, each of them with different flow and constituent concentration, may convey inflow to a node. Water is mixed at the node and a new concentration is obtained, then water leaves the node with that concentration to the outflowing pipes and to the consumption abstracted. A complete and instantaneous mixing is usually assumed in the network models.

c) Mass conservation at the network nodes. The well-known differential equation 2.1, is obtained by applying the mass conservation principle to an elementary pipe segment with a length dx in a time period dt, such that $dx = u.dt$. The same derivation can be generalized to the case of a junction where several pipes meet. Given that the flow velocity u_j in each pipe j is different, the segment length of each pipe considered in the elementary node volume will be different in order to handle the balance of the incoming and outcoming quantities in the same period of time dt, so that $dx_j = u_j.dt$. The following nodal equivalent of equation 2.1 is thus obtained:

$$\sum_{j=1}^{m}\left(\frac{dx_j}{2}A_j\right)\frac{\partial C}{\partial t} = \sum_{j=1}^{m}\left(A_j D_j\frac{\partial C}{\partial x} + Q_j C - K_j A_j C\frac{dx_j}{2}\right) - q_j C \tag{2.2}$$

where m = number of pipes connected to the node; A_j= cross sectional area of pipe j; Q_j = flow rate in pipe j; D_j= dispersion coefficient of pipe j; K_j= first order decay constant of pipe j; and q_i= flow rate abstracted at the node. For the case where two pipes of equal characteristics meet at the node and q_i=0, the equation 2.2 reduces to the equation 2.1 if dx_j and dt tend to be infinitesimally small.

d) Mass balance at the storage tanks connected to the network:

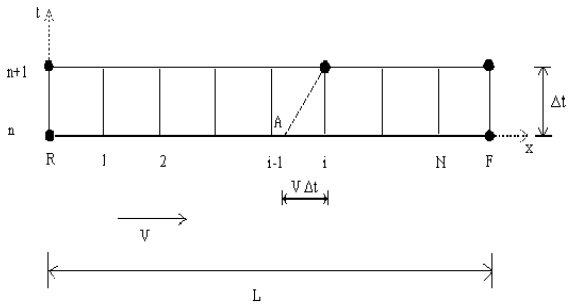


Figure 3.1: Discretization in a pipe

3. Numerical solution. In order to numerically solve eqn 2.1 for each pipe with the given boundary conditions at the network nodes, a two-stage Eulerian-Lagrangian solution is employed [2]. The space-time domain (x, t) of each pipe is discretized in a rectangular grid with time step Δt_q and gridsize Δx. The interior points are numbered from 1 to N, and the two pipe ends are called *rear node*, R, and *front node*, F (Figure 3.1). For each time segment considered in the numerical solution, the values of C for the points on the time level t^n are known and the values of C for the points on the time level t^{n+1} are to be computed. The differential equation 2.1 is split in two parts, an advective part and a dispersive part, and numerically solved in each time step in two stages.

3.1. Lagrangian stage. In the first (Lagrangian) stage the advective (or advective-reactive) part of the equationn 2.1, i.e.,

$$\frac{C^a - C^n}{\Delta t_q} = -\left(u\frac{\partial C}{\partial x}\right) \tag{3.1}$$

is solved for each pipe. The backward method of characteristics is used [5]. The points of the level t^{n+1} are projected backwards in time on the characteristic lines that pass through them until the characteristic lines cross the time level t^n. For the point i shown in Figure 3.1, for example, the projected point is A. Because of the pure advection nature of eqn 3.1, the value of C for point i at time t^{n+1} will be the same as that for point A, and can be found by interpolation between the known values for the time level t^n. The solution obtained is denoted by C^a.

This procedure is used to compute C^a for points 1 to F within each pipe. To compute C^a for the point R (the rear end of the pipe), the mass balance of the inflowing pipes connected to the same point (which is a network node) is considered assuming complete mixing. Thus the concentration at a network node i, C_i, is computed as:

$$C_i^a = \frac{\sum (QC_F^a)}{\sum Q_{out} + q_i} \tag{3.2}$$

where q_i = flow rate abstracted at the node; Q_{out} = flow rate in a pipe outflowing to the node; C^a = concentration computed in the advection stage; (QC_F^a) = flow rate in an inflowing pipe multiplied by the value of C^a for the front pipe end, F. To simulate the effect of a tank connected to the network, the concentration inside the tank is computed assuming complete mixing of the inflowing mass and considering bulk water decay.

3.2. Eulerian stage. The dispersion term is considered at this stage by numerically solving the differential equation:

$$\frac{C^{n+1}-C^a}{\Delta t_q}=\left(uD\frac{\partial^2 C}{\partial x^2}\right)^{n+1} \tag{3.3}$$

using the values of C^a calculated in the Lagrangian stage as initial conditions. Several well established numerical methods exist for the solution of eqn 3.3 in a single domain. Little attention has been given nevertheless to the application of these methods to interconnected domains such as pipeline networks. The direct application of a finite difference approximation to eqn 3.3 in a network of pipes produces a non banded system of linear equations, not amenable for an efficient numerical solution. This problem is aggravated when the network is large. Additionally, the condition given by eqn 2.2 has to be met. The application of known numerical solutions to problems in networks addresses these two difficulties with some *ad hoc* and particular procedure, due to the lack of a general approach. To overcome these difficulties, [1] proposed an approach for the consideration of the boundary conditions and for an efficient numerical solution of boundary value problems in networks, called *the numerical Green's function technique*. This technique is applied here in order to obtain an efficient numerical solution.

By definition, a Green's function is a one parameter P function that obeys the given differential equation, having the value of 1 at point P and value zero at the boundaries. It is common to denote a Green's function as $G(\xi, P)$ where ξ is the running coordinate and P is the point of application of the unit load. The reader is referred to [9] or [6] for a formal definition and theory of the Green's functions.

Using Green's functions, the sought solution of eqn 2.2 inside each network pipe, can be expressed as

$$C(\xi)=H(\xi)+G(\xi,R)C_R+G(\xi,F)C_F \tag{3.4}$$

where C_R and C_F is the (still unknown) concentration at the two pipe ends R and F respectively, $H(\xi)$ is a function that obeys eqn 3.4 with C_R=0 and C_F=0 (the so called *homogenous solution*), $G(\xi,R)=G(\xi,0)$ is the rear end Green's function and $G(\xi,F)=G(\xi,L)$ is the front end Green's function. $C(\xi)$, given by by eqn 3.4, is the sought solution of eqn 3.3 because each of the functions $H(\xi), G(\xi,R)$ and $G(\xi,F)$ obeys eqn 3.3 (the sum of any number of particular solutions of a linear differential equation is also a solution), and their sum satisfies the boundary conditions at the pipe ends. The first term on the right hand side of eqn 3.4 accounts for the initial conditions, and the other two for the boundary conditions.

$H(\xi), G(\xi,R)$ and $G(\xi,F)$ are easy to obtain numerically in each pipe. Expression 3.4 is then substituted in the balance condition at the network nodes (such as eqn 2.2) resulting in a system of equations for the values of C at the network nodes. After obtaining the values of C at the network nodes from the solution of this system, the values of C at the interior points are computed by eqn 3.4. This way the large system of equations produced by the finite difference scheme is decomposed into three easy-to-solve systems for each pipe and one (much smaller) system for the concentration at the pipe junctions.

The finite difference approximation for equation 3.3 for the interior points of a pipe can be written in the following form:

$$-\frac{\lambda}{2}C_{j-1}^{n+1}+(1+\lambda)C_j^{n+1}-\frac{\lambda}{2}C_{j+1}^{n+1}=b_j,\quad j=1,\ldots,N \tag{3.5}$$

where

$$\lambda=\frac{D\Delta t_q}{\Delta x^2};\quad b_j=\tfrac{\lambda}{2}C_{j-1}^a+(\lambda-1)C_j^a+\tfrac{\lambda}{2}C_{j+1}^a \tag{3.6}$$

Expression 3.5 represents a system of N equations with $N+2$ unknowns, that are the values of C at the points 1 to N, R and F. This system cannot be solved directly (separately) for each pipe because the number of unknowns is larger than the number of equations. Instead, the following procedure is used:

First, C is set to zero on the two pipe ends and the system of equations is solved numerically. The resulting solution vector represents *the homogeneous solution* $H(\xi)$ for the pipe and accounts for initial conditions. Then C is set to 1 on the rear reach end, C is set to zero on the front end, all b_j in eqn 3.5 are set to zero, and the system of equations is solved numerically. This way the rear end Green's function $G(\xi, R)$ is obtained. After that C is set to zero the rear reach end, C is set to 1 on the front end, all b_j in 3.5 are set to zero, and once again the system of equations is solved to obtain the front end Green's functions $G(\xi, F)$. Since 3.5 with b_j=0 is invariant under changing the order of the equations, the last two functions are symmetric, i.e., $G(\xi, R) = G(L-\xi, R)$, where L = pipe length; so the system needs to be solved only for one of them. Thus the desired solution for C is expressed as a superposition of the homogeneous solution and the two Green's functions multiplied by the still unknown values of C_R and C_F at the pipe ends, according to eqn 3.4, i.e.,

$$C_i = CH_i + GR_i C_R + GF_i C_F \quad i = 1, \ldots, N \tag{3.7}$$

where h_j = homogeneous solution for the point j; GR_j = the rear pipe end Green's function; GF_j = front pipe end Green's function; N = number of points inside the pipe, and C_R and C_F =unknown values of the desired solution for the rear and front pipe ends.

At each network node i, a unique and continuous value (the same for the ends of the pipes that join at that node) for C is supposed, say C_i. To obtain this value for the network nodes (and thus the values of C_R and C_F at the pipe ends) the equation 2.2 is used. The term $\sum_{j=1}^{m} Q_j C - q_j C$ in this equation is considered in the Lagrangian stage, so the equation can be written in finite difference form as:

$$\left(\sum_{j=1}^{m} \frac{dx_j}{2} A_j\right) \frac{C_i^{n+1} - C_i^a}{\Delta t_q} = \sum_{j=1}^{m} \left[\frac{A_j D_j}{2\Delta x_j}\left(C_{1,j}^{n+1} - C_i^{n+1} + C_{1,j}^a - C_i^a\right) - \frac{dx_j}{2} K_j A_j C_i^a\right] \tag{3.8}$$

where i denotes the network node; and $1, j$ denotes the discretization point of the pipe nearest to the network node i (which can be 1 or N depending on the numbering direction within the pipe). The value of $C_{1,j}^{n+1}$ in this equation is still unknown for each pipe j and can be expressed by eqn 3.7 thus involving the unknown values of C at the two pipe ends (which are two network nodes). Equation 3.8, written for each network node i in turn, provides a system of linear equations for the values of C at the network nodes. Once the system of equations is solved, the values of C for the intermediate points along the pipe are computed using eqn 3.7.

This way the homogeneous solution and the two numerical Green's functions in each pipe are computed from tri-diagonal systems of linear equations. The same matrix is used in each of them. The two Green's functions are symmetric, so only one of them needs to be computed. The matrix of the system of equations for the network nodes is symmetric and sparse, and reflects the structure of the network itself: for each network node there is a row in the matrix whose non zero elements correspond to the diagonal and to the nodes to which the current node is connected. Efficient sparse matrix algorithms can be applied to store and to solve systems of equations with this type of matrix. Thus the large non banded systems of equations that would otherwise produce the direct application of the finite difference scheme for the network, is decomposed in three easy-to-solve systems for each network reach and a

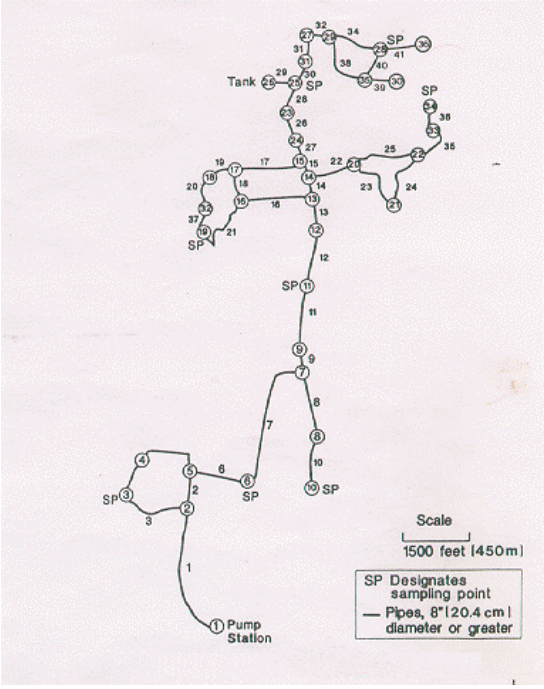


Figure 4.1: Node-link representation of the Brushy Plain-Cherry Hill networks

much smaller sparse system for the network nodes, and the solution can be computed more efficiently, especially for large networks.

4. Comparison with known models and field measurements. A public domain computer program for simulating the network hydraulics and contaminant transport in water distribution networks, called EPANET was developed by the US Environmental Protection Agency (EPA) [7]. The program uses an advection-reaction contaminant transport model. The proposed advection-dispersion-reaction model was applied to simulate the fluoride and chlorine transport in the Cherry Hill Brushy Plains service area network, for which a series of field measurements was carried out by the EPA in order to compare the observed concentration with the predictions of the EPANET model [8]. Figure 4.1 shows the node-link representation of the network and the sampling points where fluoride and chlorine concentration was measured.

The predictions of the EPANET model compare fairly well with the field measurements of fluoride concentration for sampling points 3, 6, 11, 19 and 25; but for sampling points 10, 28 and 34 the model fails to represent correctly the trend of concentration evolution, as can be seen in the corresponding graphics presented by [8]. Figure 4.2 shows the results for sampling point 10 with a $D = 0.20$ m2/s in pipes 8 and 10. It is seen that the proposed advection-diffusion model represents more realistically the concentration evolution thanks to the inclusion of dispersion.

5. Summary and conclusions. An Eulerian-Lagrangian numerical solution for the non-steady advection-dispersion-reaction constituent transport in water distribution networks is proposed. The solution employs the numerical Green's function technique to efficiently solve the system of linear equations produced by the numerical scheme in the Eulerian

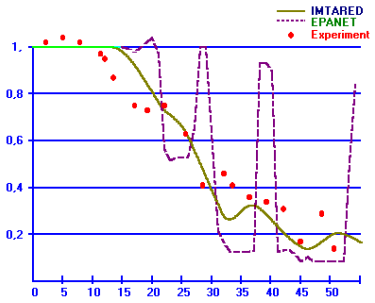


Figure 4.2: Concentration evolution obtained by the proposed model (IMTARED), the EPANET model and field measurements

stage. As a result of the application of this technique the large system of equations produced by the numerical scheme is decomposed in three tri-diagonal systems for each pipe and a smaller system of equations for the concentration at the network nodes. The numerical solution is applied to a real water distribution network for which results of simulations with the EPANET model and field observations are available. In the network pipes with medium and high flow velocities the two models give similar results. In pipes with low flow velocities the measured concentration evolution is represented better by the proposed model than by the EPANET model, due to the inclusion of dispersion.

REFERENCES

[1] A. A. Aldama, V. G. Tzatchkov, and F. I. Arreguin. The numerical Green's function technique for boundary value problems in networks. In W. Blain, editor, *Hydraulic Engineering Software VII*, pages 121–130, Southampton, England, 1998. Wessex Intitute of Technology, Computational Mechanics Publications/WIT Press.

[2] A. A. Aldama, V. G. Tzatchkov, F. I. Arreguin, and L. R. Puente. An efficient numerical solution for convection-diffusion transport in pipe networks. In *Computational Methods in Water Resources XI, Vol. 2: Computational Methods in Surface Flow and Transport Problems*, pages 119–127, Southampton, Boston, 1996. Computational Mechanics Publications.

[3] American Water Works Association Research Foundation, Denver, Colo. *Characterization and Modeling of Chlorine Decay in Distribution Systems*, 1996.

[4] American Water Works Association Research Foundation and Environmental Protection Agency. *Conference on Water Quality Modeling in Distribution Systems*, Cincinnati, Ohio., 1991.

[5] A. Baptista. *Solution of advection-dominated transport by Eulerian-Lagrangian methods using the backwards methods of characteristics*. PhD thesis, Massachusetts Institute of Technology, Mass., 1987.

[6] G. Roach. *Green's Functions*. Cambridge University Press, Cambridge, England., 2nd edition, 1982.

[7] L. Rossman. The epanet water quality model. In B. Coulbeck, editor, *Integrated Computer Applications in Water Supply*, pages 79–93, Taunton, England, 1993. Research Studies Press Ltd.

[8] L. Rossman, R. Clark, and W. Grayman. Modeling chlorine residuals in drinking-water distribution systems. *Journal of Environmental Engineering, ASCE*, 120(4):803–820., 1994.

[9] I. Stackgold. *Green's Functions and Boundary Value Problems*. John Wiley and Sons, New York, N.Y., 1979.

[10] V. Tzatchkov, A. Aldama, and F. Arreguin. Modelacion numerica de la adveccion y dispersion de solutos en redes de distribucion de agua potable. *Ingenieria Hidraulica en Mexico*, 15(3):101–115, 2000. in Spanish.

[11] V. Tzatchkov, A. Aldama, and F. Arreguin. Advection-dispersion-reaction modeling in water distribution networks. *Journal of Water Resources Planning and Management, ASCE*, 2002.

[12] V. Tzatchkov, F. Arreguin, and A. Aldama. An application of the numerical Green's function technique to advection-diffusion contaminant transport in water supply networks. In W. Blain, editor, *Hydraulic Engineering Software VII*, pages 587–596, Southampton, England, 1998. Computational Mechanics Publications/WIT Press.